Université de Montréal

Multi-player Games in the Era of Machine Learning

par

Gauthier Gidel

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des arts et des sciences
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

July, 2020

Université de Montréal
Faculté des arts et des sciences

Cette thèse intitulée:

Multi-player Games in the Era of Machine Learning

présentée par:

Gauthier Gidel

a été évaluée par un jury composé des personnes suivantes:

Ioannis Mitliagkas,	président-rapporteur
Simon Lacoste-Julien,	directeur de recherche
Yoshua Bengio,	membre du jury
Constantinos Daskalakis,	examinateur externe

Thèse acceptée le: .....

À toi qui lis ces lignes, et à la beauté qui te guide.

To you who read these lines, and to the beauty guiding you.

Résumé

Parmi tous les jeux de société joués par les humains au cours de l'histoire, le jeu de go était considéré comme l'un des plus difficiles à maîtriser par un programme informatique [Van Den Herik et al., 2002]; Jusqu'à ce que ce ne soit plus le cas [Silver et al., 2016]. Cette percée révolutionnaire [Müller, 2002, Van Den Herik et al., 2002] fût le fruit d'une combinaison sophistiquée de Recherche arborescente Monte-Carlo et de techniques d'apprentissage automatique pour évaluer les positions du jeu, mettant en lumière le grand potentiel de l'apprentissage automatique pour résoudre des jeux.

L'apprentissage antagoniste, un cas particulier de l'optimisation multiobjective, est un outil de plus en plus utile dans l'apprentissage automatique. Par exemple, les jeux à deux joueurs et à somme nulle sont importants dans le domain des réseaux génératifs antagonistes [Goodfellow et al., 2014] et pour maîtriser des jeux comme le Go ou le Poker en s'entraînant contre lui-même [Silver et al., 2017, Brown and Sandholm, 2017]. Un résultat classique de la théorie des jeux indique que les jeux convexes-concaves ont toujours un équilibre [Neumann, 1928]. Étonnamment, les praticiens en apprentissage automatique entrainent avec succès une seule paire de réseaux de neurones dont l'objectif est un problème de minimax non convexe et non concave alors que pour une telle fonction de gain, l'existence d'un équilibre de Nash n'est pas garantie en général. Ce travail est une tentative d'établir une solide base théorique pour l'apprentissage dans les jeux.

La première contribution explore le théorème minimax pour une classe particulière de jeux nonconvexes-nonconcaves qui englobe les réseaux génératifs antagonistes. Cette classe correspond à un ensemble de jeux à deux joueurs et a somme nulle joués avec des réseaux de neurones.

Les deuxième et troisième contributions étudient l'optimisation des problèmes minimax, et plus généralement, les inégalités variationnelles dans le cadre de l'apprentissage automatique. Bien que la méthode standard de descente de gradient ne parvienne pas à converger vers l'équilibre de Nash de jeux convexes-concaves simples, il existe des moyens simples d'utiliser des gradients pour obtenir des méthodes qui convergent. Nous étudierons plusieurs techniques telles que l'extrapolation, la moyenne et la quantité de mouvement à paramètre négatif.

La quatrième contribution fournit une étude empirique du comportement pratique des réseaux génératifs antagonistes. Dans les deuxième et troisième contributions, nous diagnostiquons que la méthode du gradient échoue lorsque le champ de vecteur du jeu est fortement rotatif. Cependant, une telle situation peut décrire un

pire des cas qui ne se produit pas dans la pratique. Nous fournissons de nouveaux outils de visualisation afin d'évaluer si nous pouvons détecter des rotations dans comportement pratique des réseaux génératifs antagonistes.

Abstract

Among all the historical board games played by humans, the game of go was considered one of the most difficult to master by a computer program [Van Den Herik et al., 2002]; Until it was not [Silver et al., 2016]. This odds-breaking breakthrough [Müller, 2002, Van Den Herik et al., 2002] came from a sophisticated combination of Monte Carlo tree search and machine learning techniques to evaluate positions, shedding light upon the high potential of machine learning to solve games.

Adversarial training, a special case of multiobjective optimization, is an increasingly useful tool in machine learning. For example, two-player zero-sum games are important for generative modeling (GANs) [Goodfellow et al., 2014] and mastering games like Go or Poker via self-play [Silver et al., 2017, Brown and Sandholm, 2017]. A classic result in Game Theory states that convex-concave games always have an equilibrium [Neumann, 1928]. Surprisingly, machine learning practitioners successfully train a single pair of neural networks whose objective is a nonconvex-nonconcave minimax problem while for such a payoff function, the existence of a Nash equilibrium is not guaranteed in general. This work is an attempt to put learning in games on a firm theoretical foundation.

The first contribution explores minimax theorems for a particular class of nonconvex-nonconcave games that encompasses generative adversarial networks. The proposed result is an approximate minimax theorem for two-player zero-sum games played with neural networks, including WGAN, StarCrat II, and Blotto game. Our findings rely on the fact that despite being nonconcave-nonconvex with respect to the neural networks parameters, the payoff of these games are concave-convex with respect to the actual functions (or distributions) parametrized by these neural networks.

The second and third contributions study the optimization of minimax problems, and more generally, variational inequalities in the context of machine learning. While the standard gradient descent-ascent method fails to converge to the Nash equilibrium of simple convex-concave games, there exist ways to use gradients to obtain methods that converge. We investigate several techniques such as extrapolation, averaging and negative momentum. We explore these technique experimentally by proposing a state-of-the-art (at the time of publication) optimizer for GANs called ExtraAdam. We also prove new convergence results for Extrapolation from the past, originally proposed by Popov [1980], as well as for gradient method with negative momentum.

The fourth contribution provides an empirical study of the practical landscape of GANs. In the second and third contributions, we diagnose that the gradient method breaks when the game's vector field is highly rotational. However, such a situation may describe a worst-case that does not occur in practice. We provide new visualization tools in order to exhibit rotations in practical GAN landscapes. In this contribution, we show empirically that the training of GANs exhibits significant rotations around Local Stable Stationary Points (LSSP), and we provide empirical evidence that GAN training converges to a *stable* stationary point which, is a saddle point for the generator loss, not a minimum, while still achieving excellent performance.

Keywords—Mots-clés

machine learning, game theory, adversarial training, minimax, Nash equilibrium, optimization, multi-player games, variational inequality, generative adversarial networks, extragradient, generative modeling, landscape visualization, momentum.

apprentissage statistique, théorie des jeux, apprentissage antagoniste, minimax, equilibre de Nash, optimisation, jeux a somme nulle, inégalités variationelles, réseaux génératifs antagonistes, extragradient, model génératifs, visualisation de champ de vecteurs, méthode du moment.

Contents

List of Tables

List of Figures

List of acronyms and abbreviations

CCE	coarse correlated equilibria
e.g.	*exempli gratia* [for instance]
EG	Extragradient Method
ERM	Empirical Risk Minimization
i.e.	*ide est* [that is]
GANs	Generative Adversarial Networks
LSSE	Locally Stable Stationnary Point
ML	Machine Learning
NE	Nash Equilibrium
PPM	Proximal-Point Method
resp.	respectively
SGD	Stocastic Gradient Descent
VIP	Variational Inequality Problem

Acknowledgements

Aaron, Adam, Adrien, Ahmed, Alais, Amjad, Amy, Antoine, Ariane, Aude, Audrey, Alex, Alexandre, Alberto, Anna, Anne Marie, Annabel, Aristide, Arthur, Aymeric, Benoît (Beubeu), Bérénice, Camille, Catherine, Christophe, Claire, Clément (Clemsy), Danielle, Damien, David, Denise, Dzmitry, Élodie, Emma, Elise, Elyse, Eugene, Fabian, Florent, Florestan (Floflo), Francis, François, Félix, Fernand, Francesco, Gaëtan, Gabriel, Gabriella (Gaby), Garo, Geneviève (Geugeu), Georgios, Giancarlo, Grace, Guillaume, Gul, Guy (Gitou), Hélene, Hirosuke (Hiro), Hugo, Ian, Idriss, Ioannis, Issam, Jacques (Jak), Jaime, James, Jean Baptiste, Jeanne, Jérémy, Joey, Judith, Kawtar, Kyle, Laura, Laure, Léna, Léonie, Linda, Lucie, Loïc, Loucas, Marie-Auxille, Marion, Mark, Marta, Marie Christine, Maryse, Mastro, Mathieu, Maximilian, Max, Michel, Mickaël (Mika), Mohamad, Mohammad, Morgane, Myrto, Nazia, Niao, Nicolas, Olivier, Oscar, Pascal, Pierre, Quentin, Raphaël, Reyhane, Rémi, Rim, Robert, Romain, Salem, Sharan, Sébastian, Sébastien, Serge, Sandra, Samy, Sesh, Simon, Sophie, Tanis, Tanja, Tatjana, Tess, Theodor, Thomas, Tom, Tommy, Tony, Tristan, Tweety, Utku, Valentin, Veranika, Victor, Vincent, Yan, Yann (Yannou), Yana, Yoram, Yoshua, Ylva, Waïss, Wojtek, Zhor, Ziad.

Merci a toi que j'ai peut-être apprécié, respecté, aimé, idolatré ou detesté; avec qui j'ai parlé, échangé, fait de la recherche, souri, rêvé, ri, ou pleuré. Ma thèse fut un long voyage dont tu as fait partie. Nos chemin se sont croisé, et se recroiserons peut-être.

Thanks to you who I may have appreciated, respected, loved, idolized or hated; with whom I spoke, interacted, did research, smiled, dreamt, laughed, or cried. My thesis was a long journey you were part of. Our paths crossed and may cross again.

Notation

The set of real numbers	$\mathbb{R}$
The set of complex numbers	$\mathbb{C}$
The real and imaginary part of $z \in \mathbb{C}$	$\Re(z)$ and $\Im(z)$
Scalars are lower-case letters	λ
Vectors are lower-case bold letters	$\boldsymbol{\theta}$
Matrices are upper-case bold letters	$\boldsymbol{A}$
Operators are upper-case letters	F
The spectrum of a squared matrix $\boldsymbol{A}$	$\mathrm{Sp}(\boldsymbol{A})$
The spectral radius of a squared matrix $\boldsymbol{A}$ . . .	$\rho(A)$
The largest and the smallest singular values of $\boldsymbol{A}$	$\sigma_{\min}(\boldsymbol{A})$ and $\sigma_{\max}(\boldsymbol{A})$
The identity matrix of $\mathbb{R}^{d \times d}$	$\boldsymbol{I}_d$
Standard asymptotic notations	$\mathcal{O}$, Ω and Θ

1 Introduction

«On a pas les mêmes règles pourtant c'est le même jeu»
[We do not have the same rules yet it's the same game]– Lomepal [2019]

What is the game mentioned by Lomepal [2019]? For some, it could be the game of life [Gardner, 1970], while for others, it remains a mystery. However, what is certain is that we live in a world full of games. From the simplest ones, such as rock-paper-scissor, to the most challenging ones such as chess, go, or StarCraft II, games are so complex and interesting that there exist a professional league of players and dense theoretical literature for each of them [Simon and Chase, 1988, Bozulich, 1992, Vinyals et al., 2017].

That is why a long-standing goal in artificial intelligence [Samuel, 1959, Tesauro, 1995, Schaeffer, 2000] has been to achieve superhuman performance—with a computer program—at such games of skills.

Among all the historical board games played by humans, the game of go was considered one of the most difficult to master by a computer program [Van Den Herik et al., 2002]; Until it was not [Silver et al., 2016]. This odds-breaking breakthrough [Müller, 2002, Van Den Herik et al., 2002] came from a sophisticated combination of Monte Carlo tree search and machine learning techniques to evaluate positions, shedding light upon the high potential of machine learning to solve games.

Machine learning, the science of learning mathematical models from data,

has expanded significantly in the last two decades. It has had a noticeable impact in diverse areas such as computer vision [Krizhevsky et al., 2012], speech recognition [Hinton et al., 2012], natural language processing [Sutskever et al., 2014], computational biology [Libbrecht and Noble, 2015], and medicine [Esteva et al., 2017].

Interestingly, while regarding such different domains, these success stories have a common ground: they all correspond to the estimation of a prediction function based on empirical data [Vapnik, 2006]. This learning paradigm, based on empirical risk minimization (ERM), is known as supervised learning. At a high level, estimating the correct dependence through ERM requires three main ingredients: 1) a sufficient amount of data, 2) a hypothesis class on the actual dependence function, 3) a sufficient amount of computing to find an approximate solution to the corresponding minimization problem. These three steps are subject to interdependent tradeoffs [Bottou and Bousquet, 2008]—for instance, for a given computational

budget, more data would improve our ability to generalize but would make the optimization procedure harder—are at the heart of the challenges of supervised learning.

The notable success of the applications of supervised learning mentioned above can be attributed to many factors. Among them, one can arguably say that the building of high-quality datasets [Russakovsky et al., 2015], the improved access to computational resources, and the design of scalable training methods for large models [LeCun et al., 1998] played a significant role.

However, yet powerful, supervised learning is a restrictive setting where a single learner is in a fixed environment, i.e., it has access to a large number of input-output pairs at training time that come from an independent and identically distributed process. This assumption does not consider that some other agents, i.e., computer programs or humans, maybe part of the environment and thus impact the task.

Real-world games such as Go, poker, or chess are composed of multiple players, thus not fitting into the i.i.d. ERM framework that is composed of a single learner in a fixed environment. From the players' viewpoint, the environment that conditions the way they should play depends on their opponent and thus is not fixed. From a machine learning perspective, the task in those games is to learn how to play to beat any opponent.

Recently, machine learning techniques have led to significant progress on increasingly complex domains such as classical board games (e.g., chess [Silver et al., 2018] or go [Silver et al., 2017]), card games (e.g., poker [Brown and Sandholm, 2017, 2019] or Hanabi [Foerster et al., 2019]), as well as video games (e.g., StarCraft II [Vinyals et al., 2019] or Dota 2 [Berner et al., 2019]). However, chess, go, and more generally, all the zero-sum games of skills mentioned in this introduction are just "a *Drosophilia* of reasoning" [Kasparov, 2018]. We are just scratching the surface, the combination of multi-player games and machine learning can offer.

1 Multiplayer games and Machine Learning

Adversarial formulations, or more generally multi-player games, are frameworks that aim at casting tasks into which several agents (a.k.a players) compete (or collaborate) to solve a problem. At a high level, each agent is given a set of parameters and a loss that they try to minimize. The key difference between standard supervised learning and multi-player games is that each agent's loss depends on all the players' parameters, thus entangling the minimization problem of each player.

Such frameworks include real-world games such as Backgammon, poker, or Starcraft II, but also market mechanisms [Nisan et al., 2007], auctions [Vickrey, 1961], as well as games specifically designed for a machine learning purpose such as Generative Adversarial Networks (GANs) [Goodfellow et al., 2014] that enabled a

significant breakthrough in generative modeling, the domain of representing data distributions.

1.1 Motivation: defining 'good' task losses through games

Designing computer programs that learn from games' first principles has been a well-established challenge of artificial intelligence [Samuel, 1959, Tesauro, 1995].

Until recently, state-of-the-art performances were only achieved by injecting domain-specific human knowledge [Campbell et al., 2002, Genesereth et al., 2005, Silver et al., 2016] such as specific heuristics or openings discovered by humans or data of games played by professional players.

A recent breakthrough [Silver et al., 2018] permitted to master chess, shogi, and go by using a general algorithm that learns by playing against itself without having access to any data or knowledge of other players.

It demonstrates the powerful potential of game formulation long-ago noticed by the community [Genesereth et al., 2005]: the goal of "winning" the game is simple yet challenging to achieve. The complexity arises from the competition between the two players that usually start from a quite even state, and try to eventually take the lead by taking advantage of the subtlety of some rules. To win, the best-performing players must learn to incorporate knowledge representation, reasoning, and rational decision-making.

The framework of GANs defines a game between two neural networks, a generator that aims at creating realistic images and a discriminator that tries to distinguish real images from the generated ones. The discriminator implicitly defines a divergence between two distributions through the classification task: the data distribution and the generated one. Huang et al. [2017] argue that such divergences, implicitly defined by a class of discriminators, are "good" task losses for generative modeling: they are differentiable, have a better sample efficiency than standard divergences, are easier to learn from, and can encode high-level concepts such as "paying attention to the texture in the image."

The impact of such new differentiable game formulations for learning is compelling and promising, but they currently lack theoretical foundations.

1.2 Foundations of Games for Machine Learning

Compared to supervised learning, a loss minimization problem, multi-player games are a multi-objective minimization (or maximization) problem. For example, in a Texas hold'em poker game, each player tries to maximize its own gains. However, since the game is zero-sum, the maximization of each player' gain conflict with each other and thus cannot be considered individually. Thus, in a game, the optimization of each player must be considered *jointly*.

In that case, the standard notion of optimality is called *Nash equilibrium* [Nash et al., 1950]. A Nash equilibrium is achieved when no player can decrease its loss by unilaterally changing its strategy.

In general, the existence of an equilibrium is not guaranteed [Neumann and Morgenstern, 1944]. This fact is problematic since the Nash equilibrium is a natural and intuitive notion of target for the players of a game. Ensuring that we have a well-defined notion of equilibrium is a necessary first step to eventually build-up an understanding of multi-player games. The minimax theorem [Neumann, 1928] was among the first existence results of equilibria in games and is considered to be at the heart of game theory.

> *"As far as I can see, there could be no theory of games [without] the Minimax Theorem."– Neumann [1953]*

While this quote may appear slightly dismissive at first, it has to be put in context. Neumann [1953] and Fréchet [1953] is an exchange between Maurice Fréchet and John von Neumann regarding the relative contribution of Borel [1921] and Neumann [1928] in the foundation of the field that is now called game theory. Beyond the controversy, the quote mentioned above underlines that when building a new field, it is fundamental to show that the object of interest—in that case, a Nash equilibrium—exists to eventually build a theory on top of it. For instance, the question of the computational complexity of a Nash equilibrium is closely entangled with such an existence result: since a Nash always exists its computational complexity cannot belong to the class of NP problems [Papadimitriou, 2007].

In the context of machine learning, the considerations regarding games are different. Each player's payoff functions correspond to the performance of the machine learning models that represent that player. Often their models are parametrized by finite-dimensional variables. Consequently, the payoffs are (potentially non-convex) functions of these parameters, raising the question of the existence of an equilibrium for such a game played with machine learning models.

Overall, this thesis revolves around two main questions regarding machine learning in games: what is the target, and how can it be reached in a reasonable amount of time?

2 Overview of the Thesis Structure

Besides the introduction and conclusion, this thesis includes a background section followed by four contributions that correspond to four research papers that the author of this thesis wrote during his Ph.D.

The story of the contribution develops as follows: the first contribution explores the notion of equilibrium in the context of a game played by deep machine learning models by proving a minimax theorem for a particular class of nonconvex-nonconcave zero-sum two-player game where the players are neural networks. We then get interested in game optimization in the second and third contributions, and we finally provide an empirical study of some practical games' optimization landscape.

Before jumping into the background section, we provide a more detailed summary of each section.

2.1 Defining a target for learning in games

In his seminal work, Neumann [1928] considered zero-sum two-player games. In that case, both players are competing for the same payoff function, but while one tries to maximize it, the other aims at minimizing it. The minimax theorem ensures that under mild assumptions, such a game has a *value* and an *equilibrium*: there exists an optimal strategy for each player that induces a given value for the payoff function.

Such games are convenient to build up intuition because the notion of winning, losing, and tying can be related to the value of the payoff function, i.e., if the payoff of the first player is above (resp. below, equal) the value of the game it means that the first player is currently winning (resp. losing, tying) the game.

In that setting, a general minimax theorem has been proved by Sion et al. [1958] under the assumption that each player's strategy set is convex and that the payoff function is a convex-concave function.

In the context of machine learning, the payoff function often depends on the parameters of each player. For instance, in chess, the reinforcement learning policy that would pick each move may be parametrized by a neural network, and in GANs, the discriminator and the generator are usually neural networks.

Because of this neural network parameterization, one cannot expect the payoff function to be convex-concave in general (the same way one cannot expect the loss of a regression problem to be convex in supervised learning with a deep neural network) [Choromanska et al., 2015].

In our first contribution, we propose an approximate minimax theorem certain class of problems, including Wasserstein GAN formulations wherein both the generator and the discriminator are one hidden layer RELU network. Our result relies on the fact that for many practical games, e.g., GANs or Starcraft II, the nonconvex-nonconcavity of the payoff function comes from the neural network parametrization.

Roughly, we show that a pair of larger-width one-hidden-layer ReLU networks attain the min-max value of the game attainable by distributions over smaller-width one-hidden-layer RELU networks. The underlying intuition is as follows:

neural nets have a particular structure that interleaves matrix multiplications and simple non-linearities (often based on the max operator like ReLU). The matrix multiplications in one layer of a neural net compute linear combinations of functions encoded by the other layers. In other words, neural nets are (non-)linear mixtures of their sub-networks.

2.2 Building our theoretical understanding of game optimization

The learning dynamics of differentiable games ,such as GANs, may exhibit a *cyclic behavior*. For instance, consider a game such as rock-paper-scissors. Intuitively, it makes sense that each of the agents, trying to beat their opponent using gradient information, will slowly change their strategies to the best response that currently beats their opponent, continuously switching from rock to paper and then scissors. One can show that this intuition is accurate: naively implementing the gradient method to train the agents will lead to cycles where the players will alternatively play each different actions without even converging to a Nash equilibrium of the game.

In the second contribution, we show the failure of standard gradient methods to find an equilibrium of such simple examples inspired by rock-paper-scissors. Such a failure of the gradient method on simple games leads to an immediate question: are there principled methods that address this cycling problem? We answered this question affirmatively and proposed to tap into the variational inequality literature to leverage the concept of averaging and extrapolation to design new optimization methods that address the cycling problem and tackle the current constraints of modern machine learning, such as stochastic optimization in high dimension. Our main theoretical contributions are two-fold: we first consider the convergence of averaging and extrapolation for the optimization of bilinear examples. Our second theoretical contribution is the study a variant of extragradient that we call *extrapolation from the past* originally introduced by Popov [1980] and prove novel *convergence rates* in the context VIP with Lipschitz and strongly monotone operators, and stochastic VIP with Lipschitz operator. We *prove its convergence* for strongly monotone operators and in the stochastic VIP setting. Our empirical contribution is the introduction of a novel algorithm leveraging extrapolation, that we called ExtraAdam for the training of GANs. Our experiments show substantial improvements over Adam and OptimisticAdam [Daskalakis et al., 2018] leading to state-of-the-art performance for GANs at the time of publication.

In the third contribution, we investigate the impact of Polyak's momentum [Polyak, 1964] in the optimization dynamics of such games. Momentum is known to have a detrimental role in deep learning [Sutskever et al., 2013], but its effect in games was an unexplored topic. We prove that a negative value for the momentum hyper-parameter may improve the gradient method's convergence

properties for a large class of adversarial games. Notably, for games similar to the rock-paper-scissor example described above, negative momentum is a way to fix the gradient method where each player update alternatively their state (as opposed to simultaneously). This fact is quite surprising since, in standard minimization, the momentum's optimal value is positive. It is an excellent example of counter-intuitive phenomena occurring in games with differentiable payoffs. We also propose to build new intuition on the game dynamics by using a notion of rotation. In standard minimization, the iterates are 'attracted' to the solution and thus adding momentum will use the past gradient to enhance this attraction to the solution, and thus converge faster. In games, rotations around the optimum may occur. To picture a simple dynamic, one can think of a planet orbiting around the sun. Adding positive momentum will push the planet away from the sun (the equilibrium point), while negative momentum will use past information to correct the trajectory toward the sun. This balance between accelerating the attraction and correcting the rotation explains why in games, unlike in minimization, the optimal value for the momentum can be negative.

2.3 Studying the practical vector field of games

In the second and third contribution, we theoretically study the vector field of games, i.e., the concatenation of each player's payoff gradient, and build simple examples where the gradient method fails to converge. However, going beyond simple counter-examples, one question remains: does this cycling behavior that breaks standard gradient methods happen in practice? Such a phenomenon is due to the phenomenon of rotation that only occurs in differentiable games (compared to standard minimization): due to their potential adversarial component.

Our fourth and final contribution is an empirical study aiming to bridge the gap between the simple theoretical examples previously proposed and the practice. We formalize the notions of rotation and attraction in games by relating it to the imaginary part in the eigenvalues of the Jacobian of the game's vector field. We eventually develop a technique to visualize these rotations in differentiable games and apply it to GANs. We show empirical evidence of significant rotations on several GAN formulations and datasets. Moreover, we also study the nature of the gradient method's potential equilibrium and provide empirical evidence that standard training methods for GANs converge to an equilibrium point that is not a Nash equilibrium of the game.

3 Excluded research

In order to keep this manuscript consistent and succinct, the author has decided to exclude a significant part of the publication produced during his Ph.D. Some of

the excluded research constitute relevant related follow-ups to the work discussed in thesis [Huang et al., 2017, Chavdarova et al., 2019, Azizian et al., 2020a,b, Bailey et al., 2020, Ibrahim et al., 2020]. The author of this Ph.D. thesis also worked on:

- Line-search for non-convex minimization [Vaswani et al., 2019].
- Dynamics of Recurrent Neural Networks [Kerg et al., 2019].
- Implicit regularization for linear neural networks [Gidel et al., 2019a].
- Adaptive Three Operator Splitting [Pedregosa and Gidel, 2018].
- Variants of the Frank-Wolfe algorithm [Gidel et al., 2017, 2018].

2 Background

In this chapter we present the frameworks considered in our four contributions. First we will introduce the standard single-objective minimization, then we will talk about multi-objective minimization and Nash equilibrium and finally we will present a generalization of the these frameworks based on the necessary first order stationary conditions.

1 Single Objective Optimization

Despite the contributions of this thesis regarding the optimization of multi-player games, it seemed essential to the author to give an overview of the current knowledge on single objective optimization to contrast it with the numerous open questions remaining in multi-player games optimization.

1.1 Convex Optimization

Recent advances in machine learning are largely driven by the success of gradient-based optimization methods for the training process.

A common learning paradigm is empirical risk minimization, where a (potentially non-convex) objective, that depends on the data, is minimized. In this section we introduce the standard notions present in single-objective minimization.

Let f a function and $\mathcal{X}$ a convex set. A set $\mathcal{X}$ is said to be convex if,

$$\boldsymbol{x}, \boldsymbol{y} \in \mathcal{X} \Rightarrow \gamma \boldsymbol{x} + (1-\gamma)\boldsymbol{y} \in \mathcal{X}\,,\ \forall \gamma \in [0,1]\,. \tag{1.1}$$

For simplicity, we will assume that $\mathcal{X}$ is a subset of $\mathbb{R}^d$, but most of the results provided in this work can be generalized to infinite-dimensional spaces.

Single-objective minimization is the problem of finding a solution to the following minimization problem:

$$\text{find } \boldsymbol{x}^* \text{ such that } \boldsymbol{x}^* \in \mathcal{X}^* := \arg\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x})\,. \tag{MIN}$$

The way to estimate the quality of an approximate minimizer $\boldsymbol{x}$ is to use a *merit function*. Formally, a function $g : \mathcal{X} \to \mathbb{R}$ is called a *merit function* if g is non-negative such that $g(\boldsymbol{x}) = 0 \Leftrightarrow \boldsymbol{x} \in \mathcal{X}^*$ [Larsson and Patriksson, 1994]. For

minimization, a natural merit function is the *suboptimality*: $f(\boldsymbol{x}) - f^*$ where f^* is the minimum of f. We will see in Section 2 that there exists different way to extend the notion of suboptimality to zero-sum two-player games. Some of these extensions are not a merit function.

A standard assumption on the function f is convexity. Such assumption is standard because it is a sufficient condition for the local minimal to also be global minima. It is also a convenient assumption to obtain *global* convergence rates. Such an assumption can be weakened using for instance the Polyak-Lojasiewicz condition or the quadratic growth condition (see [Karimi et al., 2016] for an extensive study of the relations between these conditions).

A function f is said to be *convex* if its value at a convex combination of point is smaller than the convex combination of its values:

$$f(\gamma\boldsymbol{x} + (1-\gamma)\boldsymbol{y}) \leq \gamma f(\boldsymbol{x}) + (1-\gamma)f(\boldsymbol{y})\,, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}\,,\ \forall \gamma \in [0,1]\,. \tag{1.2}$$

From this property, it follows that the function f is *sub-differentiable*, i.e., there exist a linear lower-bound for f at any point. Moreover, the set of the sub-differential $\partial f(\boldsymbol{x})$ at point $\boldsymbol{x}$ is defined as,

$$\partial f(\boldsymbol{x}) := \{\boldsymbol{d} \in \mathbb{R}^d \ :\ f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + (\boldsymbol{y}-\boldsymbol{x})^\top \boldsymbol{d}\,,\ \forall \boldsymbol{y} \in \mathbb{R}^d\}\,. \tag{1.3}$$

If f is convex then this set is convex and non-empty for any $\boldsymbol{x} \in \mathcal{X}$.

In this work we are mostly interested in first-order optimization and we will assume that the function f is *differentiable*. A stronger assumption on the regularity of f is the *smoothness* assumption. A function f is said to be L-*smooth* if its gradient is L-Lipschitz, i.e.,

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L\|\boldsymbol{x}-\boldsymbol{y}\|_2\,,\ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}\,. \tag{1.4}$$

This assumption is very common but may be weakened with the *Hölder continuity* condition:

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L_\nu\|\boldsymbol{x}-\boldsymbol{y}\|_2^\nu\,,\ \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}\,, \tag{1.5}$$

where L_ν is a constant defined for any $\nu \in (0,1]$. Note that L_ν may be infinite for some ν and that Hölder continuity implies that,

$$f(\boldsymbol{y}) \leq f(\boldsymbol{x}) + \nabla f(\boldsymbol{x})^\top(\boldsymbol{y}-\boldsymbol{x}) + \frac{L_\nu}{1+\nu}\|\boldsymbol{y}-\boldsymbol{x}\|_2^{1+\nu}\,, \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathcal{X}\,. \tag{1.6}$$

One last common assumption that can be made on f is *strong convexity*. A function f is said to be μ-*strongly convex* if $f - \mu\|\cdot\|_2^2$ is convex, which is equivalent to say that,

$$f(\gamma \boldsymbol{x}+(1-\gamma)\boldsymbol{y}) \leq \gamma f(\boldsymbol{x})+(1-\gamma)f(\boldsymbol{y})-\mu\gamma(1-\gamma)\|\boldsymbol{x}-\boldsymbol{y}\|_2^2\,,\ \forall \boldsymbol{x},\boldsymbol{y} \in \mathcal{X}\,,\ \forall \gamma \in [0,1]\,. \tag{1.7}$$

Note that being convex is equivalent to being 0-strongly convex.

Unconstrained Minimization Methods

If the constraint set $\mathcal{X}$ is equal to $\mathbb{R}^d$ then (MIN) is an *unconstrained* minimization problem. In that case the *necessary stationary condition* for differentiable functions is,

$$\boldsymbol{x}^* \in \mathcal{X}^* \quad \Rightarrow \quad \nabla f(\boldsymbol{x}^*) = \boldsymbol{0} \tag{1.8}$$

When the function f is convex this condition is a sufficient condition for optimality.

When the function f is differentiable, a standard method to solve (MIN) is the *gradient descent* method (GD) which dates back to Cauchy [1847]. At time $t \geq 0$, this method requires the computation of the gradient at the current iterate $\boldsymbol{x}_t$ to get the next iterate $\boldsymbol{x}_{t+1}$ with the following update rule,

$$\text{Gradient Descent:} \quad \boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)\,, \tag{1.9}$$

where $\eta_t > 0$ is called the *step-size*.[1] This method is called a *descent* method because at point $\boldsymbol{x}_t$ the direction $\nabla f(\boldsymbol{x}_t)$ is a *descent direction*,[2] meaning that for η_t small enough $f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t)$. Moreover if the function f is smooth, *gradient descent* benefits from the following descent lemma:

Lemma 1 ((1.2.13) [Nesterov, 2004])**.** *If f is a L-smooth function then for $\eta_t \leq 1/L$ we have that,*

$$f(\boldsymbol{x}_{t+1}) \leq f(\boldsymbol{x}_t) - \frac{\eta_t}{2}\|\nabla f(\boldsymbol{x}_t)\|_2^2\,,\ \forall \boldsymbol{x}_t \in \mathbb{R}^d\,. \tag{1.10}$$

This property is key in the convergence proof of *gradient descent*. Note this property does *not* require the convexity of f, actually this lemma is also heavily used in order to show properties on the iterates when the objective function f is non-convex (see §1.2)

Constrained Optimization

When the constrained set $\mathcal{X}$ is a strict subset of $\mathbb{R}^d$, then (MIN) is a *constrained optimization problem* and $\mathcal{X}$ is called the *constraint set*. In that case the standard

[1] In machine learning this quantity is also known as *learning-rate*.

[2] Actually, it can be shown that any direction $\boldsymbol{d}$ such that $\boldsymbol{d}^\top \nabla f(\boldsymbol{x}) > 0$ is a descent direction from $\boldsymbol{x}$.

gradient descent method cannot be longer used because the direction given by the gradient may not be *feasible*, i.e., $\exists \boldsymbol{x} \in \mathcal{X}$ s.t. $\boldsymbol{x} - \eta \nabla f(\boldsymbol{x}) \notin \mathcal{X} \, \forall \eta > 0$. In that case, the standard method to solve such problem is the *projected gradient method*. This method requires an additional *projection* step in order to get a feasible iterate

$$\text{Projected Gradient Descent:} \quad \boldsymbol{x}_{t+1} = P_{\mathcal{X}}[\boldsymbol{x}_t - \eta_t \nabla f(\boldsymbol{x}_t)] \,, \tag{1.11}$$

where $P_{\mathcal{X}}[\cdot]$ is the projection onto the set $\mathcal{X}$ defined as $P_{\mathcal{X}}[\boldsymbol{x}] = \arg\min_{\boldsymbol{y} \in \mathcal{X}} \|\boldsymbol{x} - \boldsymbol{y}\|_2^2$.

If the set $\mathcal{X}$ is convex, the projection is a *quadratic* minimization problem that has a unique solution. Otherwise, this projection sub-problem might be very challenging to solve. However, considering non-convex constraint sets goes way beyond the scope of this work. That is why, in the following we will always assume that the constraint set $\mathcal{X}$ is convex.

1.2 Non-Convex Single-objective Optimization

When the function f is non-convex, gradient descent may converge to a *stationary point* that is not the *global minimum* of the function. For simplicity of the discussion, in this section, we only consider the *unconstrained setting*. In that case, a *stationary point* is a point where the gradient is equal to zero. Since the gradient at a stationary point is by definition always null, we have that if $\boldsymbol{x}_t$ is a stationary point, then the following iterate $\boldsymbol{x}_{t+1}$ computed by gradient descent (1.11) is equal to $\boldsymbol{x}_t$. There are three kinds of *stationary points*: local minima, local maxima and saddle points. Let $\boldsymbol{x} \in \mathbb{R}^d$,

1. A stationary point such that there exists a neighbourhood U of $\boldsymbol{x}$ such that $f(\boldsymbol{y}) \leq f(\boldsymbol{x}) \,, \; \forall \boldsymbol{y} \in U$, is a local maximum.

2. A stationary point such that there exists a neighbourhood U of $\boldsymbol{x}$ such that $f(\boldsymbol{y}) \geq f(\boldsymbol{x}) \,, \; \forall \boldsymbol{y} \in U$, is a local minimum.

3. A stationary point such that for any neighbourhood U of $\boldsymbol{x}$, there exist $\boldsymbol{y}, \boldsymbol{y}' \in U$ such that $f(\boldsymbol{y}) \leq f(\boldsymbol{x}) \leq f(\boldsymbol{y}')$, is a saddle point.

It can be shown that with random initialization gradient descent almost surely converges to a local minimizer [Lee et al., 2016]. From an optimization perspective, this property is appealing since we aim to converge to the global minimizer of f. However, standard gradient descent can take an exponential time (exponential in the dimension d) to escape saddle [Du et al., 2017]. Fortunately, adding noise at some key moments can get rid of this exponential constant [Jin et al., 2017].

2 Multi-objective Optimization

As argued previously, single-objective minimization plays a major role in current machine learning. However, some recently introduced models require the joint minimization of several objectives.

For example, actor-critic methods can be written as a bi-level optimization problem [Pfau and Vinyals, 2016] and generative adversarial networks (GANs) [Goodfellow et al., 2014] use a two-player game formulation.

In that case, the goal of the learning procedure is to find an *equilibrium* of this multi-objective optimization problem (a.k.a. multi-player game). The notion of equilibrium points date back to Cournot [1838]. It was later formalized by Nash et al. [1950] who pioneered the field of game theory.

2.1 Minimax Problems and Two-player Games

The *two-player game problem* [Neumann and Morgenstern, 1944, Nash et al., 1950] consists in finding the following *Nash equilibrium*:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}_1(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}_2(\boldsymbol{\theta}^*, \boldsymbol{\varphi})\,. \tag{2.1}$$

One important point to notice is that the two optimization problems (with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$) in (2.1) are *coupled* and have to be considered *jointly* from an optimization point of view. In game theory $\mathcal{L}_i$ is also known as the *payoff* of the i^{th} player.

When $\mathcal{L}_1 = -\mathcal{L}_2 := \mathcal{L}$, the two-player game is called a *zero-sum game* and (2.1) can be formulated as a *saddle point problem* [Hiriart-Urruty and Lemaréchal, 1993, VII.4]:

$$\text{find } (\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \quad \text{s.t.} \quad \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}) \leq \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \leq \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*)\,, \quad \forall (\boldsymbol{\theta}, \boldsymbol{\varphi}) \in \Theta \times \Phi\,. \tag{2.2}$$

If such a pair $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ exists, then we have that,

$$\max_{\boldsymbol{\varphi}\in\Phi} \min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \tag{2.3}$$

Note that by weak duality [Rockafellar, 1970], it is generally true that,

$$\sup_{\boldsymbol{\varphi}\in\Phi} \inf_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \leq \inf_{\boldsymbol{\theta}\in\Theta} \sup_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})\,. \tag{2.4}$$

2.2 Extension to n-player Games

We can extend the two-player game framework to a game with an arbitrary number of players. A n-player game is a set of n players and their respective losses

$\mathcal{L}^{(i)} : \mathbb{R}^d \to \mathbb{R}, 1 \leq i \leq n$. Player i controls the parameter $\boldsymbol{\theta}_i \in \Theta_i \subset \mathbb{R}^{d_i}$ where $\sum_{i=1}^n d_i = d$. The *n-player game problem* consists in finding the Nash equilibrium:

$$\boldsymbol{\theta}_i^* \in \arg\min_{\boldsymbol{\theta}_i \in \Theta_i} \mathcal{L}^{(i)}(\boldsymbol{\theta}_1^*, \ldots, \boldsymbol{\theta}_{i-1}^*, \boldsymbol{\theta}_i, \boldsymbol{\theta}_{i+1}^*, \ldots, \boldsymbol{\theta}_n^*), \quad 1 \leq i \leq n. \tag{2.5}$$

Note that any non-zero-sum n-player game can be written as a $(n+1)$-player zero-sum game adding a $(n+1)$-th loss equal to minus the sum of the other losses.

2.3 Existence of Equilibria

In the case of a zero-sum game, standard results [Sion et al., 1958, Fan, 1953, Hiriart-Urruty and Lemaréchal, 1993] show that under convexity assumptions there exists a saddle point of $\mathcal{L}$. We present the result from [Hiriart-Urruty and Lemaréchal, 1993] that requires the following assumptions,

(H1) the objective function $\mathcal{L}$ is *convex-concave*, i.e., $\mathcal{L}(\cdot, \boldsymbol{\varphi})$ is convex for all $\boldsymbol{\varphi} \in \Phi$ and $\mathcal{L}(\boldsymbol{\theta}, \cdot)$ is concave for all $\boldsymbol{\theta} \in \Theta$.

(H2) The sets Θ and Φ are nonempty closed convex sets.

(H3) Either Θ is bounded or there exists $\bar{\boldsymbol{\varphi}} \in \Phi$ such that $\mathcal{L}(\boldsymbol{\theta}, \bar{\boldsymbol{\varphi}}) \to \infty$ when $\|\boldsymbol{\theta}\| \to \infty$.

(H4) Either Φ is bounded or there exists $\bar{\boldsymbol{\theta}} \in \Theta$ such that $\mathcal{L}(\bar{\boldsymbol{\theta}}, \boldsymbol{\varphi}) \to \infty$ when $\|\boldsymbol{\varphi}\| \to \infty$.

Theorem 1. *[Hiriart-Urruty and Lemaréchal, 1993, Theorem 4.3.1] Under the assumptions (H1)-(H4) the payoff function $\mathcal{L}$ has a nonempty compact set of saddle points.*

In the first contribution of this thesis, we prove a minimax theorem where the payoff function $\mathcal{L}$ is *not* convex-concave. Such result is motivated by the machine learning applications where neural networks parametrizations induce nonconvex-nonconcave payoff functions.

Beyond the zero-sum two-player game setting, results on the existence of equilibria in multi-player games,first developed by Nash et al. [1950], is a rich literature [Nash, 1951, Glicksberg, 1952, Nikaidô et al., 1955, Dasgupta and Maskin, 1986] that is outside of the scope of this thesis.

2.4 Merit functions for games

When dealing with optimization of games, the first question to ask is the question of which merit function to use. For simplicity, we focus on zero-sum two-player games.

Some previous work [Yadav et al., 2018] considered the sum of the "minimization suboptimality" $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) - \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ with the "maximization suboptimality" $\mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) - \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$:

$$g(\boldsymbol{\theta}, \boldsymbol{\varphi}) := \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) - \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}) \tag{2.6}$$

Unfortunately, as explained in [Gidel et al., 2017] this function is *not* a merit function for the problem 2.2 in general. For example, with $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \boldsymbol{\theta} \cdot \boldsymbol{\varphi}$ and $\Theta = \Phi = [-1, 1]$, then $\boldsymbol{\theta}^* = \boldsymbol{\varphi}^* = \mathbf{0}$, implying that $g(\boldsymbol{\theta}, \boldsymbol{\varphi}) = 0$ for any $(\boldsymbol{\theta}, \boldsymbol{\varphi})$. However, when $\mathcal{L}$ is *strongly convex-concave* one can lower-bound g by the distance to the optimum times a constant.

In the general case, if the domains Θ and Φ are bounded, one can define the gap function

$$G(\boldsymbol{\theta}, \boldsymbol{\varphi}) := \max_{\boldsymbol{\varphi} \in \Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) - \min_{\boldsymbol{\theta} \in \Theta} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \max_{(\boldsymbol{\theta}', \boldsymbol{\varphi}') \in \Theta \times \Phi} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}') - \mathcal{L}(\boldsymbol{\theta}', \boldsymbol{\varphi}) \tag{2.7}$$

If, the domains are not bounded this function may be infinite except at the optimum (take for instance $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \boldsymbol{\theta} \cdot \boldsymbol{\varphi}$ and $\Theta = \Phi = \mathbb{R}$). In order to contravene this issue Nesterov [2007] considered the intersections $\Theta_R := \Theta \cup B(\bar{\boldsymbol{\theta}}, R)$ and $\Phi_R := \Phi \cup B(\bar{\boldsymbol{\varphi}}, R)$ where $B(a, R)$ is a ball of radius R and center a. If there exists a Nash equilibrium, then for any given $\bar{\theta}$ and $\bar{\boldsymbol{\varphi}}$ and a large enough R, the function

$$G_R(\boldsymbol{\theta}, \boldsymbol{\varphi}) := \max_{\boldsymbol{\varphi} \in \Phi_R} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) - \min_{\boldsymbol{\theta} \in \Theta_R} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \,, \tag{2.8}$$

is a merit function.

2.5 Other multi-objective formulation

There exist other multi-objective optimization formulations that are not multi-player games. Such formulations are outside of the scope of this thesis.

However, we provide a quick overview of the main alternatives.

Bilevel Optimization

Conversely to games, where all the players have a symmetric role, bilevel optimization is a multi-objective optimization framework introducing an asymmetry between the objectives. It considers an *upper-level* objective f and a *lower-level* objective g. The *lower-level* objective is used to induce a constraint on some parameters of the *upper-level* objective:

$$\begin{aligned} &\min_{\boldsymbol{\theta} \in \Theta} f(\hat{\boldsymbol{\omega}}(\boldsymbol{\theta}), \boldsymbol{\theta}) \\ &\text{s.t.} \quad \hat{\boldsymbol{\omega}}(\boldsymbol{\theta}) \in \arg\min_{\boldsymbol{\omega} \in \Omega} g(\boldsymbol{\omega}, \boldsymbol{\theta}) \end{aligned} \tag{2.9}$$

Note that this formulation can be more general with more objective or with a stochastic formulation, see for instance Colson et al. [2007], Bard [2013] for an overview of the field and Pedregosa [2016], Gould et al. [2016], Shaban et al. [2019] for applications in a machine learning context.

Stackelberg Games

Such notion of hierarchy between the objective is related to the notion of Stackelberg games Stackelberg [1934], Conitzer and Sandholm [2006], Fiez et al. [2020] that exhibit a notion of hierarchy between the players. In its simplest form (two-player), a Stackelberg game opposes a *follower* and a *leader*. The latter can choose its strategy with the knowledge of the strategy of the follower. Thus, this problem can be formulated as a bilevel optimization problem where the *follower* corresponds to the *lower-level* objective and the *leader* corresponds to the *upper-level* objective.

2.6 Solving games with optimization

The question of algorithms to find Nash equilibrium is related to the notion of complexity of Nash equilibrium [Papadimitriou, 2007]. In general, Nash equilibria are hard to compute. For instance, simple statements such as 'are there two Nashes?' or 'is there a Nash that contains the strategy s?' are NP-hard problems for two-player games [Gilboa and Zemel, 1989].

However, computing a Nash equilibrium cannot be a NP-hard problem because a Nash equilibrium always exists [Papadimitriou, 2007]. The complexity of Nash equilibria computation belongs to a class of problems called PPAD [Daskalakis et al., 2009].

Consequently, it seems hopeless in general to design algorithms to solve games (even approximately [Papadimitriou, 2007]). However, there are some points to argue why this line of research is not futile. First, the Nash computation of zero-sum two-player games can be reformulated as a linear program. Thus, in that case, the computational challenge comes from the potentially large (or even infinite) number of strategies. For instance, in a machine learning context, the strategy spaces we may consider are finite-dimensional parameter spaces, motivating the use of gradient-based methods. Secondly, the practical instance we want to solve may not be hard. For instance, the mathematical programming literature developed a plethora of algorithms to try to find approximate solutions to NP-hard problems such as the travelling salesman problem [Bellmore and Nemhauser, 1968]. Another example is the deep learning community that successfully minimizes non-convex objectives [Zhang et al., 2017] while it is NP-hard even to check if a point is a local minimizer of the objective function [Murty and Kabadi, 1985, Nesterov, 2000].

In the particular of minimax optimization, there exists a very rich literature that deals with problems of the form

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d}\max_{\boldsymbol{\varphi}\in\mathbb{R}^p} f(\boldsymbol{\theta}) + \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\varphi} - g(\boldsymbol{\varphi})\,. \tag{2.10}$$

When f and g are convex, such formulation is the primal-dual formulation of a convex problem (see for instance [Rockafellar, 1970] for more details about convex duality). Many primal-dual algorithms have been designed to solve such particular minimax problem such as the Arrow-Hurwicz algorithm [Arrow et al., 1958, Zhu and Chan, 2008], the Chambolle-Pock Primal-Dual algorithm [Chambolle and Pock, 2011, 2016], the Accelerated Primal-Dual algorithm [Ouyang et al., 2015]. However, these algorithms heavily exploit the bilinear structure of (2.10) and thus cannot be straightforwardly extended to general minimax games.

In this thesis, we focus on methods to solve games with differentiable payoffs. While recently, due to the motivations coming from the practical applications in the context of machine learning, there is a revival of specific gradient-based method optimization for games (see discussion in Chapter 11, there the mathematical programming literature dealt with such (differentiable) game optimization problems by casting them as variational inequalities.

In the following section, we present the variational inequality framework and eventually present the standard methods such problems.

3 Variational Inequality Problem

Let $\Omega \subset \mathbb{R}^d$, and $F : \Omega \to \mathbb{R}^d$ be a continuous mapping. In this section $\|\cdot\|$ is a norm of $\mathbb{R}^d$.

The *variational inequality problem* [Harker and Pang, 1990] is:

$$\text{find } \boldsymbol{\omega}^* \in \Omega \quad \text{such that} \quad F(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \;\; \forall \boldsymbol{\omega} \in \Omega\,. \tag{VIP}$$

We call *optimal set* the set Ω^* of $\boldsymbol{\omega} \in \Omega$ verifying (VIP).

A standard assumption on F is *monotonicity*:

$$(F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}'))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') \geq 0 \quad \forall\, \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega\,. \tag{3.1}$$

If $F(\boldsymbol{\omega}) = \nabla f(\boldsymbol{\omega})$, it is equivalent to f being convex. If F can be written as (2.5), it implies that the cost functions are convex.[3]

When the operator F is monotone, we have that

$$F(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \leq F(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*)\,, \;\; \forall \boldsymbol{\omega}, \boldsymbol{\omega}^*\,. \tag{3.2}$$

Hence, in this case,

[3] The convexity of the cost functions in (2.3) is a necessary condition (not sufficient) for the operator to be monotone.

(VIP) implies a stronger formulation sometimes called *Minty variational inequality* [Crespi et al., 2005]:

$$\text{find } \boldsymbol{\omega}^* \in \Omega \quad \text{such that} \quad F(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \quad \forall \boldsymbol{\omega} \in \Omega\,. \tag{MVI}$$

This formulation is stronger in the sense that, under mild assumptions, if (MVI) holds for some $\boldsymbol{\omega}^* \in \Omega$, then (VIP) holds too [Minty, 1967]. A stronger assumption than monotonicity is *μ-strong monotonicity*,

$$(F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}'))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') \geq \mu\|\boldsymbol{\omega} - \boldsymbol{\omega}'\|^2 \quad \forall\, \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega\,. \tag{3.3}$$

Note that 0-*strong monotonicity* is equivalent to monotonicity.

3.1 Merit Functions for variational inequality problems

A *merit function* useful for our analysis can be derived from this formulation. Roughly, a merit function is a convergence measure.

A way to derive a merit function from (MVI) would be to use $g(\boldsymbol{\omega}^*) = \sup_{\boldsymbol{\omega} \in X} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}^* - \boldsymbol{\omega})$ which is zero if and only if (MVI) holds for $\boldsymbol{\omega}^*$. To deal with unbounded constraint sets (leading to a potentially infinite valued function outside of the optimal set), we use the *restricted merit function* [Nesterov, 2007]:

$$\mathrm{Err}_R(\boldsymbol{\omega}_t) := \max_{\boldsymbol{\omega} \in \Omega, \|\boldsymbol{\omega} - \boldsymbol{\omega}_0\| \leq R} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}_t - \boldsymbol{\omega})\,. \tag{3.4}$$

This function acts as merit function for (VIP) on the interior of the open ball of radius R around $\boldsymbol{\omega}_0$, as shown in Lemma 1 of Nesterov [2007]. That is, let $\Omega_R := \Omega \cap \{\boldsymbol{\omega} : \|\boldsymbol{\omega} - \boldsymbol{\omega}_0\| < R\}$. Then for any point $\hat{\boldsymbol{\omega}} \in \Omega_R$, we have

$$\mathrm{Err}_R(\hat{\boldsymbol{\omega}}) = 0 \Leftrightarrow \hat{\boldsymbol{\omega}} \in \Omega^* \cap \Omega_R. \tag{3.5}$$

The reference point $\boldsymbol{\omega}_0$ is arbitrary, but in practice it is usually the initialization point of the algorithm. R has to be big enough to ensure that Ω_R contains a solution. Err_R measures how much (MVI) is violated on the restriction Ω_R.

Such merit function is standard in the variational inequality literature. A similar one is used in [Nemirovski, 2004, Juditsky et al., 2011].

3.2 Standard algorithms to Solve Variational Inequality Problems

One very important piece in the VIP optimization is the projection P_Ω onto the set Ω:

$$P_\Omega[\boldsymbol{\omega}] \in \arg\min_{\boldsymbol{\omega}' \in \Omega} \|\omega - \omega'\|^2\,. \tag{3.6}$$

In the following we will assume that we can compute such a projection quite efficiently. Note that one can extend this projection framework to other geometries using Bregman divergences [Bregman, 1967]. For simplicity and clarity, we work with projections with respect to a norm $\|\cdot\|$.

Among the first algorithms to solve VIP is the standard projection method [Sibony, 1970],

$$\text{Projection Method:} \quad x_{t+1} = P_{\Omega}[x_t - \eta_t F(x_t)]\,. \tag{3.7}$$

This algorithm converges linearly for Lipschitz and strongly monotone operators. However, this method does not converge, in general, for monotone operators [Korpelevich, 1976]. One way to contravene this issue is to solve a sequentially less regularized problem using the projection method this technique is known as the proximal-point method (PPM) [Martinet, 1970, Rockafellar, 1976]

$$\text{PPM:} \quad x_{t+1} = \text{Solution of VIP with the operator } F_k(\boldsymbol{\omega}) := c_k F(\boldsymbol{\omega}) + (\boldsymbol{\omega} - \boldsymbol{\omega}_k)\,. \tag{3.8}$$

This method converges linearly (in terms of projection calls) when F is strongly monotone and sublinearly when F is monotone. However, the inner-outer loop structure as well as the supplementary regularization hyperparameter of this method make it less practical than the projection method.

A middle ground was achieved by Korpelevich [1976] with a method, called extragradient, that does not have inner loops and converges when F is monotone,

$$\text{Extragadient Method:} \quad \begin{cases} y_t = P_{\Omega}[x_t - \eta_t F(x_t)] \\ x_{t+1} = P_{\Omega}[x_t - \eta_t F(y_t)]\,. \end{cases} \tag{3.9}$$

Such a method is based on the idea of *extrapolation*, where the vector field used to update x_t is not computed at x_t but at an extrapolated point y_t. The idea behind the computation of y_t is that y_t roughly approximate the solution the iterates of the proximal point method. This idea of comparing y_t to the solution of the proximal point method is actually at the heart of the analysis of the method provided by Nemirovski [2004].

Note that, the idea of using an extrapolation step was to give "stability" to the gradient is prior to Korpelevich [1976]'s work (see for instance Polyak [1963, Chap. II]).

4 Neural Networks Training

In this section, we introduce the definition of (artificial) *feed-forward neural networks*. A *feed-forward* neural network is a composition of affine transformations

$W_i \cdot + b_i : \mathbb{R}^{d_i} \to \mathbb{R}_{d_{i+1}}$ and non-linearities $\sigma_i : \mathbb{R}^{d_{i+1}} \to \mathbb{R}^{d_{i+1}}$ for $1 \leq i \leq r$. The integer r is called the *depth* of the neural network and $\max_i d_i$ is called the width of the neural network. Formally, a neural network is a function $f : \mathcal{X} \to \mathcal{Y}$ where $\mathcal{X}$ is called the *input space* and $\mathcal{Y}$ is called the *output space* where the function f is defined as,

$$f = f_1 \circ \dots f_r \quad \text{where} \quad f_i(\boldsymbol{h}_i) = \boldsymbol{h}_{i+1} = \sigma_i(\boldsymbol{W}_i \boldsymbol{h}_i + \boldsymbol{b}_i)\,,\ 1 \leq i \leq r\,. \tag{4.1}$$

The vectors $\boldsymbol{h}_i \in \mathbb{R}^{d_i}$, $2 \leq i \leq r-1$ are called the hidden states. Note that $\boldsymbol{h}_1 = \boldsymbol{x}$ is the input and $\boldsymbol{h}_r = \boldsymbol{y}$ is the output. Two-layer feed-forward neural networks are known to be universal approximators, with a width going to infinity [Hornik et al., 1989].

Empirical Risk Minimization for Supervised Learning

Consider the following setting, general enough to be applied to many supervised learning problems. We have a finite set of data $(\boldsymbol{x}_i, y_i)_{1 \leq i \leq n} \in (\mathcal{X} \times \mathcal{Y})^n$ independently sampled from a distribution P. Given a loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and a class $\mathcal{F}$ of prediction function $f : \mathcal{X} \to \mathcal{Y}$, the goal of the learning procedure is to minimize the *risk*:

$$\arg\min_{f \in \mathcal{F}} \mathbb{E}_{(\boldsymbol{x},y)\sim P} \ell(f(\boldsymbol{x}), y) \tag{4.2}$$

Since, in practice, one has often only access to a *finite* number of samples, the optimization procedure can only be done on the *empirical risk*:

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{k=1}^{n} \ell(f(\boldsymbol{x}_i), y_i) \tag{4.3}$$

Here, we are not developing two issues: if f is the $0-1$ loss this problem is NP-hard [Feldman et al., 2012, Ben-David et al., 2003] and that in practice one consider surrogates losses and in order to get a solution of (4.3) with good generalization properties (i.e., being close to the minimizer of (4.2)) one usually adds a regularization to (4.3). These problems are standard issues of supervised learning and are not the direction of research of this work. That is why, in the following, we will focus on the optimization of (4.3) where ℓ is a differentiable function.

Since in modern machine learning the dimension d and the number of samples n are large, second-order method (because of large d) and batch methods (because of large n) are prohibitively expensive. That is why machine learning engendered the revival of *stochastic first order methods.* One of the most popular algorithms belonging to this class of method is the *stochastic gradient method* (SGD) [Robbins and Monro, 1951].

Stochastic Gradient Descent

Let $\mathcal{F}_{\boldsymbol{\theta}}$ be a parametrized family of function. The *stochastic gradient descent* is method similar as (1.11) but using an *unbiased estimate* of the gradient instead of the gradient itself. Let assume that we want to solve

$$\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim Q}\, \ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)\,. \tag{4.4}$$

For instance, if Q is the empirical distribution associated with the data $(\boldsymbol{x}_i, y_i)_{1\leq i\leq n}$, this problem is just a rewriting of (4.3) with an expectation. The principle of SGD is to sample $(\boldsymbol{x}, y) \sim Q$ and then to compute $\nabla_{\boldsymbol{\theta}}\ell(f_{\boldsymbol{\theta}}(\boldsymbol{x}), y)$ to update $\boldsymbol{\theta}$ with this estimate of the gradient,

$$\text{Stochastic Gradient Descent:} \quad \begin{cases} \text{Sample: } (\boldsymbol{x}, y) \sim Q \\ \text{Compute: } \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}), y)\,, \end{cases} \tag{4.5}$$

Note that this method is not a *descent* method and thus should be called *stochastic gradient method* but the acronym SGD has become standard. Under some reasonable assumption, such as the Lipschitzness of the expected gradient $\boldsymbol{\theta} \mapsto \mathbb{E}_P \nabla f_{\boldsymbol{\theta}}(\boldsymbol{x}, y)$ and the finite variance of the estimator one can show that this method does converge at a $O(1/\sqrt{t})$ rate.

Adaptive methods

Variants of SGD that scale coordinates of the gradient by some sort of averaging of the previous gradient coordinates observed during the optimization procedure have known a large success for neural networks optimization particularly because these methods provide a sort of learning rate adaptivity for each individual feature. Seminal works in this line of research proposed an algorithm called AdaGrad [Duchi et al., 2011] proving significantly better convergence guarantees than SGD when the gradients are sparse or small,

$$\text{AdaGrad:} \quad \begin{cases} \text{Sample: } (\boldsymbol{x}, y) \sim Q \\ \text{Set: } g_t := \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}), y) \text{ and } V_t := \dfrac{\operatorname{diag}(\sum_{s=1}^t g_s^2)}{t} \\ \text{Compute: } \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \dfrac{g_t}{\sqrt{V_t}}\,. \end{cases} \tag{4.6}$$

If the gradient is not sparse or small (for instance in non-convex optimization, gradients may vary a lot between early and late in learning) the learning rates suffer from a too rapid decay and performances of Adagrad deteriorate. In order to fix that issue, the non-convex optimization literature considers several variants

of Adagrad. The most popular variant of Adagrad for deep learning is arguably *Adam* [Kingma and Ba, 2015]:

$$\text{Adam:} \begin{cases} \text{Sample: } (\boldsymbol{x}, y) \sim Q \\ \text{Compute: } m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \,,\; V_t := \beta_2 V_{t-1} + (1-\beta_2) g_t^2 \\ \text{Set: } g_t := \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\boldsymbol{x}), y) \\ \text{Compute: } \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \dfrac{m_t}{\sqrt{V_t}} \,, \end{cases} \tag{4.7}$$

Note that, in practice, for all the methods presented in this section, in order to avoid singularities, a small ϵ is added to the denominator. Even if Adam has been widely adopted in practice, this method suffers from a fundamental theoretical issue (mainly due to the fact that the step-size may not decrease) and may not converge [Reddi et al., 2019].

5 Generative Adversarial Networks

The purpose of generative modeling is to generate samples from a distribution $q_{\boldsymbol{\theta}}$ that matches best the true distribution p of the data. The generative adversarial network training strategy can be understood as a *game* between two players called *generator* and *discriminator*. The former produces a sample that the latter has to classify between real or fake data. The final goal is to build a generator able to produce sufficiently realistic samples to fool the discriminator.

From a game theory point of view, GAN training is a differentiable two-player game (2.1): the discriminator $D_{\boldsymbol{\varphi}}$ aims at minimizing its cost function $\mathcal{L}^D$ and the generator $G_{\boldsymbol{\theta}}$ aims at minimizing its own cost function $\mathcal{L}^G$.

5.1 Standard GANs

In the original GAN paper [Goodfellow et al., 2014], the GAN objective is formulated as a *zero-sum game* where the cost function of the discriminator $D_{\boldsymbol{\varphi}}$ is given by the negative log-likelihood of the binary classification task between real or fake data generated from $q_{\boldsymbol{\theta}}$ by the generator,

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \quad \text{where} \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x}\sim p}[\log D_{\boldsymbol{\varphi}}(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{x}'\sim q_{\boldsymbol{\theta}}}[\log(1 - D_{\boldsymbol{\varphi}}(\mathbf{x}'))] \,. \tag{5.1}$$

However Goodfellow et al. [2014] recommend to use in practice a second formulation, called *non-saturating GAN*. This formulation is a *non-zero-sum game* where the aim is to jointly minimize

$$\mathcal{L}^G(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x}' \sim q_{\boldsymbol{\theta}}} \log D_{\boldsymbol{\varphi}}(\mathbf{x}') \text{ and } \mathcal{L}^D(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x} \sim p} \log D_{\boldsymbol{\varphi}}(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{x}' \sim q_{\boldsymbol{\theta}}} \log(1 - D_{\boldsymbol{\varphi}}(\mathbf{x}'))\,. \tag{5.2}$$

This formulation has the same *stationary points* as the zero-sum one (5.1) but are claimed to provide "much stronger gradients early in learning" [Goodfellow et al., 2014].

The distribution $q_{\boldsymbol{\theta}}$ is sampled by sampling $\boldsymbol{z}$ according to a prior distribution π (often a multivariate Gaussian distribution) and then transforming $\boldsymbol{z}$ with the generator function,

$$\boldsymbol{x} \sim p_{\boldsymbol{\theta}} \quad \Leftrightarrow \quad \boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z})\,, \quad \boldsymbol{z} \sim \pi\,. \tag{5.3}$$

5.2 Divergence minimization and Wasserstein GANs

An interesting point of view on GANs is that the GANs objective formulated as minimax are a divergence between the distribution p of the real data and the one $q_{\boldsymbol{\theta}}$ of fake data. In practice, the divergence they are minimizing are **parametric adversarial divergences** [Huang et al., 2017] of the form $\sup_{\boldsymbol{\varphi} \in \Phi} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{x}') \sim p \otimes q_{\boldsymbol{\theta}}}[\ell(f_{\boldsymbol{\varphi}}(\boldsymbol{x}), f_{\boldsymbol{\varphi}}(\boldsymbol{x}'))]$. In other words, the loss of a GAN between the distribution p of the real data and the one $q_{\boldsymbol{\theta}}$ of fake data is a parametric divergence.

One popular example is the 1-Wasserstein distance [Villani, 2009]:

$$W_1(p, q) := \inf_{\gamma \in \Gamma(p,q)} \mathbb{E}_{(\boldsymbol{x}, \boldsymbol{y}) \sim \gamma}[\|\boldsymbol{x} - \boldsymbol{y}\|_1] \tag{5.4}$$

where $\Gamma(p, q) := \{\gamma \ : \ \gamma|_p = p,\, \gamma|_q = q\}$ is the collection of all measures in the product space with marginals p and q. The dual formulation of the 1-Wasserstein distance is a maximum over 1-Lipschitz functions,

$$W_1(p, q) := \sup_{f\,,\, 1-\text{Lip}} \mathbb{E}_p[f(\boldsymbol{x})] - \mathbb{E}_q[f(\boldsymbol{x})]\,. \tag{5.5}$$

By choosing $\Delta(f_{\boldsymbol{\varphi}}(\boldsymbol{x}), f_{\boldsymbol{\varphi}}(\boldsymbol{x}')) = f_{\boldsymbol{\varphi}}(\boldsymbol{x}) - f_{\boldsymbol{\varphi}}(\boldsymbol{x}')$ and constraining $f_{\boldsymbol{\varphi}}$ to the class of 1-Lipschitz functions, we get the (parametric) Wasserstein GAN (WGAN) proposed by Arjovsky et al. [2017]:

$$\min_{\boldsymbol{\theta} \in \Theta} \max_{\boldsymbol{\varphi} \in \Phi, \|f_{\boldsymbol{\varphi}}\|_L \leq 1} \mathbb{E}_{\boldsymbol{x} \sim p}[f_{\boldsymbol{\varphi}}(\boldsymbol{x})] - \mathbb{E}_{\boldsymbol{x}' \sim q_{\boldsymbol{\theta}}}[f_{\boldsymbol{\varphi}}(\boldsymbol{x}')]. \tag{5.6}$$

3 Prologue to First Contribution

1 Article Details

Minimax Theorems for Nonconcave-Nonconvex Games Played with Neural Networks. *Gauthier Gidel, David Balduzzi, Wojciech Marian Czarnecki, Marta Garnelo and Yoram Bachrach.* This paper was submitted at NeurIPS 2020.

2 Contributions of the authors

Gauthier Gidel contributed to the original idea and the writing of the paper, the experiments and the theorems. David Balduzzi, Wojciech Marian Czarnecki, Marta Garnelo and Yoram Bachrach provided valuable feedback and helped in the genesis of the idea of the paper. David Balduzzi and Yoram Bachrach supervised and contributed to the writing of the paper. Marta Garnelo helped for the experiments.

4 Minimax Theorems for Nonconcave-Nonconvex Games Played with Neural Networks

Abstract

Adversarial training, a special case of multi-objective optimization, is an increasingly prevalent machine learning technique: some of its most notable applications include GAN-based generative modeling and self-play techniques in reinforcement learning which have been applied to complex games such as Go or Poker. In practice, a *single* pair of networks is typically trained in order to find an approximate equilibrium of a highly nonconcave-nonconvex adversarial problem. However, while a classic result in game theory states such an equilibrium exists in concave-convex games, there is no analogous guarantee if the payoff is nonconcave-nonconvex. Our main contribution is to provide an approximate minimax theorem for a large class of games where the players are ReLU neural networks including WGAN, StarCraft II and Blotto Game. Our findings rely on the fact that despite being nonconcave-nonconvex with respect to the neural networks parameters, these games are concave-convex with respect to the actual functions (or distributions) parametrized by these neural networks.

1 Introduction

Real-world games have been used as benchmarks in artificial intelligence for decades [Samuel, 1959, Tesauro, 1995], with recent progress on increasingly complex domains such as poker [Brown and Sandholm, 2017, 2019], chess, Go [Silver et al., 2017], and StarCraft II [Vinyals et al., 2019]. Simultaneously, remarkable advances in generative modeling of images [Wu et al., 2019] and speech synthesis [Bińkowski et al., 2020] have resulted from zero-sum games *explicitly* designed to facilitate via carefully constructed arms races [Goodfellow et al., 2014]. Zero-sum games also play a fundamental role in building classifiers that are robust to adversarial attacks [Madry et al., 2018].

The goal of the paper is to put learning—by neural nets—in two-player zero-sum games on a firm theoretical foundation to answer the question: *What does it mean to solve a game with neural nets?*

In single-objective optimization, performance is well-defined via a fixed objective. However, it is not obvious what counts as “optimal" in a two-player zero-sum

nonconcave-nonconvex setting. Since each player's goal is to maximize its payoff, it is natural to ask whether a player can maximize its utility independently of how the other player behaves. Neumann and Morgenstern [1944] laid the foundation of game theory with the Minimax theorem, which provides a meaningful notion of optimal behavior against an unknown adversary. For a two-player zero-sum game, such a solution concept incorporates two notions: (i) *a value* V, (ii) *a strategy for each player* such that their average gain is at least V (resp. -V) no matter what the other does. The existence of such a value and optimal strategies in concave-convex games is guaranteed in Sion et al. [1958], an extension of von Neumann's result.

From a game-theoretic perspective, minimax may not exist in nonconcave-nonconvex games. Nevertheless, machine learning practitioners typically train a *single* pair of neural networks to solve

$$\min_{\theta\in\Theta}\max_{w\in\Omega}\varphi(w,\theta) \quad \text{where} \quad (w,\theta)\mapsto\varphi(w,\theta) \text{ is nonconcave-nonconvex}\,. \tag{1.1}$$

Previous work [Arora et al., 2017, Hsieh et al., 2019, Domingo-Enrich et al., 2020] coped with this nonconcave-nonconvexity issue by transforming Eq. 1.1 into a bilinear minimax problem over the Borel distributions on Θ and Ω (a.k.a. lifting trick),

$$\min_{\mu\in\mathcal{P}(\Theta)}\max_{\nu\in\mathcal{P}(\Omega)}\langle\mu, A\nu\rangle \quad \text{where} \quad \langle\mu, A\nu\rangle := \mathbb{E}_{\theta\sim\mu, w\sim\nu}[\varphi(w,\theta)] \tag{1.2}$$

However, working on the space of distributions (a.k.a, mixed strategies) over the weights of a neural network is not practical and does not exactly correspond to the initial problem (1.1). That is why we do not consider (1.2) and put our focus on proving a minimax theorem for (1.1).

Our main contribution is Theorem 1, an approximate minimax theorem for nonconcave-nonconvex games for which Assumption 1 holds. This result contrasts with the negative result of Jin et al. [2019] who construct a nonconvex-nonconcave game where pure global minimax does not exist. The insights provided by our main theorem are three-fold; first, it provides a principled explanation of why practitioners have successfully trained a single pair of neural nets in games like GANs. Secondly, the equilibrium achieved in the theorem has a meaningful interpretation as the solution of a game where the players have *limited-capacity*. Finally, we show how latent parametrized policies used to solve matrix games such as Blotto Game or multi-agent RL problem such as Starcraft II fit the assumptions of our minimax results and, as an illustration, apply this method to solve differentiable Blotto, a game with an infinite strategy space. All the proofs of the propositions and theorems can be found in the appendix.

2 Related work

Minimax theorems in GANs. Many papers have adopted a game-theoretic perspective on GANs, motivating the computation of distributions of networks (in practice, finite collections) [Arora et al., 2017, Oliehoek et al., 2018, Hsieh et al., 2019, Grnarova et al., 2018, Domingo-Enrich et al., 2020]. However, these papers fail to explain why, in practice, it suffices to train only a single generator and discriminator (instead of collections) to achieve state-of-the-art performance [Brock et al., 2019]. Overall, even if we share motivations with the related work mentioned above (providing principled results), our results and conclusion are fundamentally different: we provide explain why using a single generator and discriminator—not a distribution over them—makes sense. We do so by proving that one can achieve a notion of nonconcave-nonconvex minimax equilibrium in GANs parametrized with neural networks.

Stackelberg games and local optimality. The literature has considered other notions of equilibrium. Recently, Fiez et al. [2020] proved results on games where the goal is to find a (local) Stackelberg equilibrium. Using that perspective, Zhang et al. [2020] and Jin et al. [2019] studied local-optimality in the context of nonconcave-nonconvex games. Our work provides a complementary perspective by providing a *global* minimax optimality theorem in a restricted—though realistic—nonconcave-nonconvex setting. Stackelberg equilibria may be meaningful in some contexts, such as adversarial training, but we argue in §1 that the minimax theorem is fundamental for defining a valid notion of solution for a large class of machine learning applications.

Parametrized strategies in games. The notion of latent matrix games mentioned in this paper is similar to the pushforward technique proposed by Dou et al. [2019]. It can also be related to the latent policies used in some multi-agent reinforcement learning (RL) applications. For instance the agent trained by Vinyals et al. [2019] to play the game of StarCraft II had policies of form $\pi(a|s, z)$ where z belongs to structured space that corresponds to a particular way to start the game or to actions it should complete during the game (e.g., the first 20 constructed units and buildings). Moreover, using parameterized function approximator to estimate strategies in games has been at the heart of multi-agent RL [Baxter et al., 2000, François-Lavet et al., 2018, Mnih et al., 2015]. Our contribution regarding latent matrix games (and more broadly latent RL policies), is the theoretical framework to study equilibrium in such parametrized nonconvex-nonconcave games and the associated approximate minimax theorem we provide (Thm. 1).

Bounded rationality. In his seminal work, Simon [1969] introduced the principled of bounded rationality. Generally speaking, it aims to capture the idea that rational agents are actually incapable of dealing with the full complexity of a realistic environment, and thus by these limitations, achieve a sub-optimal solution. Neyman [1985], Papadimitriou and Yannakakis [1994], Rubinstein and Dalgaard

[1998] modeled these limitations by constraining the computational resources of the players. Similarly, Quantal response equilibrium (QRE) [McKelvey and Palfrey, 1995] is a way to model bounded rationality: the players do not choose the best action, but assign higher probabilities to actions with higher reward. Overall, QRE, bounded rationality/computation have a similar goal as latent games: to model players that are limited by computation/memory/reasoning, however, the way the limits are modeled differs since in this work we consider equilibrium achieved with functions that have a limited representative power.

Finding a Nash equilibrium of Colonel Blotto. After its introduction by Borel [1921], finding a Nash equilibrium of the Colonel Blotto game has been an open question for 85 years. Roberson [2006] found an equilibrium solution for the continuous version of the game, later extended to the discrete symmetric case by Hart [2008]. The equilibrium computation in the general case remains open. Recently, Blotto has been used as a challenging use-case for equilibrium computation [Ahmadinejad et al., 2019]. Similarly, we consider a variant of Blotto to validate that we can find approximate equilibrium in games with a single pair of neural nets.

3 Motivation: Two-player Games in Machine Learning

A *two-player zero-sum game* is a *payoff function* $\varphi : \Omega \times \Theta \to \mathbb{R}$, that evaluates pairs of strategies (w, θ). The goal of the game is to find an *equilibrium*, i.e., a pair of strategies (ω^*, θ^*) such that,

$$\varphi(w, \theta^*) \leq \varphi(w^*, \theta^*) \leq \varphi(w^*, \theta)\,,\ \forall w \in \Omega,\, \theta \in \Theta\,. \tag{3.1}$$

The existence of such an equilibrium ensures that the order in which the players choose their respective strategy does *not* matter and that there exists a *non-exploitable* strategy,

$$\min_{\theta\in\Theta} \max_{w\in\Omega} \varphi(w, \theta) = \max_{w\in\Omega} \min_{\theta\in\Theta} \varphi(w, \theta) = \varphi(w^*, \theta^*). \tag{3.2}$$

If the function φ is concave-convex and if the sets Θ and Ω are convex and compact then Sion's Minimax Theorem [Sion et al., 1958] insures that such a Nash equilibrium does exist.

Previous theoretical work in the context of machine learning [Arora et al., 2017, Oliehoek et al., 2018, Grnarova et al., 2018, Hsieh et al., 2019] considered the model *parameters* w and θ as the strategies of the game. Arguably, the most well-know example of such a game is GANs.

Example 1. *[Goodfellow et al., 2014] A GAN is a game where the first player, the* discriminator D_w*, is a binary classifier parametrized by* $w \in \mathbb{R}^{p_1}$ *and the second player, the* generator G_θ*, is a parametrized mapping from a latent space to an output space. The payoff* φ *is then the ability of the first player to discriminate a real data distribution* p_{data} *from the generated distribution,*

$$\varphi(w, \theta) := \mathbb{E}_{x \sim p_{data}}\Big[\log D_w(x)\Big] + \mathbb{E}_{z \sim \mathcal{N}(0, I_d)}\Big[\log\Big(1 - D_w(G_\theta(z))\Big)\Big]. \tag{3.3}$$

Unfortunately, Example 1 does *not* satisfy Sion Minimax theorem's assumptions for the following reasons: (i) The parameter sets are not compact. (ii) The function φ is not concave-convex because of the non-convexity induced by the neural networks parametrization. While one can easily cope with the first issue—by for instance restricting ourself to bounded weights or by leveraging Fan's Theorem [Fan, 1953]—the second issue (ii) is an intrinsic part of learning by neural networks.

On the one hand, one cannot expect (3.2) to be valid for general nonconcave-nonconvex games [Jin et al., 2019]. On the other hand, many games in the context of machine learning have a particular structure since, as we will see in the next section, their nonconcave-nonconvexity comes from the neural network parametrization.

Two complementary perspectives on a game. Example 1 can be interpreted as a game between two players, one player, the *generator*, proposes a sample that the other player, the discriminator tries to distinguish from a real data distribution p_{data}. In that game, the parameters w and θ of the payoff function (3.3), do not explicitly correspond to any meaningful strategy – i.e., generating a sample or distinguishing data from generated samples. They respectively parametrize a discriminator and a distribution one can consider as players that have an intuitive interpretation in the GAN game.

Considering p_θ the generated distribution in (3.3), we have a duality between parameters and players

$$\varphi(\underbrace{w, \theta}_{params}) = \tilde{\varphi}(\underbrace{D_w, p_\theta}_{players}) := \mathbb{E}_{x \sim p_{data}}\Big[\log D_w(x)\Big] + \mathbb{E}_{x' \sim p_\theta}\Big[\log\Big(1 - D_w(x')\Big)\Big]. \tag{3.4}$$

A compelling aspect of this dual perspective is that even though, one *cannot* expect φ, the payoff function of the parameters w and θ, to be concave-convex, the payoff of the players $\tilde{\varphi}$ is *concave-convex*. Formally, for any $w, w' \in \Omega$, $\theta, \theta' \in \Theta$ and and $\lambda \in [0, 1]$ we have,

$$\tilde{\varphi}(\lambda D_w + (1-\lambda) D_{w'}, p_\theta) \geq \lambda \tilde{\varphi}(D_w, p_\theta) + (1-\lambda)\tilde{\varphi}(D_{w'}, p_\theta) \quad \text{(by concavity of log)}$$
$$\tilde{\varphi}(D_w, \lambda p_\theta + (1-\lambda) p_{\theta'}) = \lambda \tilde{\varphi}(D_w, p_\theta) + (1-\lambda)\tilde{\varphi}(D_w, p_{\theta'}) \quad \text{(by linearity of } p \mapsto \mathbb{E}_p\text{.)}$$

Note that the notion of convex combination for the players is quite subtle here: $\lambda D_w + (1-\lambda) D_{w'}$ corresponds to a convex combination of functions

while $\lambda p_\theta + (1-\lambda)p_{\theta'}$ corresponds to a convex combination (a.k.a mixture) of distributions.

Even if the payoff (3.4) is concave-convex with respect to (D, p), one *cannot* apply (yet) any standard minimax theorem for the following reason: given $w_1, w_2 \in \Omega$ and $\lambda \in [0,1]$ we may have

$$\nexists w \in \Omega\,, \quad \text{s.t.} \quad \lambda D_{w_1} + (1-\lambda) D_{w_2} = D_\omega\,, \tag{3.5}$$

meaning that the set of functions $\mathcal{F}_\Omega := \{D_w \,|\, w \in \Omega\}$ may *not* be convex in general. However, for the particular case of functions parametrized by ReLU neural networks we will show that the set $\mathcal{F}$ is "almost convex" (see Prop. 2 and 3). It is one of the core results used in the proof of Thm. 1.

4 An assumption for nonconcave-nonconvex games

The games arising in machine learning are not classical normal- or extensive-form games. Rather, they often use neural nets to approximate complex functions and high dimensional distributions [Brock et al., 2019, Razavi et al., 2019]. That is why they are often considered *general nonconcave-nonconvex games* (1.1). However, as illustrated in (3.4), in the machine learning context, many games have a particular structure where the players' payoff is concave-convex.

Assumption 1. *The nonconcave-nonconvex game* (1.1) *is assumed to have a* concave-convex *players' payoff, i.e., the parameters w and θ respectively parametrize f_w and g_θ such that,*

$$\varphi(\underbrace{w, \theta}_{params}) = \tilde{\varphi}(\underbrace{f_w, g_\theta}_{players}) \quad \textit{where} \quad (f,p) \mapsto \tilde{\varphi}(f,p) \textit{ is concave-convex}. \tag{4.1}$$

We call f_w and g_θ the players of the game, they can either be a parametrized function or distribution.

One example of such a game has been developed in (3.4) where the first player, D_w is a function and the second one, p_θ is a distribution over images. Another example is the Wasserstein GAN (WGAN).

Example 2. *[Arjovsky et al., 2017] The WGAN formulation is a minimax game with a payoff φ s. t.,*

$$\varphi(w, \theta) = \tilde{\varphi}(D_w, p_\theta) := \mathbb{E}_{x \sim p_{data}} D_w(x) - \mathbb{E}_{x' \sim p_\theta} D_w(x')\,, \tag{4.2}$$

where the discriminator D_w has to be 1-Lipschitz, i.e., $\|D_w\|_L \leq 1$. By bilinearity of the function $(D,p) \mapsto \mathbb{E}_p[D(x)]$ we have that $\tilde{\varphi}$ is bilinear and thus satisfies Assumption 1.

Finally, we present a way to any cast matrix game with a very large (or even infinite) number of strategies into a game that follows Assumption 1.

Using function approximation to solve matrix games. In the case of matrix games, the payoff function $\varphi : A \times B \to \mathbb{R}$ has no concave-convex structure, and the sets A and B are often even discrete. Neumann and Morgenstern [1944] introduced mixed strategies $p \in \Delta(A)$, where $\Delta(A)$ is the set of probability distributions over A, in order to guarantee the existence of an equilibrium. In game-theory, a well-known example of a challenging matrix game is the Colonel Blotto game.

Example 3 (Colonel Blotto Game)**.** *Consider two players who control armies of S_1 and S_2 soldiers respectively. Each colonel allocates their soldiers on K battlefields. A strategy for player-i is an allocation $a_i \in A_i$ and the payoff of the first player is the number of battlefields won*

$$\varphi(a_1, a_2) := \tfrac{1}{K} \textstyle\sum_{k=1}^{K} \mathbf{1}\{[a_1]_k > [a_2]_k\} \tag{4.3}$$

where $A_i := \Big\{ a \in \mathbb{N}^K \ : \ \sum_{k=1}^{K} [a]_k \leq S_i \,, \ 1 \leq k \leq K \Big\}$.

In Example 3, the number of strategies grows *exponentially fast* as K grows. Consequently, one cannot afford to work with an explicit distribution over the strategies. A tractable way to compute an equilibrium of the Colonel Blotto Game has been an open question for decades. The GANs examples (Eq.3.4 & 4.2) suggest to consider distributions implicitly defined with a generator. Given a latent space $\mathcal{Z}$, a latent distribution π on $\mathcal{Z}$ and a mapping $g_\theta : \mathcal{Z} \to A$, we can define the distribution $p_\theta \in \Delta(A)$ as

$$a \sim p_\theta \ : \ a = g_\theta(z) \,, \ z \sim \pi \,. \tag{4.4}$$

Definition 1 (Latent Matrix Game)**.** *A* latent matrix game $(\varphi, \mathcal{F}, \mathcal{G})$ *is a two-player zero-sum game where the players pick $f_w \in \mathcal{F}$ and $g_\theta \in \mathcal{G}$ and, given π and π' two fixed distributions, obtain payoffs*

$$\varphi(w, \theta) := \mathbb{E}_{z \sim \pi, \, z' \sim \pi'} \Big[\varphi\Big(f_w(z), g_\theta(z') \Big) \Big] \,.$$

The reformulation of any matrix game as a latent game satisfies Assumption 1.

Example 3 (Latent Blotto)**.** *Consider the functions $f_w : \mathbb{R}^p \to A_1$ and $g_\theta : \mathbb{R}^p \to A_2$. The payoff is*

$$\varphi(w, \theta) := \tfrac{1}{K} \textstyle\sum_{k=1}^{K} \mathbb{P}\Big([f_w(Z_1)]_k > [g_\theta(Z_2)]_k \Big) \tag{4.5}$$

where $Z_1, Z_2 \sim \mathcal{N}(0, I_p)$ are independent Gaussians and A_i is defined in (4.3).

Latent matrix games encompasses multi-agent RL games played with RL policies such as the setting used by Vinyals et al. [2019] to play StarCraft II.[1] The agent, called AlphaStar, has a latent-conditioned policy $\pi(a|s,z)$ where z belongs to a structured space that represents information about how to start constructing units and buildings, and that is sampled from an expert human player distribution: $z \sim p_{human}(z)$. Given two agents $\pi_1(a|s,z)$ and $\pi_2(a|s,z)$, the payoff in the latent game is $\varphi(\pi_1,\pi_2) = \mathbb{P}(\pi_1 \text{ beats } \pi_2)$. The classes $\mathcal{F}$ and $\mathcal{G}$ correspond to the neural architectures used to parametrize the policies; the priors π and π' are the human expert distribution p_{human}.

In that example, and more generally in multi-agent RL zero-sum games played with policies parametrized by neural networks, the payoff $\varphi(w,\theta) = \mathbb{P}(\pi_w \text{ beats } \pi_\theta)$ is a potentially nonconcave-nonconvex function of the parameters but satisfies Assumption 1.

5 Minimax Theorems

We want to prove a minimax theorem for some nonconcave-nonconvex games (1.1) that satisfy Assumption 1. We start with an informal statement of our result.

Theorem 1. *[Informal] Let φ be a nonconcave-nonconvex payoff that satisfies Assumption 1 with $\tilde{\varphi}$ bilinear and where the players f_w and g_θ are one hidden layer ReLU networks of width p. For any $\epsilon > 0$ there exists a pair (w,θ) that achieves a* notion of approximate equilibrium.

When played with ReLU networks Example 2 and 3 satisfy the hypothesis of this theorem. The proof of this Theorem is split into 3 main steps: (i) in §5.1 by using the fact that $\varphi(w,\theta) = \tilde{\varphi}(f_w, g_\theta)$ we provide the existence of a limited-capacity equilibrium in the convex hull of the space of players (see Assum. 1 for the definition of players). Note that, since we are working in a larger space (the convex hull), one *cannot* expect to achieve this equilibrium with a single pair of parameters (w,θ). (ii) in §5.2 we show that approximate equilibrium can be achieved with a relatively small convex combination. (iii) in §5.3 we show that when using ReLU networks, such small convex combination of players can be achieved by a single larger ReLU network. A formal definition of convex combination of players is provided in §5.1.

[1] Note that here we do not claim the novelty of parametrizing policies/strategies such idea has been used in may games and RL applications (see related work section).

5.1 Limited Capacity Equilibrium in the Space of Players

Recall that by Assumption 1, the nonconcave-nonconvex payoff φ can be written as,

$$\varphi(w, \theta) = \tilde{\varphi}(f_w, p_\theta) \quad \text{where} \quad (f, p) \mapsto \tilde{\varphi}(f, p) \text{ is concave-convex}. \tag{5.1}$$

The players f_w and p_θ are either functions or distributions respectively parametrized by w and θ. For instance, in the context of WGAN (Example 2), f_w would be the discriminator and p_θ would be the generated probability distribution. In that example, notice that $\tilde{\varphi}(f_w, \cdot)$ is *not* convex with respect to the generator function but only with respect to the generated distribution. Similarly, if we computed convex combinations generator's parameters, the payoff φ would not be convex. Moreover, the Lipchitz constraint in Example 2 is natural in the function space, but it is challenging to translate it into a constraint in the parameter space.[2] Overall, using (5.1) one can rewrite (1.1) as follows,

$$\min_{f \in \mathcal{F}_\Omega} \max_{g \in \mathcal{G}_\Theta} \tilde{\varphi}(f, g) \tag{5.2}$$

where $\mathcal{F}_\Omega$ and G_Θ are function or distribution spaces (depending on the application) incorporating the limited capacity constraints of the problem, e.g., Lipschitz constraint. **In the following**, for simplicity of the discussion, we discuss what the formal definitions of a convex combination are when $\mathcal{F}_\Omega$ is a function space and $\mathcal{G}_\Theta$ is a distribution space when we have no additional constraint aside from the parametrization, i.e., $\mathcal{F}_\Omega := \{f_w \mid w \in \Omega\}$ and $\mathcal{G}_\Theta := \{p_\theta \,|\, \theta \in \Theta\}$. However, these notions and our results extend if we consider that both players are distributions (e.g., in Example 3), or if we add any convex constraint on the functions or the distributions, see Example 2.

Convex combination of functions. Let us consider w_1 and $w_2 \in \Omega$, the convex combination of the players f_{w_1} and f_{w_2} is their point-wise averaging. The convex hull of $\mathcal{F}_\Omega$ can be defined as,

$$\text{hull}(\mathcal{F}_\Omega) := \{\text{Averages from } \mathcal{F}_\Omega\} = \left\{ \sum_{i=1}^{K} \lambda_i f_{w_i} \,|\, w_i \in \Omega,\ \sum_{i=1}^{K} \lambda_i = 1,\ \lambda_i \geq 0,\ K \geq 0 \right\}. \tag{5.3}$$

Convex combination of distributions. Consider latent mappings θ_1 and $\theta_2 \in \Theta$ that parametrize probability distribution p_{θ_1} and p_{θ_2} over a set $\mathcal{X}$. The *convex combination* p_λ of p_{θ_1} and p_{θ_2} with $\lambda \in [0, 1]$ is the mixture of these two probability distributions, $p_\lambda := \lambda p_{\theta_1} + (1 - \lambda) p_{\theta_2}$.

To sample from p_λ, flip a biased coin with $\mathbb{P}(\text{heads}) = \lambda$. If the result is heads then sample a strategy from p_{θ_1} and if the result is tails then sample from p_{θ_2}. The convex hull of $\mathcal{G}_\Theta$ is,

[2] In practice, parameters are clipped [Arjovsky et al., 2017] or the Lipchitz constant of the network is approximated [Miyato et al., 2018]. These approximations can be arbitrarily far from the original constraint.

$$\text{hull}(\mathcal{G}_\Theta) := \{\text{Mixtures from } \mathcal{G}_\Theta\} = \Big\{ \sum_{i=1}^{K} \lambda_i p_{\theta_i} \,|\, \theta_i \in \Theta,\, \sum_{i=1}^{K} \lambda_i = 1,\, \lambda_i \geq 0,\, K \geq 0 \Big\}. \tag{5.4}$$

The set $\text{hull}(\mathcal{G}_\Theta)$ is a subset of $\mathcal{P}(\mathcal{X})$, the set of probability distributions on $\mathcal{X}$. This set is different from the set of distributions supported on $\mathcal{G}_\Theta$ considered by Arora et al. [2017], Hsieh et al. [2019]. It contains 'smaller' mixtures because there may be many distributions supported on $\mathcal{G}_\Theta$ that correspond to the same $p \in \text{hull}(\mathcal{G}_\Theta)$. Moreover these works did not take advantage of the convexity with respect to the discriminator function (see Example 1 and 2) by considering (5.3).

Existence of an equilibrium by playing in the convex hulls. Our first result is that there exists an equilibrium by allowing functions or distributions to be picked from their convex hulls.

Proposition 1. *Let φ be a game that follows Assumption 1. If $\mathcal{G}_\Theta$ and $\mathcal{F}_\Omega$ are compact, then there exist a* value *for the game such that,*

$$\mathrm{V}(\Omega, \Theta) := \sup_{f \in \text{hull}(\mathcal{F}_\Omega)} \inf_{p \in \text{hull}(\mathcal{G}_\Theta)} \tilde{\varphi}(f, p) = \inf_{p \in \text{hull}(\mathcal{G}_\Theta)} \sup_{f \in \text{hull}(\mathcal{F}_\Omega)} \tilde{\varphi}(f, p)\,, \tag{5.5}$$

where $\text{hull}(\mathcal{G}_\Theta)$ *and* $\text{hull}(\mathcal{F}_\Omega)$ *are either defined in* (5.3) *or in* (5.4)*, depending on the type player.*

After showing that the closure of $\text{hull}(\mathcal{G}_\Theta)$ and $\text{hull}(\mathcal{F}_\Omega)$ are compact, this proposition is a corollary of Sion et al. [1958]'s minimax theorem (see §3). Note that Ω and Θ are arbitrary and that this equilibrium differs from the infinite-capacity equilibrium of the game (5.2) where we would allow f and g to be any function or distribution (i.e. with no parametrization restriction). Because we consider the convex hull of $\mathcal{F}_\Omega$ and $\mathcal{G}_\Theta$, this equilibrium is achieved with *convex combinations* (5.3) resp. (5.4)) of p_{θ_i} , $i \geq 0$ (resp. f_{w_i}) and thus there is no reason to expect to achieve this equilibrium with a single pair of weights (w, θ) in general. However in §5.2, we show that one can approximate such an equilibrium with relatively small convex combinations.

5.2 Approximate minimax equilibrium

Approximate equilibria for (5.5) are the pairs of players ϵ-close to achieving the value of the game.

Definition 2 (ϵ-equilibrium)**.** *A pair $(f^*_\epsilon, p^*_\epsilon) \in \text{hull}(\mathcal{F}) \times \text{hull}(\mathcal{G})$ is an ϵ-equilibrium if,*

$$\min_{p \in \text{hull}(\mathcal{G}_\Theta)} \tilde{\varphi}(f^*_\epsilon, p) \geq \mathrm{V}(\Omega, \Theta) - \epsilon \quad \textit{and} \quad \max_{f \in \text{hull}(\mathcal{F}_\Omega)} \tilde{\varphi}(f, p^*_\epsilon) \leq \mathrm{V}(\Omega, \Theta) + \epsilon\,.$$

Note that f^*_ϵ does not depend on p^*_ϵ and vice-versa. We will show that such approximate equilibria are achieved with finite convex combinations. Considering

$f_k \in \mathcal{F}_\Omega$ and $g'_k \in \mathcal{G}_\Theta$ (that can either be functions or distributions) we aim at finding the smallest convex combination that is an ϵ-equilibrium.

$(K^\Omega_\epsilon, K^\Theta_\epsilon) :=$ Smallest K and $K' \in \mathbb{N}$ s.t. $(\sum_{k=1}^K \lambda_k f_k, \sum_{k=1}^{K'} \lambda'_k g_k)$ is an ϵ-equilibrium.

Our goal is to provide a bound that depends on ϵ and on some properties of the classes $\mathcal{F}_\Omega$ and $\mathcal{G}_\Theta$.

Theorem 2. *Let φ a game that satisfies Assumption 1. If $\tilde{\varphi}$ is bilinear, $\|\theta\| \leq R$, $\|w\| \leq R$, $\forall w, \theta \in \Omega \times \Theta \subset \mathbb{R}^d \times \mathbb{R}^p$ and φ is L-Lipschitz then,*

$$K^\Omega_\epsilon \leq \frac{4D_w^2 p}{\epsilon^2} \ln(\frac{6RL}{\epsilon^2}) \quad \textit{and} \quad K^\Theta_\epsilon \leq \frac{4D_\theta^2 d}{\epsilon^2} \ln(\frac{6RL}{\epsilon^2}) \tag{5.6}$$

where $D_w := \max_{w,w',\theta} \varphi(w,\theta) - \varphi(w',\theta)$ and $D_\theta := \max_{w,\theta,\theta'} \varphi(w,\theta) - \varphi(w,\theta')$.

Roughly, the number K^Θ_ϵ expresses to what extent the set of distributions induced by the mappings in $\mathcal{G}_\Theta$ has to be 'convexifed' to achieve an approximate equilibrium. Note that in practice we expect this quantity to be small. For instance, in the context of GANs, if the class of discriminators $\mathcal{F}_\Omega$ contains the constant function $D(\cdot) = .5$ then $K^\Omega_\epsilon = 1$ since $\varphi(D, G) = 0$, $\forall G \in \mathcal{G}$.

5.3 Achieving a Mixture or an Average with a Single Neural Net

We showed above that under the assumption of Theorem 2, approximate equilibria can be achieved with finite convex combinations. In this section, we investigate how it is possible to achieve such approximate equilibria with a single ReLU network. Formally, such a function $g : \mathbb{R}^{d_{in}} \to \mathbb{R}^d$ can be written as,

$$g_\theta(x) = \sum_{i=1}^p a_i \,\mathrm{ReLU}(c_i^\top x + d_i) + b_i \quad \text{where} \quad a_i, b_i \in \mathbb{R}^d,\, c_i \in \mathbb{R}^{d_{in}},\, d_i \in \mathbb{R}\,. \tag{5.7}$$

We note $\theta = (a, b, c, d) \in \mathbb{R}^{p(d_{in}+1+2d)}$ and $\mathrm{ReLU}(x) = \max(x, 0)$. We present two results on the representative power of neural networks. The first one concerns *mixtures of distributions* represented by latent ReLU nets, and the second one concerns *convex combinations* of ReLU nets as functions.

Neural Nets Represent Mixtures of Smaller Nets. First, we get interested in the probability distributions p_θ induced by g_θ, defined as

$$a \sim p_\theta \;:\; a = g_\theta(z) \text{ where } z \sim U([0,1]) \text{ and } \theta \in \mathbb{R}^p\,. \tag{5.8}$$

One of the motivation of this work is to represent distribution over images usually represented by a high dimensional vector in $[0,1]^d$. That is why we will assume that our generator function take its value in $[0,1]^d$.

Proposition 2. *If $d_{in} = 1$ and if for all $\theta \in \Theta$, $z \in [0,1]$, $g_\theta(z) \in [0,1]^d$ and g_θ is constant outside of $[0,1]$, then for any θ_k, , $\|\theta_k\| \leq R$, $k = 1 \dots K$, there exists $\theta \in [-KR, KR]^{K(p+6)}$ such that $d_{TV}(\frac{1}{n}\sum_{k=1}^{K} p_{\theta_k}, p_\theta) \leq 1/R$ where d_{TV} is the total variation distance.*

Fig. A.1b (in §3) illustrates how g_θ is constructed. Unlike the universal approximation theorem, Prop. 2 shows that *a single neural network* can *represent* mixtures. On the one hand, when one wants to approximate an arbitrary continuous function, the number of required hidden units may be prohibitively large [Lu et al., 2017] as the error ϵ vanishes. On the other hand, the dimension of θ_3 in Prop. 2 does not depend on any vanishing quantity. The high-level insight is that a large enough ReLU net can represent mixtures of distributions induced by smaller ReLU nets, with a width that grows linearly with the size of the mixture.

Neural Nets represent an average of smaller Nets. If we consider averages of functions as described in (5.3), we can show that point-wise averages of functions g_θ , $\theta \in \mathbb{R}^p$ defined in (5.7) can be represented by a wider neural network.

Proposition 3. *For all $w_k \in [-R, R]^p$, $k = 1 \dots K$, there exists $w \in [-R, R]^{Kp}$ such that $\frac{1}{n}\sum_{k=1}^{K} f_{w_k} = f_w$.*

Figure A.1a shows how f_w is constructed. Similarly as the Prop. 2, Prop. 3 is a representation theorem that shows that the space $\{f_w \,|\, w \in \mathbb{R}^p\}$ is 'almost' convex.

5.4 Minimax Theorem for Nonconcave-noncavex Games Played with Neural Networks

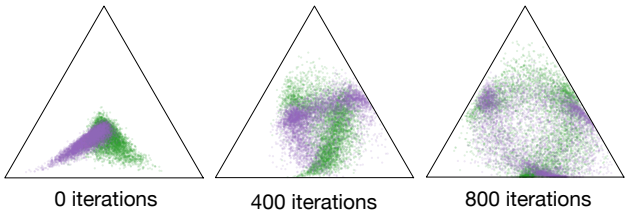
Prop. 2 and 3 give insights about the representative power of ReLU neural nets: as their width grows, ReLU nets can express larger mixtures/averages of sub-nets. Combining these properties with Thm. 2, we show that approximate equilibria can be achieved for such nonconcave-nonconvex payoff.

Theorem 1. *Let φ be a nonconcave-nonconvex game such that $\varphi(w, \theta) = \tilde{\varphi}(f_w, p_\theta)$ where p_θ is the distribution induced by g_θ defined in (5.8), and f_w and g_θ are one layer ReLU networks (5.7). If $\tilde{\varphi}$ is bilinear and $\tilde{L}$-Lipschitz, and if φ is L-Lipschitz, then for any $\epsilon > 0$, there exists $(w^*_\epsilon, \theta^*_\epsilon) \in [-R, R]^{2p}$,s.t.,*

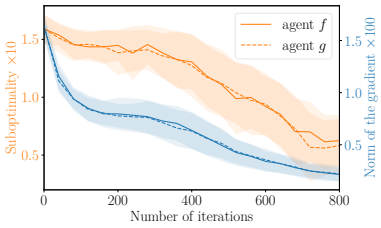
$$\min_{\substack{\theta \in \mathbb{R}^{p_\epsilon} \\ \|\theta\| \leq R_\epsilon}} \varphi(w^*_\epsilon, \theta) + \epsilon + \frac{\tilde{L}}{R} \geq \max_{\substack{w \in \mathbb{R}^{p_\epsilon} \\ \|w\| \leq R_\epsilon}} \varphi(w, \theta^*_\epsilon)\,, \tag{5.9}$$

where $p_\epsilon \geq C\epsilon\sqrt{\frac{p}{\log(R\sqrt{p}/\epsilon)}}$ and $R_\epsilon \geq R\frac{p_\epsilon}{p}$.

An explicit formula for C is provided in Appendix .3 §A as well as variants of this theorem when w parametrizes a distribution or when θ parametrizes a function. Theorem 1 shows the existence of a notion of weaker-capacity-*equilibrium*



(a) 5000 samples using the latent mapping f and g after $0, 400$, and 800 training steps. Their respective suboptimality along training has a value of $1.5, 1.2$, and .5.



(b) Performance and convergence of the agents.

Figure 4.1: Training of latent agents to play differentiable Blotto with $K = 3$. **Right:** The suboptimality corresponds to the payoff of the agent against a best response. We averaged our results over 40 random seeds.

for a nonconcave-nonconvex game where the players use a *standard fully connected architecture*. The notion of weaker-capacity is encompassed within the fact that w_ϵ and θ_ϵ are of dimension $p_\epsilon \leq p$ and are bounded in norm by $R_\epsilon \leq R$. This result differs from Arora et al. [2017, Theorem 4.3] who, only in the context of GANs, design a *specific architecture* to achieve a different notion of approximate equilibrium.

On the one hand, if $\epsilon\sqrt{p} < 1$, then the lower-bound in (5.9) is vacuous (since p_ϵ corresponds the number of non-zero parameter of the lower-capacity networks). On the other, the number of parameters of the higher-capacity networks p only needs to (roughly) grow *quadratically* with ϵ to achieve a non-vacuous bound. Hence, a consequence of Theorem 1 is that, for the nonconcave-nonconvex games of interest, *highly over parametrized networks* can provably achieve a notion of equilibrium.

6 Application: Solving Colonel Blotto Game

We apply our latent game approach (Def. 2) to solve a differentiable version of Example 3. We consider a continuous relaxation of the strategy space where $S_1 = S_2$. After renormalization we have that $A_1 = A_2 = \Delta_K$, where Δ_K is the K-dimensional simplex. It is important to notice that in that case an *allocation* corresponds to a *point on the simplex* and a *mixture of allocation* corresponds to a *distribution over the simplex*. We replace the payoff (4.5) of Latent Blotto by a differentiable one, $\varphi(w, \theta) := \mathbb{E}_{z\sim\pi,\, z'\sim\pi'}\left[\frac{1}{K}\sum_{k=1}^{K}\sigma([f_w(z) - g_\theta(z')]_k)\right]$ where σ is a sigmoid minus $1/2$ and $f_w, g_\theta : \mathbb{R}^p \to \Delta_K$. This game has been theoretically by Ferdowsi et al. [2018] when $S_1 > S_2$.

For the latent mappings, f_w anf g_θ we considered dense ReLU networks with 4 hidden layers, 16 hidden units per layer, and a K-dimensional softmax output.

We use a 16-dimensional Gaussian prior for the latent variable. We trained our agents using gradient descent ascent on the parameters of f and g with the Adam optimizer [Kingma and Ba, 2015] with $\beta_1 = .5$ and $\beta_2 = .99$.

In Fig. 4.1b, we present the performance of the agents against a best response. To compute it, we sampled 5000 strategies and computed the best response against this mixed strategy using gradient ascent on the simplex. We also computed the norm of the (stochastic) gradient used to update f. In Fig. 4.1a, we plotted samples from f at different training times. As we get closer to convergence to a non-exploitable strategy, we can see that this distribution avoids the center of the simplex (putting troops evenly on the battlefields) and the corners (focusing on a single battlefield) that are strategies easily exploitable by focusing on two battlefields, this correlates with the decrease of the gradient and of the suboptimality indicating that the agents learned how to play Blotto.

7 Discussion

Nonconcave-nonconvex games radically differ from minimization problems since equilibria may not exist in general. How, then, can neural nets regularly find meaningful solutions to games like GANs?

In this work, we partially answer this question by leveraging the structure of GANs to show that a single pair of ReLU nets can achieve a notion of limited-capacity-equilibrium. The intuition underlying our theorems is as follows: neural nets have a particular structure that interleaves matrix multiplications and simple non-linearities (often based on the max operator like ReLU). The matrix multiplications in one layer of a neural net compute linear combinations of functions encoded by the other layers. In other words, neural nets are (non-)linear mixtures of their sub-networks.

Finally, it is instructive to discuss the relative merits of the limited-capacity aspect that occurs in Theorem 1 due to the parametrization. On the one hand, if one had access to any function/distribution an infinite-capacity equilibrium would exist (because the whole function/distribution space is convex). However, this quantity may not be realistic, e.g., in GANs, the optimal infinite-capacity generator must represent the distribution of 'real-world' images. If such a concept is not tractable, it seems unrealistic to expect limited capacity agents, such as humans or computers, to find it [Papadimitriou, 2007]. On the other hand, our work shows that one can efficiently approximate some equilibria when working with neural networks. These equilibria capture the notion that agents–and humans–that play complex games have a limited capacity that seems more reasonable to play complex games such as Poker of StarCraft II that are multi-step with imperfect information. Thus in the vein of games with bounded rationality, limited-capacity equilibria seem

to be an interesting solution concept that is more realistic than infinite-capacity equilibria.

5 Prologue to the Second Contribution

1 Article Details

A Variational Inequality Perspective on Generative Adversarial Networks. *Gauthier Gidel, Hugo Berard, Gaëtan Vignoud, Pascal Vincent and Simon Lacoste-Julien.* This paper was published at ICLR 2019 [Gidel et al., 2019b].

2 Contributions of the authors

Gauthier Gidel contributed to the general writing of the paper, the results on the simple bilinear example and the proof of all the theorems of the paper. Gauthier Gidel also brought his knowledge about optimization and variational inequalities. Hugo Berard did the experiments and wrote the experimental section. He also brought his knowledge about GANs and more generally generative modeling. Gaëtan Vignoud came up with the idea of extrapolation from the past and reviewed the proofs of the paper. Simon Lacoste-Julien and Pascal Vincent supervised this project. The original idea of using extragradient and adopting a variational inequality perspective on GAN came from Simon Lacoste-Julien.

3 Modifications with respect to the published paper

We added a missing related work regarding extrapolation from the past [Popov, 1980]. We consequently modified the discussion regarding the novelty of extrapolation from the past and our related contributions.

6 A Variational Inequality Perspective on Generative Adversarial Networks

Abstract

Generative adversarial networks (GANs) form a generative modeling approach known for producing appealing samples, but they are notably difficult to train. One common way to tackle this issue has been to propose new formulations of the GAN objective. Yet, surprisingly few studies have looked at optimization methods designed for this adversarial training. In this work, we cast GAN optimization problems in the general variational inequality framework. Tapping into the mathematical programming literature, we counter some common misconceptions about the difficulties of saddle point optimization and propose to extend techniques designed for variational inequalities to the training of GANs. We apply *averaging*, *extrapolation* and a computationally cheaper variant that we call *extrapolation from the past* to the stochastic gradient method (SGD) and Adam.

1 Introduction

Generative adversarial networks (GANs) [Goodfellow et al., 2014] form a generative modeling approach known for producing realistic natural images [Karras et al., 2018] as well as high quality super-resolution [Ledig et al., 2017] and style transfer [Zhu et al., 2017]. Nevertheless, GANs are also known to be difficult to train, often displaying an unstable behavior [Goodfellow, 2016]. Much recent work has tried to tackle these training difficulties, usually by proposing new formulations of the GAN objective [Nowozin et al., 2016, Arjovsky et al., 2017]. Each of these formulations can be understood as a two-player game, in the sense of game theory [Neumann and Morgenstern, 1944], and can be addressed as a variational inequality problem (VIP) [Harker and Pang, 1990], a framework that encompasses traditional saddle point optimization algorithms [Korpelevich, 1976].

Solving such GAN games is traditionally approached by running variants of stochastic gradient descent (SGD) initially developed for optimizing supervised neural network objectives. Yet it is known that for some games [Goodfellow, 2016, §8.2] SGD exhibits oscillatory behavior and fails to converge. This oscillatory behavior, which does not arise from stochasticity, highlights a fundamental problem: while a direct application of basic gradient descent is an appropriate method for

regular minimization problems, it is *not* a sound optimization algorithm for the kind of two-player games of GANs. This constitutes a fundamental issue for GAN training, and calls for the use of more principled methods with more reassuring convergence guarantees.

Contributions. We point out that multi-player games can be cast as *variational inequality problems* (VIPs) and consequently the same applies to any GAN formulation posed as a minimax or non-zero-sum game. We present two techniques from this literature, namely *averaging* and *extrapolation*, widely used to solve VIPs but which have not been explored in the context of GANs before.[1]

We extend standard GAN training methods such as SGD or Adam into variants that incorporate these techniques (Alg. 4 is new). We also explain that the oscillations of basic SGD for GAN training previously noticed [Goodfellow, 2016] can be explained by standard variational inequality optimization results and we illustrate how *averaging* and *extrapolation* can fix this issue.

We study a variant of extragradient that we call *extrapolation from the past* originally introduced by Popov [1980]. It only requires one gradient computation per update compared to *extrapolation*, which needs to compute the gradient twice. We *prove its convergence* for strongly monotone operators and in the stochastic VIP setting.

Finally, we test these techniques in the context of GAN training. We observe a 4-6% improvement over Miyato et al. [2018] on the inception score and the Fréchet inception distance on the CIFAR-10 dataset using a WGAN-GP [Gulrajani et al., 2017] and a ResNet generator.[2]

Outline. §2 presents the background on GAN and optimization, and shows how to cast this optimization as a VIP. §3 presents standard techniques and *extrapolation from the past* to optimize variational inequalities in a batch setting. §4 considers these methods in the *stochastic* setting, yielding three corresponding variants of SGD, and provides their respective convergence rates. §5 develops how to combine these techniques with already existing algorithms. §6 discusses the related work and §7 presents experimental results.

[1]The preprints for [Mertikopoulos et al., 2019] and [Yazıcı et al., 2019], which respectively explored extrapolation and averaging for GANs, appeared after our initial preprint. See also the related work section §6.

[2]Code available at `https://gauthiergidel.github.io/projects/vip-gan.html`.

2 GAN optimization as a variational inequality problem

2.1 GAN formulations

The purpose of generative modeling is to generate samples from a distribution $q_{\boldsymbol{\theta}}$ that matches best the true distribution p of the data. The generative adversarial network training strategy can be understood as a *game* between two players called *generator* and *discriminator*. The former produces a sample that the latter has to classify between real or fake data. The final goal is to build a generator able to produce sufficiently realistic samples to fool the discriminator.

In the original GAN paper [Goodfellow et al., 2014], the GAN objective is formulated as a *zero-sum game* where the cost function of the discriminator $D_{\boldsymbol{\varphi}}$ is given by the negative log-likelihood of the binary classification task between real or fake data generated from $q_{\boldsymbol{\theta}}$ by the generator,

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \quad \text{where} \quad \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x}\sim p}[\log D_{\boldsymbol{\varphi}}(\mathbf{x})] - \mathop{\mathbb{E}}_{\mathbf{x}'\sim q_{\boldsymbol{\theta}}}[\log(1-D_{\boldsymbol{\varphi}}(\mathbf{x}'))]\,. \tag{2.1}$$

However Goodfellow et al. [2014] recommends to use in practice a second formulation, called *non-saturating GAN*. This formulation is a *non-zero-sum game* where the aim is to jointly minimize:

$$\mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x}'\sim q_{\boldsymbol{\theta}}} \log D_{\boldsymbol{\varphi}}(\mathbf{x}') \;\text{ and }\; \mathcal{L}_D(\boldsymbol{\theta}, \boldsymbol{\varphi}) := -\mathop{\mathbb{E}}_{\mathbf{x}\sim p} \log D_{\boldsymbol{\varphi}}(\mathbf{x}) - \mathop{\mathbb{E}}_{\mathbf{x}'\sim q_{\boldsymbol{\theta}}} \log(1-D_{\boldsymbol{\varphi}}(\mathbf{x}'))\,. \tag{2.2}$$

The dynamics of this formulation has the same *stationary points* as the zero-sum one (2.1) but is claimed to provide "much stronger gradients early in learning" [Goodfellow et al., 2014] .

2.2 Equilibrium

The minimax formulation (2.1) is theoretically convenient because a large literature on games studies this problem and provides guarantees on the existence of equilibria. Nevertheless, practical considerations lead the GAN literature to consider a different objective for each player as formulated in (2.2). In that case, the *two-player game problem* [Neumann and Morgenstern, 1944] consists in finding the following *Nash equilibrium*:

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \quad \text{and} \quad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi})\,. \tag{2.3}$$

Only when $\mathcal{L}_G = -\mathcal{L}_D$ is the game called a *zero-sum game* and (2.3) can be formulated as a minimax problem. One important point to notice is that the two optimization problems in (2.3) are *coupled* and have to be considered *jointly* from an optimization point of view.

Standard GAN objectives are non-convex (i.e. each cost function is non-convex), and thus such (pure) equilibria may not exist. As far as we know, not much is known about the existence of these equilibria for non-convex losses (see Heusel et al. [2017] and references therein for some results). In our theoretical analysis in §4, our assumptions (monotonicity (4.1) of the operator and convexity of the constraint set) imply the existence of an equilibrium.

In this paper, we focus on ways to optimize these games, assuming that an equilibrium exists. As is often standard in non-convex optimization, we also focus on finding points satisfying the necessary *stationary conditions*. As we mentioned previously, one difficulty that emerges in the optimization of such games is that the two different cost functions of (2.3) have to be minimized jointly in $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$. Fortunately, the optimization literature has for a long time studied so-called *variational inequality problems*, which generalize the stationary conditions for two-player game problems.

2.3 Variational inequality problem formulation

We first consider the local necessary conditions that characterize the solution of the *smooth* two-player game (2.3), defining *stationary points*, which will motivate the definition of a variational inequality. In the unconstrained setting, a *stationary point* is a couple $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ with zero gradient:

$$\|\nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,. \tag{2.4}$$

When constraints are present,[3] a *stationary point* $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ is such that the directional derivative of each cost function is non-negative in any feasible direction (i.e. there is no feasible descent direction):

$$\nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)^\top (\boldsymbol{\theta} - \boldsymbol{\theta}^*) \geq 0 \quad \text{and} \quad \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)^\top (\boldsymbol{\varphi} - \boldsymbol{\varphi}^*) \geq 0\,, \quad \forall\, (\boldsymbol{\theta}, \boldsymbol{\varphi}) \in \Theta \times \Phi. \tag{2.5}$$

Defining $\boldsymbol{\omega} := (\boldsymbol{\theta}, \boldsymbol{\varphi})$, $\boldsymbol{\omega}^* := (\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$, $\Omega := \Theta \times \Phi$, Eq. (2.5) can be compactly formulated as:

$$F(\boldsymbol{\omega}^*)^\top (\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \quad \forall \boldsymbol{\omega} \in \Omega \text{ where } F(\boldsymbol{\omega}) := \begin{bmatrix} \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\varphi}) & \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\theta}, \boldsymbol{\varphi}) \end{bmatrix}^\top. \tag{2.6}$$

These stationary conditions can be generalized to any continuous vector field: let $\Omega \subset \mathbb{R}^d$ and $F : \Omega \to \mathbb{R}^d$ be a continuous mapping. The *variational inequality problem* [Harker and Pang, 1990] (depending on F and Ω) is:

$$\text{find } \boldsymbol{\omega}^* \in \Omega \quad \text{such that} \quad F(\boldsymbol{\omega}^*)^\top (\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \quad \forall \boldsymbol{\omega} \in \Omega\,. \tag{VIP}$$

[3] An example of constraint for GANs is to clip the parameters of the discriminator [Arjovsky et al., 2017].

We call *optimal set* the set Ω^* of $\boldsymbol{\omega} \in \Omega$ verifying (VIP). The intuition behind it is that any $\boldsymbol{\omega}^* \in \Omega^*$ is a *fixed point* of the *constrained* dynamic of F (constrained to Ω).

We have thus showed that both saddle point optimization and non-zero sum game optimization, which encompass the large majority of GAN variants proposed in the literature, can be cast as VIPs. In the next section, we turn to suitable optimization techniques for such problems.

3 Optimization of Variational Inequalities (batch setting)

Let us begin by looking at techniques that were developed in the optimization literature to solve VIPs. We present the intuitions behind them as well as their performance on a simple bilinear problem (see Fig. 6.1). Our goal is to provide mathematical insights on *averaging* (§3.1) and *extrapolation* (§3.2) and propose a novel variant of the extrapolation technique that we called *extrapolation from the past* (§3.3). We consider the batch setting, i.e., the operator $F(\boldsymbol{\omega})$ defined in Eq. 2.6 yields an exact full gradient. We present extensions of these techniques to the stochastic setting later in §4.

The two standard methods studied in the VIP literature are the *gradient method* [Bruck, 1977] and the *extragradient method* [Korpelevich, 1976]. The iterates of the basic gradient method are given by $\boldsymbol{\omega}_{t+1} = P_\Omega[\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_t)]$ where $P_\Omega[\cdot]$ is the *projection onto the constraint set* (if constraints are present) associated to (VIP). These iterates are known to converge linearly under an additional assumption on the operator[4] [Chen and Rockafellar, 1997], but oscillate for a bilinear operator as shown in Fig. 6.1. On the other hand, the *uniform average* of these iterates converge for any bounded monotone operator with a $O(1/\sqrt{t})$ rate [Nedić and Ozdaglar, 2009], motivating the presentation of *averaging* in §3.1. By contrast, the *extragradient method* (extrapolated gradient) does not require any averaging to converge for monotone operators (in the batch setting), and can even converge at the faster $O(1/t)$ rate [Nesterov, 2007]. The idea of this method is to compute a lookahead step (see intuition on *extrapolation* in §3.2) in order to compute a more stable direction to follow.

3.1 Averaging

More generally, we consider a *weighted averaging* scheme with weights $\rho_t \geq 0$. This *weighted averaging* scheme have been proposed for the first time for (batch)

[4]Strong monotonicity, a generalization of strong convexity. See §1.

VIP by Bruck [1977],

$$\bar{\boldsymbol{\omega}}_T := \frac{\sum_{t=0}^{T-1} \rho_t \boldsymbol{\omega}_t}{S_T}\,, \quad S_T := \sum_{t=0}^{T-1} \rho_t\,. \tag{3.1}$$

Averaging schemes can be efficiently implemented in an online fashion noticing that,

$$\bar{\boldsymbol{\omega}}_T = (1-\tilde{\rho}_T)\bar{\boldsymbol{\omega}}_{T-1} + \tilde{\rho}_T \boldsymbol{\omega}_T \quad \text{where} \quad 0 \leq \tilde{\rho}_T \leq 1\,. \tag{3.2}$$

For instance, setting $\tilde{\rho}_T = \frac{1}{T}$ yields *uniform averaging* ($\rho_t = 1$) and $\tilde{\rho}_t = 1-\beta < 1$ yields *geometric averaging*, also known as *exponential moving averaging* ($\rho_t = \beta^{T-t}, 1 \leq t \leq T$). Averaging is experimentally compared with the other techniques presented in this section in Fig. 6.1.

In order to illustrate how averaging tackles the oscillatory behavior in game optimization, we consider a toy example where the discriminator and the generator are linear: $D_{\boldsymbol{\varphi}}(\mathbf{x}) = \boldsymbol{\varphi}^T\mathbf{x}$ and $G_{\boldsymbol{\theta}}(\mathbf{z}) = \boldsymbol{\theta}\mathbf{z}$ (implicitly defining $q_{\boldsymbol{\theta}}$). By substituting these expressions in the WGAN objective,[5] we get the following bilinear objective:

$$\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\varphi}\in\Phi, ||\boldsymbol{\varphi}||\leq 1} \boldsymbol{\varphi}^T\mathbb{E}[\mathbf{x}] - \boldsymbol{\varphi}^T\boldsymbol{\theta}\mathbb{E}[\mathbf{z}]\,. \tag{3.3}$$

A similar task was presented by Nagarajan and Kolter [2017] where they consider a quadratic discriminator instead of a linear one, and show that gradient descent is not necessarily asymptotically stable. The bilinear objective has been extensively used [Goodfellow, 2016, Mescheder et al., 2018, Yadav et al., 2018, Daskalakis et al., 2018] to highlight the difficulties of gradient descent for saddle point optimization. Yet, ways to cope with this issue have been proposed decades ago in the context of mathematical programming. For illustrating the properties of the methods of interest, we will study their behavior in the rest of §3 on a simple *unconstrained* unidimensional version of Eq. 3.3 (this behavior can be generalized to general multidimensional bilinear examples, see §2.3):

$$\min_{\theta\in\mathbb{R}} \max_{\phi\in\mathbb{R}} \; \theta\cdot\phi \qquad \text{and} \qquad (\theta^*, \phi^*) = (0,0)\,. \tag{3.4}$$

The operator associated with this minimax game is $F(\theta,\phi) = (\phi, -\theta)$. There are several ways to compute the discrete updates of this dynamics. The two most common ones are the *simultaneous* and the *alternating* gradient update rules,

$$\text{Sim. update:} \begin{cases} \theta_{t+1} = \theta_t - \eta\phi_t \\ \phi_{t+1} = \phi_t + \eta\theta_t \end{cases}, \quad \text{Alt. update:} \begin{cases} \theta_{t+1} = \theta_t - \eta\phi_t \\ \phi_{t+1} = \phi_t + \eta\theta_{t+1} \end{cases}. \tag{3.5}$$

Interestingly, these two choices give rise to completely different behaviors. The norm of the *simultaneous* updates diverges geometrically, whereas the alternating

[5] Wasserstein GAN (WGAN) proposed by Arjovsky et al. [2017] boils down to the following minimax formulation: $\min_{\boldsymbol{\theta}\in\Theta} \max_{\boldsymbol{\varphi}\in\Phi, ||D_{\boldsymbol{\varphi}}||_L\leq 1} \mathbb{E}_{\mathbf{x}\sim p}[D_{\boldsymbol{\varphi}}(\mathbf{x})] - \mathbb{E}_{\mathbf{x}'\sim q_{\boldsymbol{\theta}}}[D_{\boldsymbol{\varphi}}(\mathbf{x}')]$.

iterates are bounded but do not converge to the equilibrium. As a consequence, their respective uniform average have a different behavior, as highlighted in the following proposition (proof in Appendix B §2.1 and generalization in Appendix B, §2.3):

Proposition 1. *The* simultaneous *iterates diverge geometrically and the* alternating *iterates defined in* (3.5) *are bounded but do not converge to 0 as*

$$\text{Simultaneous: } \theta_{t+1}^2+\phi_{t+1}^2=(1+\eta^2)(\theta_t^2+\phi_t^2)\,, \quad \text{Alternating: } \theta_t^2+\phi_t^2=\Theta(\theta_0^2+\phi_0^2) \tag{3.6}$$

where $u_t=\Theta(v_t) \Leftrightarrow \exists \alpha,\beta,t_0>0$ *such that* $\forall t\geq t_0, \alpha v_t\leq u_t\leq \beta v_t$.

The uniform average $(\bar{\theta}_t,\bar{\phi}_t) := \frac{1}{t}\sum_{s=0}^{t-1}(\theta_s,\phi_s)$ *of the* simultaneous *updates (resp. the* alternating updates*) diverges (resp. converges to 0) as,*

$$\text{Sim.: } \bar{\theta}_t^2+\bar{\phi}_t^2=\Theta\left(\frac{\theta_0^2+\phi_0^2}{\eta^2 t^2}(1+\eta^2)^t\right), \quad \text{Alt.: } \bar{\theta}_t^2+\bar{\phi}_t^2=\Theta\left(\frac{\theta_0^2+\phi_0^2}{\eta^2 t^2}\right). \tag{3.7}$$

This sublinear convergence result, proved in Appendix B §2, underlines the benefits of averaging when the sequence of iterates is bounded (i.e. for *alternating* update rule). When the sequence of iterates is not bounded (i.e. for *simultaneous* updates) averaging fails to ensure convergence. This theorem also shows how *alternating* updates may have better convergence properties than *simultaneous* updates.

3.2 Extrapolation

Another technique used in the variational inequality literature to prevent oscillations is *extrapolation*. This concept is anterior to the extragradient method since Korpelevich [1976] mentions that the idea of *extrapolated* "prices" to give "stability" had been already formulated by Polyak [1963, Chap. II]. The idea behind this technique is to compute the gradient at an (extrapolated) point different from the current point from which the update is performed, stabilizing the dynamics:

$$\text{Compute extrapolated point: } \boldsymbol{\omega}_{t+1/2}=P_\Omega[\boldsymbol{\omega}_t-\eta F(\boldsymbol{\omega}_t)]\,, \tag{3.8}$$

$$\text{Perform update step: } \boldsymbol{\omega}_{t+1}=P_\Omega[\boldsymbol{\omega}_t-\eta F(\boldsymbol{\omega}_{t+1/2})]\,. \tag{3.9}$$

Note that, even in the *unconstrained case*, this method is intrinsically different from Nesterov's momentum[6] [Nesterov, 2004, Eq. 2.2.9] because of this lookahead step for the gradient computation:

$$\text{Nesterov's method: } \begin{aligned}\boldsymbol{\omega}_{t+1/2}&=\boldsymbol{\omega}_t-\eta F(\boldsymbol{\omega}_t)\,,\\ \boldsymbol{\omega}_{t+1}&=\boldsymbol{\omega}_{t+1/2}+\beta(\boldsymbol{\omega}_{t+1/2}-\boldsymbol{\omega}_t)\,.\end{aligned}$$

[6] Sutskever [2013, §7.2] showed the equivalence between "standard momentum" and Nesterov's formulation.

Nesterov's method does not converge when trying to optimize (3.4). One intuition of why *extrapolation* has better convergence properties than the standard gradient method comes from Euler's integration framework. Indeed, to first order, we have $\boldsymbol{\omega}_{t+1/2} \approx \boldsymbol{\omega}_{t+1} + o(\eta)$ and consequently, the update step (3.9) can be interpreted as a first order approximation to an *implicit method* step:

$$\text{Implicit step:} \quad \boldsymbol{\omega}_{t+1} = \boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_{t+1})\,. \tag{3.10}$$

Implicit methods are known to be more stable and to benefit from better convergence properties [Atkinson, 2003] than *explicit methods*, e.g., in §2.2 we show that (3.10) on (3.4) converges *for any* η. Though, they are usually not practical since they require to solve a potentially non-linear system at each step. Going back to the simplified WGAN toy example (3.4) from §3.1, we get the following update rules:

$$\text{Implicit:} \begin{cases} \theta_{t+1} = \theta_t - \eta\phi_{t+1} \\ \phi_{t+1} = \phi_t + \eta\theta_{t+1} \end{cases}, \text{ Extrapolation:} \begin{cases} \theta_{t+1} = \theta_t - \eta(\phi_t + \eta\theta_t) \\ \phi_{t+1} = \phi_t + \eta(\theta_t - \eta\phi_t) \end{cases}. \tag{3.11}$$

In the following proposition, we see that for $\eta < 1$, the respective convergence rates of the *implicit method* and *extrapolation* are highly similar. Keeping in mind that the latter has the major advantage of being more practical, this proposition clearly underlines the benefits of *extrapolation*. Note that Prop. 1 and 2 generalize to general unconstrained bilinear game (more details and proof in §2.3),

Proposition 2. *The squared norm of the iterates* $N_t^2 := \theta_t^2 + \phi_t^2$*, where the update rule of* θ_t *and* ϕ_t *are defined in* (3.11)*, decreases geometrically for any* $\eta < 1$ *as,*

$$\text{Implicit: } N_{t+1}^2 = \left(1 - \eta^2 + \eta^4 + \mathcal{O}(\eta^6)\right) N_t^2\,, \text{ Extrapolation: } N_{t+1}^2 = (1 - \eta^2 + \eta^4) N_t^2\,. \tag{3.12}$$

3.3 Extrapolation from the past

One issue with extrapolation is that the algorithm "wastes" a gradient (3.8). Indeed we need to compute the gradient at two different positions for every single update of the parameters. [Popov, 1980] proposed a similar technique that only requires a single gradient computation per update. The idea is to store and re-use the extrapolated gradient for the extrapolation:

$$\text{Extrapolation from the past:} \quad \boldsymbol{\omega}_{t+1/2} = P_\Omega[\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_{t-1/2})] \tag{3.13}$$

$$\text{Perform update step:} \quad \boldsymbol{\omega}_{t+1} = P_\Omega[\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_{t+1/2})] \tag{3.14}$$

$$\text{and store:} \quad F(\boldsymbol{\omega}_{t+1/2})\,.$$

A similar update scheme was proposed by Chiang et al. [2012, Alg. 1] in the context of online convex optimization and generalized by Rakhlin and Sridharan

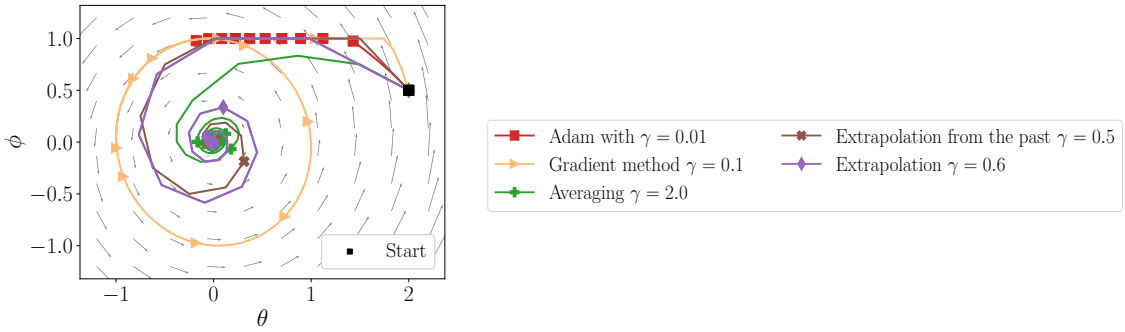


Figure 6.1: Comparison of the basic gradient method (as well as Adam) with the techniques presented in §3 on the optimization of (3.3). Only the algorithms advocated in this paper (Averaging, Extrapolation and Extrapolation from the past) converge quickly to the solution. Each marker represents 20 iterations. We compare these algorithms on a non-convex objective in §7.1.

[2013] for general online learning. Without projection, (3.13) and (3.14) reduce to the optimistic mirror descent described by Daskalakis et al. [2018]:

$$\text{Optimistic mirror descent (OMD):} \quad \boldsymbol{\omega}_{t+1/2} = \boldsymbol{\omega}_{t-1/2} - 2\eta F(\boldsymbol{\omega}_{t-1/2}) + \eta F(\boldsymbol{\omega}_{t-3/2}) \tag{3.15}$$

OMD was proposed with similar motivation as ours, namely tackling oscillations due to the game formulation in GAN training, but with an online learning perspective. Using the VIP point of view, we are able to prove a linear convergence rate for *extrapolation from the past* (see details and proof of Theorem 1 in §2.4). We also provide results on the averaged iterate for a stochastic version in §4. In comparison to the convergence results from Daskalakis et al. [2018] that hold for a bilinear objective, we provide a faster convergence rate (linear vs sublinear) on the last iterate for a general (strongly monotone) operator F and any projection on a convex Ω. One thing to notice is that the operator of a bilinear objective is *not* strongly monotone, but in that case one can use the standard extrapolation method (3.8) which converges linearly for an unconstrained bilinear game [Tseng, 1995, Cor. 3.3].

Theorem 1 (Linear convergence of *extrapolation from the past*)**.** *If F is μ-strongly monotone (see Appendix B §1 for the definition of strong monotonicity) and L-Lipschitz, then the updates* (3.13) *and* (3.14) *with $\eta = \frac{1}{4L}$ provide linearly converging iterates,*

$$\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)^t \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|_2^2, \quad \forall t \geq 0\,. \tag{3.16}$$

Algorithm 1 AvgSGD

Let $\boldsymbol{\omega}_0 \in \Omega$
for $t = 0 \dots T-1$ **do**
  $\xi_t \sim P$ *(mini-batch)*
  $\boldsymbol{d}_t \leftarrow F(\boldsymbol{\omega}_t, \xi_t)$
  $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta_t \boldsymbol{d}_t]$
end for
Return $\bar{\boldsymbol{\omega}}_T \leftarrow \frac{\sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}_t}{\sum_{t=0}^{T-1} \eta_t}$

Algorithm 2 AvgExtraSGD

for $t = 0 \dots T-1$ **do**
  $\xi_t, \xi'_t \sim P$
  $\boldsymbol{d}_t \leftarrow F(\boldsymbol{\omega}_t, \xi_t)$
  $\boldsymbol{\omega}'_t \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta_t \boldsymbol{d}_t]$
  $\boldsymbol{d}'_t \leftarrow F(\boldsymbol{\omega}'_t, \xi'_t)$
  $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta_t \boldsymbol{d}'_t]$
end for
Return $\bar{\boldsymbol{\omega}}_T \leftarrow \frac{\sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}'_t}{\sum_{t=0}^{T-1} \eta_t}$

Algorithm 3 AvgPastExtraSGD

Let $\boldsymbol{\omega}_0 \in \Omega$
for $t = 0 \dots T-1$ **do**
  $\xi_t \sim P$ *(mini-batch)*
  $\boldsymbol{\omega}'_t \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta_t \boldsymbol{d}_{t-1}]$
  $\boldsymbol{d}_t \leftarrow F(\boldsymbol{\omega}'_t, \xi_t)$
  $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta_t \boldsymbol{d}_t]$
end for
Return $\bar{\boldsymbol{\omega}}_T \leftarrow \frac{\sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}'_t}{\sum_{t=0}^{T-1} \eta_t}$

Figure 6.2: Three variants of SGD computing T updates, using the techniques introduced in §3.

4 Optimization of VIP with stochastic gradients

In this section, we consider extensions of the techniques presented in §3 to the context of a *stochastic* operator, i.e., we no longer have access to the exact gradient $F(\boldsymbol{\omega})$ but to an unbiased *stochastic* estimate of it, $F(\boldsymbol{\omega}, \xi)$, where $\xi \sim P$ and $F(\boldsymbol{\omega}) := \mathbb{E}_{\xi \sim P}[F(\boldsymbol{\omega}, \xi)]$. It is motivated by GAN training where we only have access to a finite sample estimate of the expected gradient, computed on a mini-batch. For GANs, ξ is a mini-batch of points coming from the true data distribution p and the generator distribution $q_{\boldsymbol{\theta}}$.

For our analysis, we require at least one of the two following assumptions on the stochastic operator:

Assumption 2. *Bounded variance by σ^2:* $\mathbb{E}_\xi[\|F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}, \xi)\|^2] \leq \sigma^2\,, \quad \forall \boldsymbol{\omega} \in \Omega\,.$

Assumption 3. *Bounded expected squared norm:* $\mathbb{E}_\xi[\|F(\boldsymbol{\omega}, \xi)\|^2] \leq M^2, \ \forall \boldsymbol{\omega} \in \Omega.$

Assump. 2 is standard in stochastic variational analysis, while Assump. 3 is a stronger assumption sometimes made in stochastic convex optimization. To illustrate how strong Assump. 3 is, note that it does not hold for an unconstrained bilinear objective like in our example (3.4) in §3. It is thus mainly reasonable for bounded constraint sets. Note that in practice we have $\sigma \ll M$.

We now present and analyze three algorithms that are variants of SGD that are appropriate to solve (VIP). The first one Alg. 1 (AvgSGD) is the stochastic extension of the gradient method for solving (VIP); Alg. 2 (AvgExtraSGD) uses *extrapolation* and Alg. 3 (AvgPastExtraSGD) uses *extrapolation from the past*. A fourth variant that re-use the mini-batch for the extrapolation step (ReExtraSGD, Alg. 5) is described in §4. These four algorithms return an *average* of the iterates

(typical in stochastic setting). The proofs of the theorems presented in this section are in Appendix B §6.

To handle constraints such as parameter clipping [Arjovsky et al., 2017], we gave a *projected* version of these algorithms, where $P_\Omega[\boldsymbol{\omega}']$ denotes the projection of $\boldsymbol{\omega}'$ onto Ω (see Appendix B §1). Note that when $\Omega = \mathbb{R}^d$, the projection is the identity mapping (unconstrained setting). In order to prove the convergence of these four algorithms, we will assume that F is monotone:

$$(F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}'))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') \geq 0 \quad \forall\, \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega\,. \tag{4.1}$$

If F can be written as (2.6), it implies that the cost functions are convex.[7] Note however that general GANs parametrized with neural networks lead to non-monotone VIPs.

Assumption 4. *F is* monotone *and Ω is a compact convex set, such that* $\max_{\boldsymbol{\omega},\boldsymbol{\omega}'\in\Omega} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|^2 \leq R^2$.

In that setting the quantity $g(\boldsymbol{\omega}^*) := \max_{\boldsymbol{\omega}\in\Omega} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}^* - \boldsymbol{\omega})$ is well defined and is equal to 0 if and only if $\boldsymbol{\omega}^*$ is a solution of (VIP). Moreover, if we are optimizing a *zero-sum game*, we have $\boldsymbol{\omega} = (\boldsymbol{\theta}, \boldsymbol{\varphi})$, $\Omega = \Theta \times \Phi$ and $F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = [\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \quad -\nabla_{\boldsymbol{\varphi}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})]^\top$. Hence, the quantity $h(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) := \max_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi}) - \min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*)$ is well defined and equal to 0 if and only if $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ is a *Nash equilibrium* of the game. The two functions g and h are called *merit functions* (more details on the concept of *merit functions* in §3). In the following, we call,

$$\mathrm{Err}(\boldsymbol{\omega}) := \begin{cases} \max\limits_{(\boldsymbol{\theta}',\boldsymbol{\varphi}')\in\Omega} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}') - \mathcal{L}(\boldsymbol{\theta}', \boldsymbol{\varphi}) & \text{if } F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = [\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \quad -\nabla_{\boldsymbol{\varphi}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})]^\top \\ \max\limits_{\boldsymbol{\omega}'\in\Omega} F(\boldsymbol{\omega}')^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') & \text{otherwise.} \end{cases} \tag{4.2}$$

Averaging. Alg. 1 (AvgSGD) presents the stochastic gradient method with *averaging*, which reduces to the standard (simultaneous) SGD updates for the two-player games used in the GAN literature, but returning an *average* of the iterates.

Theorem 2. *Under Assump. 2, 3 and 4, SGD with averaging (Alg. 1) with a constant step-size gives,*

$$\mathbb{E}[\mathrm{Err}(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{2\eta T} + \eta\frac{M^2 + \sigma^2}{2} \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\omega}_t\,, \quad \forall T \geq 1\,. \tag{4.3}$$

Thm. 2 uses a similar proof as [Nemirovski et al., 2009]. The constant term $\eta(M^2 + \sigma^2)/2$ in (4.3) is called the *variance term*. This type of bound is standard

[7] The convexity of the cost functions in (2.3) is a necessary condition (not sufficient) for the operator to be monotone. In the context of a zero-sum game, the convexity of the cost functions is a sufficient condition.

in stochastic optimization. We also provide in Appendix B §6 a similar $\tilde{O}(1/\sqrt{t})$ rate with an extra log factor when $\eta_t = \frac{\eta}{\sqrt{t}}$. We show that this variance term is smaller than the one of *SGD with prediction method* [Yadav et al., 2018] in §5.

Extrapolations. Alg. 2 (AvgExtraSGD) adds an extrapolation step compared to Alg. 1 in order to reduce the oscillations due to the game between the two players. A theoretical consequence is that it has a smaller variance term than (4.3). As discussed previously, Assump. 3 made in Thm. 2 for the convergence of Alg. 1 is very strong in the unbounded setting. One advantage of SGD with *extrapolation* is that Thm. 3 does not require this assumption.

Theorem 3. *[Juditsky et al., 2011, Thm. 1] Under Assump. 2 and 4, if $\mathbb{E}_\xi[F]$ is L-Lipschitz, then SGD with* extrapolation and averaging *(Alg. 2) using a constant step-size $\eta \leq \frac{1}{\sqrt{3}L}$ gives,*

$$\mathbb{E}[\mathrm{Err}(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \frac{7}{2}\eta\sigma^2 \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\omega}'_t\,, \quad \forall T \geq 1\,. \tag{4.4}$$

Since in practice $\sigma \ll M$, the variance term in (4.4) is significantly smaller than the one in (4.3). To summarize, SGD with *extrapolation* provides better convergence guarantees but requires two gradient computations and samples per iteration. This motivates our new method, Alg. 3 (AvgPastExtraSGD) which uses *extrapolation from the past* and achieves *the best of both worlds* (in theory).

Theorem 4. *Under Assump. 2 and 4, if $\mathbb{E}_\xi[F]$ is L-Lipschitz then SGD with* extrapolation from the past *using a constant step-size $\eta \leq \frac{1}{2\sqrt{3}L}$, gives that the averaged iterates converge as,*

$$\mathbb{E}[\mathrm{Err}(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \frac{13}{2}\eta\sigma^2 \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\omega}'_t \quad \forall T \geq 1\,. \tag{4.5}$$

The bound is similar to the one provided in Thm. 3 but each iteration of Alg. 3 is computationally half the cost of an iteration of Alg. 2.

5 Combining the techniques with established algorithms

In the previous sections, we presented several techniques that converge for stochastic monotone operators. These techniques can be combined in practice with existing algorithms. We propose to combine them to two standard algorithms used for training deep neural networks: the Adam optimizer [Kingma and Ba, 2015] and

the SGD optimizer [Robbins and Monro, 1951]. For the Adam optimizer, there are several possible choices on how to update the moments. This choice can lead to different algorithms in practice: for example, even in the unconstrained case, our proposed Adam with extrapolation from the past (Alg. 4) is different from Optimistic Adam [Daskalakis et al., 2018] (the moments are updated differently). Note that in the case of a two-player game (2.3), the previous convergence results can be generalized to gradient updates with a different step-size for each player by simply rescaling the objectives $\mathcal{L}_G$ and $\mathcal{L}_D$ by a different scaling factor. A detailed pseudo-code for Adam with extrapolation step (Extra-Adam) is given in Algorithm 4. Note that our interest regarding this algorithm is practical and that we do not provide any convergence proof.

Algorithm 4 Extra-Adam: proposed Adam with extrapolation step.

input: step-size η, decay rates for moment estimates β_1, β_2, access to the stochastic gradients $\nabla \ell_t(\cdot)$ and to the projection $P_\Omega[\cdot]$ onto the constraint set Ω, initial parameter $\boldsymbol{\omega}_0$, averaging scheme $(\rho_t)_{t\geq 1}$

for $t = 0 \dots T-1$ **do**

Option 1: Standard extrapolation.

Sample new mini-batch and compute stochastic gradient: $g_t \leftarrow \nabla \ell_t(\boldsymbol{\omega}_t)$

Option 2: Extrapolation from the past

Load previously saved stochastic gradient: $g_t = \nabla \ell_{t-1/2}(\boldsymbol{\omega}_{t-1/2})$

Update estimate of first moment for extrapolation: $m_{t-1/2} \leftarrow \beta_1 m_{t-1} + (1-\beta_1) g_t$

Update estimate of second moment for extrapolation: $v_{t-1/2} \leftarrow \beta_2 v_{t-1} + (1-\beta_2) g_t^2$

Correct the bias for the moments: $\hat{m}_{t-1/2} \leftarrow m_{t-1/2}/(1-\beta_1^{2t-1})$, $\hat{v}_{t-1/2} \leftarrow v_{t-1/2}/(1-\beta_2^{2t-1})$

Perform *extrapolation* step from iterate at time t: $\boldsymbol{\omega}_{t-1/2} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_{t-1/2}}{\sqrt{\hat{v}_{t-1/2}}+\epsilon}]$

Sample new mini-batch and compute stochastic gradient: $g_{t+1/2} \leftarrow \nabla \ell_{t+1/2}(\boldsymbol{\omega}_{t+1/2})$

Update estimate of first moment: $m_t \leftarrow \beta_1 m_{t-1/2} + (1-\beta_1) g_{t+1/2}$

Update estimate of second moment: $v_t \leftarrow \beta_2 v_{t-1/2} + (1-\beta_2) g_{t+1/2}^2$

Compute bias corrected for first and second moment: $\hat{m}_t \leftarrow m_t/(1-\beta_1^{2t})$, $\hat{v}_t \leftarrow v_t/(1-\beta_2^{2t})$

Perform *update* step from the iterate at time t: $\boldsymbol{\omega}_{t+1} \leftarrow P_\Omega[\boldsymbol{\omega}_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t}+\epsilon}]$

end for

Output: $\boldsymbol{\omega}_{T-1/2}$, $\boldsymbol{\omega}_T$ or $\bar{\boldsymbol{\omega}}_T = \sum_{t=0}^{T-1} \rho_{t+1} \boldsymbol{\omega}_{t+1/2} / \sum_{t=0}^{T-1} \rho_{t+1}$ (see (3.2) for online averaging)

6 Related Work

The extragradient method is a standard algorithm to optimize variational inequalities. This algorithm has been originally introduced by Korpelevich [1976] and extended by Nesterov [2007] and Nemirovski [2004]. Stochastic versions of the extragradient have been recently analyzed [Juditsky et al., 2011, Yousefian et al., 2014, Iusem et al., 2017] for stochastic variational inequalities with *bounded constraints*. A linearly convergent variance reduced version of the stochastic gradient method has been proposed by Palaniappan and Bach [2016] for strongly monotone variational inequalities. Extrapolation can also be related to *optimistic methods* [Chiang et al., 2012, Rakhlin and Sridharan, 2013] proposed in the online learning literature (see more details in §3.3). Interesting non-convex results were proved, for a new notion of regret minimization, by Hazan et al. [2017] and in the context of online learning for GANs by Grnarova et al. [2018].

Several methods to stabilize GANs consist in transforming a zero-sum formulation into a more general game that can no longer be cast as a saddle point problem. This is the case of the *non-saturating* formulation of GANs [Goodfellow et al., 2014, Fedus et al., 2018], the DCGANs [Radford et al., 2016], the *gradient penalty*[8] for WGANs [Gulrajani et al., 2017]. Yadav et al. [2018] propose an optimization method for GANs based on AltSGD using an additional momentum-based step on the generator. Daskalakis et al. [2018] proposed a method inspired from game theory. Li et al. [2017] suggest to dualize the GAN objective to reformulate it as a maximization problem and Mescheder et al. [2017] propose to add the norm of the gradient in the objective to get a better signal. Gidel et al. [2019c] analyzed a generalization of the bilinear example (3.3) with a focus put on the effect of momentum on this problem. They do not consider extrapolation (see §2.3 for more details). *Unrolling* steps [Metz et al., 2017] can be confused with extrapolation but is fundamentally different: the perspective is to try to approximate the "true generator objective function" unrolling for K steps the updates of the discriminator and then updating the generator.

Regarding the averaging technique, some recent work appear to have already successfully used *geometric averaging* (3.1) for GANs in practice, but only briefly mention it [Karras et al., 2018, Mescheder et al., 2018]. By contrast, the present work formally motivates and justifies the use of averaging for GANs by relating them to the VIP perspective, and sheds light on its underlying intuitions in §3.1. Subsequent to our first preprint, Yazıcı et al. [2019] explored averaging empirically in more depth, while Mertikopoulos et al. [2019] also investigated extrapolation, providing asymptotic convergence results (i.e. without any rate of convergence) in the context of *coherent saddle point*. The coherence assumption is slightly weaker than monotonicity.

[8] The gradient penalty is only added to the discriminator cost function. Since this gradient penalty depends also on the generator, WGAN-GP is a non-zero sum game.

7 Experiments

Our goal in this experimental section is not to provide new state-of-the art results with architectural improvements or a new GAN formulation, but to show that using the *techniques* (with theoretical guarantees in the monotone case) that we introduced earlier allows us to optimize standard GANs in a better way. These techniques, which are orthogonal to the design of new formulations of GAN optimization objectives, and to architectural choices, can potentially be used for the training of any type of GAN. We will compare the following optimization algorithms: baselines are SGD and Adam using either simultaneous updates on the generator and on the discriminator (denoted **SimAdam** and **SimSGD**) or k updates on the discriminator alternating with 1 update on the generator (denoted **AltSGD**$\{k\}$ and **AltAdam**$\{k\}$).[9] Variants that use *extrapolation* are denoted **ExtraSGD** (Alg. 2) and **ExtraAdam** (Alg. 4). Variants using *extrapolation from the past* are **PastExtraSGD** (Alg. 3) and **PastExtraAdam** (Alg. 4). We also present results using as output the *averaged* iterates, adding **Avg** as a prefix of the algorithm name when we use (uniform) *averaging*.

7.1 Bilinear saddle point (stochastic)

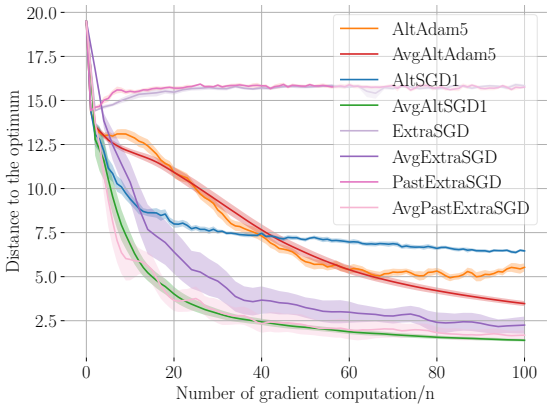


Figure 6.3: Performance of the considered stochastic optimization algorithms on the bilinear problem (7.1). Each method uses its respective optimal step-size found by grid-search.

We first test the various stochastic algorithms on a simple ($n = 10^3, d = 10^3$)

[9] In the original WGAN paper [Arjovsky et al., 2017], the authors use $k = 5$.

finite sum bilinear objective (a monotone operator) constrained to $[-1, 1]^d$:

$$\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{\theta}^\top \boldsymbol{M}^{(i)}\boldsymbol{\varphi} + \boldsymbol{\theta}^\top \boldsymbol{a}^{(i)} + \boldsymbol{\varphi}^\top \boldsymbol{b}^{(i)}\right) \tag{7.1}$$

$$\text{solved by } (\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \text{ s.t. } \begin{cases} \bar{\boldsymbol{M}}\boldsymbol{\varphi}^* = -\bar{\boldsymbol{a}} \\ \bar{\boldsymbol{M}}^T\boldsymbol{\theta}^* = -\bar{\boldsymbol{b}} \end{cases},$$

where $\bar{\boldsymbol{a}} := \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{a}^{(i)}$, $\bar{\boldsymbol{b}} := \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{b}^{(i)}$ and $\bar{\boldsymbol{M}} := \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{M}^{(i)}$. The matrices $\boldsymbol{M}^{(i)}_{kj}$, $\boldsymbol{a}^{(i)}_k$, $\boldsymbol{b}^{(i)}_k$; $1 \leq i \leq n$, $1 \leq j, k \leq d$ were randomly generated, but ensuring that $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ belongs to $[-1, 1]^d$. Results are shown in Fig. 6.3. We can see that AvgAltSGD1 and AvgPastExtraSGD perform the best on this task.

7.2 WGAN and WGAN-GP on CIFAR10

We evaluate the proposed techniques in the context of GAN training, which is a challenging stochastic optimization problem where the objectives of both players are non-convex. We propose to evaluate the Adam variants of the different optimization algorithms (see Alg. 4 for Adam with *extrapolation*) by training two different architectures on the CIFAR10 dataset [Krizhevsky and Hinton, 2009]. First, we consider a constrained zero-sum game by training the DCGAN architecture [Radford et al., 2016] with the WGAN objective and weight clipping as proposed by Arjovsky et al. [2017]. Then, we compare the different methods on a state-of-the-art architecture by training a ResNet with the WGAN-GP objective similar to Gulrajani et al. [2017]. Models are evaluated using the inception score (IS) [Salimans et al., 2016] computed on 50,000 samples. We also provide the FID [Heusel et al., 2017] and the details on the ResNet architecture in §7.3.

For each algorithm, we did an extensive search over the hyperparameters of Adam. We fixed $\beta_1 = 0.5$ and $\beta_2 = 0.9$ for all methods as they seemed to perform well. We note that as proposed by Heusel et al. [2017], it is quite important to set different learning rates for the generator and discriminator. Experiments were run with 5 random seeds for 500,000 updates of the generator.

Tab. 6.1 reports the best IS achieved on these problems by each considered method. We see that the techniques of *extrapolation* and *averaging* consistently enable improvements over the baselines (see Appendix B §7.5 for more experiments on *averaging*). Fig. 6.4 shows training curves for each method (for their best performing learning rate), as well as samples from a ResNet generator trained with ExtraAdam on a WGAN-GP objective. For both tasks, using an *extrapolation step* and averaging with Adam (ExtraAdam) outperformed all other methods. Combining ExtraAdam with averaging yields results that improve significantly over the previous state-of-the-art IS (8.2) and FID (21.7) on CIFAR10 as reported by Miyato et al. [2018] (see Tab. B.4 for FID). We also observed that methods based on *extrapolation* are less sensitive to learning rate tuning and can be used with higher learning rates with less degradation; see §7.4 for more details.

Model	WGAN (DCGAN)			WGAN-GP (ResNet)		
Method	no avg	uniform avg	EMA	no avg	uniform avg	EMA
SimAdam	*6.05 ± .12*	5.85 ± .16	6.08 ± .10	*7.51 ± .17*	7.68 ± .43	7.60 ± .17
AltAdam5	*5.45 ± .08*	5.72 ± .06	5.49 ± .05	*7.57 ± .02*	8.01 ± .05	7.66 ± .03
ExtraAdam	**6.38 ± .09**	**6.38 ± .20**	**6.37 ± .08**	7.90 ± .11	**8.47 ± .10**	8.13 ± .07
PastExtraAdam	5.98 ± .15	6.07 ± .19	6.01 ± .11	7.84 ± .06	8.01 ± .09	7.99 ± .03
OptimAdam	*5.74 ± .10*	5.80 ± .08	5.78 ± .05	*7.98 ± .08*	8.18 ± .09	8.10 ± .06

Table 6.1: Best inception scores (averaged over 5 runs) achieved on CIFAR10 for every considered Adam variant. OptimAdam is the related *Optimistic Adam* [Daskalakis et al., 2018] algorithm. EMA denotes *exponential moving average* (with $\beta = 0.9999$, see Eq. 3.2). We see that the techniques of extrapolation and averaging consistently enable improvements over the baselines (in italic).

8 Conclusion

We newly addressed GAN objectives in the framework of variational inequality. We tapped into the optimization literature to provide more principled techniques to optimize such games. We leveraged these techniques to develop practical optimization algorithms suitable for a wide range of GAN training objectives (including non-zero sum games and projections onto constraints). We experimentally verified that this could yield better trained models, improving the previous state of the art. The presented techniques address a fundamental problem in GAN training in a principled way, and are orthogonal to the design of new GAN architectures and objectives. They are thus likely to be widely applicable, and benefit future development of GANs.

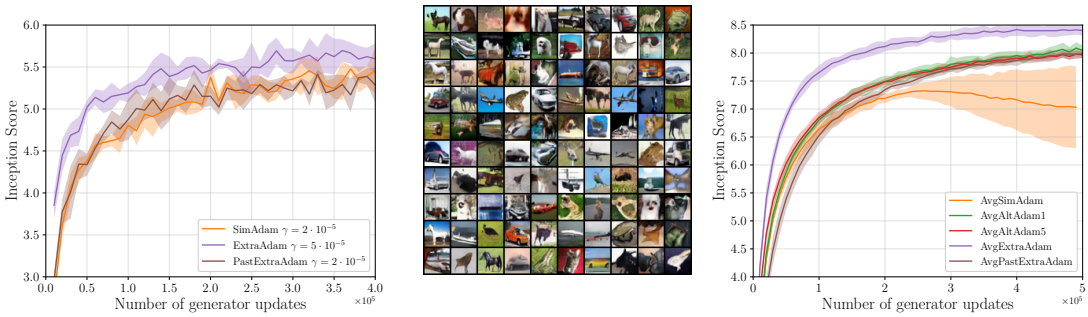


Figure 6.4: **Left:** Mean and standard deviation of the inception score computed over 5 runs for each method on WGAN trained on CIFAR10. To keep the graph readable we show only SimAdam but AltAdam performs similarly. **Middle:** Samples from a ResNet generator trained with the WGAN-GP objective using AvgExtraAdam. **Right:** WGAN-GP trained on CIFAR10: mean and standard deviation of the inception score computed over 5 runs for each method using the best performing learning rates; all experiments were run on a NVIDIA Quadro GP100 GPU. We see that ExtraAdam converges faster than the Adam baselines.

7 Prologue to the Third Contribution

1 Article Details

Negative Momentum for Improved Game Dynamics. *Gauthier Gidel**, *Reyhane Askari Hemmat**, *Mohammad Pezeshki, Rémi Le Priol, Gabriel Huang, Simon Lacoste-Julien and Ioannis Mitliagkas.* This paper was published at AISTATS 2019 [Gidel et al., 2019c].

*Equal contribution.

2 Contributions of the authors

Gauthier Gidel contributed to the general writing of the paper, the idea of all the theorems in the paper and their respective proofs and the idea of Figure 4.2 and 4.3. Reyhane Askari lead the project on the experimental part and realized the experiments with Mohammad Pezeshki. They both originally pioneered this project as a Ioannis Mitliagkas's class project. The original idea of using negative momentum comes from Ioannis Mitliagkas. Rémi Lepriol made the Right figure in 4.1 and helped on the smoothing of the paper. He also worked on improving the story and the clarity of the paper and proof-checked the appendix. Gabriel Huang made Figure 4.3 helped on the smoothing of the paper, worked on improving the story and the clarity of the paper, and proof-checked the appendix. Simon Lacoste-Julien and Ioannis Miltiagkas supervised this project.

3 Modifications with respect to the published paper

We corrected a typo in Thm. 6 (squared condition number instead of condition number; and small change in constant) and the dependence in β (the momentum parameter) in Theorem 5 for the formal statement. However, these modifications do not change our conclusions.

8 Negative Momentum for Improved Game Dynamics

Abstract

Games generalize the single-objective optimization paradigm by introducing different objective functions for different players. Differentiable games often proceed by simultaneous or alternating gradient updates. In machine learning, games are gaining new importance through formulations like generative adversarial networks (GANs) and actor-critic systems. However, compared to single-objective optimization, game dynamics is more complex and less understood. In this paper, we analyze gradient-based methods with momentum on simple games. We prove that alternating updates are more stable than simultaneous updates. Next, we show both theoretically and empirically that alternating gradient updates with a negative momentum term achieves convergence in a difficult toy adversarial problem, but also on the notoriously difficult to train saturating GANs.

1 Introduction

Recent advances in machine learning are largely driven by the success of gradient-based optimization methods for the training process. A common learning paradigm is empirical risk minimization, where a (potentially non-convex) objective, that depends on the data, is minimized. However, some recently introduced approaches require the joint minimization of several objectives. For example, actor-critic methods can be written as a bi-level optimization problem [Pfau and Vinyals, 2016] and generative adversarial networks (GANs) [Goodfellow et al., 2014] use a two-player game formulation.

Games generalize the standard optimization framework by introducing different objective functions for different optimizing agents, known as *players*. We are commonly interested in finding a local *Nash equilibrium*: a set of parameters from which no player can (locally and unilaterally) improve its objective function. Games with differentiable objectives often proceed by simultaneous or alternating gradient steps on the players' objectives. Even though the dynamics of gradient based methods is well understood for minimization problems, new issues appear in multi-player games. For instance, some stable stationary points of the dynamics may not be (local) Nash equilibria [Adolphs et al., 2018, Daskalakis and Panageas, 2018].

Motivated by a decreasing trend of momentum values in GAN literature (see Fig. 8.1), we study the effect of two particular algorithmic choices: (i) the choice between simultaneous and alternating updates, and (ii) the choice of step-size and momentum value. The idea behind our approach is that a momentum term combined with the alternating gradient method can be used to manipulate the natural oscillatory behavior of adversarial games. We summarize our main contributions as follows:

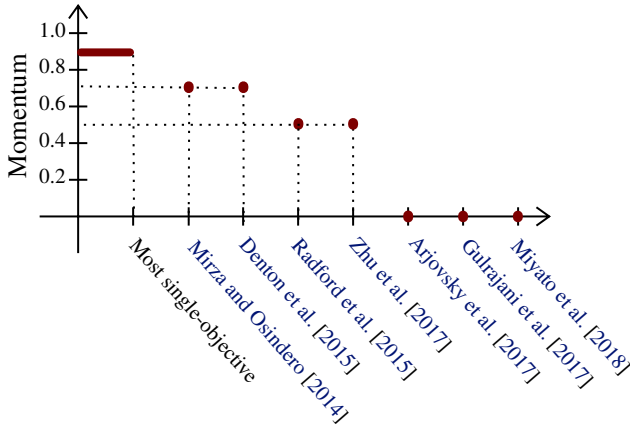
- We prove in §5 that the alternating gradient method with negative momentum is the only setting within our study parameters (Fig. 8.2) that converges on a bilinear smooth game. Using a zero or positive momentum value, or doing simultaneous updates in such games fails to converge.
- We show in §4 that, for general dynamics, when the eigenvalues of the Jacobian have a large imaginary part, negative momentum can improve the local convergence properties of the gradient method.
- We confirm the benefits of negative momentum for training GANs with the notoriously ill-behaved saturating loss on both toy settings, and real datasets.

Outline. §2 describes the fundamentals of the analytic setup that we use. §3 provides a formulation for the optimal step-size, and discusses the constraints and intuition behind it. §4 presents our theoretical results and guarantees on negative momentum. §5 studies the properties of alternating and simultaneous methods with negative momentum on a bilinear smooth game. §6 contains experimental results on toy and real datasets. Finally, in §7, we review some of the existing work on smooth game optimization as well as GAN stability and convergence.

2 Background

Notation. In this paper, scalars are lower-case letters (e.g., λ), vectors are lower-case bold letters (e.g., $\boldsymbol{\theta}$), matrices are upper-case bold letters (e.g., $\boldsymbol{A}$) and operators are upper-case letters (e.g., F). The spectrum of a squared matrix $\boldsymbol{A}$ is denoted by $\mathrm{Sp}(\boldsymbol{A})$, and its spectral radius is defined as $\rho(A) := \max\{|\lambda| \text{ for } \lambda \in \mathrm{Sp}(\boldsymbol{A})\}$. We respectively note $\sigma_{\min}(\boldsymbol{A})$ and $\sigma_{\max}(\boldsymbol{A})$ the smallest and the largest positive singular values of $\boldsymbol{A}$. The identity matrix of $\mathbb{R}^{m\times m}$ is written $\boldsymbol{I}_m$. We use $\Re$ and $\Im$ to respectively denote the real and imaginary part of a complex number. $\mathcal{O}$, Ω and Θ stand for the standard asymptotic notations. Finally, all the omitted proofs can be found in Appendix C §4.

Game theory formulation of GANs.



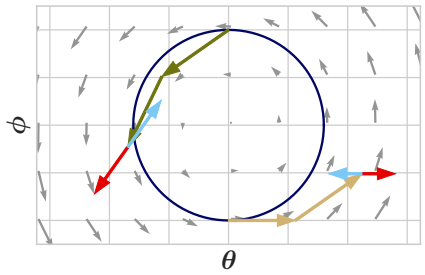


Figure 8.1: **Left:** Decreasing trend in the value of momentum used for training GANs across time. **Right:** Graphical intuition of the role of momentum in two steps of simultaneous updates (**tan**) or alternated updates (**olive**). Positive momentum (**red**) drives the iterates outwards whereas negative momentum (**blue**) pulls the iterates back towards the center, but it is only strong enough for alternated updates.

Method	β	Bounded	Converges	Bound on Δ_t
Simult. Thm. 5	>0	✗	✗	$\Omega\left((1+\eta^2\sigma^2_{\max}(A))^t\right)$
	0	✗	✗	$\Omega\left((1+\eta^2\sigma^2_{\max}(A))^t\right)$
	<0	✗	✗	$\Omega\left((1+{}^{\eta^2\sigma^2_{\max}(A)}/_{17})^t\right)$
Altern. Thm. 6	>0	✗	✗	**Conjecture:** $\Omega\left((1+\beta^2)^t\right)$
	0	✓	✗	$\Theta\left(\Delta_0\right)$
	<0	✓	✓	$\mathcal{O}\left(\Delta_0(1-{}^{\eta^2\sigma^2_{\min}(A)}/_{16})^t\right)$

Figure 8.2: Effect of gradient methods on an unconstrained bilinear example: $\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\varphi}$. The quantity Δ_t is the distance to the optimum (see formal definition in §5) and β is the momentum value.

Generative adversarial networks consist of a discriminator $D_{\boldsymbol{\varphi}}$ and a generator $G_{\boldsymbol{\theta}}$. In this game, the discriminator's objective is to tell real from generated examples. The generator's goal is to produce examples that are sufficiently close to real examples to confuse the discriminator.

From a game theory point of view, GAN training is a differentiable two-player game: the discriminator $D_{\boldsymbol{\varphi}}$ aims at minimizing its cost function $\mathcal{L}_D$ and the generator $G_{\boldsymbol{\theta}}$ aims at minimizing its own cost function $\mathcal{L}_G$. Using the same formulation as the one in Mescheder et al. [2017] and Gidel et al. [2019b], the GAN objective

has the following form,

$$\begin{cases} \boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\Theta} \mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \\ \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\Phi} \mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi}) \,. \end{cases} \tag{2.1}$$

Given such a game setup, GAN training consists of finding a local Nash Equilibrium, which is a state $(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$ in which neither the discriminator nor the generator can improve their respective cost by a small change in their parameters. In order to analyze the dynamics of gradient-based methods near a Nash Equilibrium, we look at the *gradient vector field*,

$$\boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta}) := \begin{bmatrix} \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\varphi}, \boldsymbol{\theta}) & \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{\theta}) \end{bmatrix}^\top , \tag{2.2}$$

and its associated *Jacobian* $\nabla \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})$,

$$\begin{bmatrix} \nabla^2_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\varphi}, \boldsymbol{\theta}) & \nabla_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}_D(\boldsymbol{\varphi}, \boldsymbol{\theta}) \\ \nabla_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{\theta})^T & \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\varphi}, \boldsymbol{\theta}) \end{bmatrix} . \tag{2.3}$$

Games in which $\mathcal{L}_G = -\mathcal{L}_D$ are called *zero-sum games* and (2.1) can be reformulated as a min-max problem. This is the case for the original *min-max* GAN formulation, but not the case for the *non-saturating loss* [Goodfellow et al., 2014] which is commonly used in practice.

For a zero-sum game, we note $\mathcal{L}_G = -\mathcal{L}_D = \mathcal{L}$. When the matrices $\nabla^2_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta})$ and $\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta})$ are zero, the Jacobian is anti-symmetric and has pure imaginary eigenvalues. We call games with pure imaginary eigenvalues *purely adversarial games*. This is the case in a simple bilinear game $\mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta}) := \boldsymbol{\varphi}^\top \boldsymbol{A} \boldsymbol{\theta}$. This game can be formulated as a GAN where the true distribution is a Dirac on 0, the generator is a Dirac on θ and the discriminator is linear. This setup was extensively studied in 2D by Gidel et al. [2019b].

Conversely, when $\nabla_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta})$ is zero and the matrices $\nabla^2_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta})$ and $-\nabla^2_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta})$ are symmetric and definite positive, the Jacobian is symmetric and has real positive eigenvalues. We call games with real positive eigenvalues *purely cooperative games*. This is the case, for example, when the objective function $\mathcal{L}$ is separable such as $\mathcal{L}(\boldsymbol{\varphi}, \boldsymbol{\theta}) = f(\boldsymbol{\varphi}) - g(\boldsymbol{\theta})$ where f and g are two convex functions. Thus, the optimization can be reformulated as two separated minimization of f and g with respect to their respective parameters.

These notions of *adversarial* and *cooperative* games can be related to the notions of *potential* games [Monderer and Shapley, 1996] and *Hamiltonian* games recently introduced by Balduzzi et al. [2018]: a game is a *potential game* (resp. *Hamiltonian game*) if its Jacobian is symmetric (resp. asymmetric). Our definition of *cooperative game* is a bit more general than the definition of *potential game* since some non-symmetric matrices may have positive eigenvalues. Similarly, the notion of *adversarial game* generalizes the *Hamiltonian games* since some non-antisymmetric

matrices may have pure imaginary eigenvalues, for instance,

$$\mathrm{Sp}\left(\begin{bmatrix} 0 & -1 \\ 2 & 3 \end{bmatrix}\right) = \{1, 2\}\,, \quad \mathrm{Sp}\left(\begin{bmatrix} -1 & 1 \\ -2 & 1 \end{bmatrix}\right) = \{\pm i\}\,.$$

In this work, we are interested in games *in between* purely adversarial games and purely cooperative ones, i.e., games which have eigenvalues with non-negative real part (cooperative component) and non-zero imaginary part (adversarial component). For $\boldsymbol{A} \in \mathbb{R}^{d\times p}$, a simple class of such games is parametrized by $\alpha \in [0, 1]$,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\varphi}\in\mathbb{R}^p} \; \alpha\|\boldsymbol{\theta}\|_2^2 + (1-\alpha)\boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\varphi} - \alpha\|\boldsymbol{\varphi}\|_2^2\,, \tag{2.4}$$

Simultaneous Gradient Method. Let us consider the dynamics of the simultaneous gradient method. It is defined as the repeated application of the operator,

$$F_\eta(\boldsymbol{\varphi}, \boldsymbol{\theta}) := \begin{bmatrix} \boldsymbol{\varphi} & \boldsymbol{\theta} \end{bmatrix}^\top - \eta\, \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta})\,, \quad (\boldsymbol{\varphi}, \boldsymbol{\theta}) \in \mathbb{R}^m\,, \tag{2.5}$$

where η is the learning rate. Now, for brevity we write the joint parameters $\boldsymbol{\omega} := (\boldsymbol{\varphi}, \boldsymbol{\theta}) \in \mathbb{R}^m$. For $t \in \mathbb{N}$, let $\boldsymbol{\omega}_t = (\boldsymbol{\varphi}_t, \boldsymbol{\theta}_t)$ be the t^{th} point of the sequence computed by the gradient method,

$$\boldsymbol{\omega}_t = \underbrace{F_\eta \circ \ldots \circ F_\eta}_{t}(\boldsymbol{\omega}_0) = F_\eta^{(t)}(\boldsymbol{\omega}_0)\,. \tag{2.6}$$

Then, if the gradient method converges, and its limit point $\boldsymbol{\omega}^* = (\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$ is a *fixed point* of F_η such that $\nabla v(\boldsymbol{\omega}^*)$ is positive-definite, then $\boldsymbol{\omega}^*$ is a local Nash equilibrium. Interestingly, some of the stable stationary points of gradient dynamics may not be Nash equilibrium [Adolphs et al., 2018]. In this work, we focus on the local convergence properties near the stationary points of gradient . To the best of our knowledge, there is no first order method alleviating this issue. In the following, $\boldsymbol{\omega}^*$ is a stationary point of the gradient dynamics (i.e. a point such that $\boldsymbol{v}(\boldsymbol{\omega}^*) = 0$).

3 Tuning the Step-size

Under certain conditions on a fixed point operator, linear convergence is guaranteed in a neighborhood around a fixed point.

Theorem 1 (Prop. 4.4.1 Bertsekas [1999]). *If the spectral radius* $\rho_{\max} := \rho(\nabla F_\eta(\boldsymbol{\omega}^*)) < 1$*, then, for* $\boldsymbol{\omega}_0$ *in a neighborhood of* $\boldsymbol{\omega}^*$*, the distance of* $\boldsymbol{\omega}_t$ *to the stationary point* $\boldsymbol{\omega}^*$ *converges at a linear rate of* $\mathcal{O}\Big((\rho_{\max} + \epsilon)^t\Big)\,, \forall \epsilon > 0$.

From the definition in (2.5), we have:

$$\nabla F_\eta(\boldsymbol{\omega}^*) = \boldsymbol{I}_m - \eta \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)\,, \tag{3.1}$$
$$\text{and} \quad \mathrm{Sp}(\nabla F_\eta(\boldsymbol{\omega}^*)) = \{1 - \eta\lambda | \lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))\}\ .$$

If the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ all have a positive real-part, then for small enough η, the eigenvalues of $\nabla F_\eta(\boldsymbol{\omega}^*)$ are inside a convergence circle of radius $\rho_{\max} < 1$, as illustrated in Fig. 8.3. Thm. 1 guarantees the existence of an optimal step-size η_{best} which yields a non-trivial convergence rate $\rho_{\max} < 1$. Thm. 2 gives analytic bounds on the optimal step-size η_{best}, and lower-bounds the best convergence rate $\rho_{\max}(\eta_{best})$ we can expect.

Theorem 2. *If the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ all have a positive real-part, then, the best step-size η_{best}, which minimizes the spectral radius $\rho_{\max}(\eta)$ of $\nabla F_\eta(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$, is the solution of a (convex) quadratic by parts problem, and satisfies,*

$$\max_{1\leq k\leq m} \sin(\psi_k)^2 \leq \rho_{\max}(\eta_{best})^2 \leq 1 - \Re(1/\lambda_1)\delta\,, \tag{3.2}$$
$$\textit{with} \quad \delta := \min_{1\leq k\leq m} |\lambda_k|^2 (2\Re(1/\lambda_k) - \Re(1/\lambda_1)) \tag{3.3}$$
$$\textit{and} \qquad \Re(1/\lambda_1) \leq \eta_{best} \leq 2\Re(1/\lambda_1) \tag{3.4}$$

where $(\lambda_k = r_k e^{i\psi_k})_{1\leq k\leq m} = \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*))$ are sorted such that $0 < \Re(1/\lambda_1) \leq \cdots \leq \Re(1/\lambda_m)$. Particularly, when $\eta_{best} = \Re(1/\lambda_1)$ we are in the case of the top plot of Fig.8.3 and $\rho_{\max}(\eta_{best})^2 = \sin(\psi_1)^2$.

When $\nabla \boldsymbol{v}$ is positive-definite, the best η_{best} is attained either because of one or several limiting eigenvalues. We illustrate and interpret these two cases in Fig. 8.3. In multivariate convex optimization, the optimal step-size depends on the extreme eigenvalues and their ratio, the *condition number*. Unfortunately, the notion of the condition number does not trivially extend to games, but Thm. 2 seems to indicate that the real part of the inverse of the eigenvalues play an important role in the dynamics of smooth games. We think that a notion of condition number might be meaningful for such games and we propose an illustrative example to discuss this point in §2. Note that when the eigenvalues are pure positive real numbers belonging to $[\mu, L]$, (3.2) provides the standard bound $\rho_{\max} \leq 1 - \mu/L$ obtained with a step-size $\eta = 1/L$ (see §4.2 for details).

Note that, in (3.3), we have $\delta > 0$ because (λ_k) are sorted such that, $\Re(1/\lambda_k) \geq \Re(1/\lambda_1)\,, \forall 1 \leq k \leq m$. In (3.2), we can see that if the Jacobian of $\boldsymbol{v}$ has an almost purely imaginary eigenvalue $r_j e^{\psi_j}$ then $\sin(\psi_j)$ is close to 1 and thus, the convergence rate of the gradient method may be arbitrarily close to 1. Zhang and Mitliagkas [2019] provide an analysis of the momentum method for quadratics, showing that momentum can actually help to better condition the model. One interesting point from their work is that the best conditioning is achieved when the added momentum makes the Jacobian eigenvalues turn from positive reals into complex conjugate pairs. Our goal is to use momentum to wrangle game dynamics into convergence manipulating the eigenvalues of the Jacobian.

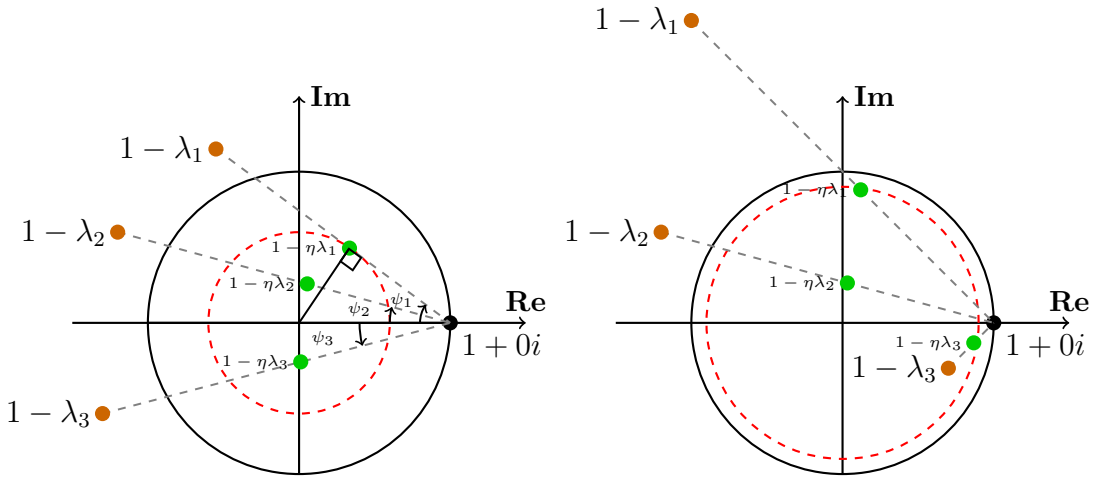


Figure 8.3: Eigenvalues λ_i of the Jacobian $\nabla \boldsymbol{v}(\boldsymbol{\phi}^*, \boldsymbol{\theta}^*)$, their trajectories $1 - \eta\lambda_i$ for growing step-sizes, and the optimal step-size. The unit circle is drawn in **black**, and the **red** dashed circle has radius equal to the largest eigenvalue $\mu_{\max}$, which is directly related to the convergence rate. Therefore, smaller red circles mean better convergence rates. **Top:** The red circle is limited by the tangent trajectory line $1 - \eta\lambda_1$, which means the best convergence rate is limited only by the eigenvalue which will pass furthest from the origin as η grows, i.e., $\lambda_i = \arg\min \Re(1/\lambda_i)$. **Bottom:** The red circle is cut (not tangent) by the trajectories at points $1 - \eta\lambda_1$ and $1 - \eta\lambda_3$. The η is optimal because any increase in η will push the eigenvalue λ_1 out of the red circle, while any decrease in step-size will retract the eigenvalue λ_3 out of the red circle, which will lower the convergence rate in any case. *Figure inspired by Mescheder et al. [2017].*

4 Negative Momentum

As shown in (3.2), the presence of eigenvalues with large imaginary parts can restrict us to small step-sizes and lead to slow convergence rates. In order to improve convergence, we add a *negative* momentum term into the update rule. Informally, one can think of negative momentum as friction that can damp oscillations. The new momentum term leads to a modification of the *parameter update operator* $F_\eta(\boldsymbol{\omega})$ of (2.5). We use a similar state augmentation as Zhang and Mitliagkas [2019] and Daskalakis and Panageas [2018] to form a compound state $(\boldsymbol{\omega}_t, \boldsymbol{\omega}_{t-1}) := (\boldsymbol{\varphi}_t, \boldsymbol{\theta}_t, \boldsymbol{\varphi}_{t-1}, \boldsymbol{\theta}_{t-1}) \in \mathbb{R}^{2m}$. The update rule (2.5) turns into the following,

$$F_{\eta,\beta}(\boldsymbol{\omega}_t, \boldsymbol{\omega}_{t-1}) = (\boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}_t) \tag{4.1}$$

$$\text{where} \quad \boldsymbol{\omega}_{t+1} := \boldsymbol{\omega}_t - \eta \boldsymbol{v}(\boldsymbol{\omega}_t) + \beta(\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t-1})\,, \tag{4.2}$$

in which $\beta \in \mathbb{R}$ is the momentum parameter. Therefore, the Jacobian of $F_{\eta,\beta}$ has the following form,

$$\begin{bmatrix} \boldsymbol{I}_n & \boldsymbol{0}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} - \eta \begin{bmatrix} \nabla \boldsymbol{v}(\boldsymbol{\omega}_t) & \boldsymbol{0}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} + \beta \begin{bmatrix} \boldsymbol{I}_n & -\boldsymbol{I}_n \\ \boldsymbol{0}_n & \boldsymbol{0}_n \end{bmatrix} \tag{4.3}$$

Note that for $\beta = 0$, we recover the gradient method.

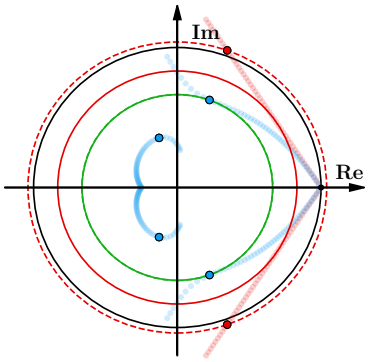


Figure 8.4: Transformation of the eigenvalues by the negative momentum method for a game introduced in (2.4) with $d = p = 1, A = 1, \alpha = 0.4, \eta = 1.55, \beta = -0.25$. Convergence circles for gradient method are in **red**, negative momentum in **green**, and unit circle in **black**. Solid convergence circles are optimized over all step-sizes, while dashed circles are at a given step-size η. For a fixed η, original eigenvalues are in **red** and negative momentum eigenvalues are in **blue**. Their trajectories as η sweeps in $[0, 2]$ are in light colors. Negative momentum helps as the new convergence circle (green) is smaller, due to shifting the original eigenvalues (red dots) towards the origin (right blue dots), while the eigenvalues due to state augmentation (left blue dots) have smaller magnitude and do not influence the convergence rate. Negative momentum allows faster convergence (green circle is inside the solid red circle) for a much broader range of step-sizes.

In some situations, if $\beta < 0$ is adjusted properly, negative momentum can improve the convergence rate to a local stationary point by pushing the eigenvalues of its Jacobian towards the origin. In the following theorem, we provide an explicit equation for the eigenvalues of the Jacobian of $F_{\eta,\beta}$.

Theorem 3. *The eigenvalues of* $\nabla F_{\eta,\beta}(\boldsymbol{\omega}^*)$ *are*

$$\mu_{\pm}(\beta, \eta, \lambda) := (1 - \eta\lambda + \beta)\frac{1 \pm \Delta^{\frac{1}{2}}}{2}, \tag{4.4}$$

where $\Delta := 1 - \frac{4\beta}{(1-\eta\lambda+\beta)^2}$, $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$ *and* $\Delta^{\frac{1}{2}}$ *is the* complex *square root of*

Δ *with positive real part*[1]*. Moreover we have the following Taylor approximation,*

$$\mu_+(\beta,\eta,\lambda) = 1-\eta\lambda-\beta\frac{\eta\lambda}{1-\eta\lambda}+O(\beta^2)\,, \tag{4.5}$$

$$\mu_-(\beta,\eta,\lambda) = \frac{\beta}{1-\eta\lambda}+O(\beta^2)\,. \tag{4.6}$$

When β is small enough, Δ is a complex number close to 1. Consequently, μ_+ is close to the original eigenvalue for gradient dynamics $1-\eta\lambda$, and μ_-, the eigenvalue introduced by the state augmentation, is close to 0. We formalize this intuition by providing the first order approximation of both eigenvalues.

In Fig. 8.4, we illustrate the effects of negative momentum on a game described in (2.4). Negative momentum shifts the original eigenvalues (trajectories in light red) by pushing them to the left towards the origin (trajectories in light blue).

Since our goal is to minimize the largest magnitude of the eigenvalues of $F_{\eta,\beta}$ which are computed in Thm. 3, we want to understand the effect of β on these eigenvalues with potential large magnitude. Let $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$, we define the (squared) magnitude $\rho_{\lambda,\eta}(\beta)$ that we want to optimize,

$$\rho_{\lambda,\eta}(\beta) := \max\left\{|\mu_+(\beta,\eta,\lambda)|^2, |\mu_-(\beta,\eta,\lambda)|^2\right\}. \tag{4.7}$$

We study the local behavior of $\rho_{\lambda,\eta}$ for small β. The following theorem shows that a well suited β decreases $\rho_{\lambda,\eta}$, which corresponds to faster convergence.

Theorem 4. *For any $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$ s.t. $\Re(\lambda) > 0$,*

$$\rho'_{\lambda,\eta}(0) > 0 \Leftrightarrow \eta \in I(\lambda) := \left(\tfrac{|\lambda|-|\Im(\lambda)|}{|\lambda|\Re(\lambda)}, \tfrac{|\lambda|+|\Im(\lambda)|}{|\lambda|\Re(\lambda)}\right).$$

Particularly, we have $\rho'_{\lambda,\Re(1/\lambda)}(0) = 2\Re(\lambda)\Re(1/\lambda) > 0$ and if $|Arg(\lambda)| \geq \frac{\pi}{4}$ then, $(\Re(1/\lambda), 2\Re(1/\lambda)) \subset I(\lambda)$.

As we have seen previously in Fig. 8.3 and Thm. 2, there are only few eigenvalues which slow down the convergence. Thm. 4 is a local result showing that a small negative momentum can improve the magnitude of the limiting eigenvalues in the following cases: when there is only one limiting eigenvalue λ_1 (since in that case the optimal step-size is $\eta_{best} = \Re(1/\lambda_1) \in I(\lambda_1)$) or when there are several limiting eigenvalues $\lambda_1, \ldots, \lambda_k$ and the intersection $I(\lambda_1) \cap \ldots \cap I(\lambda_k)$ is not empty. We point out that we do not provide any guarantees on whether this intersection is empty or not but note that if the absolute value of the argument of λ_1 is larger than $\pi/4$ then by (3.4), our theorem provides that the optimal step-size η_{best} belongs to $I(\lambda_1)$.

Since our result is local, it does not provide any guarantees on large negative values of β. Nevertheless, we numerically optimized (4.7) with respect to β and η

[1] If Δ is a negative real number we set $\Delta^{\frac{1}{2}} := i\sqrt{-\Delta}$

and found that for any non-imaginary fixed eigenvalue λ, the optimal momentum is negative and the associated optimal step-size is larger than $\hat{\eta}(\lambda)$. Another interesting aspect of negative momentum is that it admits larger step-sizes (see Fig. 8.4 and 8.5).

For a game with purely imaginary eigenvalues, when $|\eta\lambda| \ll 1$, Thm. 3 shows that $\mu_+(\beta, \eta, \lambda) \approx 1 - (1 + \beta)\eta\lambda$. Therefore, at the first order, β only has an impact on the imaginary part of μ_+. Consequently μ_+ cannot be pushed into the unit circle, and the convergence guarantees of Thm. 1 do not apply. In other words, the analysis above provides convergence rates for games without any pure imaginary eigenvalues. It excludes the purely adversarial bilinear example ($\alpha = 0$ in Eq. 2.4) that is discussed in the next section.

5 Bilinear Smooth Games

In this section we analyze the dynamics of a purely adversarial game described by,

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^d} \max_{\boldsymbol{\varphi} \in \mathbb{R}^p} \boldsymbol{\theta}^\top \boldsymbol{A} \boldsymbol{\varphi} + \boldsymbol{\theta}^\top \boldsymbol{b} + \boldsymbol{c}^\top \boldsymbol{\varphi}, \quad \boldsymbol{A} \in \mathbb{R}^{d \times p}. \tag{5.1}$$

The first order stationary condition for this game characterizes the solutions $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ as

$$\boldsymbol{A}\boldsymbol{\varphi}^* = \boldsymbol{b} \quad \text{and} \quad \boldsymbol{A}^\top \boldsymbol{\theta}^* = \boldsymbol{c}. \tag{5.2}$$

If $\boldsymbol{b}$ (resp. $\boldsymbol{c}$) does not belong to the column space of $\boldsymbol{A}$ (resp. $\boldsymbol{A}^\top$), the game (5.1) admits no equilibrium. In the following, we assume that an equilibrium does exist for this game. Consequently, there exist $\boldsymbol{b}'$ and $\boldsymbol{c}'$ such that $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{b}'$ and $\boldsymbol{c} = \boldsymbol{A}^\top \boldsymbol{c}'$. Using the translations $\boldsymbol{\theta} \to \boldsymbol{\theta} - \boldsymbol{c}'$ and $\boldsymbol{\varphi} \to \boldsymbol{\varphi} - \boldsymbol{b}'$, we can assume without loss of generality, that $p \geq d$, $\boldsymbol{b} = \boldsymbol{0}$ and $\boldsymbol{c} = \boldsymbol{0}$. We provide upper and lower bounds on the squared distance from the known equilibrium,

$$\Delta_t = \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*\|_2^2 \tag{5.3}$$

where $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ is the projection of $(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$ onto the solution space. We show in §3, Lem. 12 that, for our methods of interest, this projection has a simple formulation that only depends on the initialization $(\boldsymbol{\theta}_0, \boldsymbol{\varphi}_0)$.

We aim to understand the difference between the dynamics of simultaneous steps and alternating steps. Practitioners have been widely using the latter instead of the former when optimizing GANs despite the rich optimization literature on simultaneous methods.

5.1 Simultaneous gradient descent

We define this class of methods with momentum using the following formulas,

$$F^{\text{sim}}_{\eta,\beta}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t, \boldsymbol{\theta}_{t-1}, \boldsymbol{\varphi}_{t-1}) := (\boldsymbol{\theta}_{t+1}, \boldsymbol{\varphi}_{t+1}, \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t) \tag{5.4}$$

$$\text{where } \begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \\ \boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \eta_2 \boldsymbol{A}^\top \boldsymbol{\theta}_t + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1}) \,. \end{cases}$$

In our simple setting, the operator $F^{\text{sim}}_{\eta,\beta}$ is linear. One way to study the asymptotic properties of the sequence $(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$ is to compute the eigenvalues of $\nabla F^{\text{sim}}_{\eta,\beta}$. The following proposition characterizes these eigenvalues.

Proposition 1. *The eigenvalues of $\nabla F^{sim}_{\eta,\beta}$ are the roots of the 4^{th} order polynomials:*

$$(x-1)^2(x-\beta_1)(x-\beta_2) + \eta_1\eta_2\lambda x^2,\ \lambda \in \text{Sp}(\boldsymbol{A}^\top \boldsymbol{A}) \tag{5.5}$$

Interestingly, these roots only depend on the product $\eta_1\eta_2$ meaning that any re-scaling $\eta_1 \to \gamma\eta_1\,,\ \eta_2 \to \frac{1}{\gamma}\eta_2$ does not change the eigenvalues of $\nabla F^{\text{sim}}_{\eta,\beta}$ and consequently the asymptotic dynamics of the iterates $(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$. The magnitude of the eigenvalues described in (5.5), characterizes the asymptotic properties for the iterates of the simultaneous method (5.4). We report the maximum magnitude of these roots for a given λ and for a grid of step-sizes and momentum values in Fig C.1. We observe that they are always larger than 1, which transcribes a diverging behavior. The following theorem provides an analytical rate of divergence.

Theorem 5. *For any $\eta_1, \eta_2 \geq 0$ and $\beta_1 = \beta_2 = \beta$, the iterates of the simultaneous methods (5.4) diverge as,*

$$\Delta_t \in \begin{cases} \Omega\Big(\Delta_0(1 + \eta^2\sigma^2_{\max}(A))^t\Big) & \text{if} \quad \beta \geq 0 \\ \Omega\Big(\Delta_0(1 + \frac{\eta^2\sigma^2_{\max}(A)}{17})^t\Big) & \text{if} \quad -\dfrac{1}{16} \leq \beta < 0\,. \end{cases}$$

This theorem states that the iterates of the simultaneous method (5.4) diverge geometrically for $\beta \geq -\frac{1}{16}$. Interestingly, this geometric divergence implies that *even* a uniform averaging of the iterates (standard in game optimization to ensure convergence [Freund et al., 1999]) cannot alleviate this divergence.

5.2 Alternating gradient descent

Alternating gradient methods take advantage of the fact that the iterates $\boldsymbol{\theta}_{t+1}$ and $\boldsymbol{\varphi}_{t+1}$ are computed sequentially, to plug the value of $\boldsymbol{\theta}_{t+1}$ (instead of $\boldsymbol{\theta}_t$ for

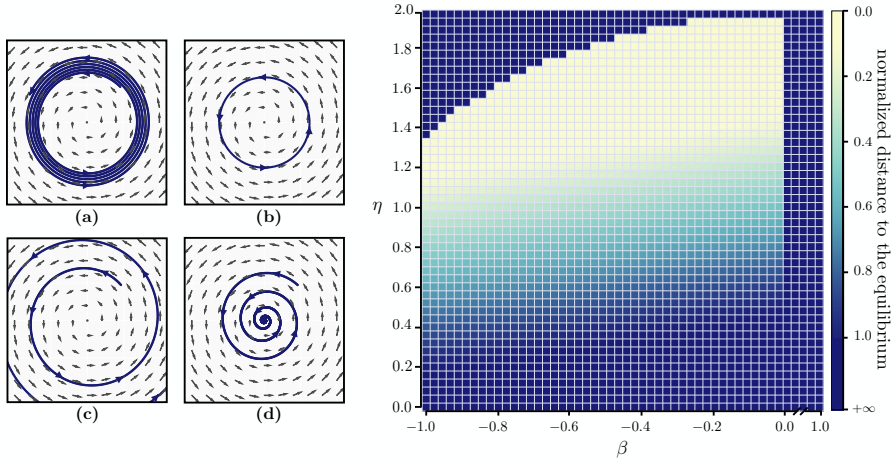


Figure 8.5: The effect of momentum in a simple min-max bilinear game where the equilibrium is at $(0, 0)$. **(left-a)** Simultaneous GD with no momentum **(left-b)** Alternating GD with no momentum. **(left-c)** Alternating GD with a momentum of $+0.1$. **(left-d)** Alternating GD with a momentum of -0.1. **(right)** A grid of experiments for alternating GD with different values of momentum (β) and step-sizes (η): While any positive momentum leads to divergence, small enough value of negative momentum allows for convergence with large step-sizes. The color in each cell indicates the normalized distance to the equilibrium after 500k iteration, such that 1.0 corresponds to the initial condition and values larger (smaller) than 1.0 correspond to divergence (convergence).

simultaneous update rule) into the update of $\boldsymbol{\varphi}_{t+1}$,

$$F^{\text{alt}}_{\eta,\beta}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t, \boldsymbol{\theta}_{t-1}, \boldsymbol{\varphi}_{t-1}) := (\boldsymbol{\theta}_{t+1}, \boldsymbol{\varphi}_{t+1}, \boldsymbol{\theta}_t, \boldsymbol{\varphi}_t) \tag{5.6}$$

$$\text{where} \begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \\ \boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \eta_2 \boldsymbol{A}^\top \boldsymbol{\theta}_{t+1} + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1}) \,. \end{cases}$$

This slight change between (5.4) and (5.6) significantly shifts the eigenvalues of the Jacobian. We first characterize them with the following proposition.

Proposition 2. *The eigenvalues of $\nabla F^{alt}_{\eta,\beta}$ are the roots of the 4^{th} order polynomials:*

$$(x-1)^2(x-\beta_1)(x-\beta_2) + \eta_1\eta_2\lambda x^3 \,,\ \lambda \in \text{Sp}(\boldsymbol{A}^\top \boldsymbol{A}) \tag{5.7}$$

The same way as in (5.5), these roots only depend on the product $\eta_1\eta_2$. The only difference is that the monomial with coefficient $\eta_1\eta_2\lambda$ is of degree 2 in (5.5) and of degree 3 in (5.7). This difference is major since, for well chosen values of negative momentum, the eigenvalues described in Prop. 2 lie in the unit disk (see Fig. C.1). As a consequence, the iterates of the alternating method with no momentum are bounded and do converge if we add some well chosen negative momentum:

Theorem 6. *If we set $\eta \leq \frac{1}{\sigma_{\max}(A)}$, $\beta_1 = -\frac{1}{2}$ and $\beta_2 = 0$ then we have*

$$\Delta_{t+1} \in O\left(\max\{\tfrac{1}{2}, 1 - \tfrac{\eta^2\sigma^2_{\min}(A)}{16}\}^t \Delta_0\right) \tag{5.8}$$

If we set $\beta_1 = 0$ and $\beta_2 = 0$, then there exists $M > 1$ such that for any $\eta_1, \eta_2 \geq 0$, $\Delta_t = \Theta(\Delta_0)$.

Our results from this section, namely Thm. 5 and Thm. 6, are summarized in Fig. 8.2, and demonstrate how alternating steps can improve the convergence properties of the gradient method for bilinear smooth games. Moreover, combining them with negative momentum can surprisingly lead to a linearly convergent method. The conjecture provided in Fig. 8.2 (divergence of the alternating method with positive momentum) is backed-up by the results provided in Fig. 8.5 and §1.1.

6 Experiments and Discussion

Min-Max Bilinear Game [Fig. 8.5] In our first experiments, we showcase the effect of negative momentum in a bilinear min-max optimization setup (2.4) where $\boldsymbol{\phi}, \boldsymbol{\theta} \in \mathbb{R}$ and $\boldsymbol{A} = 1$. We compare the effect of positive and negative momentum in both cases of alternating and simultaneous gradient steps.

Fashion MNIST and CIFAR 10 [Fig. 8.6] In our third set of experiments, we use negative momentum in a GAN setup on CIFAR-10 [Krizhevsky and Hinton, 2009] and Fashion-MNIST [Xiao et al., 2017] with *saturating loss* and alternating steps. We use residual networks for both the generator and the discriminator with no batch-normalization. Following the same architecture as Gulrajani et al. [2017], each residual block is made of two 3×3 convolution layers with $ReLU$ activation function. Up-sampling and down-sampling layers are respectively used in the generator and discriminator. We experiment with different values of momentum on the discriminator and a constant value of 0.5 for the momentum of the generator. We observe that using a negative value can generally result in samples with higher quality and inception scores. Intuitively, using negative momentum only on the discriminator slows down the learning process of the discriminator and allows for better flow of the gradient to the generator. Note that we provide an additional experiment on mixture of Gaussians in § 1.2.

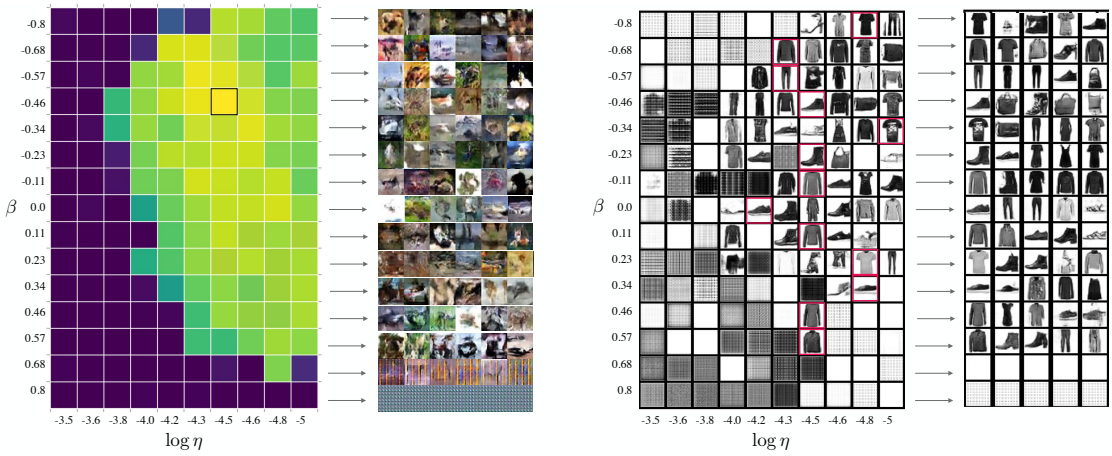


Figure 8.6: Comparison between negative and positive momentum on GANs with saturating loss on CIFAR-10 (left) and on Fashion MNIST (right) using a residual network. For each dataset, a grid of different values of momentum (β) and step-sizes (η) is provided which describes the discriminator's settings while a constant momentum of 0.5 and step-size of 10^{-4} is used for the generator. Each cell in CIFAR-10 (or Fashion MNIST) grid contains a single configuration in which its color (or its content) indicates the inception score (or a single sample) of the model. For CIFAR-10 experiments, yellow is higher while blue is the lower inception score. Along each row, the best configuration is chosen and more samples from that configuration are presented on the right side of each grid.

7 Related Work

Optimization.

From an optimization point of view, a lot of work has been done in the context of understanding momentum and its variants [Polyak, 1964, Qian, 1999, Nesterov, 2004, Sutskever et al., 2013]. Some recent studies have emphasized the importance of momentum tuning in deep learning such as Sutskever et al. [2013], Kingma and Ba [2015], and Zhang and Mitliagkas [2019], however, none of them consider using negative momentum. Among recent work, using robust control theory, Lessard et al. [2016] study optimization procedures and cover a variety of algorithms including momentum methods. Their analysis is global and they establish worst-case bounds for smooth and strongly-convex functions. Mitliagkas et al. [2016] considered negative momentum in the context of asynchronous single-objective minimization. They show that asynchronous-parallel dynamics 'bleed' into optimization updates introducing momentum-like behavior into SGD. They argue that algorithmic momentum and asynchrony-induced momentum add up to create an effective 'total momentum' value. They conclude that to attain the optimal (positive) effective momentum in an asynchronous system, one would have to reduce algorithmic momentum to small or sometimes negative values. This differs from our work where we show that for games the optimal effective momentum may be negative. Ghadimi et al. [2015] analyze momentum and provide global convergence properties for functions with Lipschitz-continuous gradients. However, all the results mentioned above are restricted to minimization problems. The purpose of our work is to try to understand how momentum influences game dynamics which is intrinsically different from minimization dynamics.

Finally, similar proof techniques based on the study of the eigenvalues of a state-augmented operator have been recently used by Daskalakis and Panageas [2018] for the study of the optimistic gradient method (OGDA). However, even though OGDA and Polyak's momentum can be seen as a variant of the gradient method with an additional term, these additional terms are different. In OGDA it is a difference between the two previous gradients, while in Polyak's method it is a difference between the two past iterates.

GANs as games. A lot of recent work has attempted to make GAN training easier with new optimization methods. Daskalakis et al. [2018] extrapolate the next value of the gradient using previous history and Gidel et al. [2019b] explore averaging and introduce a variant of the extra-gradient algorithm.

Balduzzi et al. [2018] develop new methods to understand the dynamics of general games: they decompose second-order dynamics into two components using Helmholtz decomposition and use the fact that the optimization of Hamiltonian games is well understood. It differs from our work since we do not consider any decomposition of the Jacobian but focus on the manipulation of its eigenvalues.

Recently, Liang and Stokes [2019] provide a unifying theory for smooth two-player games for non-asymptotic local convergence. They also provide theory for choosing the right step-size required for convergence.

From another perspective, Odena et al. [2018] show that in a GAN setup, the average conditioning of the Jacobian of the generator becomes ill-conditioned during training. They propose Jacobian clamping to improve the inception score and Frechet Inception Distance. Mescheder et al. [2017] provide discussion on how the eigenvalues of the Jacobian govern the local convergence properties of GANs. They argue that the presence of eigenvalues with zero real-part and large imaginary-part results in oscillatory behavior but do not provide results on the optimal step-size and on the impact of momentum. Nagarajan and Kolter [2017] also analyze the local stability of GANs as an approximated continuous dynamical system. They show that during training of a GAN, the eigenvalues of the Jacobian of the corresponding vector field are pushed away from one along the real axis.

8 Conclusion

In this paper, we show analytically and empirically that alternating updates with negative momentum is the only method within our study parameters (Fig.8.2) that converges in bilinear smooth games. We study the effects of using negative values of momentum in a GAN setup both theoretically and experimentally. We show that, for a large class of adversarial games, negative momentum may improve the convergence rate of gradient-based methods by shifting the eigenvalues of the Jacobian appropriately into a smaller convergence disk. We found that, in simple yet intuitive examples, using negative momentum makes convergence to the Nash Equilibrium easier. Our experiments support the use of negative momentum for saturating losses on mixtures of Gaussians, as well as on other tasks using CIFAR-10 and fashion MNIST. Altogether, fully stabilizing learning in GANs requires a deep understanding of the underlying highly non-linear dynamics. We believe our work is a step towards a better understanding of these dynamics. We encourage deep learning researchers and practitioners to include negative values of momentum in their hyper-parameter search.

We believe that our results explain a decreasing trend in momentum values used for training GANs in the past few years reported in Fig. 8.4. Some of the most successful papers use zero momentum [Arjovsky et al., 2017, Gulrajani et al., 2017] for architectures that would otherwise call for high momentum values in a non-adversarial setting.

9 Prologue to the Fourth Contribution

1 Article Details

A Closer Look at the Optimization Landscapes of Generative Adversarial Networks. *Hugo Berard**, *Gauthier Gidel**, *Amjad Almahairi, Pascal Vincent and Simon Lacoste-Julien.* This paper was published at ICLR 2020 [Berard et al., 2020].

*Equal contribution.

2 Contributions of the authors

Gauthier Gidel contributed to the general writing of the paper, the idea of the paper (jointly with Hugo Berard) as well as the proof of all the theorems of the paper. Gauthier Gidel also brought his knowledge about optimization. Hugo Berard co-lead the project, wrote the code, did the experiments, and wrote the experimental section. He also brought his knowledge about GANs and more generally generative modeling. Amjad Almahairi supervised Gauthier Gidel during his internship at ElementAI, worked on the writing of the paper, and on the experiments. Simon Lacoste-Julien and Pascal Vincent supervised this project.

10 A Closer Look at the Optimization Landscapes of Generative Adversarial Network

Abstract

Generative adversarial networks have been very successful in generative modeling, however they remain relatively challenging to train compared to standard deep neural networks. In this paper, we propose new visualization techniques for the optimization landscapes of GANs that enable us to study the game vector field resulting from the concatenation of the gradient of both players. Using these visualization techniques we try to bridge the gap between theory and practice by showing empirically that the training of GANs exhibits significant rotations around Local Stable Stationary Points (LSSP), similar to the one predicted by theory on toy examples.

Moreover, we provide empirical evidence that GAN training converge to a *stable* stationary point which is a saddle point for the generator loss, not a minimum, while still achieving excellent performance.

1 Introduction

Deep neural networks have exhibited remarkable success in many applications [Krizhevsky et al., 2012]. This success has motivated many studies of their non-convex loss landscape [Choromanska et al., 2015, Kawaguchi, 2016, Li et al., 2018], which, in turn, has led to many improvements, such as better initialization and optimization methods [Glorot and Bengio, 2010, Kingma and Ba, 2015].

While most of the work on studying non-convex loss landscapes has focused on single objective minimization, some recent class of models require the joint minimization of several objectives, making their optimization landscape intrinsically different. Among these models is the generative adversarial network (GAN) [Goodfellow et al., 2014] which is based on a two-player game formulation and has achieved state-of-the-art performance on some generative modeling tasks such as image generation [Brock et al., 2019].

On the theoretical side, many papers studying multi-player games have argued that one main optimization issue that arises in this case is the rotation due to the adversarial component of the game [Mescheder et al., 2018, Balduzzi et al., 2018, Gidel et al., 2019c]. This has been extensively studied on toy examples, in particular

on the so-called bilinear example [Goodfellow, 2016] (a.k.a Dirac GAN [Mescheder et al., 2018]). However, those toy examples are very far from the standard realistic setting of image generation involving deep networks and challenging datasets. To our knowledge it remains an open question if this rotation phenomenon actually occurs when training GANs in more practical settings.

In this paper, we aim at closing this gap between theory and practice. Following Mescheder et al. [2017] and Balduzzi et al. [2018], we argue that instead of studying the loss surface, we should study the *game vector field* (i.e., the concatenation of each player's gradient), which can provide better insights to the problem. To this end, we propose a new visualization technique that we call *Path-angle* which helps us observe the nature of the game vector field close to a stationary point for high dimensional models, and carry on an empirical investigation of the properties of the optimization landscape of GANs. The core questions we want to address may be summarized as the following:

Is rotation a phenomenon that occurs when training GANs on real world datasets, and do existing training methods find local Nash equilibria?

To answer this question we conducted extensive experiments by training different GAN formulations (NSGAN and WGAN-GP) with different optimizers (Adam and ExtraAdam) on three datasets (MoG, MNIST and CIFAR10). Based on our experiments and using our visualization techniques we observe that the landscape of GANs is fundamentally different from the standard loss surfaces of deep networks. Furthermore, we provide evidence that existing GAN training methods do not converge to a local Nash equilibrium.

Contributions. More precisely, our contributions are the following: (i) We propose studying empirically the game vector field (as opposed to studying the loss surfaces of each player) to understand training dynamics in GANs using a novel visualization tool, which we call *Path-angle* and that captures the rotational and attractive behaviors near local stationary points (ref. §4.2). (ii) We observe experimentally on both a mixture of Gaussians, MNIST and CIFAR10 datasets that a variety of GAN formulations have a significant rotational behavior around their locally stable stationary points (ref. §5.1). (iii) We provide empirical evidence that existing training procedures find stable stationary points that are saddle points, not minima, for the loss function of the generator (ref. § 5.2).

2 Related work

Improving the training of GANs has been an active research area in the past few years. Most efforts in stabilizing GAN training have focused on formulating new objectives [Arjovsky et al., 2017], or adding regularization terms [Gulrajani

et al., 2017, Mescheder et al., 2017, 2018]. In this work, we try to characterize the difference in the landscapes induced by different GAN formulations and how it relates to improving the training of GANs.

Recently, Nagarajan and Kolter [2017], Mescheder et al. [2018] show that a local analysis of the eigenvalues of the Jacobian of the game can provide guarantees on local stability properties. However, their theoretical analysis is based on some unrealistic assumptions such as the generator's ability to fully capture the real distribution. In this work, we assess experimentally to what extent these theoretical stability results apply in practice.

Rotations in differentiable games has been mentioned and interpreted by [Mescheder et al., 2018, Balduzzi et al., 2018] and Gidel et al. [2019c]. While these papers address rotations in games from a theoretical perspective, it was never shown that GANs, which are games with highly non-convex losses, suffered from these rotations in practice. To our knowledge, trying to quantify that GANs actually suffer from this rotational component in practice for real world dataset is novel.

The stable points of the gradient dynamics in general games have been studied independently by Mazumdar et al. [2020], Daskalakis and Panageas [2018], Adolphs et al. [2018]. They notice that the locally stable stationary point of some games are not local Nash equilibria. In order to reach a local Nash equilibrium, Adolphs et al. [2018], Mazumdar et al. [2019] develop techniques based on second order information. In this work, we argue that reaching local Nash equilibria may not be as important as one may expect and that we do achieve good performance at a locally stable stationary point.

Several works have studied the loss landscape of deep neural networks. Goodfellow et al. [2015] proposed to look at the linear path between two points in parameter space and show that neural networks behave similarly to a convex loss function along this path. Draxler et al. [2018] proposed an extension where they look at nonlinear paths between two points and show that local minima are connected in deep neural networks. Another extension was proposed by [Li et al., 2018] where they use contour plots to look at the 2D loss surface defined by two directions chosen appropriately. In this paper, we use a similar approach of following the linear path between two points to gain insight about GAN optimization landscapes. However, in this context, looking at the loss of both players along that path may be uninformative. We propose instead to look, along a linear path from initialization to best solution, at the game vector field, particularly at its angle w.r.t. the linear path, the *Path-angle*.

Another way to gain insight into the landscape of deep neural networks is by looking at the Hessian of the loss; this was done in the context of single objective minimization by [Dauphin et al., 2014, Sagun et al., 2016, 2017, Alain et al., 2019]. Compared to linear path visualizations which can give global information (but only along one direction), the Hessian provides information about the loss landscape

in several directions but only locally. The full Hessian is expensive to compute and one often has to resort to approximations such has computing only the top-k eigenvalues. While, the Hessian is symmetric and thus has real eigenvalues, the Jacobian of a game vector field is significantly different since it is in general not symmetric, which means that the eigenvalues belong to the complex plane. In the context of GANs, Mescheder et al. [2017] introduced a gradient penalty and use the eigenvalues of the Jacobian of the game vector field to show its benefits in terms of stability. In our work, we compute these eigenvalues to assess that, on different GAN formulations and datasets, existing training procedures find a locally stable stationary point that is a saddle point for the loss function of the generator.

3 Formulations for GAN optimization and their practical implications

3.1 The standard game theory formulation

From a game theory point of view, GAN training may be seen as a game between two players: the discriminator $D_{\boldsymbol{\varphi}}$ and the generator $G_{\boldsymbol{\theta}}$, each of which is trying to minimize its loss $\mathcal{L}_D$ and $\mathcal{L}_G$, respectively. Using the same formulation as Mescheder et al. [2017], the GAN objective takes the following form (for simplicity of presentation, we focus on the unconstrained formulation):

$$\boldsymbol{\theta}^* \in \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^p} \mathcal{L}_G(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) \qquad \text{and} \qquad \boldsymbol{\varphi}^* \in \arg\min_{\boldsymbol{\varphi}\in\mathbb{R}^d} \mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi})\,. \tag{3.1}$$

The solution $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ is called a *Nash equilibrium* (NE). In practice, the considered objectives are non-convex and we typically cannot expect better than a *local* Nash equilibrium (LNE), i.e. a point at which (3.1) is only locally true (see e.g. [Adolphs et al., 2018] for a formal definition). Ratliff et al. [2016] derived some derivative-based necessary and sufficient conditions for being a LNE. They show that, for being a local NE it is sufficient to be a *differential Nash equilibrium*:

Definition 1 (Differential NE)**.** *A point* $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ *is a* differential Nash equilibrium *(DNE) iff*

$$\|\nabla_{\boldsymbol{\theta}}\mathcal{L}_G(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = \|\nabla_{\boldsymbol{\varphi}}\mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)\| = 0\,,\ \nabla^2_{\boldsymbol{\theta}}\mathcal{L}_G(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \succ 0 \text{ and } \nabla^2_{\boldsymbol{\varphi}}\mathcal{L}_D(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \succ 0 \tag{3.2}$$

where $\boldsymbol{S} \succ 0$ *if and only if* $\boldsymbol{S}$ *is positive definite.*

Being a DNE is not necessary for being a LNE because a local Nash equilibrium may have Hessians that are only semi-definite. NE are commonly used in GANs to describe the goal of the learning procedure [Goodfellow et al., 2014]: in this definition, $\boldsymbol{\theta}^*$ (resp. $\boldsymbol{\varphi}^*$) is seen as a local minimizer of $\mathcal{L}_G(\cdot, \boldsymbol{\varphi}^*)$ (resp. $\mathcal{L}_D(\boldsymbol{\theta}^*, \cdot)$).

Under this view, however, the interaction between the two networks is not taken into account. This is an important aspect of the game stability that is missed in the definition of DNE (and Nash equilibrum in general). We illustrate this point in the following section, where we develop an example of a game for which gradient methods converge to a point which is a saddle point for the generator's loss and thus not a DNE for the game.

3.2 An alternative formulation based on the game vector field

In practice, GANs are trained using first order methods that compute the gradients of the losses of each player. Following Gidel et al. [2019b], an alternative point of view on optimizing GANs is to jointly consider the players' parameters $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ as a joint state $\boldsymbol{\omega} := (\boldsymbol{\theta}, \boldsymbol{\varphi})$, and to study the vector field associated with these gradients,[1] which we call the *game vector field*

$$\boldsymbol{v}(\boldsymbol{\omega}) := \begin{bmatrix} \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\omega})^\top & \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\omega})^\top \end{bmatrix}^\top \quad \text{where} \quad \boldsymbol{\omega} := (\boldsymbol{\theta}, \boldsymbol{\varphi})\,. \tag{3.3}$$

With this perspective, the notion of DNE is replaced by the notion of locally stable stationary point (LSSP). Verhulst [1989, Theorem 7.1] defines a LSSP $\boldsymbol{\omega}^*$ using the eigenvalues of the Jacobian of the game vector field $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ at that point.

Definition 2 (LSSP)**.** *A point $\boldsymbol{\omega}^*$ is a* locally stable stationary point *(LSSP) iff*

$$\boldsymbol{v}(\boldsymbol{\omega}^*) = 0 \qquad \textit{and} \qquad \Re(\lambda) > 0\,, \quad \forall \lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))\,. \tag{3.4}$$

where $\Re$ denote the real part of the eigenvalue λ belonging to the spectrum of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$.

This definition is not easy to interpret but one can intuitively understand a LSSP as a stationary point (a point $\boldsymbol{\omega}^*$ where $\boldsymbol{v}(\boldsymbol{\omega}^*) = 0$) to which all neighbouring points are attracted. We will formalize this intuition of attraction in Proposition 1. In our two-player game setting, the Jacobian of the game vector field around the LSSP has the following block-matrices form:

$$\nabla \boldsymbol{v}(\boldsymbol{\omega}^*) = \begin{bmatrix} \nabla^2_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\omega}^*) & \nabla_{\boldsymbol{\varphi}} \nabla_{\boldsymbol{\theta}} \mathcal{L}_G(\boldsymbol{\omega}^*) \\ \nabla_{\boldsymbol{\theta}} \nabla_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\omega}^*) & \nabla^2_{\boldsymbol{\varphi}} \mathcal{L}_D(\boldsymbol{\omega}^*) \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{B} \\ \boldsymbol{A} & \boldsymbol{S}_2 \end{bmatrix}. \tag{3.5}$$

When $\boldsymbol{B} = -\boldsymbol{A}^\top$, being a DNE is a sufficient condition for being of LSSP [Daskalakis and Panageas, 2018]. However, some LSSP may not be DNE [Adolphs et al., 2018], meaning that the optimal generator $\boldsymbol{\theta}^*$ could be a saddle point of $\mathcal{L}_G(\cdot, \boldsymbol{\varphi}^*)$, while the optimal joint state $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ may be a LSSP of the game. We summarize these properties in Table 10.1. In order to illustrate the

[1] Note that, in practice, the joint vector field (3.3) is *not* a gradient vector field, i.e., it cannot be rewritten as the gradient of a single function.

Zero-sum game	Non-zero-sum game
DNE $\Rightarrow$ LSSP [Daskalakis and Panageas, 2018]	DNE $\not\Rightarrow$ LSSP (Example 5, in Appendix D §1
DNE $\not\Leftarrow$ LSSP [Adolphs et al., 2018] [Daskalakis and Panageas, 2018]	DNE $\not\Leftarrow$ LSSP (Example 4)

Table 10.1: Summary of the implications between Differentiable Nash Equilibrium (DNE) and a locally stable stationnary point (LSSP): in general, being a DNE is neither necessary or sufficient for being a LSSP.

intuition behind this counter-intuitive fact, we study a simple example where the generator is 2D and the discriminator is 1D.

Example 4. *Let us consider $\mathcal{L}_G$ as a hyperbolic paraboloid (a.k.a., saddle point function) centered in $(1,1)$ where $(1,\varphi)$ is the principal descent direction and $(-\varphi,1)$ is the principal ascent direction, while $\mathcal{L}_D$ is a simple bilinear objective.*

$$\mathcal{L}_G(\theta_1,\theta_2,\varphi) = (\theta_2 - \varphi\theta_1 - 1)^2 - \tfrac{1}{2}(\theta_1 + \varphi\theta_2 - 1)^2\,, \quad \mathcal{L}_D(\theta_1,\theta_2,\varphi) = \varphi(5\theta_1 + 4\theta_2 - 9)$$

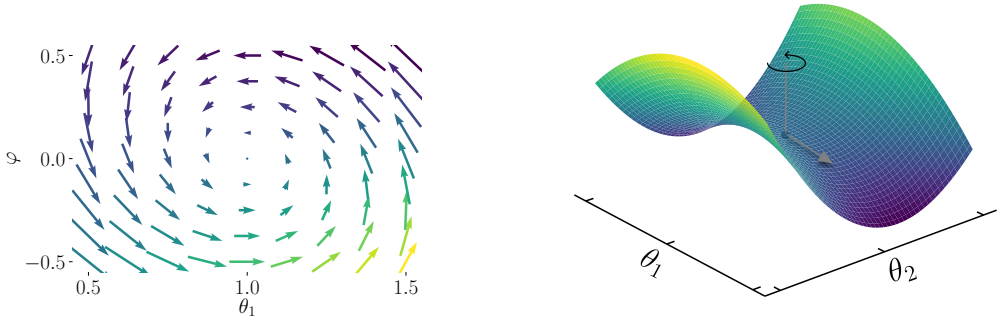
We plot $\mathcal{L}_G$ in Fig. 10.1b. Note that the discriminator φ controls the principal descent direction of $\mathcal{L}_G$.

We show (see Appendix D §1.2) that $(\theta_1^*, \theta_2^*, \varphi^*) = (1,1,0)$ is a locally stable stationary point but is not a DNE: the generator loss at the optimum $(\theta_1,\theta_2) \mapsto \mathcal{L}_G(\theta_1,\theta_2,\varphi^*) = \theta_2^2 - \frac{1}{2}\theta_1^2$ is not at a DNE because it has a clear descent direction, $(1,0)$. However, if the generator follows this descent direction, the dynamics will remain stable because the discriminator will update its parameter, rotating the saddle and making $(1,0)$ an ascent direction. We call this phenomenon *dynamic stability*: the loss $\mathcal{L}_G(\cdot,\varphi^*)$ is unstable for a fixed φ^* but becomes stable when φ dynamically interacts with the generator around φ^*.

A mechanical analogy for this dynamic stability phenomenon is a ball in a rotating saddle—even though the gravity pushes the ball to escape the saddle, a quick enough rotation of the saddle would trap the ball at the center (see [Thompson et al., 2002] for more details). This analogy has been used to explain Paul's trap [Paul, 1990]: a counter-intuitive way to trap ions using a dynamic electric field. In Example 4, the parameter φ explicitly controls the rotation of the saddle.

This example illustrates the fact that the DNE corresponds to a notion of *static stability*: it is the stability of one player's loss given the other player is fixed. Conversely, LSSP captures a notion of *dynamic stability* that considers both players jointly.

By looking at the game vector field we capture these interactions. Fig. 10.1b only captures a snapshot of the generator's loss surface for a fixed φ and indicates static instability (the generator is at a saddle point of its loss). In Fig. 10.1a, however, one can see that, starting from any point, we will rotate around the stationary point $(\varphi^*, \theta_1^*) = (0,1)$ and eventually converge to it.



(a) 2D projection of the vector field.

(b) Landscape of the generator loss.

Figure 10.1: Visualizations of Example 4. Left: projection of the game vector field on the plane $\theta_2 = 1$. Right: Generator loss. The descent direction is $(1, \varphi)$ (in grey). As the generator follows this descent direction, the discriminator changes the value of φ, making the saddle rotate, as indicated by the circular black arrow.

The visualization of the game vector field reveals an interesting behavior that does not occur in single objective minimization: close to a LSSP, the parameters rotate around it. Understanding this phenomenon is key to grasp the optimization difficulties arising in games. In the next section, we formally characterize the notion of rotation around a LSSP and in §4 we develop tools to visualize it in high dimensions. Note that gradient methods may converge to saddle points in single objective minimization, but these are not *stable* stationary points, unlike in our game example.

3.3 Rotation and attraction around locally stable stationary points in games

In this section, we formalize the notions of rotation and attraction around LSSP in games, which we believe may explain some difficulties in GAN training. The local stability of a LSSP is characterized by the eigenvalues of the Jacobian $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ because we can linearize $\boldsymbol{v}(\boldsymbol{\omega})$ around $\boldsymbol{\omega}^*$:

$$\boldsymbol{v}(\boldsymbol{\omega}) \approx \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)(\boldsymbol{\omega} - \boldsymbol{\omega}^*). \tag{3.6}$$

If we assume that (3.6) is an equality, we have the following theorem.

Proposition 1. *Let us assume that* (3.6) *is an equality and that* $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ *is diagonalizable, then there exists a basis* $\boldsymbol{P}$ *such that the coordinates* $\tilde{\boldsymbol{\omega}}_j(t) := [\boldsymbol{P}(\boldsymbol{\omega}(t) - \boldsymbol{\omega}^*)]_j$ *where* $\boldsymbol{\omega}(t)$ *is a solution of* (3.6) *have the following behavior: for* $\lambda_j \in \operatorname{Sp} \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ *we have,*

1. If $\lambda_j \in \mathbb{R}$, *we observe* pure attraction*:* $\quad \tilde{\boldsymbol{\omega}}_j(t) = e^{-\lambda_j t} \tilde{\boldsymbol{\omega}}_j(0)$.

2. *If* $\Re(\lambda_j) = 0$*, we observe* pure rotation*:*

$$\begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(t) \\ \tilde{\boldsymbol{\omega}}_{j+1}(t) \end{bmatrix} = \begin{bmatrix} \cos|\lambda_j t| & \sin|\lambda_j t| \\ -\sin|\lambda_j t| & \cos|\lambda_j t| \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(0) \\ \tilde{\boldsymbol{\omega}}_{j+1}(0) \end{bmatrix}.$$

3. *Otherwise, we observe both:*

$$\begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(t) \\ \tilde{\boldsymbol{\omega}}_{j+1}(t) \end{bmatrix} = e^{-\operatorname{Re}(\lambda_j)t} \begin{bmatrix} \cos\operatorname{Im}(\lambda_j t) & \sin\operatorname{Im}(\lambda_j t) \\ -\sin\operatorname{Im}(\lambda_j t) & \cos\operatorname{Im}(\lambda_j t) \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(0) \\ \tilde{\boldsymbol{\omega}}_{j+1}(0) \end{bmatrix}.$$

Note that we re-ordered the eigenvalues such that the complex conjugate eigenvalues form pairs: if $\lambda_j \notin \mathbb{R}$ *then* $\lambda_{j+1} = \bar{\lambda}_j$*.*

Matrices in 2. and 3. are rotations matrices. They induce a rotational behavior illustrated in Fig 10.1a.

This proposition shows that the dynamics of $\boldsymbol{\omega}(t)$ can be decomposed in a particular basis into attractions and rotations over components that do not interact between each other. Rotation does not appear in single objective minimization around a local minimum, because the eigenvalues of the Hessian of the objective are always real. Mescheder et al. [2017] discussed that difficulties in training GANs may be a result of the imaginary part of the eigenvalues of the Jacobian of the game vector field and Gidel et al. [2019c] mentioned that games have a natural oscillatory behavior. This cyclic behavior has been explained in [Balduzzi et al., 2018] by a non-zero Hamiltonian component in the Helmholtz decomposition of the Jacobian of the game vector field. All these explanations are related to the spectral properties of this Jacobian. The goal of Proposition 1 is to provide a formal definition to the notions of *rotation* and *attraction* we are dealing with in this paper.

In the following section, we introduce a new tool in order to assess the magnitude of the rotation around a LSSP compared to the attraction to this point.

4 Visualization for the vector field landscape

Neural networks are parametrized by a large number of variables and visualizations are only possible using low dimensional plots (1D or 2D). We first present a standard visualization tool for deep neural network loss surfaces that we will exploit in §4.2.

4.1 Standard visualizations for the loss surface

One way to visualize a neural network's loss landscape is to follow a parametrized path $\boldsymbol{\omega}(\alpha)$ that connects two parameters $\boldsymbol{\omega}, \boldsymbol{\omega}'$ (often one is chosen early in learning and another one is chosen late in learning, close to a solution). A path is a continuous function $\boldsymbol{\omega}(\cdot)$ such that $\boldsymbol{\omega}(0) = \boldsymbol{\omega}$ and $\boldsymbol{\omega}(1) = \boldsymbol{\omega}'$. Goodfellow et al. [2015] considered a linear path $\boldsymbol{\omega}(\alpha) = \alpha\boldsymbol{\omega} + (1-\alpha)\boldsymbol{\omega}'$. More complex paths can be considered to assess whether different minima are connected [Draxler et al., 2018].

4.2 Proposed visualization: Path-angle

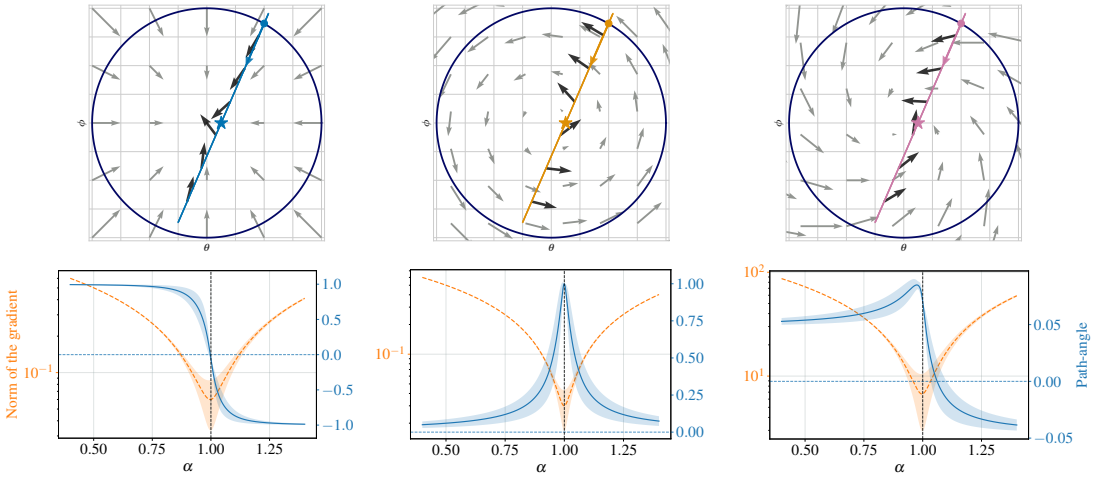
We propose to study the linear path between parameters early in learning and parameters late in learning. We illustrate the extreme cases for the game vector field along this path in simple examples in Figure 10.2(a-c): pure attraction occurs when the vector field perfectly points to the optimum (Fig. 10.2a) and pure rotation when the vector field is orthogonal to the direction to the optimum (Fig. 10.2b). In practice, we expect the vector field to be in between these two extreme cases (Fig. 10.2c). In order to determine in which case we are, around a LSSP, in practice, we propose the following tools.

Path-norm.. We first ensure that we are in a neighborhood of a stationary point by computing the norm of the vector field. Note that considering independently the norm of each player may be misleading: even though the gradient of one player may be close to zero, it does not mean that we are at a stationary point since the other player might still be updating its parameters. **Path-angle.**. Once we are close to a final point $\boldsymbol{\omega}'$, i.e., in a neighborhood of a LSSP, we propose to look at the angle between the vector field (3.3) and the linear path from $\boldsymbol{\omega}$ to $\boldsymbol{\omega}'$. Specifically, we monitor the cosine of this angle, a quantity we call *Path-angle*:

$$c(\alpha) := \frac{\langle \boldsymbol{\omega}' - \boldsymbol{\omega}, \boldsymbol{v}_\alpha \rangle}{\|\boldsymbol{\omega}' - \boldsymbol{\omega}\| \|\boldsymbol{v}_\alpha\|} \quad \text{where} \quad \boldsymbol{v}_\alpha := \boldsymbol{v}(\alpha\boldsymbol{\omega}' + (1-\alpha)\boldsymbol{\omega}) \,, \; \alpha \in [a, b] \,. \tag{4.1}$$

Usually $[a, b] = [0, 1]$, but since we are interested in the landscape around a LSSP, it might be more informative to also consider further extrapolated points around $\boldsymbol{\omega}'$ with $b > 1$.

Eigenvalues of the Jacobian.. Another important tool to gain insights on the behavior close to a LSSP, as discussed in §3.2, is to look at the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$. We propose to compute the top-k eigenvalues of this Jacobian. When all the eigenvalues have positive real parts, we conclude that we have reached a LSSP, and if some eigenvalues have large imaginary parts, then the game has a strong rotational behavior (Thm. 1). Similarly, we can also compute the top-k eigenvalues of the diagonal blocks of the Jacobian, which correspond to the Hessian of each player. These eigenvalues can inform us on whether we have converged to a LSSP that is not a LNE.



(a) Attraction only **(b)** Rotation only **(c)** Rotation and attraction

Figure 10.2: Above: game vector field (in grey) for different archetypal behaviors. The equilibrium of the game is at $(0, 0)$. Black arrows correspond to the directions of the vector field at different linear interpolations between two points: $\bullet$ and $\star$. **Below**: path-angle $c(\alpha)$ for different archetypal behaviors (right y-axis, in blue). The left y-axis in orange correspond to the norm of the gradients. Notice the "bump" in path-angle (close to $\alpha = 1$), characteristic of rotational dynamics.

An important advantage of the Path-angle relative to the computation of the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ is that it only requires computing gradients (and not second order derivatives, which may be prohibitively computationally expensive for deep networks). Also, it provides information along a whole path between two points and thus, more global information than the Jacobian computed at a single point. In the following section, we use the Path-angle to study the archetypal behaviors presented in Thm 1.

4.3 Archetypal behaviors of the Path-angle around a LSSP

Around a LSSP, we have seen in (3.6) that the behavior of the vector field is mainly dictated by the Jacobian matrix $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$. This motivates the study of the behavior of the Path-angle $c(\alpha)$ where the Jacobian is a constant matrix:

$$\boldsymbol{v}(\boldsymbol{\omega}) = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{B} \\ \boldsymbol{A} & \boldsymbol{S}_2 \end{bmatrix} (\boldsymbol{\omega} - \boldsymbol{\omega}^*) \quad \text{and thus} \quad \nabla \boldsymbol{v}(\boldsymbol{\omega}) = \begin{bmatrix} \boldsymbol{S}_1 & \boldsymbol{B} \\ \boldsymbol{A} & \boldsymbol{S}_2 \end{bmatrix} \quad \forall \boldsymbol{\omega}\,. \tag{4.2}$$

Depending on the choice of $\boldsymbol{S}_1, \boldsymbol{S}_2, \boldsymbol{A}$ and $\boldsymbol{B}$, we cover the following cases:

- $\boldsymbol{S}_1, \boldsymbol{S}_2 \succ 0, \boldsymbol{A} = \boldsymbol{B} = 0$: eigenvalues are real. Thm. 1 ensures that we only have *attraction*. Far from $\boldsymbol{\omega}^*$, the gradient points to $\boldsymbol{\omega}^*$ (See Fig. 10.2a) and thus $c(\alpha) = 1$ for $\alpha \ll 1$ and $c(\alpha) = -1$ for $\alpha \gg 1$. Since $\boldsymbol{\omega}'$ is not exactly $\boldsymbol{\omega}^*$, we observe a *quick sign switch* of the Path-angle around $\alpha = 1$. We plotted

the average Path-angle over different approximate optima in Fig. 10.2a (see appendix for details).

- $\boldsymbol{S}_1, \boldsymbol{S}_2 = 0, \boldsymbol{A} = -\boldsymbol{B}^\top$: eigenvalues are pure imaginary. Thm. 1 ensures that we only have *rotations*. Far from the optimum the gradient is orthogonal to the direction that points to $\boldsymbol{\omega}$ (See Fig. 10.2b). Thus, $c(\alpha)$ vanishes for $\alpha \ll 1$ and $\alpha \gg 1$. Because $\boldsymbol{\omega}'$ is not exactly $\boldsymbol{\omega}^*$, around $\alpha = 1$, the gradient is tangent to the circles induced by the rotational dynamics and thus $c(\alpha) = \pm 1$. That is why in Fig. 10.2b we observe a *bump* in $c(\alpha)$ when α is close to 1.

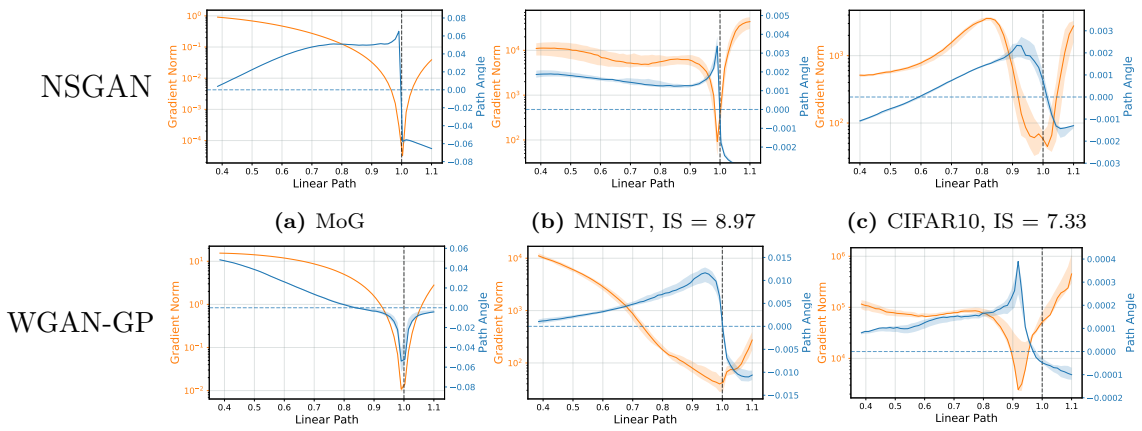
- General high dimensional LSSP (3.4). The dynamics display both attraction and rotation. We observe a combination of the sign switch due to the attraction and the bump due to the rotation. The higher the bump, the closer we are to pure rotations. Since we are performing a low dimensional visualization, we actually project the gradient onto our direction of interest. That is why the Path-angle is significantly smaller than 1 in Fig. 10.2c.

5 Numerical results on GANs

Losses. We focus on two common GAN loss formulations: we consider both the original non-saturating GAN (NSGAN) formulation proposed in Goodfellow et al. [2014] and the WGAN-GP objective described in Gulrajani et al. [2017].

Datasets. We first propose to train a GAN on a toy task composed of a 1D mixture of 2 Gaussians (MoG) with 10,000 samples. For this task both the generator and discriminator are neural networks with 1 hidden layer and ReLU activations. We also train a GAN on MNIST, where we use the DCGAN architecture [Radford et al., 2016] with spectral normalization(see Appendix D §3.2 for details). Finally we also look at the optimization landscape of a state of the art ResNet on CIFAR10 [Krizhevsky and Hinton, 2009].

Optimization methods. For the mixture of Gaussian (MoG) dataset, we used the full-batch extragradient method [Korpelevich, 1976, Gidel et al., 2019b]. We also tried to use standard batch gradient descent, but this led to unstable results indicating that gradient descent might indeed be unable to converge to stable stationary points due to the rotations (see Appendix D §3.4). On MNIST and CIFAR10, we tested both Adam [Kingma and Ba, 2015] and ExtraAdam [Gidel et al., 2019b]. The observations made on models trained with both methods are very similar. ExtraAdam gives slightly better performance in terms of inception score [Salimans et al., 2016], and Adam sometimes converge to unstable points, thus we decided to only include the observations on ExtraAdam, for more details on the observations on Adam (see §3.5). As recommended by Heusel et al. [2017], we



(a) MoG **(b)** MNIST, IS = 8.97 **(c)** CIFAR10, IS = 7.33

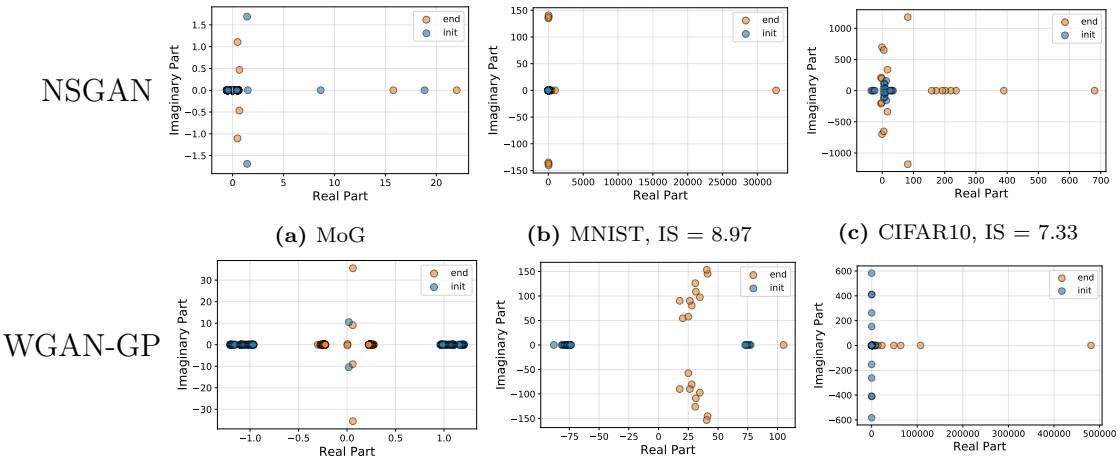
(d) MoG **(e)** MNIST, IS = 9.46 **(f)** CIFAR10, IS = 7.65

Figure 10.3: Path-angle for NSGAN (**top row**) and WGAN-GP (**bottom row**) trained on the different datasets, see Appendix D §3.3 for details on how the path-angle is computed. For MoG the ending point is a generator which has learned the distribution. For MNIST and CIFAR10 we indicate the Inception score (IS) at the ending point of the interpolation. Notice the "bump" in path-angle (close to $\alpha = 1.0$), characteristic of games rotational dynamics, and absent in the minimization problem (d). Details on error bars in Appendix D §3.3.

chose different learning rates for the discriminator and the generator. All the hyperparameters and precise details about the experiments can be found in Appendix D §3.1.

5.1 Evidence of rotation around locally stable stationary points in GANs

We first look, for all the different models and datasets, at the path-angles between a random initialization (initial point) and the set of parameters during training achieving the best performance (end point) (Fig. 10.3), and at the eigenvalues of the Jacobian of the game vector field for the same end point (Fig. 10.4). We're mostly interested in looking at the optimization landscape around LSSPs, so we first check if we are actually close to one. To do so we look at the gradient norm around the end point, this is shown by the orange curves in Fig.10.3, we can see that the norm of the gradient is quite small for all the models meaning that we are close to a stationary point. We also need to check that the point is stable, to do so we look at the eigenvalues of the Game in Fig. 10.4, if all the eigenvalues have positive real parts then the point is also stable. We observe that most of the time, the model has reached a LSSP. However we can see that this is not always the case, for example in Fig. 10.4d some of the eigenvalues have a negative real part. We still include those results since although the point is unstable it gives similar



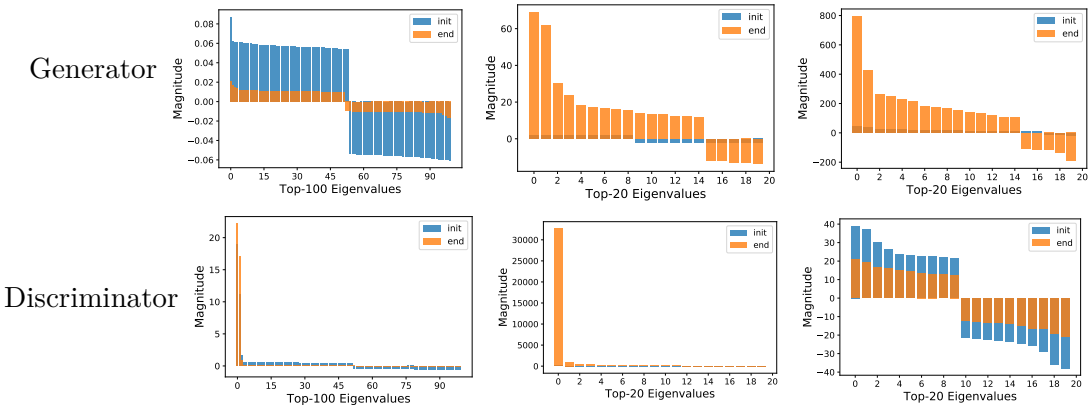
(a) MoG **(b)** MNIST, IS = 8.97 **(c)** CIFAR10, IS = 7.33

(d) MoG **(e)** MNIST, IS = 9.46 **(f)** CIFAR10, IS = 7.65

Figure 10.4: Eigenvalues of the Jacobian of the game for NSGAN (**top row**) and WGAN-GP (**bottom row**) trained on the different datasets. Large imaginary eigenvalues are characteristic of rotational behavior. Notice that NSGAN and WGAN-GP objectives lead to very different landscapes (see how the eigenvalues of WGAN-GP are shifted to the right of the imaginary axis). This could explain the difference in performance between NSGAN and WGAN-GP.

performance to a LSSP.

Our first observation is that all the GAN objectives on both datasets have a non zero rotational component. This can be seen by looking at the Path-angle in Fig. 10.3, where we always observe a bump, and this is also confirmed by the large imaginary part in the eigenvalues of the Jacobian in Fig. 10.4. The rotational component is clearly visible in Fig. 10.3d, where we see no sign switch and a clear bump similar to Fig. 10.2b. On MNIST and CIFAR10, with NSGAN and WGAN-GP (see Fig. 10.3), we observe a combination of a bump and a sign switch similar to Fig. 10.2c. Also Fig. 10.4 clearly shows the existence of imaginary eigenvalues with large magnitude. Fig. 10.4c and 10.4e. We can see that while almost all models exhibit rotations, the distribution of the eigenvalues are very different. In particular the complex eigenvalues for NSGAN seems to be much more concentrated on the imaginary axis while WGAN-GP tends to spread the eigenvalues towards the right of the imaginary axis Fig. 10.4e. This shows that different GAN objectives can lead to very different landscapes, and has implications in terms of optimization, in particular that might explain why WGAN-GP performs slightly better than NSGAN.

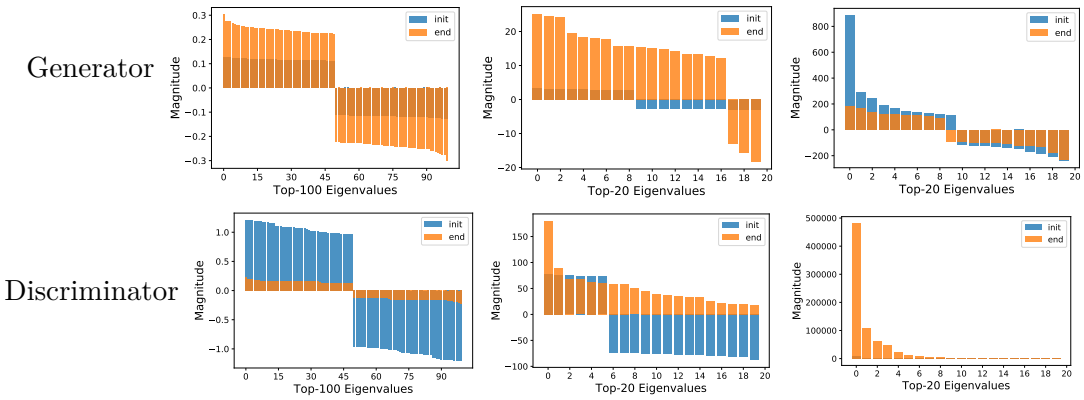


(a) MoG **(b)** MNIST, IS = 8.97 **(c)** CIFAR10, IS = 7.33

Figure 10.5: NSGAN. Top k-Eigenvalues of the Hessian of each player (in terms of magnitude) in descending order. Top Eigenvalues indicate that the Generator does not reach a local minimum but a saddle point (for CIFAR10 actually both the generator and discriminator are at saddle points). Thus the training algorithms converge to LSSPs which are not Nash equilibria.

5.2 The locally stable stationary points of GANs are not local Nash equilibria

As mentioned at the beginning of §5.1, the points we are considering are most of the times LSSP. To check if these points are also local Nash equilibria (LNE) we compute the eigenvalues of the Hessian of each player independently. If all the eigenvalues of each player are positive, it means that we have reached a DNE. Since the computation of the full spectrum of the Hessians is expensive, we restrict ourselves to the top-k eigenvalues with largest magnitude: exhibiting one significant negative eigenvalue is enough to indicate that the point considered is not in the neighborhood of a LNE. Results are shown in Fig. 10.5 and Fig. 10.6, from which we make several observations. First, we see that the generator never reaches a local minimum but instead finds a saddle point. This means that the algorithm converges to a LSSP which is not a LNE, while achieving good results with respect to our evaluation metrics. This raises the question whether convergence to a LNE is actually needed or if converging to a LSSP is sufficient to reach a good solution. We also observe a large difference in the eigenvalues of the discriminator when using the WGAN-GP v.s. the NSGAN objective. In particular, we find that the discriminator in NSGAN converges to a solution with very large positive eigenvalues compared to WGAN-GP. This shows that the discriminator in NSGAN converges to a much sharper minimum. This is consistent with the fact that the gradient penalty acts as a regularizer on the discriminator and prevents it from becoming too sharp.



(a) MoG **(b)** MNIST, IS = 9.46 **(c)** CIFAR10, IS = 7.65

Figure 10.6: WGAN-GP. Top k-Eigenvalues of the Hessian of each player (in terms of magnitude) in descending order. Top Eigenvalues indicate that the Generator does not reach a local minimum but a saddle point. Thus the training algorithms converge to LSSPs which are not Nash equilibria.

6 Discussion

Across different GAN formulations, standard optimization methods and datasets, we consistently observed that GANs do not converge to local Nash equilibria. Instead the generator often ends up being at a saddle point of the generator loss function. However, in practice, these LSSP achieve really good generator performance metrics, which leads us to question whether we need a Nash equilibrium to get a generator with good performance in GANs and whether such DNE with good performance does actually exist. Moreover, we have provided evidence that the optimization landscapes of GANs typically have rotational components specific to games. We argue that these rotational components are part of the reason why GANs are challenging to train, in particular that the instabilities observed during training may come from such rotations close to LSSP. It shows that simple low dimensional examples, such as for instance Dirac GAN, does capture some of the arising challenges for training large scale GANs, thus, motivating the practical use of method able to handle strong rotational components, such as extragradient [Gidel et al., 2019b], averaging [Yazıcı et al., 2019], optimism [Daskalakis et al., 2018] or gradient penalty based methods [Mescheder et al., 2017, Gulrajani et al., 2017].

11 Conclusions, Discussions, and Perspectives

1 Summary and Conclusions

In this thesis, we have investigated several aspects of learning in multi-player games. In the first contribution, we tackled the issue of the existence of equilibrium in nonconvex-nonconcave games by introducing a novel realistic assumption. In practice, although the payoff function may be nonconvex-nonconcave with respect to the model parameters, it is convex-concave as a function of the models themselves.

The investigation of the notion of equilibria in games has recently been an active research topic in the community. For instance, defining what is local optimality in games is a complex question that has been examined by [Jin et al., 2019, Zhang et al., 2020]. Alternative notions of equilibria such as Stackelberg equilibria have been studied by Fiez et al. [2020]. To our knowledge, our theorem is a novel existence result of equilibrium for nonconvex-nonconcave payoffs that contrast with the counter-example proposed by Jin et al. [2019]. Our work's main takeaway is that games like GANs of Starcraft II have an equilibrium because they satisfy the specific assumptions presented in the first contribution.

The rest of this thesis focus on the optimization of games. In the second contribution, we proposed a principled optimization perspective for GANs based on the variational inequality literature. In that work, we introduced a novel optimization algorithm leveraging the idea of extrapolation, providing state-of-the-art results on GANs at the time.

In the third contribution, we used linear algebra tools to analyze the impact of momentum in the optimization of games. One of the conclusions drawn from this work was relatively surprising: unlike in standard minimization for which the optimal momentum hyperparameter is close to 1, in games, the optimal hyperparameter can be negative.

At the time of publication of the second and third publication, the theoretical study of the optimization of games in the context of machine learning was at its premises [Gidel et al., 2017, Palaniappan and Bach, 2016, Mescheder et al., 2017, Daskalakis et al., 2018]. In these works we have identified that the simple bilinear unconstrained matrix game was an interesting case study. We showed, among other things, that the extragradient method and the alternating gradient method with negative momentum were converging linearly on that problem.

The analysis of this simple example, also proposed by [Daskalakis et al., 2018],

opened a rich research direction and many publications subsequently used that case study to test and develop new optimization methods for games [Abernethy et al., 2019, Chavdarova et al., 2019, Yazıcı et al., 2019, Mokhtari et al., 2020, Ibrahim et al., 2020, Azizian et al., 2020a,b]. Moreover, it sheds into light the fact that, in minimax games, some first-order methods can convergence linearly even in the absence of *strong convex-concavity*.

In the final contribution, we proposed an empirical study of the landscape of the optimization vector field of practical GANs. The study of the practical landscapes was not new at the time [Choromanska et al., 2015, Kawaguchi, 2016, Li et al., 2018].

The novelty of this work comes from the fact that we focused on games that are multi-objective optimization problems. We argued that the tools developed for the study of standard minimization landscapes were not suited to the games' optimization landscape. We gave strong empirical evidence that the landscape of GANs had some rotational components that cannot occur in standard minimization. Such observation is a strong argument in favor of principled methods to tackle rotations such as extrapolation or negative momentum. Moreover, we also provided empirical evidence that standard training methods were converging to stationary points that are locally stable but not local Nash equilibria. The theoretical existence of such locally stable stationary points that are not local Nash equilibria has been previously noticed by Mazumdar et al. [2020], Adolphs et al. [2018], Daskalakis and Panageas [2018]. However, there was no prior evidence that it could happen in practice.

Since Nash equilibria (and their local version) is the standard notion of equilibrium in games, this phenomenon is relatively counter-intuitive and raises the question of whether (local) Nash equilibria are needed to achieve good performance in GANs.

2 Discussions and Perspectives

While this thesis provides significant progress toward understanding multi-player games in the context of machine learning, important open questions remain. We will divide these questions into three categories: a theoretical study of games in ML, optimization of games, and the design of new practical formulations at the intersection of games and ML.

Theoretical study of games in ML. Many ML applications involve games with more than two players such as Diplomacy [Kraus and Lehmann, 1988, Kraus et al., 1989, Paquette et al., 2019, Anthony et al., 2020], DOTA [Berner et al., 2019], Texas Hold'em poker [Brown and Sandholm, 2019] or market designs [Lay and Barbu, 2010, Balduzzi, 2014]. Understanding what is (local) optimality in such complex games where each player has a non-convex loss is a fascinating question.

In the context of ML, an interesting notion to explore is the concept of coarse correlated equilibria (CCE) [Aumann, 1974]. CCE are a compelling notion because they can be computed in polynomial time for many compactly represented multi-player games [Papadimitriou, 2007] and because such notion is more relevant than Nash equilibria in cooperation games such as the prisoner dilemma [Axelrod and Hamilton, 1981].

Optimization of games. Optimization with first-order methods is quite well understood in the context of minimization. For strongly convex and Lipschitz objective, the community has defined a notion of *conditioning* of the problem. The condition number is a value that represents the difficulty of the problem (in terms of optimization). A classical result [Nesterov, 2004] shows that first-order methods *cannot* converge faster than a geometric rate depending on the condition number of the problem. Method that achieves this lower bound has been proposed by [Polyak, 1964] and Nesterov [1983]. However, in minimax games, and more generally in multi-player games, the gap between lower and upper convergence bound is far from being bridged [Azizian et al., 2020a, Ibrahim et al., 2020, Zhang et al., 2019]. One key component to take into account in the conditioning of a game is that some methods such as extragradient converge linearly even in the absence of strong convex-concavity [Tseng, 1995, Azizian et al., 2020a]. Thus, the conditioning of a game must incorporate a notion of interaction between the players.

Another challenging technical problem in game optimization is the notion of merit function. Actually, as discussed in § 2.4, the standard merit function G defined in (2.7) may be infinite almost everywhere when dealing with unbounded constraints sets. In that case, a convergence analysis using that merit function is not possible. Thus, one interesting research direction is to identify new gap functions to analyze such optimization problems. One way would be to use a *perturbed gap function* such as the one used by [Monteiro and Svaiter, 2011].

One important question remaining is the problem of convergence guarantees in the context of nonconvex-nonconcave minimax optimization (and more generally in the context of multi-player games with non-convex payoffs). Unlike in non-convex minimization, where a notion of global guarantee can be obtained, in games, there is currently no proof of convergence guarantees for a first-order algorithm (see §1.2 for more details). Some negative results have been recently provided by Letcher [2020] and Hsieh et al. [2020]. However, as argued in the first contribution, the multi-player games encountered in practical ML applications have a particular structure, exhibited in the first contribution, that could potentially be leveraged to prove some positive results.

Design of new game for ML formulations. Generative adversarial networks [Goodfellow et al., 2014] and adversarial training [Madry et al., 2018] are arguably the two most popular adversarial training formulation in machine learning. However, such formulations' potential is significant and could be leveraged to tackle challenging ML problems such as the inference of causal relationship [Ar-

jovsky et al., 2019, Ahuja et al., 2020] or the generation of adversarial examples with no interaction with the attacked model [Bose et al., 2020].

Another interesting direction to explore is the design of cooperation games using ML. The machine learning community is just starting to get interested in such challenges [Bard et al., 2020, Foerster et al., 2019], though the game theory literature has studied the notion of cooperation in games decades ago [Axelrod and Hamilton, 1981] and could provide many case studies and insights to design cooperative games for ML.

Mot de la fin. The problems presented here are fascinating challenges that can constitute rich and diverse research directions for many years but also push forward the understanding of how humans think and learn.

Bibliography

J. Abernethy, K. A. Lai, and A. Wibisono. Last-iterate convergence rates for min-max optimization. *arXiv preprint arXiv:1906.02027*, 2019.

L. Adolphs, H. Daneshmand, A. Lucchi, and T. Hofmann. Local saddle point optimization: A curvature exploitation approach. *arXiv preprint arXiv:1805.05751*, 2018.

A. Ahmadinejad, S. Dehghani, M. Hajiaghayi, B. Lucier, H. Mahini, and S. Seddighin. From duels to battlefields: Computing equilibria of Blotto and other games. *Mathematics of Operations Research*, 2019.

K. Ahuja, K. Shanmugam, K. Varshney, and A. Dhurandhar. Invariant risk minimization games. *arXiv preprint arXiv:2002.04692*, 2020.

G. Alain, N. Le Roux, and P.-A. Manzagol. Negative eigenvalues of the hessian in deep neural networks. *arXiv*, 2019.

C. D. Aliprantis and K. C. Border. *Infinite Dimensional Analysis A Hitchhiker's Guide*. Springer, 2006.

T. Anthony, T. Eccles, A. Tacchetti, J. Kramár, I. Gemp, T. C. Hudson, N. Porcel, M. Lanctot, J. Pérolat, R. Everett, et al. Learning to play no-press diplomacy with best response policy iteration. *arXiv preprint arXiv:2006.04635*, 2020.

M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.

M. Arjovsky, L. Bottou, I. Gulrajani, and D. Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2019.

S. Arora, R. Ge, Y. Liang, T. Ma, and Y. Zhang. Generalization and equilibrium in generative adversarial nets (GANs). In *ICML*, 2017.

K. J. Arrow, H. Azawa, L. Hurwicz, and H. Uzawa. *Studies in linear and non-linear programming*, volume 2. Stanford University Press, 1958.

K. E. Atkinson. *An introduction to numerical analysis*. John Wiley & Sons, 2003.

R. J. Aumann. Subjectivity and correlation in randomized strategies. *Journal of mathematical Economics*, 1(1):67–96, 1974.

R. Axelrod and W. D. Hamilton. The evolution of cooperation. *science*, 211(4489): 1390–1396, 1981.

W. Azizian, I. Mitliagkas, S. Lacoste-Julien, and G. Gidel. A tight and unified analysis of extragradient for a whole spectrum of differentiable games. In *AISTATS*, 2020a.

W. Azizian, D. Scieur, I. Mitliagkas, S. Lacoste-Julien, and G. Gidel. Accelerating smooth games by manipulating spectral shapes. In *AISTATS*, 2020b.

J. P. Bailey, G. Gidel, and G. Piliouras. Finite regret and cycles with fixed step-size via alternating gradient descent-ascent. In *COLT*, 2020.

D. Balduzzi. Cortical prediction markets. In *AAMAS*, 2014.

D. Balduzzi, S. Racaniere, J. Martens, J. Foerster, K. Tuyls, and T. Graepel. The mechanics of n-player differentiable games. In *ICML*, 2018.

J. F. Bard. *Practical bilevel optimization: algorithms and applications*, volume 30. Springer Science & Business Media, 2013.

N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, et al. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216, 2020.

J. Baxter, A. Tridgell, and L. Weaver. Learning to play chess using temporal differences. *Machine Learning*, 40(3):243–263, 2000.

M. Bellmore and G. L. Nemhauser. The traveling salesman problem: a survey. *Operations Research*, 16(3):538–558, 1968.

S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *Journal of Computer and System Sciences*, 66(3):496–514, 2003.

H. Berard, G. Gidel, A. Almahairi, P. Vincent, and S. Lacoste-Julien. A closer look at the optimization landscapes of generative adversarial networks. In *ICLR*, 2020.

C. Berner, G. Brockman, B. Chan, V. Cheung, P. Dębiak, C. Dennison, D. Farhi, Q. Fischer, S. Hashme, C. Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

D. P. Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

M. Bińkowski, J. Donahue, S. Dieleman, A. Clark, E. Elsen, N. Casagrande, L. C. Cobo, and K. Simonyan. High fidelity speech synthesis with adversarial networks. In *ICLR*, 2020.

E. Borel. La théorie du jeu et les équations intégrales à noyau symétrique. *Comptes rendus de l'Académie des Sciences*, 173(1304-1308):58, 1921.

A. J. Bose, G. Gidel, H. Berrard, A. Cianflone, P. Vincent, S. Lacoste-Julien, and W. L. Hamilton. Adversarial example games. *arXiv preprint arXiv:2007.00720*, 2020.

L. Bottou and O. Bousquet. The tradeoffs of large scale learning. In *NeurIPS*, pages 161–168, 2008.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

R. Bozulich. *The go player's almanac*. Ishi Press, 1992.

L. M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR computational mathematics and mathematical physics*, 7(3):200–217, 1967.

A. Brock, J. Donahue, and K. Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *ICLR*, 2019.

N. Brown and T. Sandholm. Safe and nested subgame solving for imperfect-information games. In *NeurIPS*, 2017.

N. Brown and T. Sandholm. Superhuman AI for multiplayer poker. *Science*, 2019.

R. E. Bruck. On the weak convergence of an ergodic iteration for the solution of variational inequalities for monotone operators in hilbert space. *Journal of Mathematical Analysis and Applications*, 1977.

M. Campbell, A. J. Hoane Jr, and F.-h. Hsu. Deep blue. *Artificial intelligence*, 134(1-2):57–83, 2002.

A. Cauchy. Méthode générale pour la résolution des systemes d'équations simultanées. *Comp. Rend. Sci. Paris*, 25(1847):536–538, 1847.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40 (1):120–145, 2011.

A. Chambolle and T. Pock. On the ergodic convergence rates of a first-order primal–dual algorithm. *Mathematical Programming*, 159(1-2):253–287, 2016.

T. Chavdarova, G. Gidel, F. Fleuret, and S. Lacoste-Julien. Reducing noise in gan training with variance reduced extragradient. In *NeurIPS*, pages 393–403, 2019.

G. H. Chen and R. T. Rockafellar. Convergence rates in forward–backward splitting. *SIAM Journal on Optimization*, 1997.

C.-K. Chiang, T. Yang, C.-J. Lee, M. Mahdavi, C.-J. Lu, R. Jin, and S. Zhu. Online optimization with gradual variations. In *COLT*, pages 6–1, 2012.

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *AISTATS*, pages 192–204, 2015.

B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of operations research*, 2007.

V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90, 2006.

A. A. Cournot. *Recherches sur les principes mathématiques de la théorie des richesses*. 1838.

G. P. Crespi, A. Guerraggio, and M. Rocca. Minty variational inequality and optimization: Scalar and vector case. In A. Eberhard, N. Hadjisavvas, and D. T. Luc, editors, *Generalized Convexity, Generalized Monotonicity and Applications*, 2005.

P. Dasgupta and E. Maskin. The existence of equilibrium in discontinuous economic games, i: Theory. *The Review of economic studies*, 53(1):1–26, 1986.

C. Daskalakis and I. Panageas. The limit points of (optimistic) gradient descent in min-max optimization. In *NeurIPS*, 2018.

C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

C. Daskalakis, A. Ilyas, V. Syrgkanis, and H. Zeng. Training GANs with optimism. In *ICLR*, 2018.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. In *NeurIPS*, 2014.

E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *NeurIPS*, 2015.

C. Domingo-Enrich, S. Jelassi, A. Mensch, G. Rotskoff, and J. Bruna. A mean-field analysis of two-player zero-sum games. *arXiv preprint arXiv:2002.06277*, 2020.

Z. Dou, X. Yan, D. Wang, and X. Deng. Finding mixed strategy nash equilibrium for continuous games through deep learning. *arXiv preprint arXiv:1910.12075*, 2019.

F. Draxler, K. Veschgini, M. Salmhofer, and F. Hamprecht. Essentially no barriers in neural network energy landscape. In *ICML*, 2018.

S. S. Du, C. Jin, J. D. Lee, M. I. Jordan, A. Singh, and B. Poczos. Gradient descent can take exponential time to escape saddle points. In *NeurIPS*, pages 1067–1077, 2017.

J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *JMLR*, 12(7), 2011.

A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun. Dermatologist-level classification of skin cancer with deep neural networks. *nature*, 542(7639):115–118, 2017.

K. Fan. Minimax theorems. *Proceedings of the National Academy of Sciences of the United States of America*, 1953.

W. Fedus, M. Rosca, B. Lakshminarayanan, A. M. Dai, S. Mohamed, and I. Goodfellow. Many paths to equilibrium: GANs do not need to decrease a divergence at every step. In *ICLR*, 2018.

V. Feldman, V. Guruswami, P. Raghavendra, and Y. Wu. Agnostic learning of monomials by halfspaces is hard. *SIAM Journal on Computing*, 41(6):1558–1590, 2012.

A. Ferdowsi, A. Sanjab, W. Saad, and T. Basar. Generalized colonel Blotto game. In *2018 Annual American Control Conference (ACC)*, 2018.

T. Fiez, B. Chasnov, and L. J. Ratliff. Implicit learning dynamics in stackelberg games: Equilibria characterization, convergence analysis, and empirical study, 2020.

J. Foerster, F. Song, E. Hughes, N. Burch, I. Dunning, S. Whiteson, M. Botvinick, and M. Bowling. Bayesian action decoder for deep multi-agent reinforcement learning. In *ICML*, 2019.

V. François-Lavet, P. Henderson, R. Islam, M. G. Bellemare, and J. Pineau. An introduction to deep reinforcement learning. *arXiv preprint arXiv:1811.12560*, 2018.

M. Fréchet. Commentary on the three notes of Emile Borel. *Econometrica: Journal of the Econometric Society*, pages 118–124, 1953.

Y. Freund, R. E. Schapire, et al. Adaptive game playing using multiplicative weights. *Games and Economic Behavior*, 1999.

M. Gardner. Mathematical games: The fantastic combinations of john conway's new solitaire game "life". *Scientific American*, 223(4):120–123, 1970.

M. Genesereth, N. Love, and B. Pell. General game playing: Overview of the AAAI competition. *AI magazine*, 26(2):62–62, 2005.

E. Ghadimi, H. R. Feyzmahdavian, and M. Johansson. Global convergence of the heavy-ball method for convex optimization. In *ECC*, 2015.

G. Gidel, T. Jebara, and S. Lacoste-Julien. Frank-Wolfe algorithms for saddle point problems. *AISTATS*, 2017.

G. Gidel, F. Pedregosa, and S. Lacoste-Julien. Frank-wolfe splitting via augmented lagrangian method. In *AISTATS*, 2018.

G. Gidel, F. Bach, and S. Lacoste-Julien. Implicit regularization of discrete gradient dynamics in linear neural networks. In *NeurIPS*, 2019a.

G. Gidel, H. Berard, P. Vincent, and S. Lacoste-Julien. A variational inequality perspective on generative adversarial nets. In *ICLR*, 2019b.

G. Gidel, R. A. Hemmat, M. Pezeshki, R. Lepriol, G. Huang, S. Lacoste-Julien, and I. Mitliagkas. Negative momentum for improved game dynamics. In *AISTATS*, 2019c.

I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

I. L. Glicksberg. A further generalization of the kakutani fixed point theorem, with application to nash equilibrium points. *Proceedings of the American Mathematical Society*, 3(1):170–174, 1952.

X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

I. Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv:1701.00160*, 2016.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NeurIPS*, 2014.

I. J. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. In *ICLR*, 2015.

S. Gould, B. Fernando, A. Cherian, P. Anderson, R. S. Cruz, and E. Guo. On differentiating parameterized argmin and argmax problems with application to bi-level optimization. *arXiv preprint arXiv:1607.05447*, 2016.

P. Grnarova, K. Y. Levy, A. Lucchi, T. Hofmann, and A. Krause. An online learning approach to generative adversarial networks. In *ICLR*, 2018.

I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *NeurIPS*, 2017.

J. Y. Halpern and R. Pass. Algorithmic rationality: Game theory with costly computation. *Journal of Economic Theory*, 156:246–268, 2015.

P. T. Harker and J.-S. Pang. Finite-dimensional variational inequality and nonlinear complementarity problems: a survey of theory, algorithms and applications. *Mathematical programming*, 1990.

S. Hart. Discrete colonel Blotto and general lotto games. *International Journal of Game Theory*, 2008.

E. Hazan, K. Singh, and C. Zhang. Efficient regret minimization in non-convex games. In *ICML*, 2017.

M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *NeurIPS*, 2017.

G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

J.-B. Hiriart-Urruty and C. Lemaréchal. Convex analysis and minimization algorithms. 1993.

K. Hornik, M. Stinchcombe, H. White, et al. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989.

Y.-P. Hsieh, C. Liu, and V. Cevher. Finding mixed nash equilibria of generative adversarial networks. In *ICML*, 2019.

Y.-P. Hsieh, P. Mertikopoulos, and V. Cevher. The limits of min-max optimization algorithms: convergence to spurious non-critical sets. *arXiv preprint arXiv:2006.09065*, 2020.

G. Huang, H. Berard, A. Touati, G. Gidel, P. Vincent, and S. Lacoste-Julien. Parametric adversarial divergences are good task losses for generative modeling. *arXiv preprint arXiv:1708.02511*, 2017.

A. Ibrahim, W. Azizian, G. Gidel, and I. Mitliagkas. Linear lower bounds and conditioning of differentiable games. In *ICML*, 2020.

A. Iusem, A. Jofré, R. I. Oliveira, and P. Thompson. Extragradient method with variance reduction for stochastic variational inequalities. *SIAM Journal on Optimization*, 2017.

C. Jin, R. Ge, P. Netrapalli, S. M. Kakade, and M. I. Jordan. How to escape saddle points efficiently. In *ICML*, 2017.

C. Jin, P. Netrapalli, and M. I. Jordan. What is local optimality in nonconvex-nonconcave minimax optimization? *arXiv preprint arXiv:1902.00618*, 2019.

A. Juditsky, A. Nemirovski, and C. Tauvel. Solving variational inequalities with stochastic mirror-prox algorithm. *Stochastic Systems*, 2011.

E. Kalai. Bounded rationality and strategic complexity in repeated games. In *Game theory and applications*, pages 131–157. Elsevier, 1990.

H. Karimi, J. Nutini, and M. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *ICLR*, 2018.

G. Kasparov. Chess, a drosophila of reasoning. *Sci*, 362(6419):1087–1087, 2018.

K. Kawaguchi. Deep learning without poor local minima. In *NeurIPS*, pages 586–594, 2016.

G. Kerg, K. Goyette, M. P. Touzel, G. Gidel, E. Vorontsov, Y. Bengio, and G. Lajoie. Non-normal recurrent neural network (nnrnn): learning long time dependencies while improving expressivity with transient dynamics. In *NeurIPS*, 2019.

D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

G. Korpelevich. The extragradient method for finding saddle points and other problems. *Matecon*, 1976.

S. Kraus and D. Lehmann. Diplomat, an agent in a multi agent environment: An overview. In *Seventh Annual International Phoenix Conference on Computers an Communications. 1988 Conference Proceedings*, pages 434–438. IEEE, 1988.

S. Kraus, D. Lehmann, and E. Ephrati. An automated diplomacy player. *Heuristic Programming in Artificial Intelligence: The 1st Computer Olympia*, pages 134–153, 1989.

A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.

T. Larsson and M. Patriksson. A class of gap functions for variational inequalities. *Math. Program.*, 1994.

N. Lay and A. Barbu. Supervised aggregation of classifiers using artificial prediction markets. In *ICML*, 2010.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Y. LeCun, C. Cortes, and C. Burges. Mnist handwritten digit database. 2010.

C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *CVPR*, 2017.

J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *COLT*, pages 1246–1257, 2016.

L. Lessard, B. Recht, and A. Packard. Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization*, 2016.

A. Letcher. On the impossibility of global convergence in multi-loss optimization. *arXiv preprint arXiv:2005.12649*, 2020.

H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

Y. Li, A. Schwing, K.-C. Wang, and R. Zemel. Dualing GANs. In *NeurIPS*, 2017.

T. Liang and J. Stokes. Interaction matters: A note on non-asymptotic local convergence of generative adversarial networks. In *AISTATS*, 2019.

M. W. Libbrecht and W. S. Noble. Machine learning applications in genetics and genomics. *Nature Reviews Genetics*, 16(6):321–332, 2015.

R. J. Lipton and N. E. Young. Simple strategies for large zero-sum games with applications to complexity theory. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 734–740, 1994.

A. Lomepal. Môme, 2019.

Z. Lu, H. Pu, F. Wang, Z. Hu, and L. Wang. The expressive power of neural networks: A view from the width. In *NeurIPS*, 2017.

A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

B. Martinet. Brève communication. régularisation d'inéquations variationnelles par approximations successives. *Revue française d'informatique et de recherche opérationnelle. Série rouge*, 4(R3):154–158, 1970.

E. Mazumdar, L. J. Ratliff, and S. S. Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, 2020.

E. V. Mazumdar, M. I. Jordan, and S. S. Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and economic behavior*, 10(1):6–38, 1995.

P. Mertikopoulos, B. Lecouat, H. Zenati, C.-S. Foo, V. Chandrasekhar, and G. Piliouras. Optimistic mirror descent in saddle-point problems: Going the extra (gradient) mile. In *ICLR*, 2019.

L. Mescheder, S. Nowozin, and A. Geiger. The numerics of GANs. In *NeurIPS*, 2017.

L. Mescheder, A. Geiger, and S. Nowozin. Which Training Methods for GANs do actually Converge? In *ICML*, 2018.

L. Metz, B. Poole, D. Pfau, and J. Sohl-Dickstein. Unrolled generative adversarial networks. In *ICLR*, 2017.

G. J. Minty. On the generalization of a direct method of the calculus of variations. *Bulletin of the American Mathematical Society*, 73(3):315–321, 1967.

M. Mirza and S. Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

I. Mitliagkas, C. Zhang, S. Hadjis, and C. Ré. Asynchrony begets momentum, with an application to deep learning. In *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2016.

T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 2015.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. MIT press, 2012.

A. Mokhtari, A. Ozdaglar, and S. Pattathil. A unified analysis of extra-gradient and optimistic gradient methods for saddle point problems: Proximal point approach. In *AISTATS*, 2020.

D. Monderer and L. S. Shapley. Potential games. *Games and economic behavior*, 1996.

R. D. Monteiro and B. F. Svaiter. Complexity of variants of tseng's modified fb splitting and korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems. *SIAM Journal on Optimization*, 21(4):1688–1720, 2011.

M. Müller. Computer go. *Artificial Intelligence*, 134(1-2):145–179, 2002.

K. G. Murty and S. N. Kabadi. Some np-complete problems in quadratic and nonlinear programming. Technical report, 1985.

V. Nagarajan and J. Z. Kolter. Gradient descent GAN optimization is locally stable. In *NeurIPS*, 2017.

J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286–295, 1951.

J. F. Nash et al. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 1950.

A. Nedić and A. Ozdaglar. Subgradient methods for saddle-point problems. *J Optim Theory Appl*, 2009.

A. Nemirovski. Prox-method with rate of convergence $O(1/t)$ for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM J. Optim.*, 2004.

A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on optimization*, 2009.

Y. Nesterov. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. In *Sov. Math. Dokl*, volume 27, 1983.

Y. Nesterov. Squared functional systems and optimization problems. In *High performance optimization*, pages 405–440. Springer, 2000.

Y. Nesterov. *Introductory lectures on convex optimization: A basic course*, volume 87. Springer Science & Business Media, 2004.

Y. Nesterov. Dual extrapolation and its applications to solving variational inequalities and related problems. *Math. Program.*, 2007.

J. v. Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100 (1):295–320, 1928.

J. v. Neumann. Communication on the Borel notes. *Econometrica: journal of the Econometric Society*, 1953.

J. v. Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

A. Neyman. Bounded complexity justifies cooperation in the finitely repeated prisoners' dilemma. *Economics letters*, 19(3):227–229, 1985.

H. Nikaidô, K. Isoda, et al. Note on non-cooperative convex games. *Pacific Journal of Mathematics*, 5(Suppl. 1):807–815, 1955.

N. Nisan et al. Introduction to mechanism design (for computer scientists). *Algorithmic game theory*, 9:209–242, 2007.

S. Nowozin, B. Cseke, and R. Tomioka. f-GAN: Training generative neural samplers using variational divergence minimization. In *NeurIPS*, 2016.

A. Odena, J. Buckman, C. Olsson, T. B. Brown, C. Olah, C. Raffel, and I. Goodfellow. Is generator conditioning causally related to gan performance? In *ICML*, 2018.

F. A. Oliehoek, R. Savani, J. Gallego, E. van der Pol, and R. Groß. Beyond local nash equilibria for adversarial networks. In *Benelux Conference on Artificial Intelligence*, pages 73–89. Springer, 2018.

Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr. An accelerated linearized alternating direction method of multipliers. *SIAM Journal on Imaging Sciences*, 8 (1):644–681, 2015.

B. Palaniappan and F. Bach. Stochastic variance reduction methods for saddle-point problems. In *NeurIPS*, 2016.

C. H. Papadimitriou. The complexity of finding nash equilibria. *Algorithmic game theory*, 2:30, 2007.

C. H. Papadimitriou and M. Yannakakis. On complexity as bounded rationality. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 726–733, 1994.

P. Paquette, Y. Lu, S. S. Bocco, M. Smith, O.-G. Satya, J. K. Kummerfeld, J. Pineau, S. Singh, and A. C. Courville. No-press diplomacy: Modeling multi-agent gameplay. In *NeurIPS*, pages 4474–4485, 2019.

W. Paul. Electromagnetic traps for charged and neutral particles. *Reviews of modern physics*, 1990.

B. A. Pearlmutter. Fast exact multiplication by the hessian. *Neural computation*, 6(1):147–160, 1994.

F. Pedregosa. Hyperparameter optimization with approximate gradient. In *ICML*, 2016.

F. Pedregosa and G. Gidel. Adaptive three operator splitting. In *ICML*, 2018.

D. Pfau and O. Vinyals. Connecting generative adversarial networks and actor-critic methods. *arXiv preprint arXiv:1610.01945*, 2016.

B. T. Polyak. Gradient methods for minimizing functionals. *Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki*, 3(4):643–653, 1963.

B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964.

L. D. Popov. A modification of the arrow-hurwicz method for search of saddle points. *Mathematical notes of the Academy of Sciences of the USSR*, 1980.

N. Qian. On the momentum term in gradient descent learning algorithms. *Neural networks*, 1999.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *ICLR*, 2016.

A. Rakhlin and K. Sridharan. Online learning with predictable sequences. In *COLT*, 2013.

L. J. Ratliff, S. A. Burden, and S. S. Sastry. On the characterization of local nash equilibria in continuous games. In *IEEE Transactions on Automatic Control*, 2016.

A. Razavi, A. van den Oord, and O. Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NeurIPS*, 2019.

S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *ICLR*, 2019.

H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

B. Roberson. The colonel Blotto game. *Economic Theory*, 2006.

R. T. Rockafellar. *Convex analysis*. Princeton university press, 1970.

R. T. Rockafellar. Monotone operators and the proximal point algorithm. *SIAM journal on control and optimization*, 14(5):877–898, 1976.

A. Rubinstein and C.-j. Dalgaard. *Modeling bounded rationality*. MIT press, 1998.

O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

L. Sagun, L. Bottou, and Y. LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. *arXiv*, 2016.

L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou. Empirical analysis of the hessian of over-parametrized neural networks. *arXiv*, 2017.

T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training GANs. In *NeurIPS*, 2016.

A. L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, 1959.

J. Schaeffer. The games computers (and people) play. In *Advances in computers*, volume 52, pages 189–266. Elsevier, 2000.

A. Shaban, C.-A. Cheng, N. Hatch, and B. Boots. Truncated back-propagation for bilevel optimization. In *AISTATS*, 2019.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

M. Sibony. Méthodes itératives pour les équations et inéquations aux dérivées partielles non linéaires de type monotone. *Calcolo*, 7(1-2):65–183, 1970.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587): 484–489, 2016.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, et al. Mastering the game of go without human knowledge. *nature*, 2017.

D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419): 1140–1144, 2018.

H. Simon and W. Chase. Skill in chess. In *Computer chess compendium*, pages 175–188. Springer, 1988.

H. A. Simon. *The sciences of the artificial*. MIT press, 1969.

M. Sion et al. On general minimax theorems. *Pacific Journal of mathematics*, 8 (1):171–176, 1958.

H. v. Stackelberg. *Marktform und gleichgewicht*. J. springer, 1934.

I. Sutskever. *Training recurrent neural networks*. University of Toronto Toronto, Canada, 2013.

I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *ICML*, 2013.

I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NeurIPS*, 2014.

G. Tesauro. Temporal difference learning and td-gammon. *Communications of the ACM*, 1995.

R. Thompson, T. Harmon, and M. Ball. The rotating-saddle trap: A mechanical analogy to rf-electric-quadrupole ion trapping? *Canadian journal of physics*, 2002.

P. Tseng. On linear convergence of iterative methods for the variational inequality problem. *Journal of Computational and Applied Mathematics*, 60(1-2):237–252, 1995.

H. J. Van Den Herik, J. W. Uiterwijk, and J. Van Rijswijck. Games solved: Now and in the future. *Artificial Intelligence*, 134(1-2):277–311, 2002.

V. Vapnik. *Estimation of dependences based on empirical data*. Springer Science & Business Media, 2006.

S. Vaswani, A. Mishkin, I. Laradji, M. Schmidt, G. Gidel, and S. Lacoste-Julien. Painless stochastic gradient: Interpolation, line-search, and convergence rates. In *NeurIPS*, 2019.

F. Verhulst. *Nonlinear differential equations and dynamical systems*. Springer Science & Business Media, 1989.

W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

C. Villani. The wasserstein distances. In *Optimal Transport*, pages 93–111. Springer, 2009.

O. Vinyals, T. Ewalds, S. Bartunov, P. Georgiev, A. S. Vezhnevets, M. Yeo, A. Makhzani, H. Küttler, J. Agapiou, J. Schrittwieser, et al. Starcraft II: A new challenge for reinforcement learning. *arXiv preprint arXiv:1708.04782*, 2017.

O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, R. Powell, T. Ewalds, P. Georgiev, et al. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, pages 1–5, 2019.

Y. Wu, J. Donahue, D. Balduzzi, K. Simonyan, and T. Lillicrap. LOGAN: Latent optimisation for generative adversarial networks. *arXiv preprint arXiv:1912.00953*, 2019.

H. Xiao, K. Rasul, and R. Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

A. Yadav, S. Shah, Z. Xu, D. Jacobs, and T. Goldstein. Stabilizing adversarial nets with prediction methods. In *ICLR*, 2018.

Y. Yazıcı, C.-S. Foo, S. Winkler, K.-H. Yap, G. Piliouras, and V. Chandrasekhar. The unusual effectiveness of averaging in gan training. In *ICLR*, 2019.

F. Yousefian, A. Nedić, and U. V. Shanbhag. Optimal robust smoothing extragradient algorithms for stochastic variational inequality problems. In *CDC*. IEEE, 2014.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

F. Zhang. *The Schur complement and its applications*. Springer Science & Business Media, 2006.

G. Zhang, P. Poupart, and Y. Yu. Optimality and stability in non-convex smooth games. *arXiv preprint arXiv:2002.11875*, 2020.

J. Zhang and I. Mitliagkas. Yellowfin and the art of momentum tuning. 2019.

J. Zhang, M. Hong, and S. Zhang. On lower iteration complexity bounds for the saddle point problems. *arXiv preprint arXiv:1912.07481*, 2019.

J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pages 2223–2232, 2017.

M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. *UCLA CAM Report*, 34, 2008.

A Minimax Theorem for Nonconcave-Nonconvex Games Played with Neural Networks

1 Relevance of the Minimax theorem in the Context of Machine Learning

A notorious ML application which has a minimax formulation is *adversarial training* where a classifier is trained to be robust against adversarial attack. From a game-theoretic perspective, the adversarial attack is picked after the classifier f is set and thus it corresponds to a best response. From a learning perspective, the goal is to learn to be robust to adversarial attacks *specifically designed* against the current classifier. Such an equilibrium is called a Stackelberg Equilibrium [Conitzer and Sandholm, 2006].

In games with imperfect information such as Colonel Blotto, Poker, or StarCraft II the players must commit to a strategy without the knowledge of the strategy picked by their opponent. In that case, the agents cannot design attacks specific to their opponent, because such attacks may be exploitable strategies. It is thus strictly equivalent to consider that the players simultaneously pick their respective strategies and then reveal them. Thus, a meaningful notion of playing the game must have a value and an equilibrium.

In machine learning applications, each player is trained using local information (though gradient or RL based methods). Because the behavior of the players changes slowly, they cannot have access to the best response against their opponent. In order to illustrate that point, let us consider the example of Generative Adversarial Networks. The two agents (the generator and the discriminator) are usually sequentially updated using a gradient method with similar step-sizes. During training, one cannot expect an agent to find a best response in a single (or few) gradient steps. To sum-up, since local updates are performed one must expect to reach a point (if it exists) that is locally stable. In this work, we show that there actually exists a *global* approximate equilibrium for a large class of parametrized games.

2 Interpretation of Equilibria in Latent Games

In latent games, players embed in mapping spaces in order to solve the game. When we consider a standard normal form game φ that we try to solve using

mappings to approximate mixtures of strategies, we are actually playing a limited capacity version of the game that heavily depends on the expressivity of the mappings in the classes $\mathcal{F}$ and $\mathcal{G}$.

Such a limitation may be interpreted as limitations on the skills of the players. It intuitively makes sense that such limitations would change the optimal way to play the game: the optimal way to play StarCraft II is different for players that can perform 10 versus 100 actions per second. Thus, if the goal is to train agents to compete with humans, one needs to set a class $\mathcal{G}$ that (roughly) corresponds to human skills. Setting "fair" constraints on the RL agents trained to play the game of StarCraft II has been an important issue Vinyals et al. [2019] and can be understood as setting the right class $\mathcal{G}$ in a latent game.

Similarly a player would not play poker the same way if they had no memory of their opponents' behavior in previous games.

Similarly, in the context of Generative Adversarial Networks, it has been argued that setting a restricted function class for the discriminator provides a more meaningful loss and describes an achievable learning task for the generator Arora et al. [2017], Huang et al. [2017]. The final task is to generate pictures that are realistic according to the human metric. Such task is way looser – and thus easier to achieve – than for instance minimizing the KL divergence or the Wasserstein distance between the real data distribution and the generated distribution.

To sum-up, the equilibrium of a latent game provides a notion of limited-capacity-equilibrium that can define a target that correspond to agents with expressive and realistic behavior. In many tasks, our goal is to train agents that outperform human using human realistic limitations: it is important to constrain the agent in order to prevent it to play 10^5 actions per minute but it is also important to constrain its opponent because we would like opponent to try to exploit the main agent in a semantically meaningful way and not by designing very specific 'adversarial example' strategies –e.g., very precise positions of units that breaks the vision system of the main agent – that a human player could not perform.

This idea of modeling the limitations of realistic players play suboptimally is related to the notion of games with bounded rationality [Simon, 1969, Rubinstein and Dalgaard, 1998, Papadimitriou and Yannakakis, 1994, Kalai, 1990] or bounded computation [Halpern and Pass, 2015]. However, bounded rationality models players that do not optimize their reward function [Rubinstein and Dalgaard, 1998], the corresponding literature aims to model a process a choice for players not always maximizing their reward. Bounded computation refers to studies of games where players pay for the (time) complexity of the strategy they use. The notion of limited-capacity in latent games is a limitation on the representative power of the function (or distribution) spaces. The literature has not thoroughly considered *limitations on representational power* – a gap that is critical to address, given that neural nets are now a major workhorse in AI and ML.

3 Proof of results from Section 5

3.1 Proof of Proposition 1

Before proving this proposition let us state Sion's minimax theorem.

Theorem 1 (Minimax theorem [Sion et al., 1958])**.** *If U and V are convex and compact sets and if the sublevel sets of $\varphi(\cdot, v)$ and $-\varphi(u, \cdot)$ are convex then,*

$$\max_{u\in U}\min_{v\in V}\varphi(u,v) = \min_{u\in U}\max_{v\in V}\varphi(u,v) \tag{3.1}$$

Let us now state our proposition.

Proposition 1. *Let φ be a game that follows Assumption 1. If $\mathcal{G}_\Theta$ and $\mathcal{F}_\Omega$ are compact, then there exist a* value *for the game such that,*

$$\mathrm{V}(\Omega,\Theta) := \sup_{f\in\mathrm{hull}(\mathcal{F}_\Omega)}\inf_{p\in\mathrm{hull}(\mathcal{G}_\Theta)}\tilde{\varphi}(f,p) = \inf_{p\in\mathrm{hull}(\mathcal{G}_\Theta)}\sup_{f\in\mathrm{hull}(\mathcal{F}_\Omega)}\tilde{\varphi}(f,p)\,, \tag{5.5}$$

where $\mathrm{hull}(\mathcal{G}_\Theta)$ *and* $\mathrm{hull}(\mathcal{F}_\Omega)$ *are either defined in (5.3) or in (5.4), depending on the type player.*

Proof. For simplicity and conciseness we note, $\mathcal{F} = \mathcal{F}_\Omega$ and $\mathcal{G} = \mathcal{G}_\Theta$. The sets $\mathrm{hull}(\mathcal{F})$ and $\mathrm{hull}(\mathcal{G})$ are convex by construction. However, they are not compact in general. However, since $\mathcal{G}$ is assumed to be a compact set we then have that under mild assumptions (namely, that $\mathcal{F}$ and $\mathcal{G}$ belong to a completely metrizable locally convex space) that the closure of $\mathrm{hull}(\mathcal{G})$ is compact [Aliprantis and Border, 2006, Theorem 5.20]. Thus, we can apply Sion's theorem to get,

$$\min_{p\in\mathrm{closure}(\mathrm{hull}(\mathcal{G}))}\max_{f\in\mathrm{closure}(\mathrm{hull}(\mathcal{F}))}\tilde{\varphi}(f,p) = \max_{f\in\mathrm{closure}(\mathrm{hull}(\mathcal{F}))}\min_{p\in\mathrm{closure}(\mathrm{hull}(\mathcal{G}))}\tilde{\varphi}(f,p) \tag{3.2}$$

□

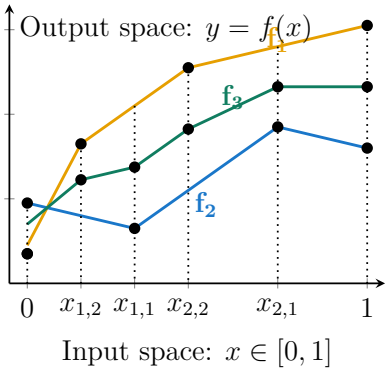
Moreover there exists $(w_i)_{i\geq 0}$, $(\theta_i)_{i\geq 0}$, $\lambda_i \geq 0\,, \sum_{i\geq 0}\lambda_i = 1$ and $\rho_i \geq 0\,, \sum_{i\geq 0}\rho_i = 1$ such that,

$$V(\Omega,\Theta) = \tilde{\varphi}\Big(\sum_{i\geq 0}\lambda_i f_{w_i}, \sum_{i\geq 0}\rho_i p_{\theta_i}\Big)\,. \tag{3.3}$$

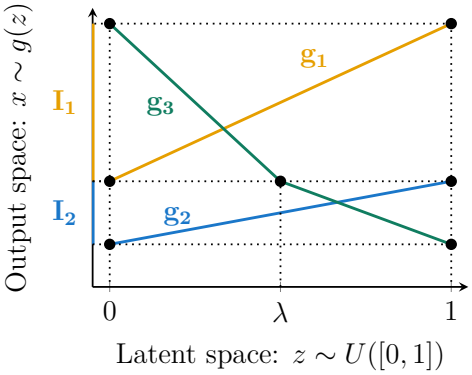
This comes from the fact that any element in closure(hull($\mathcal{F}$)) can be written as $\sum_{i\geq 0}\lambda_i f_{w_i}$:

Lemma 1. *Let U be a compact set that belongs to a completely metrizable locally convex space. Then the closure of the convex hull of U is compact and we have that* $\mathrm{closure}(\mathrm{hull}(U)) = \{\sum_{i\geq 0}\lambda_i u_i\,,\ \lambda_i \geq 0\,,\ \sum_{i\geq 0}\lambda_i = 1\,,\ u_i \in U\}$.

Proof. Let us consider a sequence $(u_n) \in conv(U)^{\mathbb{N}}$, we have $u_n = \sum_{i=0}^{K_n}\lambda_{i,n}u_{i,n}$ where $u_{i,n} \in U\,,\ \forall i, n \in \mathbb{N}$. Since $\lambda_{i,n} \in [0,1]$ and $u_{i,n} \in U$ that are compact sets these sequences have a convergent subsequence. By Cantor diagonalization process, (x_n) has a convergent subsequence. □



(a) ReLU networks of width p can represent any p-piecewise linear function. The point-wise average f_3 of two ReLU neural networks f_1 and f_2 is a piecewise linear function that can be represented by a wider ReLU neural network.



(b) Latent mixture of g_1 and g_2: For $k \in \{1,2,3\}$ the transformation g_k maps the uniform distribution on $[0,1]$ into a distribution on the output space: $x \sim p_{g_i}$ iff $x = g_i(z)$, $z \sim U([0,1])$. In that case, p_{g_1} and p_{g_2} are respectively the uniform distribution over I_1 and I_2. The function g_3 represents a distribution that puts half of its mass uniformly on I_1 and the other half on I_2.

Figure A.1: Difference between pointwise averaging of function and latent mixture of mapping.

3.2 Proof of Theorem 2

We will prove a result a bit more general that the result stated in the main paper,

Theorem 2. *Let φ a game that satisfies the assumptions of Proposition 1. If $\tilde{\varphi}$ is bilinear and φ is L-Lipschitz then,*

$$K_\epsilon^\Omega \leq \frac{4D_w}{\epsilon^2} \ln(\mathcal{N}(\Theta, \tfrac{\epsilon}{2L})) \quad \textit{and} \quad K_\epsilon^\Theta \leq \frac{4D_\theta}{\epsilon^2} \ln(\mathcal{N}(\Omega, \tfrac{\epsilon}{2L})) \tag{3.4}$$

where $\mathcal{N}(\mathcal{H}, \epsilon')$ is the number of ϵ'-balls necessary to cover the set A and the quantities D_w and D_θ are defined as $D_w := \max_{w,w',\theta} \varphi(w,\theta) - \varphi(w',\theta)$ and $D_\theta := \max_{w,\theta,\theta'} \varphi(w,\theta) - \varphi(w,\theta')$.

In the literature,the quantity $\mathcal{N}(\mathcal{H}, \epsilon')$ is called covering number of the set $\mathcal{H}$. By definition of compactness, it is finite when $\mathcal{H}$ is compact. It is a complexity measure of the set $\mathcal{H}$ that has been extensively studied in the context of generalization bounds [Mohri et al., 2012, Shalev-Shwartz and Ben-David, 2014].

Proof. This proof is largely inspired from the proof of [Lipton and Young, 1994, Theorem 2] and [Arora et al., 2017, Theorem B.3]. The difference with [Arora et al., 2017, Theorem B.3] is that we make appear a notion of condition number and we provide this proof in a context more general than [Arora et al., 2017, Theorem B.3] who was focusing on GANs.

One way to insure that D_w amd D_θ have bounded value is by assuming that Θ and Ω have a finite diameter, we then have that the values of D_w and D_θ are respectively bounded by $L\,\mathrm{diam}(\Theta)$ and $L\,\mathrm{diam}(\Omega)$. Note that is practice one also may have that the payoff is bounded between -1 (losing) and 1 (winning).

By Proposition 1 we have that there exists f^* and p^* such

$$V(\Omega,\Theta)=\tilde{\varphi}\Big(f^*,p^*\Big)\,, \tag{3.5}$$

where,

$$f^*:=\sum_{k=1}^{\infty}\lambda_k f_{w_k}\text{ and }p^*:=\sum_{k=1}^{\infty}\rho_k p_{\theta_k} \tag{3.6}$$

where $w_k,\theta_k\in\Omega\times\Theta\,,\ \rho_k,\lambda_k>0\,,\ \sum_{k=1}^{\infty}\lambda_k=\sum_{k=1}^{\infty}\rho_k=1$.

Now let us consider the mixture

$$f_n^*:=\frac{1}{n}\sum_{k=1}^{n}f_{w_k} \tag{3.7}$$

where $w_k, 1\leq k\leq n$ are defined in (3.6) and are drawn independently from the multinomial of weights $(\lambda_k)_{k=1\ldots\infty}$.

Let assume that Θ has a finite covering number $\mathcal{N}(\Theta,\epsilon/(2L))$ (in the following we will show that if Ω is compact then, its covering number is finite and we will give an explicit bound on it when $\Omega\subset\mathbb{R}^p$). Let us recall that covering number of Θ is the smallest number of ϵ balls needed to cover Θ. Let us consider $\theta_i\,,\ 1\leq i\leq\mathcal{N}(\Theta,\frac{\epsilon}{2L})$ the center of these balls where L is the Lipschitz constant of φ. Using Hoeffding's inequality, for any $\theta_i\,,\ 1\leq i\leq\mathcal{N}(\Theta,\frac{\epsilon}{2L})$ we have that,

$$\mathbb{P}(\tilde{\varphi}(f_n^*,p_{\theta_i})-\tilde{\varphi}(f^*,p_{\theta_i})<\epsilon/2)\leq e^{\frac{-n\epsilon^2}{2D_w^2}} \tag{3.8}$$

where D_w is a bound on the variations of φ defined as

$$D_w:=\max_{w,w',\theta}\varphi(w,\theta)-\varphi(w',\theta)\,. \tag{3.9}$$

Note that because we assumed that $\tilde{\varphi}$ is bilinear, the bound on the variations of φ is also valid for the variations of $\tilde{\varphi}$. More precisely, we have

$$\tilde{\varphi}(\sum_i\lambda_i f_{w_i},p_\theta)-\tilde{\varphi}(\sum_i\lambda_i' f_{w_i'},p_\theta)=\sum_i\lambda_i(\varphi(w_i,\theta)-\varphi(w_i',\theta)) \tag{3.10}$$

$$\leq\sum_i\lambda_i D_w=D_w. \tag{3.11}$$

Thus, using standard union bounds,

$$\mathbb{P}\Big(\tilde{\varphi}(f_n^*,p_{\theta_i})-\tilde{\varphi}(f^*,p_{\theta_i})<\epsilon/2\,,\ \forall 1\leq i\leq\mathcal{N}(\mathcal{G},\tfrac{\epsilon}{2\tilde{L}})\Big)$$

$$\leq\mathcal{N}(\mathcal{G},\tfrac{\epsilon}{2\tilde{L}})e^{\frac{-n\epsilon^2}{2D_w^2}} \tag{3.12}$$

Let us now consider

$$\hat{p}_n \in \arg \min_{p \in \text{closure}(\text{hull}(\mathcal{G}))} \tilde{\varphi}(f_n^*, p) = \arg \min_{\theta \in \Theta} \tilde{\varphi}(f_n^*, p_\theta) \tag{3.13}$$

Note that this minimum is achieved with $q \in \mathcal{G}$ because we assumed that the function $\tilde{\varphi}$ is bilinear (and thus a minimum with respect to a convex hull is always achieved at an atom).[1] Thus there exists $\hat{\theta}_n \in \Theta$ such that $\hat{p}_n = p_{\hat{\theta}_n}$.

Since φ is L-Lipschitz and since we have that (θ_i) is an $\frac{\epsilon}{2L}$-covering there exists a θ_i that is $\frac{\epsilon}{2L}$-close to $\hat{\theta}_n$ and thus,

$$|\tilde{\varphi}(f_n^*, p_{\theta_i}) - \min_{p \in \text{closure}(\text{hull}(\mathcal{G}))} \tilde{\varphi}(f_n^*, p)| = \left| \sum_{k=1}^{n} \tfrac{1}{n} (\varphi(w_k, \theta_i) - \varphi(w_k, \hat{\theta}_n)) \right| \leq \epsilon/2 \,. \tag{3.14}$$

When we have that $\tilde{\varphi}(f^*, p_{\theta_i}) - \tilde{\varphi}(f_n^*, p_{\theta_i}) < \epsilon/2$ (which is true with high probability) we have,

$$\begin{aligned} \min_{p \in \text{closure}(\text{hull}(\mathcal{G}))} \varphi(f_n^*, p) &\geq \tilde{\varphi}(f_n^*, p_{\theta_i}) - \epsilon/2 && (3.15) \\ &> \tilde{\varphi}(f^*, p_{\theta_i}) - \epsilon && (3.16) \\ &= V(\Omega, \Theta) - \epsilon && (3.17) \end{aligned}$$

Thus for $n > \frac{4D_w^2}{\epsilon^2} \ln(\mathcal{N}(\Theta, \frac{\epsilon}{2L}))$ we have,

$$\mathbb{P}\Big(\min_{p \in \text{closure}(\text{hull}(\mathcal{G}))} \varphi(f_n^*, p) > V(\Omega, \Theta) - \epsilon \Big) < 1 \tag{3.18}$$

Since this probability is strictly smaller than one, for any $\epsilon' > 0$, among all the possible sampled f_n^* there exist at least one such that

$$\min_{p \in \text{closure}(\text{hull}(\mathcal{G}))} \varphi(f_n^*, p) > \mathrm{V}_L - \epsilon \,. \tag{3.19}$$

Thus,

$$K_\epsilon^\Omega \leq \frac{4D_w^2}{\epsilon^2} \ln(\Theta, \tfrac{\epsilon}{2L}) \,. \tag{3.20}$$

A similarly we can prove a bound on K_ϵ^Θ.

□

Then, we will use a simple bound for the covering number $\Theta \subset \mathbb{R}^d$ that can be found in Shalev-Shwartz and Ben-David [2014],

$$\log \mathcal{N}(\Theta, \tfrac{\epsilon}{2L}) \leq d \log(\tfrac{4LR\sqrt{d}}{\epsilon}) \,. \tag{3.21}$$

that leads to

$$K_\epsilon^\Omega \leq \frac{4D_w^2 d}{\epsilon^2} \log(\frac{4LR\sqrt{d}}{\epsilon}) \tag{3.22}$$

[1] Note that we could get rid of the bilinear assumption by replacing the covering number of Θ by the covering number of hull($\mathcal{G}$). However the asymptotic behavior of the latter (when $\epsilon \to 0$) may be challenging to bound. We thus decided to focus on bilinear examples since the covering number for finite dimensional compact sets is a well studied quantity.

3.3 Proof of Proposition 3

Proposition 3. *For all $w_k \in [-R,R]^p$, $k = 1 \dots K$, there exists $w \in [-R,R]^{Kp}$ such that $\frac{1}{n}\sum_{k=1}^K f_{w_k} = f_w$.*

Proof. We will prove this result for an abitrary convex combination. Let us recall that a two-layers ReLU network of width W can be written as

$$g(x) = \sum_{i=1}^{W} a_i \,\mathrm{ReLU}(c_i^\top x + d_i) + b_i \tag{3.23}$$

where $a_i, b_i \in \mathbb{R}^{d_{out}}$, $c_i \in \mathbb{R}^d$ and $d_i \in \mathbb{R}$. Then, let us consider K such functions with p parameters, then any convex combination of these K functions can be written as,

$$f(x) = \sum_{k=1}^{K}\sum_{i=1}^{W_k} \lambda_k (a_{i,k} \,\mathrm{ReLU}(c_{i,k}^\top x + d_{i,k}) + b_{i,k}) \tag{3.24}$$

where $\lambda_k \geq 0$, $1 \leq k \leq K$ and $\sum_{k=1}^K \lambda_k = 1$.

Setting $\tilde{a}_{i,k} := \lambda_k a_{i,k}$ and $\tilde{b}_{i,k} := \lambda_k b_{i,k}$, we have that

$$f(x) = \sum_{(i,k)} \tilde{a}_{i,k} \,\mathrm{ReLU}(c_{i,k}^\top x + d_{i,k}) + \tilde{b}_{i,k} \tag{3.25}$$

which is a ReLU network with $K \cdot p$ parameters. □

3.4 Proof of Proposition 2

Proposition 2. *If $d_{in} = 1$ and if for all $\theta \in \Theta$, $z \in [0,1]$, $g_\theta(z) \in [0,1]^d$ and g_θ is constant outside of $[0,1]$, then for any θ_k, , $\|\theta_k\| \leq R$, $k = 1 \dots K$, there exists $\theta \in [-KR, KR]^{K(p+6)}$ such that $d_{TV}(\frac{1}{n}\sum_{k=1}^K p_{\theta_k}, p_\theta) \leq 1/R$ where d_{TV} is the total variation distance.*

Proof. We will prove the first part of this theorem for an arbitrary number K of mappings. Let g be a two-layers ReLU network of width p, the probability distribution π_g induced by g verifies,

$$\pi_g(S) = \ell(g^{-1}(S)), \quad \forall S \text{ measurable in } [0,1]^{d_{out}}. \tag{3.26}$$

where ℓ is the Lesbegues measure on $[0,1]^d$. The convex combination π of $\pi_{g_1}, \dots, \pi_{g_K}$ verifies,

$$\begin{aligned}\pi(S) &:= \Big(\sum_{k=1}^{K} \lambda_k \pi_{g_k}\Big)(S) \\ &= \sum_{1 \leq k \leq K} \lambda_k \ell(g_k^{-1}(S))\end{aligned}$$

Using the properties of the Lesbegues measure we have that $\forall \lambda > 0, b \in \mathbb{R}$

$$\lambda \ell(U) = \ell(\lambda U + b)\,, \tag{3.27}$$

where λU is the dilation of the set U and $U + b$ its translation by b. thus, we have that for any $b_k \in \mathbb{R}\,,\ k = 1 \dots K$,

$$\pi(S) = \sum_{1 \leq k \leq K} \ell(\lambda_k g_k^{-1}(S) + b_k)$$

Now notice that,

$$\lambda_k g_k^{-1}(S) + b_k = \{\lambda_k x + b_k \,:\, x \in [0,1]\,,\ g_k(x) \in S\} \tag{3.28}$$

$$= \{x \,:\, x \in [b_k, \lambda_k + b_k]\,,\ g_k(x/\lambda_k + b_k) \in S\} \tag{3.29}$$

Then, setting $b_k := \sum_{i=0}^{k-1} \lambda_k \in [0,1]$, we get by construction that $b_{k+1} = b_k + \lambda_k$ and thus that the sets $S_k := [b_k, \lambda_k + b_k]$ are a partition of $[0,1]$. Finally, if we note $\tilde{g}$ the function,

$$\tilde{g}(x) = g_k(x/\lambda_k + b_k) \text{ if } x \in [b_k, b_{k+1}] \tag{3.30}$$

We have by construction (and by the fact that S_k are disjoints)

$$\pi_{\tilde{g}}(S) = \sum_{1 \leq k \leq K} \ell(\lambda_k g_k^{-1}(S) + b_k) = \Big(\sum_{k=1}^{K} \lambda_k \pi_{g_k} \Big)(S) \tag{3.31}$$

However, the proof is not over because $\tilde{g}$ is not continuous and thus is not a ReLU network. We will now construct a ReLU network that approximate the distribution induced by $\tilde{g}$. Let us recall that we assumed that $g_k(x) = g_k(0)\,,\ \forall x < 0$ and $g_k(x) = g_k(1)\,,\ \forall x > 1$. Let us introduce the approximated "step" function h_k that is a ReLU net with 3 parameters.

$$h_\delta(x) := \frac{1}{\delta} \lfloor x \rfloor_+ - \frac{1}{\delta} \lfloor x - \delta \rfloor_+ = \begin{cases} 0 \text{ if } x < 0 \\ 1 \text{ if } x > \delta \\ x/\delta \text{ otherwise.} \end{cases} \tag{3.32}$$

Thus we can introduce the ReLU net $\tilde{g}_k$ defined as

$$\tilde{g}_k(x) := g_k(x/\lambda_k + b_k) - g_k(0) h_\delta(-x + b_k) - g_k(1) h_\delta(x + b_{k+1}) \tag{3.33}$$

$$= \begin{cases} 0 \text{ if } x < b_k \text{ or } x > b_{k+1} \\ g_k(x/\lambda_k + b_k) \text{ if } b_k + \delta < x < b_{k+1} - \delta \end{cases} \tag{3.34}$$

Finally we have that the sum of $\tilde{g}_k$ for $k = 1, \dots K$ is a ReLU neural network with $K(p+6)$ parameters such that

$$TV(\pi, \pi_{\sum_k \tilde{g}_k}) = \sup_S |\pi(S) - \pi_{\sum_k \tilde{g}_k}(S)|$$

$$\leq K\delta$$

Moreover, since g_k has p parameters in $[-R, R]$ we have that $\tilde{g}_k$ has $p+6$ parameters that are in $[-R/\lambda_k, R/\lambda_k]$. Since we assumed that the parameters of the ReLU network should be bounded by KR we have that we cannot pick the parameters $g_k(0)/\delta$ and$g_k(1)/\delta$ larger than KR.

Thus by setting $\lambda_k = 1/K$, there exists a ReLU network with $K(p+6)$ parameters in $[-KR, KR]$ such that,

$$\begin{aligned} TV(\pi, \pi_{\sum_k \tilde{g}_k}) &= \sup_S |\pi(S) - \pi_{\sum_k \tilde{g}_k}(S)| \\ &\leq \frac{1}{R} \end{aligned}$$

□

3.5 Proof of Theorem 1

Theorem 1. *Let φ be a L-Lipschitz nonconcave-nonconvex game with values bounded by D that follows Assumption 1 for which the payoff $\tilde{\varphi}$ is bilinear and $\tilde{L}$ Lipschiz. The players are assumed to be parametrized by ReLU networks $g : \mathbb{R}^d \to \mathbb{R}^{d_{out}}$ with parameters smaller than R, and satisfies one of the three following cases:*

- *both players are functions and at least one of the function set is bounded (for instance by clipping the parameters). For any $\epsilon > 0$ there exists $(w_\epsilon^*, \theta_\epsilon^*) \in [-R, R]^{2p}$ s.t.,*

$$\min_{\substack{\theta \in \mathbb{R}^{p_\epsilon} \\ \|\theta\| \leq R}} \varphi(w_\epsilon^*, \theta) + \epsilon \geq \max_{\substack{w \in \mathbb{R}^{p_\epsilon} \\ \|w\| \leq R}} \varphi(w, \theta_\epsilon^*)\,. \tag{3.35}$$

  where $p_\epsilon \geq \frac{\epsilon}{2D}\sqrt{\frac{p}{\log(4L\sqrt{p}/\epsilon)}}$.

- *The first player is a distribution with $d_{in} = 1$ and the second player is a function. This is for instance the setting of WGAN (Example 2).*

$$\min_{\substack{\theta \in \mathbb{R}^{p_\epsilon} \\ \|\theta\| \leq R}} \varphi(w_\epsilon^*, \theta) + \epsilon + \frac{\tilde{L}}{R} \geq \max_{\substack{w \in \mathbb{R}^{p_\epsilon} \\ \|w\| \leq R_\epsilon}} \varphi(w, \theta_\epsilon^*)\,. \tag{3.36}$$

  where $p_\epsilon \geq \frac{\epsilon}{2D}\sqrt{\frac{p}{\log(4L\sqrt{p}/\epsilon)}} - 6$ and $R_\epsilon \geq R^{\frac{p_\epsilon}{p}}$

- *both players are distributions with $d_{in} = 1$. this is for instance the setting of the Blotto game (Examples 3).*

$$\min_{\substack{\theta \in \mathbb{R}^{p_\epsilon} \\ \|\theta\| \leq R}} \varphi(w_\epsilon^*, \theta) + \epsilon + \frac{2\tilde{L}}{R} \geq \max_{\substack{w \in \mathbb{R}^{p_\epsilon} \\ \|w\| \leq R_\epsilon}} \varphi(w, \theta_\epsilon^*)\,. \tag{3.37}$$

  where $p_\epsilon \geq \frac{\epsilon}{2D}\sqrt{\frac{p}{\log(4L\sqrt{p}/\epsilon)}} - 6$ and $R_\epsilon \geq R^{\frac{p_\epsilon}{p}}$

Proof. Let $\epsilon > 0$ and let us consider ReLU networks with p_ϵ parameters in $[-R_\epsilon, R_\epsilon]$ (we will set those quantities later). For simplicity here $\Omega = \Theta = [-R_\epsilon, R_\epsilon]^p$. Theorem 2 says that an ϵ-equilibrium can be achieved with a uniform convex combination of K_ϵ networks.

Let us consider the case where the first player is a function and the second player is a distribution.

For the first player, one can apply Proposition 3 to say that such a convex combination of K_ϵ functions can be expressed with a larger network that has $K_\epsilon \cdot p_\epsilon$ parameters in $[-R_\epsilon, R_\epsilon]$.

For the second player, once can apply Proposition 2 to get that that such a uniform convex combination of K_ϵ functions can be expressed up to precision $1/K_\epsilon R_\epsilon$ with a larger network that has $K_\epsilon \cdot p_\epsilon$ parameters in $[-K_\epsilon R_\epsilon, K_\epsilon R_\epsilon]$.

Thus we get that a sufficient condition for ϵ-approximate equilibrium of the game φ to be achieved by a ReLU network with p parameters in $[-R, R]$ is that,

$$p \geq (p_\epsilon + 6)K_\epsilon \quad \text{and} \quad R \geq K_\epsilon R_\epsilon \tag{3.38}$$

Let us set,

$$p_\epsilon := \left\lfloor \frac{\epsilon}{2D}\sqrt{\frac{p}{\ln(4LR\sqrt{p}/\epsilon)}} - 6 \right\rfloor \quad \text{and} \quad R_\epsilon := R\frac{p_\epsilon}{p} \tag{3.39}$$

Using the fact that in Theorem 2, $K_\epsilon \leq \frac{4D^2}{\epsilon^2} p_\epsilon \log\left(\frac{4LR_\epsilon\sqrt{p_\epsilon}}{\epsilon}\right)$ we have that

$$(p_\epsilon + 6)K_\epsilon \leq \frac{\epsilon^2}{4D^2}\frac{p}{\ln(4LR\sqrt{p}/\epsilon)}\frac{4D^2}{\epsilon^2}\log\left(\frac{4LR\sqrt{p_\epsilon}}{\epsilon}\right) \leq p \tag{3.40}$$

where we used the fact that $p_\epsilon \leq p$ and $R_\epsilon \leq R$. Moreover, since $p \geq (p_\epsilon + 6)K_\epsilon$ we have that,

$$K_\epsilon R_\epsilon \leq \frac{p}{p_\epsilon + 6} R_\epsilon \leq R\,. \tag{3.41}$$

Finally, since in Proposition 2 we approximate such uniform convex combination up to a TV distance $1/R$ and since we assumed that φ was $\tilde{L}$-Lipschitz (with respect to the TV distance) we have the additional $\frac{\tilde{L}}{R}$ term. □

B A Variational Inequality Perspective on Generative Adversarial Networks

1 Definitions

In this section, we recall usual definitions and lemmas from convex analysis. We start with the definitions and lemmas regarding the projection mapping.

1.1 Projection mapping

Definition 3. *The projection P_Ω onto Ω is defined as,*

$$P_\Omega(\boldsymbol{\omega}') \in \arg\min_{\boldsymbol{\omega}' \in \Omega} \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 \,. \tag{1.1}$$

When Ω is a convex set, this projection is unique. This is a consequence of the following lemma that we will use in the following sections: the *non-expansiveness* of the projection onto a convex set.

Lemma 2. *Let Ω a convex set, the projection mapping $P_\Omega : \mathbb{R}^d \to \Omega$ is nonexpansive, i.e.,*

$$\|P_\Omega(\boldsymbol{\omega}) - P_\Omega(\boldsymbol{\omega}')\|_2 \leq \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2 \,, \quad \forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega \,. \tag{1.2}$$

This is standard convex analysis result which can be found for instance in [Boyd and Vandenberghe, 2004]. The following lemma is also standard in convex analysis and its proof uses similar arguments as the proof of Lemma 2.

Lemma 3. *Let $\boldsymbol{\omega} \in \Omega$ and $\boldsymbol{\omega}^+ := P_\Omega(\boldsymbol{\omega} + \boldsymbol{u})$, then for all $\boldsymbol{\omega}' \in \Omega$ we have,*

$$\|\boldsymbol{\omega}^+ - \boldsymbol{\omega}'\|_2^2 \leq \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 + 2\boldsymbol{u}^\top(\boldsymbol{\omega}^+ - \boldsymbol{\omega}') - \|\boldsymbol{\omega}^+ - \boldsymbol{\omega}\|_2^2 \,. \tag{1.3}$$

Proof of Lemma 3. We start by simply developing,

$$\begin{aligned}
\|\boldsymbol{\omega}^+ - \boldsymbol{\omega}'\|_2^2 &= \|(\boldsymbol{\omega}^+ - \boldsymbol{\omega}) + (\boldsymbol{\omega} - \boldsymbol{\omega}')\|_2^2 \\
&= \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 + 2(\boldsymbol{\omega}^+ - \boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') + \|\boldsymbol{\omega}^+ - \boldsymbol{\omega}\|_2^2 \\
&= \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 + 2(\boldsymbol{\omega}^+ - \boldsymbol{\omega})^\top(\boldsymbol{\omega}^+ - \boldsymbol{\omega}') - \|\boldsymbol{\omega}^+ - \boldsymbol{\omega}\|_2^2 \,.
\end{aligned}$$

Then since $\boldsymbol{\omega}^+$ is the projection onto the convex set Ω of $\boldsymbol{\omega} + \boldsymbol{u}$, we have that $(\boldsymbol{\omega}^+ - (\boldsymbol{\omega} + \boldsymbol{u}))^\top(\boldsymbol{\omega}^+ - \boldsymbol{\omega}') \leq 0 \,,\ \forall \boldsymbol{\omega}' \in \Omega$, leading to the result of the Lemma. □

1.2 Smoothness and Monotonicity of the operator

Another important property used is the Lipschitzness of an operator.

Definition 4. *A mapping $F : \mathbb{R}^p \to \mathbb{R}^d$ is said to be L-Lipschitz if,*

$$\|F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}')\|_2 \leq L\|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2\,, \quad \forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega\,. \tag{1.4}$$

In this paper, we also use the notion of *strong monotonicity*, which is a generalization for operators of the notion of strong convexity. Let us first recall the definition of the latter,

Definition 5. *A differentiable function $f : \Omega \to \mathbb{R}$ is said to be* μ-strongly convex *if*

$$f(\boldsymbol{\omega}) \geq f(\boldsymbol{\omega}') + \nabla f(\boldsymbol{\omega}')^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') + \frac{\mu}{2}\|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 \quad \forall \boldsymbol{\omega}, \boldsymbol{\omega}' \in \Omega\,. \tag{1.5}$$

Definition 6. *A function $(\boldsymbol{\theta}, \boldsymbol{\varphi}) \mapsto \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$ is said convex-concave if $\mathcal{L}(\cdot, \boldsymbol{\varphi})$ is convex for all $\boldsymbol{\varphi} \in \Phi$ and $\mathcal{L}(\boldsymbol{\theta}, \cdot)$ is concave for all $\boldsymbol{\theta} \in \Theta$. An $\mathcal{L}$ is said to be μ-strongly convex-concave if $(\boldsymbol{\theta}, \boldsymbol{\varphi}) \mapsto \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) - \frac{\mu}{2}\|\boldsymbol{\theta}\|_2^2 + \frac{\mu}{2}\|\boldsymbol{\varphi}\|_2^2$ is convex-concave.*

If a function f (resp. $\mathcal{L}$) is strongly convex (resp. strongly convex-concave), its gradient ∇f (resp. $[\nabla_{\boldsymbol{\theta}}\mathcal{L} \;\; -\nabla_{\boldsymbol{\varphi}}\mathcal{L}]^\top$) is strongly monotone, i.e.,

Definition 7. *For $\mu > 0$, an operator $F : \Omega \to \mathbb{R}^d$ is said to be μ-strongly monotone if*

$$(F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}'))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}') \geq \mu\|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2\,. \tag{1.6}$$

2 Gradient methods on unconstrained bilinear games

In this section, we will prove the results provided in §3, namely Proposition 1, Proposition 2 and Theorem 1. For Proposition 1 and 2, let us recall the context. We wanted to derive properties of some gradient methods on the following simple illustrative example

$$\min_{\theta \in \mathbb{R}} \max_{\phi \in \mathbb{R}} \;\; \theta \cdot \phi \tag{2.1}$$

2.1 Proof of Proposition 1

Let us first recall the proposition:

Proposition 1. *The* simultaneous *iterates diverge geometrically and the* alternating *iterates defined in* (3.5) *are bounded but do not converge to 0 as*

$$\text{Simultaneous: } \theta_{t+1}^2+\phi_{t+1}^2=(1+\eta^2)(\theta_t^2+\phi_t^2)\,, \quad \text{Alternating: } \theta_t^2+\phi_t^2=\Theta(\theta_0^2+\phi_0^2) \tag{2.2}$$

where $u_t=\Theta(v_t) \Leftrightarrow \exists \alpha,\beta,t_0>0$ *such that* $\forall t\geq t_0, \alpha v_t\leq u_t\leq \beta v_t$.

The uniform average $(\bar{\theta}_t,\bar{\phi}_t) := \frac{1}{t}\sum_{s=0}^{t-1}(\theta_s,\phi_s)$ *of the* simultaneous *updates (resp. the* alternating updates*) diverges (resp. converges to 0) as,*

$$\text{Sim.: } \bar{\theta}_t^2+\bar{\phi}_t^2=\Theta\left(\frac{\theta_0^2+\phi_0^2}{\eta^2t^2}(1+\eta^2)^t\right), \text{ Alt.: } \bar{\theta}_t^2+\bar{\phi}_t^2=\Theta\left(\frac{\theta_0^2+\phi_0^2}{\eta^2t^2}\right). \tag{2.3}$$

Proof. Let us start with the *simultaneous* update rule:

$$\begin{cases}\theta_{t+1}=\theta_t-\eta\phi_t\\ \phi_{t+1}=\phi_t+\eta\theta_t\,.\end{cases} \tag{2.4}$$

Then we have,

$$\theta_{t+1}^2+\phi_{t+1}^2=(\theta_t-\eta\phi_t)^2+(\phi_t+\eta\theta_t)^2 \tag{2.5}$$
$$=(1+\eta^2)(\theta_t^2+\phi_t^2)\,. \tag{2.6}$$

The update rule (2.4) also gives us,

$$\begin{cases}\eta\phi_t=\theta_t-\theta_{t+1}\\ \eta\theta_t=\phi_{t+1}-\phi_t\,.\end{cases} \tag{2.7}$$

Summing (2.7) for $0\leq t\leq T-1$ to get telescoping sums, we get

$$(\eta^2T^2)(\bar{\phi}_T^2+\bar{\theta}_T^2)=(\theta_0-\theta_T)^2+(\phi_0-\phi_T)^2 \tag{2.8}$$
$$=((1+\eta^2)^T+1)(\theta_0^2+\phi_0^2)-2\theta_0\theta_T-2\phi_0\phi_T \tag{2.9}$$
$$=\Theta\left((1+\eta^2)^T((\theta_0^2+\phi_0^2)\right). \tag{2.10}$$

Let us continue with the *alternating* update rule

$$\begin{cases}\theta_{t+1}=\theta_t-\eta\phi_t\\ \phi_{t+1}=\phi_t+\eta\theta_{t+1}=\phi_t+\eta(\theta_t-\eta\phi_t)\end{cases} \tag{2.11}$$

Then we have,

$$\begin{bmatrix}\theta_{t+1}\\ \phi_{t+1}\end{bmatrix}=\begin{bmatrix}1 & -\eta\\ \eta & 1-\eta^2\end{bmatrix}\begin{bmatrix}\theta_t\\ \phi_t\end{bmatrix}. \tag{2.12}$$

By simple linear algebra, for $\eta<2$, the matrix $M:=\begin{bmatrix}1 & -\eta\\ \eta & 1-\eta^2\end{bmatrix}$ has two complex conjugate eigenvalues which are

$$\lambda_\pm=1-\eta\frac{\eta\pm i\sqrt{4-\eta^2}}{2} \tag{2.13}$$

and their squared magnitude is equal to $\det(M) = 1 - \eta^2 + \eta^2 = 1$. We can diagonalize M meaning that there exists P an invertible matrix such that $M = P^{-1}\operatorname{diag}(\lambda_+, \lambda_-)P$. Then, we have

$$\begin{bmatrix}\theta_t \\ \phi_t\end{bmatrix} = M^t \begin{bmatrix}\theta_0 \\ \phi_0\end{bmatrix} = P^{-1}\operatorname{diag}(\lambda_+^t, \lambda_-^t)P \begin{bmatrix}\theta_0 \\ \phi_0\end{bmatrix} \tag{2.14}$$

and consequently,

$$\theta_t^2 + \phi_t^2 = \left\|\begin{bmatrix}\theta_t \\ \phi_t\end{bmatrix}\right\|_{\mathbb{C}}^2 = \left\|P^{-1}\operatorname{diag}(\lambda_+^t, \lambda_-^t)P \begin{bmatrix}\theta_0 \\ \phi_0\end{bmatrix}\right\|_{\mathbb{C}}^2 \leq \|P^{-1}\|\|P\|(\theta_0^2 + \phi_0^2) \tag{2.15}$$

where $\|\cdot\|_{\mathbb{C}}$ is the norm in $\mathbb{C}^2$ and $\|P\| := \max_{u\in\mathbb{C}^2} \frac{\|Pu\|_{\mathbb{C}}}{\|u\|_{\mathbb{C}}}$ is the induced matrix norm. The same way we have,

$$\theta_0^2 + \phi_0^2 = \left\|M^{-t}\begin{bmatrix}\theta_t \\ \phi_t\end{bmatrix}\right\|_{\mathbb{C}}^2 = \left\|P^{-1}\operatorname{diag}(\lambda_+^{-t}, \lambda_-^{-t})P \begin{bmatrix}\theta_t \\ \phi_t\end{bmatrix}\right\|_{\mathbb{C}}^2 \leq \|P^{-1}\|\|P\|(\theta_t^2 + \phi_t^2) \tag{2.16}$$

Hence, if $\theta_0^2 + \phi_0^2 > 0$, the sequence (θ_t, ϕ_t) is bounded but do not converge to 0. Moreover the update rule gives us,

$$\begin{cases}\eta\phi_t = \theta_t - \theta_{t+1} \\ \eta\theta_t = \phi_t - \phi_{t-1}\end{cases} \Rightarrow \begin{cases}\dfrac{\eta}{T}\displaystyle\sum_{t=0}^{T-1}\phi_t = \dfrac{\theta_0 - \theta_T}{T} \\ \dfrac{\eta}{T}\displaystyle\sum_{t=0}^{T-1}\theta_t = \dfrac{\phi_{T-1} - \phi_0 + \eta\theta_0}{T}\end{cases} \Rightarrow \begin{cases}\bar{\phi}_T = \dfrac{\theta_0 - \theta_T}{\eta T} \\ \bar{\theta}_T = \dfrac{\phi_{T-1} - \phi_0 + \eta\theta_0}{\eta T}\end{cases} \tag{2.17}$$

Consequently, since $\theta_t^2 + \phi_t^2 = \Theta(\theta_0^2 + \phi_0^2)$,

$$\sqrt{\bar{\theta}_t^2 + \bar{\phi}_t^2} = \Theta\left(\frac{\sqrt{\theta_0^2 + \phi_0^2}}{\eta t}\right) \tag{2.18}$$

□

2.2 Implicit and extrapolation method

In this section, we will prove a slightly more precise proposition than Proposition 2,

Proposition 2. *The squared norm of the iterates $N_t^2 := \theta_t^2 + \phi_t^2$, where the update rule of θ_t and ϕ_t is defined in (3.11), decrease geometrically for any $0 < \eta < 1$ as,*[1]

$$\text{Implicit: } N_{t+1}^2 = \frac{N_t^2}{1+\eta^2}, \quad \text{Extrapolation: } N_{t+1}^2 = (1 - \eta^2 + \eta^4)N_t^2, \quad \forall t \geq 0 \tag{2.19}$$

[1]Note that the relationship (2.19) holds actually for *any* η for the implicit method, and thus the decrease is geometric for any non-zero step size.

Proof. Let us recall the update rule for the implicit method

$$\begin{cases} \theta_{t+1} = \theta_t - \eta\phi_{t+1} \\ \phi_{t+1} = \phi_t + \eta\theta_{t+1} \end{cases} \Rightarrow \begin{cases} (1+\eta^2)\theta_{t+1} = \theta_t - \eta\phi_t \\ (1+\eta^2)\phi_{t+1} = \phi_t + \eta\theta_t \end{cases} \tag{2.20}$$

Then,

$$(1+\eta^2)^2(\theta_{t+1}^2 + \phi_{t+1}^2) = (\theta_t - \eta\phi_t)^2 + (\phi_t + \eta\theta_t)^2 \tag{2.21}$$

$$= \theta_t^2 + \phi_t^2 + +\eta^2(\theta_t^2 + \phi_t^2), \tag{2.22}$$

implying that

$$\theta_{t+1}^2 + \phi_{t+1}^2 = \frac{\theta_t^2 + \phi_t^2}{1+\eta^2}, \tag{2.23}$$

which is valid for *any* η.

For the extrapolation method, we have the update rule

$$\begin{cases} \theta_{t+1} = \theta_t - \eta(\phi_t + \eta\theta_t) \\ \phi_{t+1} = \phi_t + \eta(\theta_t - \eta\phi_t) \end{cases} \tag{2.24}$$

Implying that,

$$\theta_{t+1}^2 + \phi_{t+1}^2 = (\theta_t - \eta(\phi_t + \eta\theta_t))^2 + (\phi_t + \eta(\theta_t - \eta\phi_t))^2 \tag{2.25}$$

$$= \theta_t^2 + \phi_t^2 - 2\eta^2(\theta^2 + \phi^2) + \eta^2((\theta_t - \eta\phi_t)^2 + (\phi_t + \eta\theta_t)^2) \tag{2.26}$$

$$= (1 - \eta^2 + \eta^4)(\theta_t^2 + \phi_t^2) \tag{2.27}$$

□

2.3 Generalization to general unconstrained bilinear objective

In this section, we will show how to simply extend the study of the algorithm of interest provided in §3 on the general unconstrained bilinear example,

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\varphi}\in\mathbb{R}^p} \; \boldsymbol{\theta}^\top \boldsymbol{A}\boldsymbol{\varphi} - \boldsymbol{b}^\top\boldsymbol{\theta} - \boldsymbol{c}^\top\boldsymbol{\varphi} \tag{2.28}$$

where, $\boldsymbol{A} \in \mathbb{R}^{d\times p}$, $\boldsymbol{b} \in \mathbb{R}^d$ and $\boldsymbol{c} \in \mathbb{R}^p$. The only assumption we will make is that this problem is feasible which is equivalent to say that there exists a solution $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ to the system

$$\begin{cases} \boldsymbol{A}\boldsymbol{\varphi}^* = \boldsymbol{b} \\ \boldsymbol{A}^\top\boldsymbol{\theta}^* = \boldsymbol{c}\,. \end{cases} \tag{2.29}$$

In this case, we can re-write (2.28) as

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^d} \max_{\boldsymbol{\varphi}\in\mathbb{R}^p} \; (\boldsymbol{\theta} - \boldsymbol{\theta}^*)^\top \boldsymbol{A}(\boldsymbol{\varphi} - \boldsymbol{\varphi}^*) + c \tag{2.30}$$

where $c := -\boldsymbol{\theta}^{*\top} \boldsymbol{A} \boldsymbol{\varphi}^*$ is a constant that does not depend on $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$.

First, let us show that we can reduce the study of simultaneous, alternating, extrapolation and implicit updates rules for (2.28) to the study of the respective unidimensional updates (3.5) and (3.11).

This reduction has already been proposed by Gidel et al. [2019c]. For completeness, we reproduce here similar arguments. The following lemma is a bit more general than the result provided by Gidel et al. [2019c]. It states that the study of a wide class of *unconstrained* first order method on (2.28) can be reduced to the study of the method on (2.1), with potentially rescaled step-sizes.

Before explicitly stating the lemma, we need to introduce a bit of notation to encompass easily our several methods in a unified way. First, we let $\boldsymbol{\omega}_t := (\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$, where the index t here is a more general index which can vary more often than the one in §3. For example, for the extrapolation method, we could consider $\boldsymbol{\omega}_1 = \boldsymbol{\omega}'_{0+1/2}$ and $\boldsymbol{\omega}_2 = \boldsymbol{\omega}'_1$, where $\boldsymbol{\omega}'$ was the sequence defined for the extragradient. For the alternated updates, we can consider $\boldsymbol{\omega}_1 = (\boldsymbol{\theta}'_1, \boldsymbol{\varphi}'_0)$ and $\boldsymbol{\omega}_2 = (\boldsymbol{\theta}'_1, \boldsymbol{\varphi}'_1)$ (this also defines $\boldsymbol{\theta}_2 = \boldsymbol{\theta}'_1$), where $\boldsymbol{\theta}'$ and $\boldsymbol{\varphi}'$ were the sequences originally defined for alternated updates. We are thus ready to state the lemma.

Lemma 4. *Let us consider the following very general class of first order methods on (2.28), i.e.,*

$$\begin{aligned} \boldsymbol{\theta}_t &\in \boldsymbol{\theta}_0 + \mathit{span}(F_{\boldsymbol{\theta}}(\boldsymbol{\omega}_0), \dots, F_{\boldsymbol{\theta}}(\boldsymbol{\omega}_t)), \quad \forall t \in \mathbb{N}, \\ \boldsymbol{\varphi}_t &\in \boldsymbol{\varphi}_0 + \mathit{span}(F_{\boldsymbol{\varphi}}(\boldsymbol{\omega}_0), \dots, F_{\boldsymbol{\varphi}}(\boldsymbol{\omega}_t)), \quad \forall t \in \mathbb{N}, \end{aligned} \tag{2.31}$$

where $\boldsymbol{\omega}_t := (\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$ and $F_{\boldsymbol{\theta}}(\boldsymbol{\omega}_t) := \boldsymbol{A}\boldsymbol{\varphi}_t - \boldsymbol{b}$, $F_{\boldsymbol{\varphi}}(\boldsymbol{\omega}_t) = \boldsymbol{A}^\top \boldsymbol{\theta}_t - \boldsymbol{c}$. Then, we have

$$\boldsymbol{\theta}_t = \boldsymbol{U}^\top (\tilde{\boldsymbol{\theta}}_t - \boldsymbol{\theta}^*) \quad \textit{and} \quad \boldsymbol{\varphi}_t = \boldsymbol{V}^\top (\tilde{\boldsymbol{\varphi}}_t - \boldsymbol{\varphi}^*), \tag{2.32}$$

where $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$ (SVD decomposition) and the couples $([\tilde{\boldsymbol{\theta}}_t]_i, [\tilde{\boldsymbol{\varphi}}]_i)_{1 \le i \le r}$ follow the update rule of the same method on a unidimensional problem (2.1). In particular, for our methods of interest, the couples $([\tilde{\boldsymbol{\theta}}_t]_i, [\tilde{\boldsymbol{\varphi}}]_i)_{1 \le i \le r}$ follow the same update rule with a respective step-size $\sigma_i \eta$, where σ_i are the singular values on the diagonal of $\boldsymbol{D}$.

Proof. Our general class of first order methods can be written with the following update rules:

$$\begin{aligned} \boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_0 + \sum_{s=0}^{t+1} \lambda_{st} \boldsymbol{A}(\boldsymbol{\varphi}_s - \boldsymbol{\varphi}^*) \\ \boldsymbol{\varphi}_{t+1} &= \boldsymbol{\varphi}_0 + \sum_{s=0}^{t+1} \mu_{st} \boldsymbol{A}^\top (\boldsymbol{\theta}_s - \boldsymbol{\theta}^*), \end{aligned}$$

where λ_{it}, $\mu_{it} \in \mathbb{R}$, $0 \le i \le t+1$. We allow the dependence on t for the algorithm coefficients λ and μ (for example, the alternating rule would zero out some of the

coefficients depending on whether we are updating $\boldsymbol{\theta}$ or $\boldsymbol{\varphi}$ at the current iteration). Notice also that if both $\lambda_{(t+1)t}$ and $\mu_{(t+1)t}$ are non-zero, we have an *implicit* scheme.

Thus, using the SVD of $\boldsymbol{A} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top$, we get

$$\begin{aligned}
\boldsymbol{U}^\top(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*) &= \boldsymbol{U}^\top(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) + \sum_{s=0}^{t+1} \lambda_{st} \boldsymbol{D}\boldsymbol{V}^\top(\boldsymbol{\varphi}_s - \boldsymbol{\varphi}^*) \\
\boldsymbol{V}^\top(\boldsymbol{\varphi}_{t+1} - \boldsymbol{\varphi}^*) &= \boldsymbol{V}^\top(\boldsymbol{\varphi}_0 - \boldsymbol{\varphi}^*) + \sum_{s=0}^{t+1} \mu_{st} \boldsymbol{D}^\top\boldsymbol{U}^\top(\boldsymbol{\theta}_s - \boldsymbol{\theta}^*)\,,
\end{aligned}$$

which is equivalent to

$$\begin{cases}
\tilde{\boldsymbol{\theta}}_{t+1} = \tilde{\boldsymbol{\theta}}_0 + \sum_{s=0}^{t+1} \lambda_{st} \boldsymbol{D}\tilde{\boldsymbol{\varphi}}_s \\
\tilde{\boldsymbol{\varphi}}_{t+1} = \tilde{\boldsymbol{\varphi}}_0 + \sum_{s=0}^{t+1} \mu_{st} \boldsymbol{D}^\top\tilde{\boldsymbol{\theta}}_s\,,
\end{cases} \tag{2.33}$$

where $\boldsymbol{D}$ is a rectangular matrix with zeros except on a diagonal block of size r. Thus, each coordinate of $\tilde{\boldsymbol{\theta}}_{t+1}$ and $\tilde{\boldsymbol{\varphi}}_{t+1}$ are updated independently, reducing the initial problem to r unidimensional problems,

$$\begin{cases}
[\tilde{\boldsymbol{\theta}}_{t+1}]_i = [\tilde{\boldsymbol{\theta}}_0]_i + \sum_{s=0}^{t+1} \lambda_{st}\sigma_i[\tilde{\boldsymbol{\varphi}}_s]_i \\
[\tilde{\boldsymbol{\varphi}}_{t+1}]_i = [\tilde{\boldsymbol{\varphi}}_0]_i + \sum_{s=0}^{t+1} \mu_{st}\sigma_i[\tilde{\boldsymbol{\theta}}_s]_i
\end{cases} \qquad 1 \le i \le r\,, \tag{2.34}$$

where $\sigma_1 \ge \ldots \ge \sigma_r > 0$ are the positive diagonal coefficients of $\boldsymbol{D}$.

Finally, for the coordinate i where the diagonal coefficient of $\boldsymbol{D}$ is equal to 0, we can notice that the sequence $([\tilde{\boldsymbol{\theta}}_t]_i, [\tilde{\boldsymbol{\varphi}}_t]_i)$ is constant. Moreover, we have the freedom to chose any $[\boldsymbol{\theta}^*]_i \in \mathbb{R}$ and $[\boldsymbol{\varphi}^*]_i \in \mathbb{R}$ as a coordinate of the solution of (2.28). We thus set them respectively equal to $[\boldsymbol{\theta}_0]_i$ and $[\boldsymbol{\varphi}_0]_i$. The update rule (2.34) corresponds to the update rule of the general first order method considered on this proof on the unidimensional problem (2.1).

Note that the only additional restriction is that the coefficients (λ_{st}) and (σ_{st}) (that are the same for $1 \le i \le r$) are rescaled by the singular values of $\boldsymbol{A}$. In practice, for our methods of interest with a step-size η, it corresponds to the study of r unidimensional problem with a respective step-size $\sigma_i\eta$, $1 \le i \le r$. □

From this lemma, an extension of Proposition 1 and 2 directly follows to the general unconstrained bilinear objective (2.28). We note

$$N_t^2 := \mathrm{dist}(\boldsymbol{\theta}_t, \Theta^*)^2 + \mathrm{dist}(\boldsymbol{\varphi}_t, \Phi^*)^2\,, \tag{2.35}$$

where (Θ^*, Φ^*) is the set of solutions of (2.28). The following corollary is divided in two points, the first point is a result from Gidel et al. [2019c] (note that the result on

the average is a straightforward extension of the one provided in Proposition 1 and was not provided by Gidel et al. [2019c]), the second result is new. Very similar asymptotic upper bounds regarding extrapolation and implicit methods can be derived by Tseng [1995] computing the exact values of the constant τ_1 and τ_2 (and noticing that $\tau_3 = \infty$) introduced in [Tseng, 1995, Eq. 3 & 4] for the unconstrained bilinear case. However, since Tseng [1995] works in a very general setting, the bound are not as tight as ours and his proof technique is a bit more technical. Our reduction above provides here a simple proof for our simple setting.

Corollary 1. • *Gidel et al. [2019c]: The* simultaneous *iterates diverge geometrically and the* alternating *iterates are bounded but do not converge to 0 as,*

$$\text{Simultaneous: } N_{t+1}^2 = (1 + (\sigma_{\min}(\boldsymbol{A})\eta)^2)N_t^2\,, \quad \text{Alternating: } N_t^2 = \Theta(N_0^2)\,, \tag{2.36}$$

where $u_t = \Theta(v_t) \Leftrightarrow \exists \alpha, \beta, t_0 > 0$ *such that* $\forall t \geq t_0$, $\alpha v_t \leq u_t \leq \beta v_t$. *The uniform average* $(\bar{\theta}_t, \bar{\phi}_t) := \frac{1}{t}\sum_{s=0}^{t-1}(\theta_s, \phi_s)$ *of the* simultaneous *updates (resp. the* alternating updates*) diverges (resp. converges to 0) as,*

$$\text{Sim.: } \bar{N}_t^2 \leq \Theta\left(\frac{N_0^2}{t^2}(1 + (\sigma_{\min}(\boldsymbol{A})\eta)^2)^t\right), \text{ Alt.: } \bar{N}_t^2 = \Theta\left(\frac{N_0^2}{t^2}\right).$$

- Extrapolation *and* Implicit method*: The iterates respectively generated by the update rules* (3.8) *and* (3.10) *on a bilinear unconstrained problem* (2.28) *do converge linearly for any* $0 < \eta < \frac{1}{\sigma_{\max}(\boldsymbol{A})}$ *at a rate,*[2]

$$\text{Implicit: } N_{t+1}^2 \leq \frac{N_t^2}{1 + (\sigma_{\min}(\boldsymbol{A})\eta)^2}\,, \quad \forall t \geq 0 \tag{2.37}$$

$$\text{Extrapolation: } N_{t+1}^2 \leq (1 - (\sigma_{\min}(\boldsymbol{A})\eta)^2 + (\sigma_{\min}(\boldsymbol{A})\eta)^4)N_t^2\,, \quad \forall t \geq 0\,. \tag{2.38}$$

Particularly, for $\eta = \frac{1}{2\sigma_{\max}(\boldsymbol{A})}$ *we get for the extrapolation method,*

$$\text{Extrapolation: } N_{t+1}^2 \leq (1 - \tfrac{1}{8\kappa})^t N_0^2\,, \quad \forall t \geq 0\,. \tag{2.39}$$

where $\kappa := \frac{\sigma_{\max}(\boldsymbol{A})^2}{\sigma_{\min}(\boldsymbol{A})^2}$ *is the condition number of* $\boldsymbol{A}^\top \boldsymbol{A}$.

2.4 Extrapolation from the past for strongly convex objectives

Let us recall what we call *projected extrapolation form the past*, where we used the notation $\boldsymbol{\omega}'_t = \boldsymbol{\omega}_{t+1/2}$ for compactness,

$$\text{Extrapolation from the past: } \boldsymbol{\omega}'_t = P_\Omega[\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}'_{t-1})] \tag{2.40}$$

$$\text{Perform update step: } \boldsymbol{\omega}_{t+1} = P_\Omega[\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}'_t)] \text{ and store: } F(\boldsymbol{\omega}'_t) \tag{2.41}$$

[2] As before, the inequality (2.38) for the implicit scheme is actually valid for any step-size.

where $P_\Omega[\cdot]$ is the projection onto the constraint set Ω. An operator $F : \Omega \to \mathbb{R}^d$ is said to be μ-strongly monotone if

$$(F(\boldsymbol{\omega}) - F(\boldsymbol{\omega}'))^\top (\boldsymbol{\omega} - \boldsymbol{\omega}') \geq \mu \|\boldsymbol{\omega} - \boldsymbol{\omega}'\|_2^2 \,. \tag{2.42}$$

If F is strongly monotone, we can prove the following theorem:

Theorem 1. *If F is μ-strongly monotone (see Appendix B §1 for the definition of strong monotonicity) and L-Lipschitz, then the updates (3.13) and (3.14) with $\eta = \frac{1}{4L}$ provide linearly converging iterates,*

$$\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)^t \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|_2^2 \,, \quad \forall t \geq 0 \,. \tag{2.43}$$

Proof. In order to prove this theorem, we will prove a slightly more general result,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 + \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right) (\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 + \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2) \,. \tag{2.44}$$

with the convention that $\boldsymbol{\omega}'_0 = \boldsymbol{\omega}'_{-1} = \boldsymbol{\omega}'_{-2}$. It implies that

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 \leq \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 + \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)^t \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|_2^2 \,. \tag{2.45}$$

Let us first proof three technical lemmas.

Lemma 5. *If F is μ-strongly monotone, we have*

$$\mu \left(\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 - 2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \right) \leq 2F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*) \,, \quad \forall\, \boldsymbol{\omega}^* \in \Omega^* \,. \tag{2.46}$$

Proof. By strong monotonicity and optimality of $\boldsymbol{\omega}^*$,

$$2\mu\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*\|_2^2 \leq 2F(\boldsymbol{\omega}^*)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*) + 2\mu\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*\|_2^2 \leq 2F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*) \tag{2.47}$$

and then we use the inequality $2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}^*\|_2^2 \geq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 - 2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2$ to get the result claimed. □

Lemma 6. *If F is L-Lipschitz, we have for any $\boldsymbol{\omega} \in \Omega$,*

$$2\eta_t F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 + \eta_t^2 L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \,. \tag{2.48}$$

Proof. Applying Lemma 3 for $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}^+, \boldsymbol{\omega}') = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}'_t), \boldsymbol{\omega}_{t+1}, \boldsymbol{\omega})$ and $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}^+, \boldsymbol{\omega}') = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}'_{t-1}), \boldsymbol{\omega}'_t, \boldsymbol{\omega}_{t+1})$, we get,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2 \tag{2.49}$$

and

$$\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_{t-1})^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \,. \tag{2.50}$$

Summing (2.49) and (2.50) we get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}) && (2.51)\\ &\quad - 2\eta_t F(\boldsymbol{\omega}'_{t-1})^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1}\|_2^2 && (2.52)\\ &= \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}) - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1}\|_2^2\\ &\quad - 2\eta_t(F(\boldsymbol{\omega}'_{t-1})-F(\boldsymbol{\omega}'_t))^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1})\,. && (2.53)\end{aligned}$$

Then, we can use the Young's inequality $2a^\top b \leq \|a\|_2^2 + \|b\|_2^2$ to get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}) + \eta_t^2\|F(\boldsymbol{\omega}'_{t-1})-F(\boldsymbol{\omega}'_t)\|_2^2\\ &\quad + \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1}\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_{t+1}\|_2^2 && (2.54)\\ &= \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega})\\ &\quad + \eta_t^2\|F(\boldsymbol{\omega}'_{t-1})-F(\boldsymbol{\omega}'_t)\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2\\ &\leq \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}) + \eta_t^2 L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2\,. && (2.55)\end{aligned}$$

□

Lemma 7. *For all $t \geq 0$, if we set $\boldsymbol{\omega}'_{-2} = \boldsymbol{\omega}'_{-1} = \boldsymbol{\omega}'_0$ we have*

$$\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 \leq 4\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_t\|_2^2 + 4\eta_{t-1}^2L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2 - \|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2\,. \quad (2.56)$$

Proof. We start with $\|a+b\|_2^2 \leq 2\|a\|^2 + 2\|b\|^2$.

$$\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 \leq 2\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_t\|_2^2 + 2\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_{t-1}\|_2^2\,. \quad (2.57)$$

Moreover, since the projection is contractive we have that

$$\begin{aligned}\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_{t-1}\|_2^2 &\leq \|\boldsymbol{\omega}_{t-1}-\eta_{t-1}F(\boldsymbol{\omega}'_{t-1})-\boldsymbol{\omega}_{t-1}-\eta_{t-1}F(\boldsymbol{\omega}'_{t-2})\|_2^2 && (2.58)\\ &= \eta_{t-1}^2\|F(\boldsymbol{\omega}'_{t-1})-F(\boldsymbol{\omega}'_{t-2})\|_2^2 && (2.59)\\ &\leq \eta_{t-1}^2L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2\,. && (2.60)\end{aligned}$$

Combining (2.57) and (2.60) we get,

$$\begin{aligned}\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 &= 2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 && (2.61)\\ &\leq 4\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_t\|_2^2 + 4\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_{t-1}\|_2^2 - \|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 && (2.62)\\ &\leq 4\|\boldsymbol{\omega}_t-\boldsymbol{\omega}'_t\|_2^2 + 4\eta_{t-1}^2L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2 - \|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2\,. && (2.63)\end{aligned}$$

□

Proof of Theorem 1. Let $\boldsymbol{\omega}^* \in \Omega^*$ be an optimal point of (VIP). Combining Lemma 5 and Lemma 6 we get,

$$\begin{aligned}\eta_t\mu\left(\|\boldsymbol{\omega}_t-\boldsymbol{\omega}^*\|_2^2 - 2\|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2\right) &\leq \|\boldsymbol{\omega}_t-\boldsymbol{\omega}^*\|_2^2 - \|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}^*\|_2^2\\ &\quad + \eta_t^2L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2\end{aligned}$$

leading to,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 \leq (1 - \eta_t\mu)\,\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 + \eta_t^2 L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 - (1 - 2\eta_t\mu)\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \tag{2.64}$$

Now using Lemma 7 we get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 \leq (1 - \eta_t\mu)\,\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 + \eta_t^2 L^2 (4\eta_{t-1}^2 L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2 \\ - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2) - (1 - 2\eta_t\mu - 4\eta_t^2 L^2)\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2\end{aligned} \tag{2.65}$$

Now with $\eta_t = \frac{1}{4L} \leq \frac{1}{4\mu}$ we get,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 + \frac{1}{16}\left(\frac{1}{4}\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2 - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2\right)$$

Hence, using the fact that $\frac{\mu}{4L} \leq \frac{1}{4}$ we get,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}^*\|_2^2 + \tfrac{1}{16}\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \leq \left(1 - \frac{\mu}{4L}\right)\left(\|\boldsymbol{\omega}_t - \boldsymbol{\omega}^*\|_2^2 + \tfrac{1}{16}\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2\right). \tag{2.66}$$

□

3 More on merit functions

In this section, we will present how to handle an *unbounded* constraint set Ω with a more refined *merit function* than (4.2) used in the main paper. Let F be the continuous operator and Ω be the constraint set associated with the VIP,

$$\text{find } \boldsymbol{\omega}^* \in \Omega \quad \text{such that} \quad F(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \;\; \forall \boldsymbol{\omega} \in \Omega\,. \tag{VIP}$$

When the operator F is monotone, we have that

$$F(\boldsymbol{\omega}^*)^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \leq F(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*)\,, \;\; \forall \boldsymbol{\omega}, \boldsymbol{\omega}^*\,. \tag{3.1}$$

Hence, in this case (VIP) implies a stronger formulation sometimes called *Minty variational inequality* [Crespi et al., 2005]:

$$\text{find } \boldsymbol{\omega}^* \in \Omega \quad \text{such that} \quad F(\boldsymbol{\omega})^\top(\boldsymbol{\omega} - \boldsymbol{\omega}^*) \geq 0\,, \;\; \forall \boldsymbol{\omega} \in \Omega\,. \tag{MVI}$$

This formulation is stronger in the sense that if (MVI) holds for some $\boldsymbol{\omega}^* \in \Omega$, then (VIP) holds too. A *merit function* useful for our analysis can be derived from this formulation. Roughly, a merit function is a convergence measure. More formally, a function $g : \Omega \to \mathbb{R}$ is called a *merit function* if g is non-negative such that $g(\boldsymbol{\omega}) = 0 \Leftrightarrow \boldsymbol{\omega} \in \Omega^*$ [Larsson and Patriksson, 1994]. A way to derive a merit function from (MVI) would be to use $g(\boldsymbol{\omega}^*) = \sup_{\boldsymbol{\omega}\in\Omega} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}^* - \boldsymbol{\omega})$ which is

zero if and only if (MVI) holds for $\boldsymbol{\omega}^*$. To deal with unbounded constraint sets (leading to a potentially infinite valued function outside of the optimal set), we use the *restricted merit function* [Nesterov, 2007]:

$$\mathrm{Err}_R(\boldsymbol{\omega}_t) := \max_{\boldsymbol{\omega}\in\Omega, \|\boldsymbol{\omega}-\boldsymbol{\omega}_0\|\leq R} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}_t - \boldsymbol{\omega})\,. \tag{3.2}$$

This function acts as merit function for (VIP) on the interior of the open ball of radius R around $\boldsymbol{\omega}_0$, as shown in Lemma 1 of Nesterov [2007]. That is, let $\Omega_R := \Omega \cap \{\boldsymbol{\omega} : \|\boldsymbol{\omega} - \boldsymbol{\omega}_0\| < R\}$. Then for any point $\hat{\boldsymbol{\omega}} \in \Omega_R$, we have:

$$\mathrm{Err}_R(\hat{\boldsymbol{\omega}}) = 0 \Leftrightarrow \hat{\boldsymbol{\omega}} \in \Omega^* \cap \Omega_R. \tag{3.3}$$

The reference point $\boldsymbol{\omega}_0$ is arbitrary, but in practice it is usually the initialization point of the algorithm. R has to be big enough to ensure that Ω_R contains a solution. Err_R measures how much (MVI) is violated on the restriction Ω_R. Such merit function is standard in the variational inequality literature. A similar one is used in [Nemirovski, 2004, Juditsky et al., 2011]. When F is derived from the gradients (2.5) of a zero-sum game, we can define a more interpretable merit function. One has to be careful though when extending properties from the minimization setting to the saddle point setting (e.g. the merit function used by Yadav et al. [2018] is vacuous for a bilinear game as explained in App 3.2).

In the appendix, we adopt a set of assumptions a little more general than the one in the main paper:

Assumption 5.
- *F is* monotone *and Ω is convex and closed.*
- *R is set big enough such that $R > \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}^*\|$ and F is a* monotone *operator.*

Contrary to Assumption 4, in Assumption 5 the constraint set in no longer assumed to be bounded. Assumption 5 is implied by Assumption 4 by setting R to the diameter of Ω, and is thus more general.

3.1 More general merit functions

In this appendix, we will note $\mathrm{Err}_R^{(\mathrm{VI})}$ the *restricted merit function* defined in (3.2). Let us recall its definition,

$$\mathrm{Err}_R^{(\mathrm{VI})}(\boldsymbol{\omega}_t) := \max_{\boldsymbol{\omega}\in\Omega, \|\boldsymbol{\omega}-\boldsymbol{\omega}_0\|\leq R} F(\boldsymbol{\omega})^\top(\boldsymbol{\omega}_t - \boldsymbol{\omega})\,. \tag{3.4}$$

When the objective is a saddle point problem i.e.,

$$F(\boldsymbol{\theta}, \boldsymbol{\varphi}) = [\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \;\; -\nabla_{\boldsymbol{\varphi}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})]^\top \tag{3.5}$$

and $\mathcal{L}$ is *convex-concave* (see Definition 6 in §1), we can use another merit function than (3.4) on Ω_R that is more interpretable and more directly related to the cost function of the minimax formulation:

$$\mathrm{Err}_R^{(\mathrm{SP})}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t) := \max_{\substack{\boldsymbol{\varphi}\in\Phi, \boldsymbol{\theta}\in\Theta \\ \|(\boldsymbol{\theta},\boldsymbol{\varphi})-(\boldsymbol{\theta}_0,\boldsymbol{\varphi}_0)\|\leq R}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}_t)\,. \tag{3.6}$$

In particular, if the equilibrium $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*) \in \Omega^* \cap \Omega_R$ and we have that $\mathcal{L}(\cdot, \boldsymbol{\varphi}^*)$ and $-\mathcal{L}(\boldsymbol{\theta}^*, \cdot)$ are μ-strongly convex (see §1), then the merit function for saddle points upper bounds the distance for $(\boldsymbol{\theta}, \boldsymbol{\varphi}) \in \Omega_R$ to the equilibrium as:

$$\mathrm{Err}_R^{(\mathrm{SP})}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \geq \frac{\mu}{2}(\|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\varphi} - \boldsymbol{\varphi}^*\|_2^2)\,. \tag{3.7}$$

In the appendix, we provide our convergence results with the merit functions (3.4) and (3.6), depending on the setup:

$$\mathrm{Err}_R(\boldsymbol{\omega}) := \begin{cases} \mathrm{Err}_R^{(\mathrm{SP})}(\boldsymbol{\omega}) & \text{if } F \text{ is a SP operator (2.5)} \\ \mathrm{Err}_R^{(\mathrm{VI})}(\boldsymbol{\omega}) & \text{otherwise.} \end{cases} \tag{3.8}$$

3.2 On the importance of the merit function

In this section, we illustrate the fact that one has to be careful when extending results and properties from the minimization setting to the minimax setting (and consequently to the variational inequality setting). Another candidate as a merit function for saddle point optimization would be to naturally extend the suboptimality $f(\boldsymbol{\omega}) - f(\boldsymbol{\omega}^*)$ used in standard minimization (i.e. find $\boldsymbol{\omega}^*$ the minimizer of f) to the gap $P(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}^*) - \mathcal{L}(\boldsymbol{\theta}^*, \boldsymbol{\varphi})$. In a previous analysis of a modification of the stochastic gradient descent (SGD) method for GANs, Yadav et al. [2018] gave their convergence rate on P that they called the "primal-dual" gap. Unfortunately, if we do not assume that the function $\mathcal{L}$ is strongly convex-concave (a stronger assumption defined in §1 and which fails for bilinear objective e.g.), P may not be a *merit function*. It can be 0 for a non optimal point, see for instance the discussion on the differences between (3.6) and P in [Gidel et al., 2017, Section 3]. In particular, for the simple 2D bilinear example $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = \boldsymbol{\theta} \cdot \boldsymbol{\varphi}$, we have that $\boldsymbol{\theta}^* = \boldsymbol{\varphi}^* = 0$ and thus $P(\boldsymbol{\theta}, \boldsymbol{\varphi}) = 0 \quad \forall \boldsymbol{\theta}, \boldsymbol{\varphi}\,.$

3.3 Variational inequalities for non-convex cost functions

When the cost functions defined in (2.3) are non-convex, the operator F is no longer monotone. Nevertheless, (VIP) and (MVI) can still be defined, though a solution to (MVI) is less likely to exist. We note that (VIP) is a local condition for F (as only evaluating F at the points $\boldsymbol{\omega}^*$). On the other hand, an appealing property of (MVI) is that it is a global condition. In the context of minimization of a function

f for example (where $F = \nabla f$), if $\boldsymbol{\omega}^*$ solves (MVI) then $\boldsymbol{\omega}^*$ is a *global* minimum of f (and not just a stationary point for the solution of (MVI); see Proposition 2.2 from Crespi et al. [2005]). A less restrictive way to consider variational inequalities in the non-monotone setting is to use a local version of (MVI). If the cost functions are locally convex around the optimal couple $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ and if our iterates eventually fall and stay into that neighborhood, then we can consider our restricted merit function $\mathrm{Err}_R(\cdot)$ with a well suited constant R and apply our convergence results for monotone operators.

4 Another way of implementing extrapolation to SGD

We now introduce another way to combine extrapolation and SGD. This extension is very similar to AvgExtraSGD Alg. 2, the only difference is that it re-uses the mini-batch sample of the extrapolation step for the update of the current point. The intuition is that it correlates the estimator of the gradient of the extrapolation step and the one of the update step leading to a better correction of the oscillations which are also due to the stochasticity. One emerging issue (for the analysis) of this method is that since $\boldsymbol{\omega}'_t$ depend on ξ_t, the quantity $F(\boldsymbol{\omega}'_t, \xi_t)$ is a biased estimator of $F(\boldsymbol{\omega}'_t)$.

Algorithm 5 Re-used mini-batches for stochastic extrapolation (ReExtraSGD)

Let $\boldsymbol{\omega}_0 \in \Omega$
for $t = 0 \dots T-1$ **do**
  Sample $\xi_t \sim P$
  $\boldsymbol{\omega}'_t := P_\Omega[\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}_t, \xi_t)]$ ▷ Extrapolation step
  $\boldsymbol{\omega}_{t+1} := P_\Omega[\boldsymbol{\omega}_t - \eta_t F\big(\boldsymbol{\omega}'_t, \xi_t\big)]$ ▷ Update step with the **same** sample
end for
Return $\bar{\boldsymbol{\omega}}_T = \sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}'_t / \sum_{t=0}^{T-1} \eta_t$

Theorem 3. *Assume that $\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_0\| \leq R$, $\forall t \geq 0$ where $(\boldsymbol{\omega}'_t)_{t\geq 0}$ are the iterates of Alg. 5. Under Assumption 2 and 5, for any $T \geq 1$, Alg. 5 with constant step-size $\eta \leq \frac{1}{\sqrt{2}L}$ has the following convergence properties:*

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \eta \frac{\sigma^2 + 4L^2(4R^2 + \sigma^2)}{2} \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\omega}'_t \,.$$

Particularly, $\eta_t = \frac{\eta}{\sqrt{T}}$ gives $\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{O(1)}{\sqrt{T}}$.

The assumption that the sequence of the iterates provided by the algorithm is bounded is strong, but has also been made for instance in [Yadav et al., 2018]. The proof of this result is provided in §6.

5 Variance comparison between AvgSGD and SGD with prediction method

To compare the variance term of AvgSGD in (4.3) with the one of the *SGD with prediction method* [Yadav et al., 2018], we need to have the same convergence certificate. Fortunately, their proof can be adapted to our convergence criterion (using Lemma 8 in §6), revealing an extra $\sigma^2/2$ in the variance term from their paper. The resulting variance can be summarized with our notation as $(M^2(1+L)+\sigma^2)/2$ where the L is the Lipschitz constant of the operator F. Since $M \gg \sigma$, their variance term is then $1+L$ time larger than the one provided by the AvgSGD method.

6 Proof of Theorems

This section is dedicated on the proof of the theorems provided in this paper in a slightly more general form working with the merit function defined in (3.8). First we prove an additional lemma necessary to the proof of our theorems.

Lemma 8. *Let F be a monotone operator and let $(\boldsymbol{\omega}_t), (\boldsymbol{\omega}'_t), (\boldsymbol{z}_t), (\Delta_t), (\xi_t)$ and (ζ_t) be six random sequences such that, for all $t \geq 0$*

$$2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq N_t - N_{t+1} + \eta_t^2(M_1(\boldsymbol{\omega}_t, \xi_t) + M_2(\boldsymbol{\omega}'_t, \zeta_t)) + 2\eta_t \Delta_t^\top(\boldsymbol{z}_t - \boldsymbol{\omega})\,,$$

where $N_t = N(\boldsymbol{\omega}_t, \boldsymbol{\omega}'_{t-1}, \boldsymbol{\omega}'_{t-2}) \geq 0$ and we extend $(\boldsymbol{\omega}'_t)$ with $\boldsymbol{\omega}'_{-2} = \boldsymbol{\omega}'_{-1} = \boldsymbol{\omega}'_0$. Let also assume that with $N_0 \leq R$, $\mathbb{E}[\|\Delta_t\|_2^2] \leq \sigma^2$, $\mathbb{E}[\Delta_t | \boldsymbol{z}_t, \Delta_0, \ldots, \Delta_{t-1}] = 0$, $\mathbb{E}[M_1(\boldsymbol{\omega}_t, \xi_t)] \leq M_1$ and $\mathbb{E}[M_2(\boldsymbol{\omega}'_t, \zeta_t)] \leq M_2$, then,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{S_T} + \frac{M_1 + M_2 + \sigma^2}{2S_T} \sum_{t=0}^{T-1} \eta_t^2 \tag{6.1}$$

where $\bar{\boldsymbol{\omega}}_T := \sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}'_t / S_T$ and $S_T := \sum_{t=0}^{T-1} \eta_t$.

Proof of Lemma 8. We sum (8) for $0 \leq t \leq T-1$ to get,

$$2\sum_{t=0}^{T-1} \eta_t F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \sum_{t=0}^{T-1} \left[(N_t - N_{t+1}) + \eta_t^2 ((M_1(\boldsymbol{\omega}_t, \xi_t) + M_2(\boldsymbol{\omega}'_t, \zeta_t)) + 2\eta_t \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{\omega}) \right]. \tag{6.2}$$

We will then upper bound each sum in the right-hand side,

$$\Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{\omega}) = \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{u}_t) + \Delta_t^\top (\boldsymbol{u}_t - \boldsymbol{\omega})$$

where $\boldsymbol{u}_{t+1} := P_\Omega(\boldsymbol{u}_t - \eta_t \Delta_t)$ and $\boldsymbol{u}_0 := \boldsymbol{\omega}_0$. Then,

$$\|\boldsymbol{u}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{u}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t \Delta_t^\top (\boldsymbol{u}_t - \boldsymbol{\omega}) + \eta_t^2 \|\Delta_t\|_2^2$$

leading to

$$2\eta_t \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{\omega}) \leq 2\eta_t \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{u}_t) + \|\boldsymbol{u}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{u}_{t+1} - \boldsymbol{\omega}\|_2^2 + \eta_t^2 \|\Delta_t\|_2^2 \tag{6.3}$$

Then noticing that $\boldsymbol{z}_0 := \boldsymbol{\omega}_0$, back to (6.2) we get a telescoping sum,

$$2\sum_{t=0}^{T-1} \eta_t F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq 2N_0 + \sum_{t=0}^{T-1} \Big[\eta_t^2 ((M_1(\boldsymbol{\omega}_t, \xi_t) + M_2(\boldsymbol{\omega}'_t, \zeta_t)) \tag{6.4}$$

$$+ \|\Delta_t\|_2^2) + 2\eta_t \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{u}_t) \Big]. \tag{6.5}$$

If F is the operator of a convex-concave saddle point (2.5), we get, with $\boldsymbol{\omega}'_t = (\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$

$$\begin{aligned} F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) &\geq \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)^\top (\boldsymbol{\theta}_t - \boldsymbol{\theta}) - \nabla_{\boldsymbol{\varphi}} \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)^\top (\boldsymbol{\varphi}_t - \boldsymbol{\varphi}) \\ &\geq \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}) - \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t) + \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}_t) \\ &\qquad \text{(by convexity and concavity)} \\ &= \mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}_t) \end{aligned}$$

then by convexity of $\mathcal{L}(\cdot, \boldsymbol{\varphi})$ and concavity of $\mathcal{L}(\boldsymbol{\theta}, \cdot)$, we have that,

$$2S_T \sum_{t=0}^{T-1} \frac{\eta_t}{S_T} F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \geq 2S_T \sum_{t=0}^{T-1} \frac{\eta_t}{S_T} (\mathcal{L}(\boldsymbol{\theta}_t, \boldsymbol{\varphi}) - \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}_t)) \tag{6.6}$$

$$\geq \bar{2} S_T (\mathcal{L}(\bar{\boldsymbol{\theta}}_t, \boldsymbol{\varphi}) - \mathcal{L}(\bar{\boldsymbol{\theta}}, \boldsymbol{\varphi}_t)) \tag{6.7}$$

Otherwise if the operator F is just monotone since

$$F(\boldsymbol{\omega}'_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \geq F(\boldsymbol{\omega}')^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \tag{6.8}$$

we have that

$$2S_T \sum_{t=0}^{T-1} \eta_t F(\boldsymbol{\omega}_t')^\top (\boldsymbol{\omega}_t' - \boldsymbol{\omega}) \geq 2S_T \sum_{t=0}^{T-1} \eta_t F(\boldsymbol{\omega}')^\top (\boldsymbol{\omega}_t' - \boldsymbol{\omega}) \tag{6.9}$$

$$= 2S_T F(\boldsymbol{\omega}')^\top (\bar{\boldsymbol{\omega}}_t - \boldsymbol{\omega}) \tag{6.10}$$

In both cases, we can now maximize the left hand side respect to $\boldsymbol{\omega}$ (since the RHS does not depend on $\boldsymbol{\omega}$) to get,

$$2S_T \operatorname{Err}_R(\bar{\boldsymbol{\omega}}_t) \leq 2R^2 + \sum_{t=0}^{T-1} \left[\eta_t^2 ((M_1(\boldsymbol{\omega}_t, \xi_t) + M_2(\boldsymbol{\omega}_t', \zeta_t)) + \|\Delta_t\|_2^2) + 2\eta_t \Delta_t^\top (\boldsymbol{z}_t - \boldsymbol{u}_t) \right]. \tag{6.11}$$

Then taking the expectation, since $\mathbb{E}[\Delta_t | \boldsymbol{z}_t, \boldsymbol{u}_t] = \mathbb{E}[\Delta_t | \boldsymbol{z}_t, \Delta_0, \dots, \Delta_{t-1}] = 0$, $\mathbb{E}_{\zeta_t}[\|\Delta_t\|_2^2] \leq \sigma^2$, $\mathbb{E}_{\xi_t}[M_1(\boldsymbol{\omega}_t, \xi_t)] \leq M_1$ and $\mathbb{E}_{\zeta_t}[M_2(\boldsymbol{\omega}_t', \zeta_t)] \leq M_2$, we get that,

$$\mathbb{E}[\operatorname{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{S_T} + \frac{M_1 + M_2 + \sigma^2}{2S_T} \sum_{t=0}^{T-1} \eta_t^2 \tag{6.12}$$

□

6.1 Proof of Thm. 2

First let us state Theorem 2 in its general form,

Theorem 2. *Under Assumption 2, 3 and 5, Alg. 1 with constant step-size η has the following convergence rate for all $T \geq 1$,*

$$\mathbb{E}[\operatorname{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{2\eta T} + \eta \frac{M^2 + \sigma^2}{2} \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T} \sum_{t=0}^{T-1} \boldsymbol{\omega}_t \,. \tag{6.13}$$

Particularly, $\eta = \frac{R}{\sqrt{T(M^2+\sigma^2)}}$ gives $\mathbb{E}[\operatorname{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R\sqrt{M^2+\sigma^2}}{\sqrt{T}}$.

Proof of Theorem 2. Let any $\boldsymbol{\omega} \in \Omega$ such that $\|\boldsymbol{\omega}_0 - \boldsymbol{\omega}\|_2 \leq R$,

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 &= \|P_\Omega(\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}_t, \xi_t)) - \boldsymbol{\omega}\|_2^2 \\
&\leq \|\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}_t, \xi_t)) - \boldsymbol{\omega}\|_2^2 \\
&\quad \text{(projections are non-contractive, Lemma 2)} \\
&= \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}_t, \xi_t)^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) + \|\eta_t F(\boldsymbol{\omega}_t, \xi_t)\|_2^2
\end{aligned}$$

Then we can make appear the quantity $F(\boldsymbol{\omega}_t)^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega})$ on the left-hand side,

$$2\eta_t F(\boldsymbol{\omega}_t)^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 + \eta_t^2 \|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 \tag{6.14}$$

$$+ 2\eta_t (F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t))^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \tag{6.15}$$

we can sum (6.14) for $0 \leq t \leq T-1$ to get,

$$2\sum_{t=0}^{T-1} \eta_t F(\boldsymbol{\omega}_t)^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \leq \\ \sum_{t=0}^{T-1} \left[(\|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|^2) + \eta_t^2 \|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + 2\eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \right] \quad (6.16)$$

where we noted $\Delta_t := F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t)$.
By monotonicity, $F(\boldsymbol{\omega}_t)^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \geq F(\boldsymbol{\omega})^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega})$ we get,

$$2S_T F(\boldsymbol{\omega})^\top (\bar{\boldsymbol{\omega}}_T - \boldsymbol{\omega}) \leq \sum_{t=0}^{T-1} \Big[(\|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|^2) \quad (6.17)$$

$$+ \eta_t^2 \|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + 2\eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \Big], \quad (6.18)$$

where $S_T := \sum_{t=0}^{T-1} \eta_t$ and $\bar{\boldsymbol{\omega}}_T := \frac{1}{S_T} \sum_{t=0}^{T-1} \eta_t \boldsymbol{\omega}_t$.
We will then upper bound each sum in the right hand side,

$$\Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) = \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{u}_t) + \Delta_t^\top (\boldsymbol{u}_t - \boldsymbol{\omega})$$

where $\boldsymbol{u}_{t+1} := P_\Omega(\boldsymbol{u}_t - \eta_t \Delta_t)$ and $\boldsymbol{u}_0 = \boldsymbol{\omega}_0$. Then,

$$\|\boldsymbol{u}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{u}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t \Delta_t^\top (\boldsymbol{u}_t - \boldsymbol{\omega}) + \eta_t^2 \|\Delta_t\|_2^2$$

leading to

$$2\eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{\omega}) \leq 2\eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{u}_t) + \|\boldsymbol{u}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{u}_{t+1} - \boldsymbol{\omega}\|_2^2 + \eta_t^2 \|\Delta_t\|_2^2 \quad (6.19)$$

Then noticing that $\boldsymbol{u}_0 := \boldsymbol{\omega}_0$, back to (6.17) we get a telescoping sum,

$$\begin{aligned} 2S_T F(\boldsymbol{\omega})^\top (\bar{\boldsymbol{\omega}}_T - \boldsymbol{\omega}) &\leq 2\|\boldsymbol{\omega}_0 - \boldsymbol{\omega}\|^2 + \sum_{t=0}^{T-1} \eta_t^2 (\|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + \|\Delta_t\|_2^2) \\ &\quad + 2\sum_{t=0}^{T-1} \eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{u}_t) \\ &\leq 2R + \sum_{t=0}^{T-1} \eta_t^2 (\|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + \|\Delta_t\|_2^2) + 2\sum_{t=0}^{T-1} \eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{u}_t) \end{aligned}$$

Then the right hand side does not depends on $\boldsymbol{\omega}$, we can maximize over $\boldsymbol{\omega}$ to get,

$$2S_T \operatorname{Err}_R(\bar{\boldsymbol{\omega}}_T) \leq 2R + \sum_{t=0}^{T-1} \eta_t^2 (\|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + \|\Delta_t\|_2^2) + 2\sum_{t=0}^{T-1} \eta_t \Delta_t^\top (\boldsymbol{\omega}_t - \boldsymbol{u}_t) \quad (6.20)$$

Noticing that $\mathbb{E}[\Delta_t|\boldsymbol{\omega}_t, \boldsymbol{u}_t] = 0$ (the estimates of F are unbiased), by Assumption 3 $\mathbb{E}[(\|F(\boldsymbol{\omega}_t, \xi_t)\|_2^2] \leq M^2$ and by Assumption 2 $\mathbb{E}[\|\Delta_t\|_2^2] \leq \sigma^2$ we get,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R}{S_T} + \frac{M^2 + \sigma^2}{2S_T} \sum_{t=0}^{T-1} \eta_t^2 \tag{6.21}$$

particularly for $\eta_t = \eta$ and $\eta_t = \frac{\eta}{\sqrt{t+1}}$ we respectively get,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{2R}{\eta T} + \frac{\eta}{2}(M^2 + \sigma^2) \tag{6.22}$$

and

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{4R}{\eta\sqrt{T+1} - 1} + 2\eta \ln(T+1)\frac{M^2 + \sigma^2}{\sqrt{T+1} - 1} \tag{6.23}$$

□

6.2 Proof of Thm. 3

Theorem 3. *Under Assumption 2 and 5, if $\mathbb{E}_\xi[F]$ is L-Lipschitz, then Alg. 2 with a constant step-size $\eta \leq \frac{1}{\sqrt{3}L}$ has the following convergence rate for any $T \geq 1$,*

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \frac{7}{2}\eta\sigma^2 \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1} \boldsymbol{\omega}'_t\,. \tag{6.24}$$

Particularly, $\eta = \frac{\sqrt{2}R}{\sigma\sqrt{7T}}$ gives $\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{\sqrt{14}R\sigma}{\sqrt{T}}$.

Proof of Thm. 3. Let any $\boldsymbol{\omega} \in \Omega$ such that $\|\boldsymbol{\omega}_0 - \boldsymbol{\omega}\|_2 \leq R$. Then, the update rules become $\boldsymbol{\omega}_{t+1} = P_\Omega(\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}'_t, \zeta_t))$ and $\boldsymbol{\omega}'_t = P_\Omega(\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_t, \xi_t))$. We start by applying Lemma 3 for $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}', \boldsymbol{\omega}^+) = (\boldsymbol{\omega}_t, -\eta F(\boldsymbol{\omega}'_t, \zeta_t), \boldsymbol{\omega}, \boldsymbol{\omega}_{t+1})$ and $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}', \boldsymbol{\omega}^+) = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}_t, \xi_t), \boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}'_t)$,

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2 \\
\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2
\end{aligned}$$

Then, summing them we get

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 &- 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) \\
&- 2\eta_t F(\boldsymbol{\omega}_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}'_t\|_2^2
\end{aligned} \tag{6.25}$$

leading to

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 &- 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\
&+ 2\eta_t(F(\boldsymbol{\omega}'_t, \zeta_t) - F(\boldsymbol{\omega}_t, \xi_t))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}'_t\|_2^2
\end{aligned}$$

Then with $2\boldsymbol{a}^\top\boldsymbol{b} \leq \|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2$ we get

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\ - \|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 + \eta_t^2\|F(\boldsymbol{\omega}'_t, \zeta_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2\end{aligned}$$

Using the inequality $\|\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c}\|_2^2 \leq 3(\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2 + \|\boldsymbol{c}\|_2^2)$ we get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 \\ + 3\eta_t^2(\|F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\|_2^2 \\ + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}_t)\|_2^2)\end{aligned}$$

Then we can use the L-Lipschitzness of F to get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \zeta_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 \\ + 3\eta_t^2(\|F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 \\ + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\|_2^2 + L^2\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2)\end{aligned}$$

As we restricted the step-size to $\eta_t \leq \frac{1}{\sqrt{3}L}$ we get,

$$\begin{aligned}2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \\ + 2\eta_t(F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\ + 3\eta_t^2\|F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 + 3\eta_t^2\|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\|_2^2\end{aligned}$$

We get a particular case of (8) so we can use Lemma 8 where $N_t = \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2$, $M_1(\boldsymbol{\omega}_t, \xi_t) = 3\|F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2$, $M_2(\boldsymbol{\omega}'_t, \zeta_t) = 3\|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\|_2^2$, $\Delta_t = F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)$ and $\boldsymbol{z}_t = \boldsymbol{\omega}'_t$. By Assumption 2, $M_1 = M_2 = 3\sigma^2$ and by the fact that

$$\begin{aligned}\mathbb{E}[F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\,|\boldsymbol{\omega}'_t, \Delta_0, \ldots, \Delta_{t-1}] &= \mathbb{E}[\mathbb{E}[F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \zeta_t)\,|\boldsymbol{\omega}'_t]|\Delta_0, \ldots, \Delta_{t-1}] \\ &= 0\end{aligned}$$

the hypothesis of Lemma 8 hold and we get,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{S_T} + \frac{7\sigma^2}{2S_T}\sum_{t=0}^{T-1}\eta_t^2 \tag{6.26}$$

□

6.3 Proof of Thm. 4

Theorem 4. *Under Assumption 2, if $\mathbb{E}_\xi[F]$ is L-Lipschitz, then AvgPastExtraSGD (Alg. 3) with a constant step-size $\eta \leq \frac{1}{2\sqrt{3}L}$ has the following convergence rate for any $T \geq 1$,*

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \frac{13}{2}\eta\sigma^2 \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\omega}'_t\,. \tag{6.27}$$

Particularly, $\eta = \frac{\sqrt{2}R}{\sigma\sqrt{13T}}$ gives $\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{\sqrt{26}R\sigma}{\sqrt{T}}$.

First let us recall the update rule

$$\begin{cases} \boldsymbol{\omega}_{t+1} = P_{\Omega}[\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}'_t, \xi_t)] \\ \boldsymbol{\omega}'_{t+1} = P_{\Omega}[\boldsymbol{\omega}_{t+1} - \eta_{t+1} F(\boldsymbol{\omega}'_t, \xi_t)] \,. \end{cases} \tag{6.28}$$

Lemma 9. *We have for any $\boldsymbol{\omega} \in \Omega$,*

$$\begin{aligned} 2\eta F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \\ + 3\eta_t^2 L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 + 3\eta_t^2 \Big[\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 \\ + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2 \Big] \,. \end{aligned} \tag{6.29}$$

Proof. Applying Lemma 3 for $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}^+, \boldsymbol{\omega}') = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}'_t, \xi_t), \boldsymbol{\omega}_{t+1}, \boldsymbol{\omega})$ and $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}^+, \boldsymbol{\omega}') = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}), \boldsymbol{\omega}'_t, \boldsymbol{\omega}_{t+1})$, we get,

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2 \tag{6.30}$$

and

$$\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1})^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \,. \tag{6.31}$$

Summing (6.30) and (6.31) we get,

$$\begin{aligned} \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) \\ &\quad - 2\eta_t F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1})^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 \\ &= \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 \\ &\quad - 2\eta_t (F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_t, \xi_t))^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) \,. \end{aligned} \tag{6.32}$$

Then, we can use the inequality of arithmetic and geometric means $2a^\top b \leq \|a\|_2^2 + \|b\|_2^2$ to get,

$$\begin{aligned} \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\ &\quad + \eta_t^2 \|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2 \\ &\quad + \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 \qquad (6.33) \\ &= \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\ &\quad + \eta_t^2 \|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \,. \qquad (6.34) \end{aligned}$$

Using the inequality $\|\boldsymbol{a} + \boldsymbol{b} + \boldsymbol{c}\|_2^2 \leq 3(\|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2 + \|\boldsymbol{c}\|_2^2)$ we get,

$$\begin{aligned} \|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2 &\leq 3(\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 \\ &\quad + \|F(\boldsymbol{\omega}'_{t-1}) - F(\boldsymbol{\omega}'_t)\|_2^2 \\ &\quad + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2) \qquad (6.35) \\ &\leq 3(\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 + L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \\ &\quad + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2) \,, \qquad (6.36) \end{aligned}$$

where we used the L-Lipschitzness of F for the last inequality.
Combining (6.34) with (6.36) we get,

$$\begin{aligned}\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top (\boldsymbol{\omega}'_t - \boldsymbol{\omega}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \\ + 3\eta_t^2 L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 + 3\eta_t^2 \Big[\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 \\ + \|F(\boldsymbol{\omega}'_t) - F(\boldsymbol{\omega}'_t, \xi_t)\|_2^2\Big]\,. \end{aligned} \tag{6.37}$$

□

Lemma 10. *For all $t \geq 0$, if we set $\boldsymbol{\omega}'_{-2} = \boldsymbol{\omega}'_{-1} = \boldsymbol{\omega}'_0$ we have*

$$\begin{aligned}\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \leq 4\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 + 12\eta_{t-1}^2 \Big(\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 \\ + L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2 \\ + \|F(\boldsymbol{\omega}'_{t-2}) - F(\boldsymbol{\omega}'_{t-2}, \xi_{t-2})\|_2^2)\Big) \\ - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2\,. \end{aligned} \tag{6.38}$$

Proof. We start with $\|a + b\|_2^2 \leq 2\|a\|^2 + 2\|b\|^2$.

$$\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \leq 2\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 + 2\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_{t-1}\|_2^2\,. \tag{6.39}$$

Moreover, since the projection is contractive we have that

$$\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_{t-1}\|_2^2 \leq \|\boldsymbol{\omega}_{t-1} - \eta_{t-1} F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - \boldsymbol{\omega}_{t-1} - \eta_{t-1} F(\boldsymbol{\omega}'_{t-2}, \xi_{t-2})\|_2^2 \tag{6.40}$$

$$= \eta_{t-1}^2 \|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-2}, \xi_{t-2})\|_2^2 \tag{6.41}$$

$$\begin{aligned}\leq 3\eta_{t-1}^2 \Big(\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 + L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2 \\ + \|F(\boldsymbol{\omega}'_{t-2}) - F(\boldsymbol{\omega}'_{t-2}, \xi_{t-2})\|_2^2)\Big)\,. \end{aligned} \tag{6.42}$$

where in the last line we used the same inequality as in (6.36). Combining (6.38) and (6.42) we get,

$$\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 = 2\|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \tag{6.43}$$

$$\leq 4\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 + 4\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_{t-1}\|_2^2 - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2 \tag{6.44}$$

$$\begin{aligned}\leq 4\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 + 12\eta_{t-1}^2 \Big(\|F(\boldsymbol{\omega}'_{t-1}, \xi_{t-1}) - F(\boldsymbol{\omega}'_{t-1})\|_2^2 \\ + L^2 \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_{t-2}\|_2^2 + \|F(\boldsymbol{\omega}'_{t-2}) - F(\boldsymbol{\omega}'_{t-2}, \xi_{t-2})\|_2^2)\Big) \\ - \|\boldsymbol{\omega}'_{t-1} - \boldsymbol{\omega}'_t\|_2^2\,. \end{aligned} \tag{6.45}$$

□

Proof of Theorem 4. Combining Lemma 10 and Lemma 9 we get,

$$\begin{aligned}2\eta_t F(\boldsymbol{\omega}'_t,\xi_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}) \le{}& \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2-\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}\|_2^2\\ &+36\eta_t^2\eta_{t-1}^2L^2\Big(\|F(\boldsymbol{\omega}'_{t-1},\xi_{t-1})-F(\boldsymbol{\omega}'_{t-1})\|_2^2\\ &\quad+L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2\\ &\quad+\|F(\boldsymbol{\omega}'_{t-2})-F(\boldsymbol{\omega}'_{t-2},\xi_{t-2})\|_2^2\Big)\\ &-3\eta_t^2L^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2+(12\eta_t^2L^2-1)\|\boldsymbol{\omega}'_t-\boldsymbol{\omega}_t\|_2^2\\ &+3\eta_t^2\Big[\|F(\boldsymbol{\omega}'_{t-1},\xi_{t-1})-F(\boldsymbol{\omega}'_{t-1})\|_2^2\\ &\quad+\|F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\|_2^2\Big]. \end{aligned}\tag{6.46}$$

Then for $\eta_t \le \frac{1}{2\sqrt{3}L}$ we have $36\eta_t^2\eta_{t-1}^2L^4 \le 3\eta_{t-1}^2L^2$,

$$\begin{aligned}2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega}) \le{}& \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2-\|\boldsymbol{\omega}_{t+1}-\boldsymbol{\omega}\|_2^2\\ &+3L^2(\eta_{t-1}^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2-\eta_t^2\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_t\|_2^2)\\ &+2\eta_t(F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t))^\top(\boldsymbol{\omega}'_t-\boldsymbol{\omega})\\ &+3\eta_t^2\Big[|F(\boldsymbol{\omega}'_{t-2},\xi_{t-2})-F(\boldsymbol{\omega}'_{t-2})\|_2^2\\ &\quad+2\|F(\boldsymbol{\omega}'_{t-1},\xi_{t-1})-F(\boldsymbol{\omega}'_{t-1})\|_2^2\\ &\quad+\|F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\|_2^2\Big]. \end{aligned}\tag{6.47}$$

We can then use Lemma 8 where

$$\begin{aligned}N_t &= \|\boldsymbol{\omega}_t-\boldsymbol{\omega}\|_2^2+3L^3\eta_{t-1}\|\boldsymbol{\omega}'_{t-1}-\boldsymbol{\omega}'_{t-2}\|_2^2,\\ M_1(\boldsymbol{\omega}_t,\xi_t) &= 0\\ M_2(\boldsymbol{\omega}'_t,\xi_t) &= 3\|F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\|_2^2+6\|F(\boldsymbol{\omega}'_{t-1})-F(\boldsymbol{\omega}'_{t-1},\xi_{t-1})\|_2^2\\ &\quad+3\|F(\boldsymbol{\omega}'_{t-2})-F(\boldsymbol{\omega}'_{t-2},\xi_{t-2})\|_2^2\\ \Delta_t &= F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\\ \boldsymbol{z}_t &= \boldsymbol{\omega}'_t. \end{aligned}$$

By Assumption 2, $M_2 = 12\sigma^2$ and by the fact that

$$\begin{aligned}&\mathbb{E}[F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\,|\boldsymbol{\omega}'_t,\Delta_0,\dots,\Delta_{t-1}] &(6.48)\\ &\quad=\mathbb{E}[\mathbb{E}[F(\boldsymbol{\omega}'_t)-F(\boldsymbol{\omega}'_t,\xi_t)\,|\boldsymbol{\omega}'_t]|\Delta_0,\dots,\Delta_{t-1}]=0 &(6.49)\end{aligned}$$

the hypothesis of Lemma 8 hold and we get,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \quad \le \frac{R^2}{S_T}+\frac{13\sigma^2}{2S_T}\sum_{t=0}^{T-1}\eta_t^2 \tag{6.50}$$

□

6.4 Proof of Theorem 3

Theorem 3 has been introduced in §4. This theorem is about Algorithm 5 which consists in another way to implement extrapolation to SGD. Let us first restate this theorem,

Theorem 3. *Assume that* $\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_0\| \leq R$, $\forall t \geq 0$ *where* $(\boldsymbol{\omega}'_t)_{t\geq 0}$ *are the iterates of Alg. 5. Under Assumption 2 and 5, for any* $T \geq 1$, *Alg. 5 with constant step-size* $\eta \leq \frac{1}{\sqrt{2}L}$ *has the following convergence properties:*

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{\eta T} + \eta\frac{\sigma^2 + 4L^2(4R^2+\sigma^2)}{2} \quad \textit{where} \quad \bar{\boldsymbol{\omega}}_T := \frac{1}{T}\sum_{t=0}^{T-1}\boldsymbol{\omega}'_t\,.$$

Particularly, $\eta_t = \frac{\eta}{\sqrt{T}}$ *gives* $\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{O(1)}{\sqrt{T}}$.

Proof of Thm. 3. Let any $\boldsymbol{\omega} \in \Omega$ such that $\|\boldsymbol{\omega}_0 - \boldsymbol{\omega}\|_2 \leq R$. Then, the update rules become $\boldsymbol{\omega}_{t+1} = P_\Omega(\boldsymbol{\omega}_t - \eta_t F(\boldsymbol{\omega}'_t, \xi_t))$ and $\boldsymbol{\omega}'_t = P_\Omega(\boldsymbol{\omega}_t - \eta F(\boldsymbol{\omega}_t, \xi_t))$. We start the same way as the proof of Thm. 3 by applying Lemma 3 for $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}', \boldsymbol{\omega}^*) = (\boldsymbol{\omega}_t, -\eta F(\boldsymbol{\omega}'_t, \xi_t), \boldsymbol{\omega}, \boldsymbol{\omega}_{t+1})$ and $(\boldsymbol{\omega}, \boldsymbol{u}, \boldsymbol{\omega}', \boldsymbol{\omega}^+) = (\boldsymbol{\omega}_t, -\eta_t F(\boldsymbol{\omega}_t, \xi_t), \boldsymbol{\omega}_{t+1}, \boldsymbol{\omega}'_t)$,

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}_t\|_2^2 \\
\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}\|_2^2 &\leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}_{t+1}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2
\end{aligned}$$

Then, summing them we get

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 &- 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top(\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}) \\
&- 2\eta_t F(\boldsymbol{\omega}_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}'_t\|_2^2 \quad (6.51)
\end{aligned}$$

leading to

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 &- 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\
&+ 2\eta_t(F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}_t, \xi_t))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}_{t+1}) - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}'_t\|_2^2
\end{aligned}$$

Then with $2\boldsymbol{a}^\top\boldsymbol{b} \leq \|\boldsymbol{a}\|_2^2 + \|\boldsymbol{b}\|_2^2$ we get

$$\begin{aligned}
\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \\
+\eta_t^2\|F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}_t, \xi_t)\|_2^2 - \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2
\end{aligned}$$

Using the Lipschitz assumption we get

$$\|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - 2\eta_t F(\boldsymbol{\omega}'_t, \xi_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) + (\eta_t^2 L^2 - 1)\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2$$

Then we add $2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega})$ in both sides to get,

$$2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \\ - 2\eta_t(F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}'_t))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) + (\eta_t^2 L^2 - 1)\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 \quad (6.52)$$

Here, unfortunately we cannot use Lemma 8 because $F(\boldsymbol{\omega}'_t, \xi_t)$ is biased. We will then deal with the quantity $A = (F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}'))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega})$. We have that,

$$\begin{aligned} A &= (F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}_t, \xi_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}'_t) + (F(\boldsymbol{\omega}_t) - F(\boldsymbol{\omega}'_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}'_t) \\ &\quad + (F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t) + (F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}_t) \\ &\leq 2L\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}\|_2 + \|F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)\|\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2 \\ &\quad + (F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}_t) \\ &\quad \text{(Using Cauchy-Schwarz and the } L\text{-Lip of } F) \end{aligned}$$

Then using $2\|a\|\|b\| \leq \delta\|a\|_2^2 + \frac{1}{\delta}\|b\|_2^2$, for $\delta = 4$,

$$\begin{aligned} -2\eta_t(F(\boldsymbol{\omega}'_t, \xi_t) - F(\boldsymbol{\omega}'_t))^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \frac{1}{2}\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|^2 + 8\eta_t^2 L^2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}\|_2^2 \\ + 4\eta_t^2\|F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)\|_2^2 \\ + \frac{1}{4}\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_t\|_2^2 \\ + 2\eta_t(F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}_t) \end{aligned}$$

leading to,

$$\begin{aligned} 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \\ + 2\eta_t(F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}_t) \\ + (\eta_t^2 L^2 - \tfrac{1}{4})\|\boldsymbol{\omega}_t - \boldsymbol{\omega}'_t\|_2^2 \\ + 4\eta_t^2(2L^2\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}\|_2^2 + \|F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)\|_2^2) \end{aligned}$$

If one assumes finally that $\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_0\|_2 \leq R$ (assumption of the theorem) and that $\eta_t \leq \frac{1}{2L}$ we get,

$$\begin{aligned} 2\eta_t F(\boldsymbol{\omega}'_t)^\top(\boldsymbol{\omega}'_t - \boldsymbol{\omega}) \leq \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2 - \|\boldsymbol{\omega}_{t+1} - \boldsymbol{\omega}\|_2^2 \\ + 2\eta_t(F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t))^\top(\boldsymbol{\omega} - \boldsymbol{\omega}_t) \\ + 4\eta_t^2(4L^2R^2 + \|F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)\|_2^2) \end{aligned}$$

where we used that $\|\boldsymbol{\omega}'_t - \boldsymbol{\omega}\|_2 \leq \|\boldsymbol{\omega}'_t - \boldsymbol{\omega}_0\|_2 + \|\boldsymbol{\omega}_0 - \boldsymbol{\omega}\|_2 \leq 2R$. Once again this equation is a particular case of Lemma 8 where $N_t = \|\boldsymbol{\omega}_t - \boldsymbol{\omega}\|_2^2$, $M_1(\boldsymbol{\omega}_t, \xi_t) = 4(4L^2R^2 + \|F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)\|_2^2)$, $M_2(\boldsymbol{\omega}'_t, \zeta_t) = 0$, $\boldsymbol{z}_t = \boldsymbol{\omega}_t$ and $\Delta_t = F(\boldsymbol{\omega}_t, \xi_t) - F(\boldsymbol{\omega}_t)$. By Assumption 2 $\mathbb{E}[M_1(\boldsymbol{\omega}_t, \xi_t)] \leq 16L^2R^2 + 4\sigma^2$ and

$\mathbb{E}[\Delta_t|\boldsymbol{\omega}_t, \Delta_0, \ldots, \Delta_{t-1}] = \mathbb{E}[\mathbb{E}[\Delta_t|\boldsymbol{\omega}_t]|\Delta_0, \ldots, \Delta_{t-1}] = 0$ so we can use Lemma 8 and get,

$$\mathbb{E}[\mathrm{Err}_R(\bar{\boldsymbol{\omega}}_T)] \leq \frac{R^2}{S_T} + \frac{\sigma^2 + 16L^2R^2 + 4\sigma^2}{2S_T} \sum_{t=0}^{T-1} \eta_t^2 \,. \tag{6.53}$$

□

7 Additional experimental results

7.1 Toy non-convex GAN (2D and deterministic)

We now consider a task similar to [Mescheder et al., 2018] where the discriminator is linear $D_{\boldsymbol{\varphi}}(\boldsymbol{\omega}) = \boldsymbol{\varphi}^T\boldsymbol{\omega}$, the generator is a Dirac distribution at $\boldsymbol{\theta}$, $q_{\boldsymbol{\theta}} = \delta_{\boldsymbol{\theta}}$ and the distribution we try to match is also a Dirac at $\boldsymbol{\omega}^*$, $p = \delta_{\boldsymbol{\omega}^*}$. The minimax formulation from Goodfellow et al. [2014] gives:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\varphi}} -\log\left(1 + e^{-\boldsymbol{\varphi}^T\boldsymbol{\omega}^*}\right) - \log\left(1 + e^{\boldsymbol{\varphi}^T\boldsymbol{\theta}}\right) \tag{7.1}$$

Note that as observed by Nagarajan and Kolter [2017], this objective is concave-concave, making it hard to optimize. We compare the methods on this objective where we take $\boldsymbol{\omega}^* = -2$, thus the position of the equilibrium is shifted towards the position $(\boldsymbol{\theta}, \boldsymbol{\varphi}) = (-2, 0)$. The convergence and the gradient vector field are shown in Figure B.1. We observe that depending on the initialization, some methods can fail to converge but *extrapolation* (3.11) seems to perform better than the other methods.

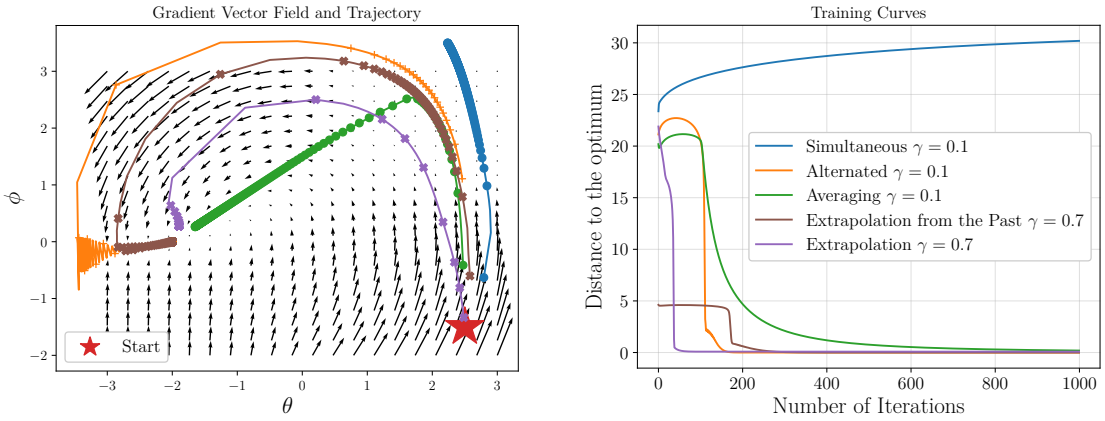


Figure B.1: Comparison of five algorithms (described in Section 3) on the non-convex GAN objective (7.1), using the optimal step-size for each method. **Left**: The gradient vector field and the dynamics of the different methods. **Right**:The distance to the optimum as a function of the number of iterations.

7.2 DCGAN with WGAN-GP objective

In addition to the results presented in section §7.2, we also trained the DCGAN architecture with the WGAN-GP objective. The results are shown in Table B.2. The best results are achieved with *uniform averaging* of AltAdam5. However, its iterations require to update the discriminator 5 times for every generator update.

Generator
Input: $z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
Linear $128 \to 512 \times 4 \times 4$
Batch Normalization
ReLU
transposed conv. (kernel: 4×4, $512 \to 256$, stride: 2, pad: 1)
Batch Normalization
ReLU
transposed conv. (kernel: 4×4, $256 \to 128$, stride: 2, pad: 1)
Batch Normalization
ReLU
transposed conv. (kernel: 4×4, $128 \to 3$, stride: 2, pad: 1)
$Tanh(\cdot)$
Discriminator
Input: $x \in \mathbb{R}^{3\times32\times32}$
conv. (kernel: 4×4, $1 \to 64$; stride: 2; pad:1)
LeakyReLU (negative slope: 0.2)
conv. (kernel: 4×4, $64 \to 128$; stride: 2; pad:1)
Batch Normalization
LeakyReLU (negative slope: 0.2)
conv. (kernel: 4×4, $128 \to 256$; stride: 2; pad:1)
Batch Normalization
LeakyReLU (negative slope: 0.2)
Linear $128 \times 4 \times 4 \times 4 \to 1$

Table B.1: DCGAN architecture used for our CIFAR-10 experiments. When using the gradient penalty (WGAN-GP), we remove the Batch Normalization layers in the discriminator.

With a small drop in best final score, ExtraAdam can train WGAN-GP significantly faster (see Fig. B.2 right) as the discriminator and generator are updated only twice.

7.3 FID scores for ResNet architecture with WGAN-GP objective

In addition to the inception scores, we also computed the FID scores [Heusel et al., 2017] using 50,000 samples for the ResNet architecture with the WGAN-GP objective; the results are presented in Table B.4. We see that the results and conclusions are similar to the one obtained from the inception scores, adding an extrapolation step as well as using Exponential Moving Average (EMA) consistently

Model	WGAN-GP (DCGAN)	
Method	no averaging	uniform avg
SimAdam	*6.00 ± .07*	6.01 ± .08
AltAdam5	*6.25 ± .05*	**6.51 ± .05**
ExtraAdam	6.22 ± .04	6.35 ± .05
PastExtraAdam	6.27 ± 0.06	6.23 ± 0.13

Table B.2: Best inception scores (averaged over 5 runs) achieved on CIFAR10 for every considered Adam variant. We see that the techniques of extrapolation and averaging consistently enable improvements over the baselines (in italic).

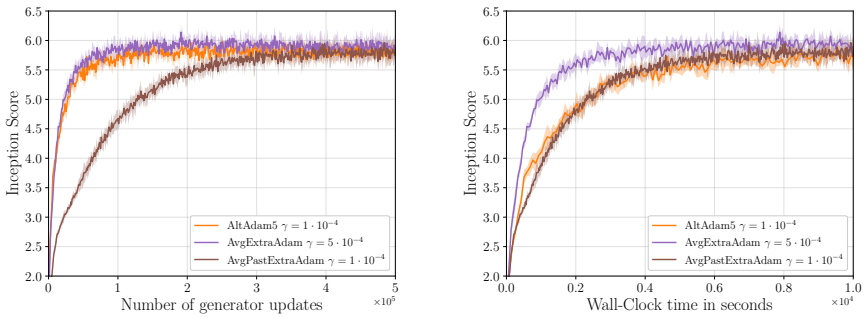


Figure B.2: DCGAN architecture with WGAN-GP trained on CIFAR10: mean and standard deviation of the inception score computed over 5 runs for each method using the best performing learning rate plotted over number of generator updates (**Left**) and wall-clock time (**Right**); all experiments were run on a NVIDIA Quadro GP100 GPU. We see that ExtraAdam converges faster than the Adam baselines.

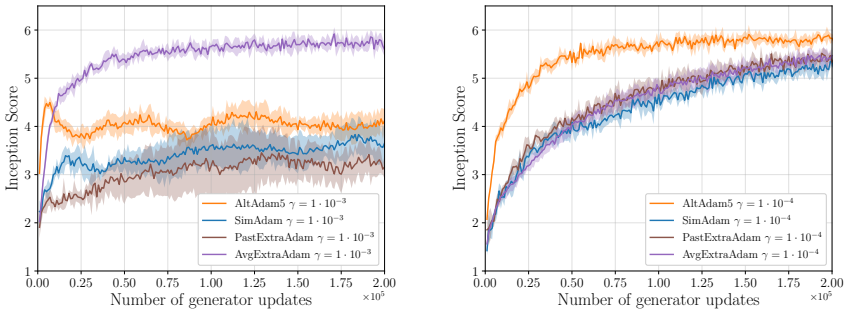
improves the FID scores. However, contrary to the results from the inception score, we observe that uniform averaging does not necessarily improve the performance of the methods. This could be due to the fact that the samples produced using uniform averaging are more blurry and FID is more sensitive to blurriness; see §7.3 for more details about the effects of uniform averaging.

Generator
Input: $z \in \mathbb{R}^{128} \sim \mathcal{N}(0, I)$
Linear $128 \to 128 \times 4 \times 4$
ResBlock $128 \to 128$
ResBlock $128 \to 128$
ResBlock $128 \to 128$
Batch Normalization
ReLU
transposed conv. (kernel: 3×3, $128 \to 3$, stride: 1, pad: 1)
$Tanh(\cdot)$
Discriminator
Input: $x \in \mathbb{R}^{3\times32\times32}$
ResBlock $3 \to 128$
ResBlock $128 \to 128$
ResBlock $128 \to 128$
ResBlock $128 \to 128$
Linear $128 \to 1$

Table B.3: ResNet architecture used for our CIFAR-10 experiments. When using the gradient penalty (WGAN-GP), we remove the Batch Normalization layers in the discriminator.

Model	WGAN-GP (ResNet)		
Method	no averaging	uniform avg	EMA
SimAdam	*23.74 ± 2.79*	26.29 ± 5.56	21.89 ± 2.51
AltAdam5	*21.65 ± 0.66*	19.91 ± 0.43	20.69 ± 0.37
ExtraAdam	19.42 ± 0.15	18.13 ± 0.51	$\mathbf{16.78 \pm 0.21}$
PastExtraAdam	19.95 ± 0.38	22.45 ± 0.93	17.85 ± 0.40
OptimAdam	*18.88 ± 0.55*	21.23 ± 1.19	16.91 ± 0.32

Table B.4: Best FID scores (averaged over 5 runs) achieved on CIFAR10 for every considered Adam variant. OptimAdam is the related *Optimistic Adam* [Daskalakis et al., 2018] algorithm. We see that the techniques of extrapolation and EMA consistently enable improvements over the baselines (in italic).



(a) learning rate of 10^{-3} **(b)** learning rate of 10^{-4}

Figure B.3: Inception score on CIFAR10 for WGAN-GP (DCGAN) over number of generator updates for different learning rates. We can see that AvgExtraAdam is less sensitive to the choice of learning rate.

7.4 Comparison of the methods with the same learning rate

In this section, we compare how the methods presented in §7 perform with the same step-size. We follow the same protocol as in the experimental section §7, we consider the DCGAN architecture with WGAN-GP experiment described in App §7.2. In Figure B.3 we plot the inception score provided by each training method as a function of the number of generator updates. Note that these plots advantage **AltAdam5** a bit because each iteration of this algorithm is a bit more costly (since it perform 5 discriminator updates for each generator update). Nevertheless, the goal of this experiment is not to show that **AltAdam5** is faster but to show that **ExtraAdam** is less sensitive to the choice of learning rate and can be used with higher learning rates with less degradation. In Figure B.4, we compare the sample quality on the DCGAN architecture with the WGAN-GP objective of **AltAdam5** and **AvgExtraAdam** for different step-sizes. We notice that for **AvgExtraAdam**, the sample quality does not significantly change whereas the sample quality of **AltAdam5** seems to be really sensitive to step-size tunning. We think that robustness to step-size tuning is a key property for an optimization algorithm in order to save as much time as possible to tune other hyperparameters of the learning procedure such as regularization.

(a) AvgExtraAdam with $\eta = 10^{-3}$

(b) AvgExtraAdam with $\eta = 10^{-4}$

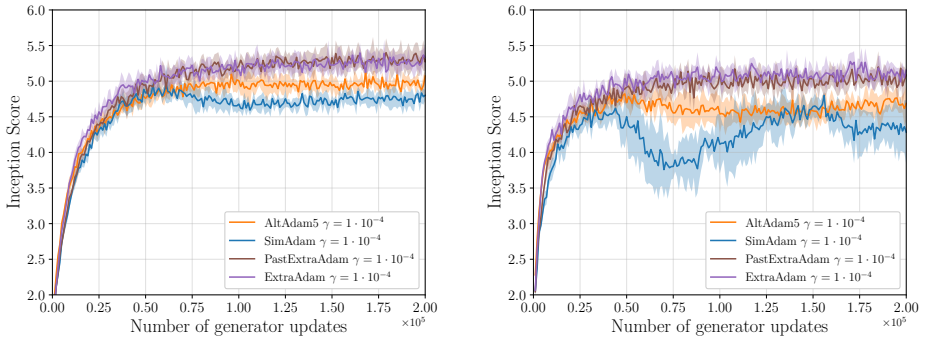
(c) AltAdam with $\eta = 10^{-3}$

(d) AltAdam with $\eta = 10^{-4}$

Figure B.4: Comparison of the samples quality on the WGAN-GP (DCGAN) experiment for different methods and learning rate η.

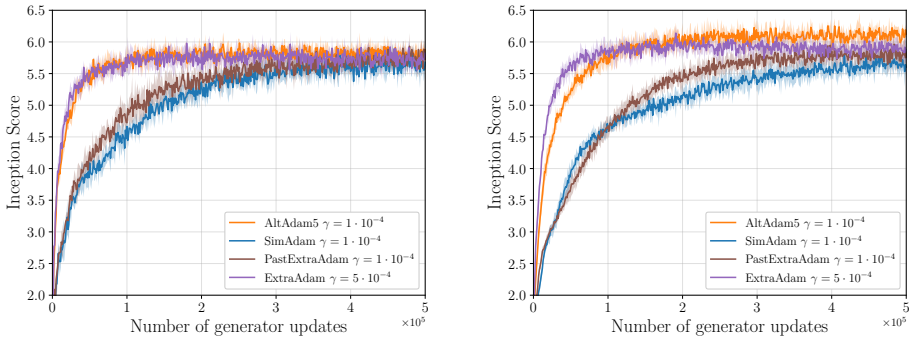
7.5 Comparison of the methods with and without uniform averaging

In this section, we compare how uniform averaging affect the performance of the methods presented in §7. We follow the same protocol as in the experimental section §7, we consider the DCGAN architecture with the WGAN and weight clipping objective as well as the WGAN-GP objective. In Figure B.5 and B.6, we plot the inception score provided by each training method as a function of the number of generator updates with and without uniform averaging. We notice that uniform averaging seems to improve the inception score, nevertheless it looks like the sample are a bit more blurry (see Figure B.7). This is confirmed by our result (Figure B.8) on the Fréchet Inception Distance (FID) which is more sensitive to blurriness. A similar observation about FID was made in §7.3.



(a) with averaging **(b)** without averaging

Figure B.5: Inception Score on CIFAR10 for WGAN over number of generator updates with and without averaging. We can see that averaging improve the inception score.



(a) with averaging **(b)** without averaging

Figure B.6: Inception score on CIFAR10 for WGAN-GP (DCGAN) over number of generator updates

(a) PastExtraAdam without averaging

(b) PastExtraAdam with averaging

(c) AltAdam5 without averaging

(d) AltAdam5 with averaging

Figure B.7: Comparison of the samples of a WGAN trained with the different methods with and without averaging. Although averaging improves the inception score, the samples seem more blurry

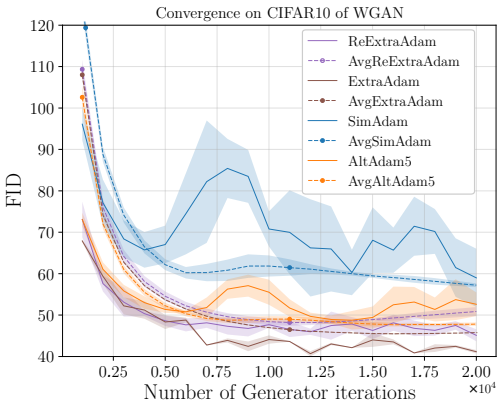


Figure B.8: The Fréchet Inception Distance (FID) from Heusel et al. [2017] computed using 50,000 samples, on the WGAN experiments. ReExtraAdam refers to Alg. 5 introduced in §4. We can see that averaging performs worse than when comparing with the Inception Score. We observed that the samples generated by using averaging are a little more blurry and that the FID is more sensitive to blurriness, thus providing an explanation for this observation.

8 Hyperparameters

(DCGAN) WGAN Hyperparameters	
Batch size	$= 64$
Number of generator update	$= 500,000$
Adam β_1	$= 0.5$
Adam β_2	$= 0.9$
Weight clipping for the discriminator	$= 0.01$
Learning rate for generator	$= 5 \times 10^{-5}$ (for ExtraAdam)
	$= 2 \times 10^{-5}$ (for the other algorithms)
Learning rate for discriminator	$= 5 \times 10^{-4}$ (for ExtraAdam)
	$= 2 \times 10^{-4}$ (for the other algorithms)
β for EMA	$= 0.999$

(DCGAN) WGAN-GP Hyperparameters	
Batch size	$= 64$
Number of generator update	$= 500,000$
Adam β_1	$= 0.5$
Adam β_2	$= 0.9$
Gradient penalty	$= 10$
Learning rate for generator	$= 5 \times 10^{-4}$ (for ExtraAdam)
	$= 1 \times 10^{-4}$ (for the other algorithms)
Learning rate for discriminator	$= 5 \times 10^{-4}$ (for ExtraAdam)
	$= 1 \times 10^{-4}$ (for the other algorithms)
β for EMA	$= 0.999$

(ResNet) WGAN-GP Hyperparameters	
Batch size	$= 64$
Number of generator update	$= 500,000$
Adam β_1	$= 0.5$
Adam β_2	$= 0.9$
Gradient penalty	$= 10$
Learning rate for generator	$= 5 \times 10^{-5}$ (for ExtraAdam)
	$= 2 \times 10^{-5}$ (for the other algorithms)
Learning rate for discriminator	$= 5 \times 10^{-4}$ (for ExtraAdam)
	$= 2 \times 10^{-4}$ (for the other algorithms)
β for EMA	$= 0.9999$

C Negative Momentum for Improved Game Dynamics

1 Additional Figures

1.1 Maximum magnitude of the eigenvalues gradient descent with negative momentum on a bilinear objective

In Figure C.1 we numerically (using the formula provided in Proposition 1 and 2) computed the maximum magnitude of the eigenvalues gradient descent with negative momentum on a bilinear objective as a function of the step size η and the momentum β. We can notice that on one hand, for simultaneous gradient method, no value of η and β provide a maximum magnitude smaller than 1, causing a divergence of the algorithm. On the other hand, for alternating gradient method there exists a sweet spot where the maximum magnitude of the eigenvalues of the operator is smaller than 1 insuring that this method does converge linearly (since the Jacobian of a bilinear minmax proble is constant).

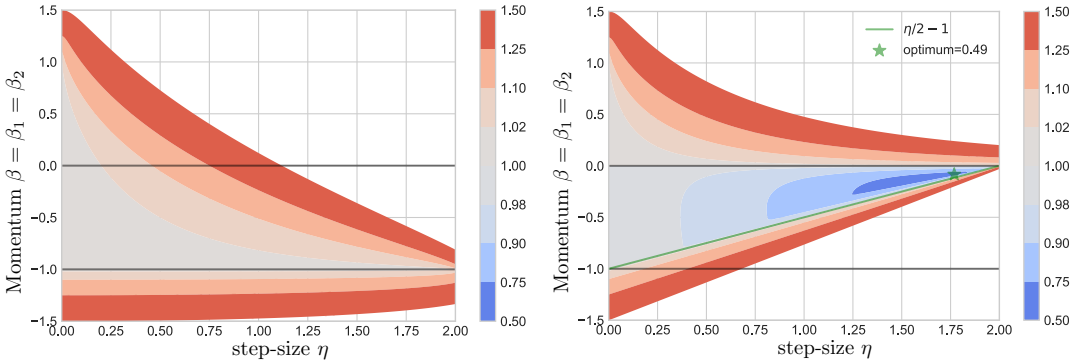


Figure C.1: Contour plot of the maximum magnitude of the eigenvalues of the polynomial $(x-1)^2(x-\beta)^2+\eta^2x^2$ (**left**, simultaneous) and $(x-1)^2(x-\beta)^2+\eta^2x^3$ (**right**, alternated) for different values of the step-size η and the momentum β. Note that compared to (5.5) and (5.7) we used $\beta_1 = \beta_2 = \beta$ and we defined $\eta := \sqrt{\eta_1\eta_2\lambda}$ without loss of generality. On the left, magnitudes are always larger than 1, and equal to 1 for $\beta = -1$. On the right, magnitudes are smaller than 1 for $\frac{\eta}{2} - 1 \leq \beta \leq 0$ and greater than 1 elsewhere.

1.2 Mixture of Gaussian

[Fig. C.2] In this set of experiments we evaluate the effect of using negative momentum for a GAN with *saturating loss* and alternating steps. The data in this experiment comes from eight Gaussian distributions which are distributed uniformly around the unit circle. The goal is to force the generator to generate 2-D samples that are coming from *all* of the 8 distributions. Although this looks like a simple task, many GANs fail to generate diverse samples in this setup. This experiment shows whether the algorithm prevents mode collapse or not.

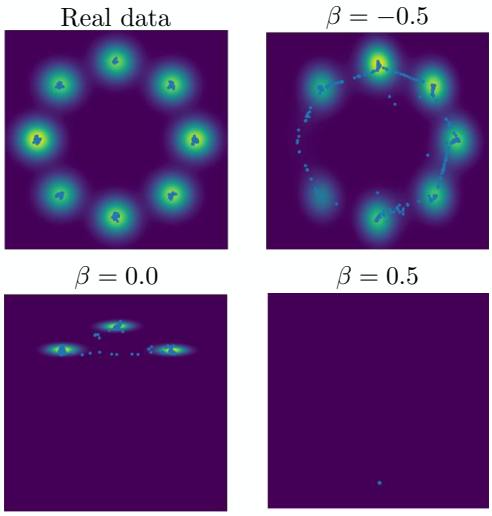


Figure C.2: The effect of negative momentum for a mixture of 8 Gaussian distributions in a GAN setup. Real data and the results of using SGD with zero momentum on the Generator and using negative / zero / positive momentum (β) on the Discriminator are depicted.

We use a fully connected network with 4 hidden *ReLU* layers where each layer has 256 hidden units. The latent code of the generator is an 8-dimensional multivariate Gaussian. The model is trained for 100,000 iterations with a learning rate of 0.01 for stochastic gradient descent along with values of zero, -0.5 and 0.5 momentum. We observe that negative momentum considerably improves the results compared to positive or zero momentum.

2 Discussion on Momentum and Conditioning

In this section, we analyze the effect of the conditioning of the problem on the optimal value of momentum. Consider the following formulation as an extension of the bilinear min-max game discussed in §5, Eq. 2.4 ($p = d = n$),

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^n}\max_{\boldsymbol{\varphi}\in\mathbb{R}^n}\ \alpha\|\boldsymbol{D}^{1/2}\boldsymbol{\theta}\|_2^2+(1-\alpha)\boldsymbol{\theta}^\top\boldsymbol{A}\boldsymbol{\varphi}-\alpha\|\boldsymbol{D}^{1/2}\boldsymbol{\varphi}\|_2^2\,,\quad \alpha\in[0,1],\ \boldsymbol{A}\in\mathbb{R}^{n\times n}\,, \tag{2.1}$$

where $\boldsymbol{D}$ is a square diagonal positive-definite matrix,

$$\boldsymbol{D}=\begin{bmatrix} d_{1,1} & 0 & 0 & . & . & . & 0\\ 0 & d_{2,2} & 0 & . & . & . & 0\\ 0 & 0 & d_{3,3} & . & . & . & 0\\ 0 & 0 & 0 & . & . & . & 0\\ 0 & 0 & 0 & . & . & . & d_{n,n}\end{bmatrix} \text{ and } \forall j\in\{1,n-1\},\ d_{j+1,j+1}\geq d_{j,j}>0, \tag{2.2}$$

and its condition number is $\kappa(\boldsymbol{D})=d_{n,n}/d_{1,1}$. Thus, we can re-write the vector field and the Jacobian as a function of α and $\boldsymbol{D}$,

$$\boldsymbol{v}(\boldsymbol{\varphi},\boldsymbol{\theta},\boldsymbol{\alpha},\boldsymbol{D})=\begin{bmatrix}-(1-\alpha)\boldsymbol{\theta}+2\alpha\boldsymbol{D}\boldsymbol{\varphi}\\ 2\alpha\boldsymbol{D}\boldsymbol{\theta}+(1-\alpha)\boldsymbol{\varphi}\end{bmatrix},\ \nabla\boldsymbol{v}(\boldsymbol{\varphi},\boldsymbol{\theta},\boldsymbol{\alpha},\boldsymbol{D})=\begin{bmatrix}2\alpha\boldsymbol{D} & (\alpha-1)\boldsymbol{I}_n\\ (1-\alpha)\boldsymbol{I}_n & 2\alpha\boldsymbol{D}\end{bmatrix}. \tag{2.3}$$

The corresponding eigenvalues λ of the Jacobian are,

$$\lambda=2\alpha\ d_{j,j}\pm(1-\alpha)i. \tag{2.4}$$

For simplicity, in the following we will note $\nabla F_{\eta,\beta}$ for $\nabla F_{\eta,\beta}(\boldsymbol{\varphi},\boldsymbol{\theta},\boldsymbol{\alpha},\boldsymbol{D})$.

Using Thm. (3), the eigenvalues of $\nabla F_{\eta,\beta}$ are,

$$\mu_+(\beta,\eta,\lambda)=(1-\eta\lambda+\beta)\frac{1+\Delta^{\frac{1}{2}}}{2}\quad\text{and}\quad\mu_-(\beta,\eta,\lambda)=(1-\eta\lambda+\beta)\frac{1-\Delta^{\frac{1}{2}}}{2}\,. \tag{2.5}$$

where $\Delta:=1-\frac{4\beta}{(1-\eta\lambda+\beta)^2}$ and $\Delta^{\frac{1}{2}}$ is the *complex* square root of Δ with positive real part.

Hence the spectral radius of $\nabla F_{\eta,\beta}$ can be explicitly formulated as a function of β and η,

$$\rho(\nabla F_{\eta,\beta})=\max_{\lambda\in\mathrm{Sp}(\nabla F_{\eta,\beta})}\max\left\{|\mu_+(\beta,\eta,\lambda)|,|\mu_-(\beta,\eta,\lambda)|\right\} \tag{2.6}$$

In Figure C.3, we numerically computed the optimal β that minimizes $\rho_{max}(\nabla F_{\eta,\beta})$ as a function of the step-size η, for $n=2$, $d_{1,1}=1/\kappa$ and $d_{2,2}=1$. To balance the game between the adversarial part and the cooperative part, we normalize the matrix $\boldsymbol{D}$ such that the sum of its diagonal elements is n. It can be seen that there is a competition between the type of the game (adversarial and cooperative) versus the conditioning of the matrix $\boldsymbol{D}$. In a more cooperative regime, increasing κ results in more positive values of momentum which is consistent with the intuition that cooperative games are almost minimization problems where the optimum value for the momentum is known [Polyak, 1964]

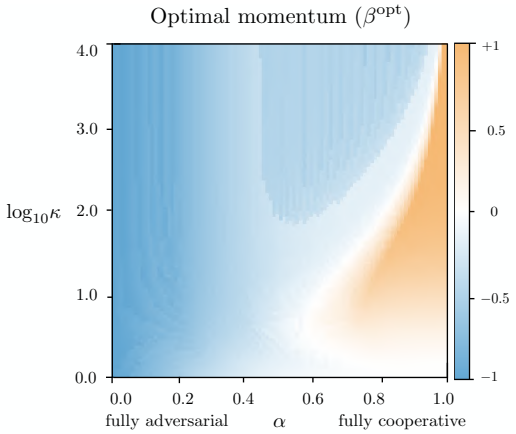


Figure C.3: Plot of the optimal value of momentum by for different α's and condition numbers ($log_{10}\kappa$). Blue/white/orange regions correspond to negative/zero/positive values of the optimal momentum, respectively.

to be $\beta = \left(\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}\right)^2$. Interestingly, even if the condition number of $\boldsymbol{D}$ is large, when the game is adversarial enough, the optimum value for the momentum is negative. This experimental setting seems to suggest the existence of a multidimensional condition number taking into account the difficulties introduced by the ill conditioning of $\boldsymbol{D}$ as well as the adversarial component of the game.

3 Lemmas and Definitions

Recall that the spectral radius $\rho(A)$ of a matrix A is the maximum magnitude of its eigenvalues.

$$\rho(A) := \max\{|\lambda| \ : \ \lambda \in \mathrm{Sp}(A)\}\,. \tag{3.1}$$

For a symmetric matrix, this is equal to the spectral norm, which is the operator norm induced by the vector 2-norm. However, we are dealing with general matrices, so these two values may be different. The spectral radius is always smaller than the spectral norm, but it's not a norm itself, as illustrated by the example below:

$$\text{If } \boldsymbol{A} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \text{ then } \left(\mathrm{Sp}(\boldsymbol{A}) = \{0\} \implies \rho(\boldsymbol{A}) = 0\right)$$
$$\text{but } \left(\boldsymbol{A}^\top \boldsymbol{A} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \implies \|\boldsymbol{A}\|_2 = 1\right)$$

where we used the fact that the spectral norm is also the square root of the largest singular value.

In this section we will introduce three lemmas that we will use in the proofs of §4.

The first lemma is about the determinant of a block matrix.

Lemma 11. *Let A, B, C, D four matrices such that C and D commute. Then*

$$\begin{vmatrix} A & B \\ C & D \end{vmatrix} = \left| AD - BC \right| \tag{3.2}$$

where $\left|A\right|$ is the determinant of A.

Proof. See [Zhang, 2006, Section 0.3]. □

The second lemma is about the iterates of the simultaneous and the alternating methods introduced in §5 for the bilinear game. It shows that we can pick a subspace where the iterates will remain.

Lemma 12. *Let $(\boldsymbol{\theta}_t, \boldsymbol{\varphi}_t)$ the updates computed by the simultaneous (resp. alternating) gradient method with momentum (5.4) (resp. (5.6)). There exists are couple $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ solution of (5.1) only depending on $(\boldsymbol{\theta}_0, \boldsymbol{\varphi}_0)$ such that,*

$$\boldsymbol{\theta}_t - \boldsymbol{\theta}^* \in \mathit{span}(\boldsymbol{A}\boldsymbol{A}^\top) \quad \textit{and} \quad \boldsymbol{\varphi}_t - \boldsymbol{\varphi}^* \in \mathit{span}(\boldsymbol{A}^\top \boldsymbol{A})\,, \quad \forall t \geq 0\,. \tag{3.3}$$

Proof of Lemma 12. Let us start with the simultaneous updates (5.4).

Let $\boldsymbol{U}^\top \boldsymbol{D} \boldsymbol{V} = \boldsymbol{A}$ the SVD of $\boldsymbol{A}$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal matrices and

$$D = \begin{bmatrix} \operatorname{diag}(\sigma_1, \dots, \sigma_r) & \mathbf{0}_{r,p-r} \\ \mathbf{0}_{d-r,r} & \mathbf{0}_{d-r,p-r} \end{bmatrix} \tag{3.4}$$

where r is the rank of A and $\sigma_1 \geq \cdots \geq \sigma_r > 0$ are the (positive) singular values of A. The update rules (5.4) implies that,

$$\begin{cases} \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \\ \boldsymbol{\varphi}_{t+1} = \boldsymbol{\varphi}_t + \eta_2 \boldsymbol{A}^\top \boldsymbol{\theta}_t + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1}) \end{cases} \tag{3.5}$$

$$\Rightarrow \begin{cases} \boldsymbol{U}\boldsymbol{\theta}_{t+1} = \boldsymbol{U}\boldsymbol{\theta}_t - \eta_1 \boldsymbol{D}\boldsymbol{V}\boldsymbol{\varphi}_t + \beta_1 \boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1}) \\ \boldsymbol{V}\boldsymbol{\varphi}_{t+1} = \boldsymbol{V}\boldsymbol{\varphi}_t + \eta_2 \boldsymbol{D}^\top \boldsymbol{U}\boldsymbol{\theta}_t + \beta_2 \boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1}) \end{cases} \tag{3.6}$$

Consequently, for any $\boldsymbol{\theta}_0 \in \mathbb{R}^d$ and $\boldsymbol{\varphi}_0 \in \mathbb{R}^p$ we have that,

$$\boldsymbol{A}^\top \left(\boldsymbol{U}^\top \begin{bmatrix} 0 \\ \vdots \\ 0 \\ [\boldsymbol{U}\boldsymbol{\theta}_0]_{r+1} \\ \vdots \\ [\boldsymbol{U}\boldsymbol{\theta}_0]_d \end{bmatrix} \right) = \mathbf{0} \quad \text{and} \quad \boldsymbol{A} \left(\boldsymbol{V}^\top \begin{bmatrix} 0 \\ \vdots \\ 0 \\ [\boldsymbol{V}\boldsymbol{\varphi}_0]_{r+1} \\ \vdots \\ [\boldsymbol{V}\boldsymbol{\varphi}_0]_d \end{bmatrix} \right) = \mathbf{0} \tag{3.7}$$

Since the solutions $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ of (5.1) verify the following first order conditions:

$$\boldsymbol{A}^\top \boldsymbol{\theta}^* = \mathbf{0} \quad \text{and} \quad \boldsymbol{A}\boldsymbol{\varphi}^* = \mathbf{0} \tag{3.8}$$

One can set $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ as in (3.7) to be a couple of solution of (5.1) such that $\boldsymbol{U}(\boldsymbol{\theta}_0 - \boldsymbol{\theta}^*) \in span(D)$ and $\boldsymbol{V}(\boldsymbol{\varphi}_0 - \boldsymbol{\varphi}^*) \in span(D)$. By an immediate recurrence, using (3.5) we have that for any initialization $(\boldsymbol{\theta}_0, \boldsymbol{\varphi}_0)$ there exists a couple $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ such that that for any $t \geq 0$,

$$\boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*) \in span(\boldsymbol{D}) \quad \text{and} \quad \boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*) \in span(\boldsymbol{D}^\top) \tag{3.9}$$

Consequently,

$$\boldsymbol{\theta}_t - \boldsymbol{\theta}^* \in span(\boldsymbol{A}) = span(\boldsymbol{A}\boldsymbol{A}^\top) \text{ and } \boldsymbol{\varphi}_t - \boldsymbol{\varphi}^* \in span(\boldsymbol{A}^\top) = span(\boldsymbol{A}^\top \boldsymbol{A}) \,, \; t \geq 0 \tag{3.10}$$

The proof for the alternated updates (5.6) are the same since we only use the fact that the iterates stay on the span of interest. □

Lemma 13. *Let $\boldsymbol{M} \in \mathbb{R}^{m \times m}$ and $(\boldsymbol{u}_t)$ a sequence such that, $\boldsymbol{u}_{t+1} = \boldsymbol{M}\boldsymbol{u}_t$, then we have three cases of interest for the spectral radius $\rho(M)$:*

- *If $\rho(M) < 1$, and M is diagonalizable, then $\|\boldsymbol{u}_t\|_2 \in O((\rho(\boldsymbol{M}))^t \|\boldsymbol{u}_0\|_2)$.*
- *If $\rho(M) > 1$, then there exist $\boldsymbol{u}_0$ such that $\|\boldsymbol{u}_t\|_2 \in \Omega(\rho(\boldsymbol{M}))^t \|\boldsymbol{u}_0\|_2$.*
- *If $|\lambda| = 1 \,, \; \forall \lambda \in Sp(M)$, and M is diagonalizable then $\|\boldsymbol{u}_t\|_2 \in \Theta(\|\boldsymbol{u}_0\|_2)$.*

Proof. For that section we note $\|\cdot\|_2$ the ℓ_2 norm of $\mathbb{C}^m$:

- If $\rho(M) < 1$:

  We have for $t \geq 0$ and any $\boldsymbol{u}_0 \in \mathbb{R}^m$,

  $$\|\boldsymbol{u}_t\|_2 = \|\boldsymbol{M}^t \boldsymbol{u}_0\|_2 \leq \|\boldsymbol{M}^t\| \|\boldsymbol{u}_0\|_2 \tag{3.11}$$

  Then we can diagonalize $\boldsymbol{M} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$ where $\boldsymbol{P}$ is invertible and $\boldsymbol{D}$ is a diagonal matrix. Hence using $\|\cdot\|_2$ as the norm of $\mathbb{C}^m$ (because $\boldsymbol{P}$ can belong to $\mathbb{C}^{m \times m}$) we have that,

  $$\begin{aligned} \|\boldsymbol{u}_t\|_2 \|\boldsymbol{P}\boldsymbol{D}^t\boldsymbol{P}^{-1}\| \|\boldsymbol{u}_0\|_2 &\leq \|\boldsymbol{P}\| \|\boldsymbol{P}^{-1}\| \|\boldsymbol{D}^t\| \|\boldsymbol{u}_0\|_2 && (3.12) \\ &\leq \|\boldsymbol{P}\| \|\boldsymbol{P}^{-1}\| \rho(M)^t \|\boldsymbol{u}_0\|_2 && (3.13) \\ &= O((\rho(\boldsymbol{M}))^t \|\boldsymbol{u}_0\|_2) \,. && (3.14) \end{aligned}$$

- If $\rho(M) > 1$: We have for $t \geq 0$ and any $\boldsymbol{u}_0 \in \mathbb{R}^m$,

  $$\|\boldsymbol{u}_t\|_2 = \|\boldsymbol{M}^t \boldsymbol{u}_0\|_2 \tag{3.15}$$

But we know that there exist a $\boldsymbol{u}_0 \in \mathbb{R}^m$ that only depends on M such that $\|\boldsymbol{M}^t \boldsymbol{u}_0\|_2 = \|\boldsymbol{M}^t\| \|\boldsymbol{u}_0\|_2$ (explicitly $\boldsymbol{u}_0$ is the eigenvector associated with the largest eigenvalue of $\boldsymbol{M}^\top \boldsymbol{M}$). But, using [Bertsekas, 1999, Proposition A.15] we know that $\rho(\boldsymbol{M}) \leq \|\boldsymbol{M}\|_2$. Then we have that,

$$\|\boldsymbol{u}_t\|_2 \geq \rho(\boldsymbol{M})^t \|\boldsymbol{u}_0\|_2 \tag{3.16}$$

- If $|\lambda| = 1\,,\ \forall \lambda \in Sp(M)$, we can diagonalize $\boldsymbol{M}$ such that $\boldsymbol{M} = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1}$ where $\boldsymbol{P}$ is invertible and $\boldsymbol{D}$ is a diagonal matrix with complex values of magnitude 1.

  We have for $t \geq 0$ and any $\boldsymbol{u}_0 \in \mathbb{R}^m$,

$$\begin{aligned} \|\boldsymbol{u}_t\|_2 &= \|\boldsymbol{M}^t \boldsymbol{u}_0\|_2 && (3.17)\\ &= \|\boldsymbol{P}\boldsymbol{D}^t\boldsymbol{P}^{-1}\boldsymbol{u}_0\|_2 && (3.18)\\ &\leq \|\boldsymbol{P}\| \|\boldsymbol{D}^t\| \|\boldsymbol{P}^{-1}\| \|\boldsymbol{u}_0\|_2 = \|\boldsymbol{P}\| \|\boldsymbol{P}^{-1}\| \|\boldsymbol{u}_0\|_2 && (3.19) \end{aligned}$$

  Similarly,

$$\begin{aligned} \|\boldsymbol{u}_0\|_2 &= \|\boldsymbol{M}^{-t} \boldsymbol{u}_t\|_2 && (3.20)\\ &= \|\boldsymbol{P}\boldsymbol{D}^{-t}\boldsymbol{P}^{-1}\boldsymbol{u}_t\|_2 && (3.21)\\ &\leq \|\boldsymbol{P}\| \|\boldsymbol{D}^t\| \|\boldsymbol{P}^{-1}\| \|\boldsymbol{u}_t\|_2 = \|\boldsymbol{P}\| \|\boldsymbol{P}^{-1}\| \|\boldsymbol{u}_t\|_2 && (3.22) \end{aligned}$$

□

4 Proofs of the Theorems and Propositions

4.1 Proof of Thm. 1

Let us recall the Theorem proposed by Bertsekas [1999, Proposition 4.4.1]. We also provide a convergence rate that was not previously stated in [Bertsekas, 1999].

Theorem 1. *If the spectral radius* $\rho_{\max} := \rho(\nabla F_\eta(\boldsymbol{\omega}^*)) < 1$*, then, for* $\boldsymbol{\omega}_0$ *in a neighborhood of* $\boldsymbol{\omega}^*$*, the distance of* $\boldsymbol{\omega}_t$ *to the stationary point* $\boldsymbol{\omega}^*$ *converges at a linear rate of* $\mathcal{O}\Big((\rho_{\max} + \epsilon)^t\Big)\,,\ \forall \epsilon > 0$*.*

Proof. For brevity let us write $x_t := (\phi_t, \theta_t)$ for $t \geq 0$ and $x^* := (\phi^*, \theta^*)$. Let $\epsilon > 0$.

By Proposition A.15 [Bertsekas, 1999] there exists a norm $\|\cdot\|$ such that its induced matrix norm has the following property:

$$\|\nabla F_\eta(x^*)\| \leq \rho(\nabla F_\eta(x^*)) + \frac{\epsilon}{2}\,. \tag{4.1}$$

Then by definition of the sequence (x_t) and since x^* is a fixed point of F_η, we have that,

$$\|x_{t+1} - x^*\| = \|F_\eta(x_t) - F_\eta(x^*)\| \tag{4.2}$$

Since F_η is assumed to be continuously differentiable by the mean value theorem we have that

$$F_\eta(x_t) = F_\eta(x^*) + \nabla F_\eta(\tilde{x}_t)(x_t - x^*)\,, \tag{4.3}$$

for some $\tilde{x}_t \in [x_t, x^*]$. Then,

$$\|x_{t+1} - x^*\| \leq \|\nabla F_\eta(\tilde{x}_t)\|\|x_t - x^*\| \tag{4.4}$$

where $\|\nabla F_\eta(\tilde{x}_t)\|$ is the induced matrix norm of $\|\cdot\|$.

Since the induced norm of a square matrix is continuous on its elements and since we assumed that ∇F_η was continuous, there exists $\delta > 0$ such that,

$$\|\nabla F_\eta(x) - \nabla F_\eta(x^*)\| \leq \frac{\epsilon}{2}\,, \quad \forall x \;:\; \|x - x^*\| \leq \delta\,. \tag{4.5}$$

Finally, we get that if $\|x_t - x^*\| \leq \delta$, then,

$$\begin{aligned}
\|x_{t+1} - x^*\| &\leq \|\nabla F_\eta(\tilde{x}_t)\|\|x_t - x^*\| && (4.6)\\
&\leq \left(\|\nabla F_\eta(x^*)\| + \|\nabla F_\eta(\tilde{x}_t) - \nabla F_\eta(x^*)\|\right)\|x_t - x^*\| && (4.7)\\
&\leq \left(\rho(\nabla F_\eta(x^*)) + \frac{\epsilon}{2} + \frac{\epsilon}{2}\right)\|x_t - x^*\| && (4.8)
\end{aligned}$$

where in the last line we used (4.1) and (4.5). Consequently, if $\rho(\nabla F_\eta(x^*) < 1$ and if $\|x_0 - x^*\| \leq \delta$, we have that,

$$\|x_t - x^*\| \leq \left(\rho(\nabla F_\eta(x^*)) + \epsilon\right)^t \|x_0 - x^*\| \leq \delta\,, \quad \forall \epsilon > 0\,. \tag{4.9}$$

□

4.2 Proof of Thm. 2

We are interested in the optimal step-size for the Simultaneous gradient with no momentum. Define the step-size associated to one eigenvalue $\lambda \in \mathbb{C}$ by $\hat{\eta}(\lambda) := \frac{\Re(\lambda)}{|\lambda|^2}$.

Theorem 2. *If the eigenvalues of $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ all have a positive real-part, then, the best step-size η_{best}, which minimizes the spectral radius $\rho_{\max}(\eta)$ of $\nabla F_\eta(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$, is the solution of a (convex) quadratic by parts problem, and satisfies,*

$$\max_{1\leq k\leq m} \sin(\psi_k)^2 \leq \rho_{\max}(\eta_{best})^2 \leq 1 - \Re(1/\lambda_1)\delta\,, \tag{4.10}$$

$$\textit{with} \quad \delta := \min_{1\leq k\leq m} |\lambda_k|^2(2\Re(1/\lambda_k) - \Re(1/\lambda_1)) \tag{4.11}$$

$$\textit{and} \qquad \Re(1/\lambda_1) \leq \eta_{best} \leq 2\Re(1/\lambda_1) \tag{4.12}$$

where $(\lambda_k = r_k e^{i\psi_k})_{1\leq k\leq m} = \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*))$ are sorted such that $0 < \Re(1/\lambda_1) \leq \cdots \leq \Re(1/\lambda_m)$. Particularly, when $\eta_{best} = \Re(1/\lambda_1)$ we are in the case of the top plot of Fig.8.3 and $\rho_{\max}(\eta_{best})^2 = \sin(\psi_1)^2$.

Proof. The eigenvalues of ∇F_η are $1-\eta\lambda$, for $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\varphi}, \boldsymbol{\theta}))$. Our goal is to solve

$$\rho_{\max} := \min_{\eta \geq 0} \max_{1 \leq i \leq m} |1 - \eta\lambda_i|^2 \tag{4.13}$$

where $\{\lambda_1, \ldots, \lambda_m\}$ is the spectrum of $\nabla \boldsymbol{v}(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$. we can develop the magnitude to get,

$$f_i(\eta) := |1 - \eta\lambda_i|^2 = 1 - 2\eta\Re(\lambda_i) + \eta^2|\lambda_i|^2 \tag{4.14}$$

The function $\eta \mapsto \max_{1 \leq i \leq n} f_i(\eta)$ is a convex function quadratic by part. This function goes to $+\infty$ as η gets larger, so it reaches its minimum over $[0, \infty)$. We can notice that each function f_i reaches its minimum for $\eta_i = \frac{\Re(\lambda_i)}{|\lambda_i|^2} = \Re(1/\lambda_i)$. Consequently, if we order the eigenvalues such that,

$$\eta_1 \leq \ldots \leq \eta_m \tag{4.15}$$

we have that

$$f_i'(\eta_1) \leq 0\,, \quad 1 \leq i \leq m \quad \text{and} \quad f_1(x) \geq 1\,, \forall x \geq 2\eta_1 \tag{4.16}$$

As a result,

$$\eta_1 \leq \eta_{best} \leq 2\eta_1 \tag{4.17}$$

Moreover, it is easy to notice that,

$$|1 - \eta_1\lambda_1|^2 = \min_{\eta \geq 0} |1 - \eta\lambda_1|^2 \leq \min_{\eta \geq 0} \max_{1 \leq k \leq m} |1 - \eta\lambda_i|^2 \tag{4.18}$$

Then developing $|1 - \eta_1\lambda_1|^2$, we get that,

$$|1 - \eta_1\lambda_1|^2 = |1 - \tfrac{\Re(\lambda_1)}{|\lambda_1|^2}\lambda_1|^2 = 1 - \tfrac{\Re(\lambda_1)^2}{|\lambda_1|^2} = \sin(\psi_1)^2 \tag{4.19}$$

where $\lambda_1 = r_1 e^{i\psi_1}$. Moreover, we also have that

$$\begin{aligned}
\rho_{\max} &= \min_{\eta \geq 0} \max_{1 \leq i \leq n} |1 - \eta\lambda_i|^2 && (4.20)\\
&\leq \max_{1 \leq k \leq m} |1 - \eta_1\lambda_i|^2 && (4.21)\\
&= 1 - \eta_1 \min_{1 \leq k \leq m} 2\Re(\lambda_k) - \eta_1|\lambda_k|^2 = 1 - \Re(1/\lambda_1)\delta && (4.22)
\end{aligned}$$

This upper bound is then achieved for $\eta = \Re(1/\lambda_1)$. Moreover is $\mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*) \subset [\mu, L]$ we have that, $\lambda_1 = L$ and that

$$\delta \geq \min_{\lambda \in [\mu, L]} 2\lambda - \lambda^2/L = 2\mu - \mu^2/L \tag{4.23}$$

Consequently we recover the standard upper bound $\rho_{\max}^2 \leq 1 - 2\frac{\mu}{L} + \frac{\mu^2}{L} = (1 - \mu/L)^2$ provided in the convex case.

□

4.3 Proof of Thm. 3

We are now interested in the eigenvalues of the Simultaneous Gradient Method with Momentum.

Theorem 3. *The eigenvalues of* $\nabla F_{\eta,\beta}(\boldsymbol{\omega}^*)$ *are*

$$\mu_\pm(\beta,\eta,\lambda) := (1-\eta\lambda+\beta)\frac{1\pm\Delta^{\frac{1}{2}}}{2}, \tag{4.24}$$

where $\Delta := 1-\frac{4\beta}{(1-\eta\lambda+\beta)^2}$, $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$ *and* $\Delta^{\frac{1}{2}}$ *is the* complex *square root of* Δ *with positive real part*[1]*. Moreover we have the following Taylor approximation,*

$$\mu_+(\beta,\eta,\lambda) = 1-\eta\lambda-\beta\frac{\eta\lambda}{1-\eta\lambda}+O(\beta^2)\,, \tag{4.25}$$

$$\mu_-(\beta,\eta,\lambda) = \frac{\beta}{1-\eta\lambda}+O(\beta^2)\,. \tag{4.26}$$

Proof. The Jacobian of $F_{\eta,\beta}$ is

$$M := \begin{bmatrix} \boldsymbol{I}_n - \eta\nabla\boldsymbol{v}(\boldsymbol{\omega}^*) + \beta\boldsymbol{I}_n & -\beta\boldsymbol{I}_n \\ \boldsymbol{I}_n & \boldsymbol{0}_n \end{bmatrix} \tag{4.27}$$

Its characteristic polynomial can be written:

$$\chi_M(X) = \det(X\boldsymbol{I}_{2n}-M) = \begin{vmatrix} (X-1-\beta)\boldsymbol{I}_n+\eta T & \beta\boldsymbol{I}_n \\ -\boldsymbol{I}_n & X\boldsymbol{I}_n \end{vmatrix} \tag{4.28}$$

where $\nabla\boldsymbol{v}(\boldsymbol{\omega}^*) = PTP^{-1}$ and T is an upper-triangular matrix. Finally by Lemma 11 we have that,

$$\chi_M(X) = \left|X((X-1-\beta)\boldsymbol{I}_n+\eta T)+\beta\boldsymbol{I}_n\right| = \prod_{i=1}^{n}\left(X((X-1-\beta)+\eta\lambda_i)+\beta\right) \tag{4.29}$$

where

$$T = \begin{bmatrix} \lambda_1 & * & \dots & * \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & * \\ 0 & \dots & 0 & \lambda_n \end{bmatrix}.$$

Let λ one of the λ_i we have,

$$X((X-1-\beta)+\eta\lambda)+\beta = X^2-(1-\eta\lambda+\beta)X+\beta \tag{4.30}$$

[1] If Δ is a negative real number we set $\Delta^{\frac{1}{2}} := i\sqrt{-\Delta}$

The roots of this polynomial are

$$\mu_+(\lambda) = \frac{1 - \eta\lambda + \beta + \sqrt{\Delta}}{2} \quad \text{and} \quad \mu_-(\lambda) = \frac{1 - \eta\lambda + \beta - \sqrt{\Delta}}{2} \tag{4.31}$$

where $\Delta := (1 - \eta\lambda + \beta)^2 - 4\beta$ and $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$. This can be rewritten as,

$$\mu_\pm(\beta, \eta, \lambda) := (1 - \eta\lambda + \beta)\frac{1 \pm \Delta^{\frac{1}{2}}}{2} \tag{4.32}$$

where $\Delta := 1 - \frac{4\beta}{(1-\eta\lambda+\beta)^2}$, $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*))$ and $\Delta^{\frac{1}{2}}$ is the *complex* square root of Δ with real positive part (if Δ is a real negative number, we set $\Delta^{\frac{1}{2}} := i\sqrt{-\Delta}$). Moreover we have the following Taylor approximation,

$$\mu_+(\beta, \eta, \lambda) = 1 - \eta\lambda - \beta\frac{\eta\lambda}{1 - \eta\lambda} + O(\beta^2) \quad \text{and} \quad \mu_-(\beta, \eta, \lambda) = \frac{\beta}{1 - \eta\lambda} + O(\beta^2). \tag{4.33}$$

□

4.4 Proof of Thm. 4

We are interested in the impact of small Momentum values on the convergence rate of Simultaneous Gradient Method.

Theorem 4. *For any* $\lambda \in \mathrm{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$ *s.t.* $\Re(\lambda) > 0$*,*

$$\rho'_{\lambda,\eta}(0) > 0 \Leftrightarrow \eta \in I(\lambda) := \left(\tfrac{|\lambda| - |\Im(\lambda)|}{|\lambda|\Re(\lambda)}, \tfrac{|\lambda| + |\Im(\lambda)|}{|\lambda|\Re(\lambda)}\right) .$$

Particularly, we have $\rho'_{\lambda,\Re(1/\lambda)}(0) = 2\Re(\lambda)\Re(1/\lambda) > 0$ *and if* $|Arg(\lambda)| \geq \frac{\pi}{4}$ *then,* $(\Re(1/\lambda), 2\Re(1/\lambda)) \subset I(\lambda)$*.*

Proof. Recall the definitions of μ_+ and μ_- from Thm. 3, and the definition of the radius:

$$\rho_{\lambda,\eta}(\beta) := \max\left\{|\mu_+|^2, |\mu_-|^2\right\} \tag{4.34}$$

When β is close to 0, μ_- is close also to 0 whereas μ_+ is close to $1 - \eta\lambda$. In general $1 - \eta\lambda \neq 0$, so around 0, $\rho_{\lambda,\eta}(\beta) = |\mu_+(\beta)|^2 = \mu_+(\beta)\bar{\mu}_+(\beta)$. The special case where $1 - \eta\lambda = 0$ is excluded from this analysis because it means that the eigenvalue λ is not one constraining the learning rate as seen in Thm. 2. Computing

the derivative of ρ give us

$$\rho'_{\lambda,\eta}(0) = (\mu_+\bar{\mu}_+)'(0) = \mu_+(0)\bar{\mu}'_+(0) + \bar{\mu}_+(0)\mu'_+(0) \tag{4.35}$$
$$= 2\Re(\mu_+(0)\bar{\mu}'_+(0)) \tag{4.36}$$
$$= 2\Re\left((1-\eta\lambda)\frac{-\eta\bar{\lambda}}{1-\eta\bar{\lambda}}\right) \tag{4.37}$$
$$= -2\eta\Re\left(\frac{\bar{\lambda}(1-\eta\lambda)^2}{|1-\eta\lambda|^2}\right) \tag{4.38}$$
$$= \frac{-2\eta}{|1-\eta\lambda|^2}\left[\Re(\lambda) - 2\eta|\lambda|^2 + \eta^2|\lambda|^2\Re(\lambda)\right] \tag{4.39}$$

which leads to,

$$\rho'_{\lambda,\eta}(0) = 2\frac{2\eta^2|\lambda|^2 - \eta\Re(\lambda)(1+\eta^2|\lambda|^2)}{|1-\eta\lambda|^2} \tag{4.40}$$

The sign of $\rho'_{\lambda,\eta}(0)$ is determined by the sign of

$$2\eta|\lambda|^2 - \Re(\lambda)(1+\eta^2|\lambda|^2) = -\Re(\lambda)|\lambda|^2\eta^2 + 2|\lambda|^2\eta - \Re(\lambda) \tag{4.41}$$

This quadratic function is strictly positive on the open interval $\left(\frac{|\lambda|-|\Im(\lambda)|}{|\lambda|\Re(\lambda)}, \frac{|\lambda|+|\Im(\lambda)|}{|\lambda|\Re(\lambda)}\right)$.

Moreover since $\Re(1/\lambda) = \frac{\Re(\lambda)}{|\lambda|^2}$, we have that $|1-\lambda\Re(1/\lambda)|^2 = 1 - \Re(\lambda)\Re(1/\lambda)$ (see Eq. 4.19) and then,

$$\rho'_{\lambda,\Re(1/\lambda)}(0) = 2\Re(\lambda)\Re(1/\lambda). \tag{4.42}$$

Finally writting $\lambda = re^{i\psi}$ we get that,

$$\frac{|\lambda| - |\Im(\lambda)|}{|\lambda|\Re(\lambda)} = \frac{1-|\sin(\psi)|}{r\cos(\psi)} = \Re(1/\lambda)\frac{1-|\sin(\psi)|}{1-|\sin(\psi)|^2} \tag{4.43}$$

and

$$\frac{|\lambda| + |\Im(\lambda)|}{|\lambda|\Re(\lambda)} = \frac{1+|\sin(\psi)|}{r\cos(\psi)} = \Re(1/\lambda)\frac{1+|\sin(\psi)|}{1-|\sin(\psi)|^2}$$

Consequently,

$$I(\lambda) = \left(\frac{\Re(1/\lambda)}{1+|\sin(\psi)|}, \frac{\Re(1/\lambda)}{1-|\sin(\psi)|}\right) \tag{4.44}$$

and $|\arg(\lambda)| \geq \frac{\pi}{4}$ implies that $\left(\frac{2}{3}\Re(1/\lambda), 2\Re(1/\lambda)\right)$ □

4.5 Proof of Thm. 5

We are now in the special case of a bilinear game. We first consider the simultaneous gradient step with momentum The operator $F_{\eta,\beta}$ is defined as:

$$F^{\text{sim}}_{\eta,\beta}\begin{bmatrix}\boldsymbol{\theta}_t\\ \boldsymbol{\varphi}_t\\ \boldsymbol{\theta}_{t-1}\\ \boldsymbol{\varphi}_{t-1}\end{bmatrix} := \begin{bmatrix}\boldsymbol{\theta}_t - \eta_1\boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})\\ \boldsymbol{\varphi}_t + \eta_2\boldsymbol{A}^\top\boldsymbol{\theta}_t + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1})\\ \boldsymbol{\theta}_t\\ \boldsymbol{\varphi}_t\end{bmatrix}. \tag{4.45}$$

Proposition 1. *The eigenvalues of $\nabla F^{sim}_{\eta,\beta}$ are the roots of the 4^{th} order polynomials:*

$$(x-1)^2(x-\beta_1)(x-\beta_2)+\eta_1\eta_2\lambda x^2\,,\ \lambda\in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A}). \tag{4.46}$$

Particularly, when $\beta_1=\beta_2=0$ and $\eta_1=\eta_2=\eta$ we have,

$$P_\lambda(x)=x^2(x^2-2x+1+\eta^2\lambda)\ ,\ \lambda\in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A}) \tag{4.47}$$

Proof. $F^{\mathrm{sim}}_{\eta,\beta}$ is a linear operator belonging to $\mathbb{R}^{d\times p}$, for notational compactness let us call $m:=d+p$. Let us recall that $\boldsymbol{I}_m$ and $\boldsymbol{0}_{d,p}$ are respectively the identity of $\mathbb{R}^{m\times m}$ and the zero matrix of $\mathbb{R}^{d\times p}$.

$$\nabla F^{\mathrm{sim}}_{\eta,\beta}=\begin{bmatrix}\boldsymbol{I}_m & \boldsymbol{0}_m\\ \boldsymbol{I}_m & \boldsymbol{0}_m\end{bmatrix}+\begin{bmatrix}\boldsymbol{0}_d & -\eta_1\boldsymbol{A} & \boldsymbol{0}_m\\ \eta_2\boldsymbol{A}^\top & \boldsymbol{0}_p & \\ \boldsymbol{0}_m & & \boldsymbol{0}_m\end{bmatrix}+\begin{bmatrix}\beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p} & -\beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \boldsymbol{0}_{p,d} & \beta_2\boldsymbol{I}_p & \boldsymbol{0}_{p,d} & -\beta_2\boldsymbol{I}_p\\ \boldsymbol{0}_m & & \boldsymbol{0}_m & \end{bmatrix} \tag{4.48}$$

Leading to the compressed form

$$\nabla F^{\mathrm{sim}}_{\eta,\beta}=\begin{bmatrix}(1+\beta)\boldsymbol{I}_d & -\eta\boldsymbol{A} & -\beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \eta\boldsymbol{A}^\top & (1+\beta)\boldsymbol{I}_p & \boldsymbol{0}_{p,d} & -\beta_2\boldsymbol{I}_p\\ \boldsymbol{I}_m & & \boldsymbol{0}_m & \end{bmatrix} \tag{4.49}$$

Then the characteristic polynomial of this matrix is equal to,

$$\chi(X):=\begin{vmatrix}(X-1-\beta_1)\boldsymbol{I}_d & \eta\boldsymbol{A} & \beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ -\eta\boldsymbol{A}^\top & (X-1-\beta_2)\boldsymbol{I}_p & \boldsymbol{0}_{p,d} & \beta_2\boldsymbol{I}_p\\ -\boldsymbol{I}_m & & X\boldsymbol{I}_m & \end{vmatrix} \tag{4.50}$$

Then we can use Lemma 11 to compute this determinant,

$$\chi(X)=\det\left(X\begin{bmatrix}(X-1-\beta_1)\boldsymbol{I}_d & \eta_1\boldsymbol{A}\\ -\eta_2\boldsymbol{A}^\top & (X-1-\beta_2)\boldsymbol{I}_p\end{bmatrix}+\begin{bmatrix}\beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \boldsymbol{0}_{p,d} & \beta_2\boldsymbol{I}_p\end{bmatrix}\right) \tag{4.51}$$

$$=\begin{vmatrix}(X(X-1-\beta_1)+\beta_1)\boldsymbol{I}_d & \eta_1X\boldsymbol{A}\\ -\eta_2X\boldsymbol{A}^\top & (X(X-1-\beta_2)+\beta_2)\boldsymbol{I}_p\end{vmatrix} \tag{4.52}$$

$$=\begin{vmatrix}(X-\beta_1)(X-1)\boldsymbol{I}_d+\eta_1\eta_2\frac{X^2}{X(X-1-\beta_2)+\beta_2}\boldsymbol{A}^\top\boldsymbol{A} & \eta_1X\boldsymbol{A}\\ \boldsymbol{0}_{d,p} & (X-\beta_2)(X-1)\boldsymbol{I}_p\end{vmatrix} \tag{4.53}$$

Where for the last equality we added to the first block column the second one multiplied by $\eta_2\boldsymbol{A}^\top\frac{X}{X(X-1-\beta_2)+\beta_2}$. It's now time to introduce r the rank of $\boldsymbol{A}$. We can diagonalize $\boldsymbol{A}^\top\boldsymbol{A}=\boldsymbol{U}^\top diag(\lambda_1,\dots,\lambda_r,0,\dots,0)\boldsymbol{U}$ to get the determinant of a triangular matrix,

$$\chi(X)=P_1(X)P_2(X) \tag{4.54}$$

where

$$P_1(X) := ((X-\beta_1)(X-1))^{d-r}((X-\beta_2)(X-1))^{p-r} \tag{4.55}$$

$$P_2(X) := \prod_{k=1}^{r}\left[(X-\beta_1)(X-1)(X-\beta_2)(X-1)+\eta_1\eta_2X^2\lambda_k\right]. \tag{4.56}$$

This is the characteristic polynomial we were seeking, taking into account the null singular values of A.

In particular, when $\beta_1=\beta_2=0$, we get,

$$\chi(X) = X^m(X-1)^{m-2r}\prod_{k=1}^{r}((X-1)^2+\eta_1\eta_2\lambda_k) \tag{4.57}$$

□

Theorem 5. *For any $\eta_1,\eta_2\geq 0$ and $\beta_1=\beta_2=\beta$, the iterates of the simultaneous methods (5.4) diverge as,*

$$\Delta_t \in \begin{cases} \Omega\Big(\Delta_0(1+\eta^2\sigma_{\max}^2(A))^t\Big) & \text{if} \quad \beta\geq 0 \\ \Omega\Big(\Delta_0(1+\frac{\eta^2\sigma_{\max}^2(A)}{17})^t\Big) & \text{if} \quad -\dfrac{1}{16}\leq\beta<0\,. \end{cases}$$

Proof of Thm. 5. We report the maximum magnitudes of the eigenvalues of the polynomial from Prop. 1 in Fig. C.1. We observe that they are larger than 1. We now prove it in several cases. Let us start with the simpler case $\beta_1=\beta_2=0$. Using Lemma 12, there exists $(\boldsymbol{\theta}^*,\boldsymbol{\varphi}^*)$ such that for any $t\geq 0$,

$$\boldsymbol{\theta}_t-\boldsymbol{\theta}^*\in \mathit{span}(\boldsymbol{A})=\mathit{span}(\boldsymbol{A}\boldsymbol{A}^\top) \;\text{ and }\; \boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*\in\mathit{span}(\boldsymbol{A}^\top)=\mathit{span}(\boldsymbol{A}^\top\boldsymbol{A})\,,\; t\geq 0 \tag{4.58}$$

Then, we have,

$$\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}^*\|^2 = \|\boldsymbol{\theta}_t-\boldsymbol{\theta}^*-2\eta\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)\|^2 \tag{4.59}$$

$$= \|\boldsymbol{\theta}_t-\boldsymbol{\theta}^*\|^2-2\eta(\boldsymbol{\theta}_t-\boldsymbol{\theta}^*)\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)+\eta^2\|\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)\|^2 \tag{4.60}$$

$$\overset{(4.58)}{\geq} \|\boldsymbol{\theta}_t-\boldsymbol{\theta}^*\|^2-2\eta(\boldsymbol{\theta}_t-\boldsymbol{\theta}^*)\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)+\eta^2\sigma_{\min}^2(\boldsymbol{A})\|\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*\|^2 \tag{4.61}$$

where in line 1 we used that $\boldsymbol{A}\boldsymbol{\varphi}^*=0$ and in line 3 we used that $\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*$ is orthogonal to the null space of $\boldsymbol{A}$, so that we lower bound the product by the smallest non-zero singular value $\sigma_{\min}(\boldsymbol{A})$. The same way, we get:

$$\|\boldsymbol{\varphi}_{t+1}-\boldsymbol{\varphi}^*\|^2 = \|\boldsymbol{\varphi}-\boldsymbol{\varphi}^*\|^2+2\eta(\boldsymbol{\theta}_t-\boldsymbol{\theta}^*)\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)+2\eta^2\|\boldsymbol{A}^\top(\boldsymbol{\theta}-\boldsymbol{\theta}^*)\|^2 \tag{4.62}$$

$$\overset{(4.58)}{\geq} \|\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*\|^2+2\eta(\boldsymbol{\theta}_t-\boldsymbol{\theta}^*)\boldsymbol{A}(\boldsymbol{\varphi}_t-\boldsymbol{\varphi}^*)+\eta^2\sigma_{\min}^2(\boldsymbol{A})\|\boldsymbol{\theta}_t-\boldsymbol{\theta}^*\|^2 \tag{4.63}$$

Summing (4.61) and (4.63), we get

$$\Delta_{t+1} \geq (1 + \eta^2 \sigma_{\min}^2(\boldsymbol{A}))\Delta_t \tag{4.64}$$

where $\sigma_{\min}^2(\boldsymbol{A})$ is the minimal (positive) squared singular value of $\boldsymbol{A}$.

Now we can try to handle the case where $\beta_1 = \beta_2 = \beta \neq 0$. To prove Thm. 5 we will prove the following Proposition

Proposition 2. *Let $F_{\eta,\beta}^{sim}$ the operator defined in (5.4).*

- *For $\beta \geq 0$ its radial spectrum is lower bounded by $1 + \eta_1\eta_2\sigma_{\max}^2(A)$.*
- *For $-1/16 \leq \beta < 0$ its radial spectrum is lower bounded by $1 + \eta_1\eta_2\sigma_{\max}^2(A)/17$.*

Proof of Proposition 2. Let us use Proposition 1 to get that the eigenvalues of our linear operator are the solutions of

$$(x-1)^2(x-\beta)^2 + \eta^2\lambda x^2\,, \quad \lambda \in Sp(\boldsymbol{A}^\top\boldsymbol{A})\,. \tag{4.65}$$

Let us fix $\lambda > 0$ belonging to $Sp(\boldsymbol{A}^\top\boldsymbol{A})$. For simplicity, let us note $\alpha^2 = \eta^2\lambda$. We can then notice that this polynomial can be factorized as

$$(x-1)^2(x-\beta)^2 + (\alpha x)^2 = ((x-1)(x-\beta) + i\alpha x)\,((x-1)(x-\beta) - i\alpha x) \tag{4.66}$$

Then the roots of these 2 quadratic polynomials are

$$z_1 = \frac{1+\beta+i\alpha + ((1+\beta+i\alpha)^2 - 4\beta)^{1/2}}{2} \tag{4.67}$$

$$z_2 = \frac{1+\beta+i\alpha - ((1+\beta+i\alpha)^2 - 4\beta)^{1/2}}{2} \tag{4.68}$$

$$z_3 = \frac{1+\beta-i\alpha + ((1+\beta-i\alpha)^2 - 4\beta)^{1/2}}{2} \tag{4.69}$$

$$\text{and } z_4 = \frac{1+\beta-i\alpha - ((1+\beta-i\alpha)^2 - 4\beta)^{1/2}}{2}\,. \tag{4.70}$$

where $\pm z^{1/2}$ are the complex square roots of z with positive imaginary part. Our goal is going to be to show that z_1 has a magnitude larger than 1.

We are going to use the fact that

$$\Re(z^{1/2}) = \sqrt{\frac{|z| + \Re(z)}{2}} \quad \text{and} \quad \Im(z^{1/2}) = \sqrt{\frac{|z| - \Re(z)}{2}} \tag{4.71}$$

Let us first assume that $\beta < 0$. We have that,

$$\Re(z^{1/2}) = \sqrt{\frac{\sqrt{((1-\beta)^2-\alpha^2)^2+4\alpha^2(1+\beta)^2}+(1-\beta)^2-\alpha^2}{2}} \tag{4.72}$$

$$= \sqrt{\frac{\sqrt{((1-\beta)^2+\alpha^2)^2+16\alpha^2\beta}+(1-\beta)^2-\alpha^2}{2}} \tag{4.73}$$

$$\geq \sqrt{\frac{(1-\beta)^2+\alpha^2+16\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2}+(1-\beta)^2-\alpha^2}{2}} \tag{4.74}$$

$$= \sqrt{(1-\beta)^2+8\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2}} \tag{4.75}$$

$$\geq 1-\beta+8\frac{\alpha^2\beta}{(1-\beta)(\alpha^2+(1-\beta)^2)} \tag{4.76}$$

where for the two inequalities we used $\sqrt{1+x} \geq 1+x\,, \forall x \leq 0$. With the same ideas we can lower bound the Imaginary part of $z^{1/2}$,

$$\Im(z^{1/2}) = \sqrt{\frac{\sqrt{((1-\beta)^2-\alpha^2)^2+4\alpha^2(1+\beta)^2}-(1-\beta)^2+\alpha^2}{2}} \tag{4.77}$$

$$= \sqrt{\frac{\sqrt{((1-\beta)^2+\alpha^2)^2+16\alpha^2\beta}-(1-\beta)^2+\alpha^2}{2}} \tag{4.78}$$

$$\geq \sqrt{\frac{(1-\beta)^2+\alpha^2+16\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2}-(1-\beta)^2+\alpha^2}{2}} \tag{4.79}$$

$$= \sqrt{\alpha^2+8\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2}} \tag{4.80}$$

$$\geq \alpha+8\frac{\alpha\beta}{\alpha^2+(1-\beta)^2} \tag{4.81}$$

Consequently we can use (4.76) and (4.81) to lower bound the magnitude of z_1 (defined in Eq. 4.67) as,

$$|z_1|^2 = \Re(z_1)^2+\Im(z_1)^2 \tag{4.82}$$

$$\geq \left(1+4\frac{\alpha^2\beta}{(1-\beta)(\alpha^2+(1-\beta)^2)}\right)^2+\left(\alpha+4\frac{\alpha\beta}{\alpha^2+(1-\beta)^2}\right)^2 \tag{4.83}$$

$$\geq 1+8\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2}+\alpha^2+8\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2} \tag{4.84}$$

$$= 1+\alpha^2+16\alpha^2\beta \tag{4.85}$$

For $-1/16 \leq \beta < 0$ we have that $\alpha^2 + 16\frac{\alpha^2\beta}{\alpha^2+(1-\beta)^2} \geq \frac{\alpha^2}{17}$. Hence,

$$|z_1|^2 \geq 1 + \frac{\alpha^2}{17}\,, \quad \forall -1/16 \leq \beta < 0 \tag{4.86}$$

Let us now consider the case $\beta \geq 0$. By using the fact that $\sqrt{a+b} \geq \sqrt{a}\,,\ \forall a, b \geq 0$ we have that,

$$\Re(z^{1/2}) = \sqrt{\frac{\sqrt{((1-\beta)^2+\alpha^2)^2 + 16\alpha^2\beta} + (1-\beta)^2 - \alpha^2}{2}} \geq 1 - \beta \tag{4.87}$$

and the same way,

$$\Im(z^{1/2}) = \sqrt{\frac{\sqrt{((1-\beta)^2-\alpha^2)^2 + 4\alpha^2(1+\beta)^2} - (1-\beta)^2 + \alpha^2}{2}} \geq \alpha \tag{4.88}$$

I then quickly leads to

$$|z_1|^2 \geq 1 + \alpha^2\,. \tag{4.89}$$

□

To conclude this proof we just need to combine Proposition 2 with Lemma 13 saying that if the spectral radius is strictly larger than 1 then the iterates diverge. □

4.6 Proof of Thm. 6

Proposition 2. *The eigenvalues of $\nabla F^{alt}_{\eta,\beta}$ are the roots of the 4^{th} order polynomials:*

$$(x-1)^2(x-\beta_1)(x-\beta_2) + \eta_1\eta_2\lambda x^3\,,\ \lambda \in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A}) \tag{4.90}$$

Particularly for $\beta_1 = \beta_2 = 0$ and $\eta_1\eta_2 = \eta^2$ we get

$$P_\lambda(x) = x^2((x-1)^2 + \eta^2\lambda x^3\,, \quad \lambda \in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A}) \tag{4.91}$$

Giving the following set of eigenvalues,

$$\{0\} \cup \left\{1 + \eta\frac{-\eta\lambda \pm \sqrt{\eta^2\lambda^2 - 4\lambda}}{2} \ :\ \lambda \in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A})\right\} \tag{4.92}$$

Particularly for $\beta_1 = -\frac{1}{2}$ and $\beta_2 = 0$ we get

$$x[(x-1)^2(x+\frac{1}{2}) + \eta^2\lambda x^2]\,, \quad \lambda \in \mathrm{Sp}(\boldsymbol{A}^\top\boldsymbol{A}) \tag{4.93}$$

proof of Proposition 2. Let us recall the definition of $F^{\text{alt}}_{\eta,\beta}$, (for compactness we note $F^{\text{alt}}_{\eta,\beta} = F^{\text{alt}}_{\eta_1,\eta_2,\beta_1,\beta_2}$)

$$F^{\text{alt}}_{\eta,\beta}\begin{bmatrix}\boldsymbol{\theta}_t\\ \boldsymbol{\varphi}_t\\ \boldsymbol{\theta}_{t-1}\\ \boldsymbol{\varphi}_{t-1})\end{bmatrix} := \begin{bmatrix}\boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})\\ \boldsymbol{\varphi}_t + \eta_2 \boldsymbol{A}^\top \boldsymbol{\theta}_{t+1} + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1})\\ \boldsymbol{\theta}_t\\ \boldsymbol{\varphi}_t\end{bmatrix} \tag{4.94}$$

$$= \begin{bmatrix}\boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})\\ \boldsymbol{\varphi}_t + \eta_2 \boldsymbol{A}^\top(\boldsymbol{\theta}_t - \eta_1 \boldsymbol{A}\boldsymbol{\varphi}_t + \beta_1(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})) + \beta_2(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1})\\ \boldsymbol{\theta}_t\\ \boldsymbol{\varphi}_t\end{bmatrix} \tag{4.95}$$

Hence, the matrix $F^{\text{alt}}_{\eta,\beta}$ is,

$$F^{\text{alt}}_{\eta,\beta} = \begin{bmatrix}(1+\beta_1)\boldsymbol{I}_d & -\eta_1\boldsymbol{A} & -\beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ (1+\beta_1)\eta_2\boldsymbol{A}^\top & (1+\beta_2)\boldsymbol{I}_p - \eta_1\eta_2\boldsymbol{A}^\top\boldsymbol{A} & -\beta_1\eta_2\boldsymbol{A}^\top & -\beta_2\boldsymbol{I}_p\\ & \boldsymbol{I}_m & \boldsymbol{0}_m & \end{bmatrix} \tag{4.96}$$

Then the characteristic polynomial of $F^{\text{alt}}_{\eta,\beta}$ is equal to

$$\chi(X) = \begin{vmatrix}(X-1-\beta_1)\boldsymbol{I}_d & \eta_1\boldsymbol{A} & \beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ -(1+\beta_1)\eta_2\boldsymbol{A}^\top & (X-1-\beta_2)\boldsymbol{I}_p + \eta_1\eta_2\boldsymbol{A}^\top\boldsymbol{A} & \beta_1\eta_2\boldsymbol{A}^\top & \beta_2\boldsymbol{I}_d\\ -\boldsymbol{I}_d & \boldsymbol{0}_{d,p} & X\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \boldsymbol{0}_{p,d} & -\boldsymbol{I}_p & \boldsymbol{0}_{p,d} & X\boldsymbol{I}_p\end{vmatrix} \tag{4.97}$$

$$= \begin{vmatrix}(X-1-\beta_1+\frac{\beta_1}{X})\boldsymbol{I}_d & \eta_1\boldsymbol{A} & \beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ -(1+\beta_1+\frac{\beta_1}{X})\eta_2\boldsymbol{A}^\top & (X-1-\beta_2+\frac{\beta_2}{X})\boldsymbol{I}_p + \eta_1\eta_2\boldsymbol{A}^\top\boldsymbol{A} & \beta_1\eta_2\boldsymbol{A}^\top & \beta_2\boldsymbol{I}_p\\ \boldsymbol{0}_d & \boldsymbol{0}_{d,p} & X\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \boldsymbol{0}_{d,p} & \boldsymbol{0}_p & \boldsymbol{0}_{p,d} & X\boldsymbol{I}_p\end{vmatrix} \tag{4.98}$$

Where in the last line we added the third block column multiplied by $\frac{1}{X}$ to the first one.Then we have

$$\chi(X) = \begin{vmatrix}(X-1-\beta_1+\frac{\beta_1}{X})\boldsymbol{I}_d & \eta_1\boldsymbol{A} & \beta_1\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ -X\eta_2\boldsymbol{A}^\top & (X-1-\beta_2+\frac{\beta_2}{X})\boldsymbol{I}_p & \boldsymbol{0}_{p,d} & \beta_2\boldsymbol{I}_p\\ \boldsymbol{0}_d & \boldsymbol{0}_{d,p} & X\boldsymbol{I}_d & \boldsymbol{0}_{d,p}\\ \boldsymbol{0}_{d,p} & \boldsymbol{0}_p & \boldsymbol{0}_{p,d} & X\boldsymbol{I}_p\end{vmatrix} \tag{4.99}$$

where we added to the second block line the first block line by $-\eta_2\boldsymbol{A}^\top$. Then our

determinant is triangular by squared blocks of size $m \times m$ and we can write,

$$\chi(X) = \det(X\boldsymbol{I}_m) \begin{vmatrix} (X - 1 - \beta_1 + \frac{\beta_1}{X})\boldsymbol{I}_d & \eta_1 \boldsymbol{A} \\ -X\eta_2 \boldsymbol{A}^\top & (X - 1 - \beta_2 + \frac{\beta_2}{X})\boldsymbol{I}_p \end{vmatrix} \tag{4.100}$$

$$= \begin{vmatrix} (X(X - 1 - \beta_1) + \beta_1)\boldsymbol{I}_d & X\eta_1 \boldsymbol{A} \\ -X^2\eta_2 \boldsymbol{A}^\top & (X(X - 1 - \beta_2) + \beta_2)\boldsymbol{I}_p \end{vmatrix} \tag{4.101}$$

$$= \begin{vmatrix} (X - 1)(X - \beta_1)\boldsymbol{I}_d & X\eta_1 \boldsymbol{A} \\ -X^2\eta_2 \boldsymbol{A}^\top & (X - 1)(X - \beta_2)\boldsymbol{I}_p \end{vmatrix} \tag{4.102}$$

$$= \begin{vmatrix} (X - 1)(X - \beta_1)\boldsymbol{I}_d + \eta_1\eta_2 \boldsymbol{A}^\top \boldsymbol{A} \frac{X^3}{(X-1)(X-\beta_2)} & X\eta_1 \boldsymbol{A} \\ \boldsymbol{0}_{p,d} & (X - 1)(X - \beta_2)\boldsymbol{I}_p \end{vmatrix} \tag{4.103}$$

Now we can diagonalize $\boldsymbol{A}^\top \boldsymbol{A}$ to get,

$$\chi(X) = (X - \beta_1)^{d-r}(X - \beta_2)^{p-r}(X - 1)^{p+d-2r} \tag{4.104}$$

$$\cdot \prod_{k=1}^{r} \left((X - 1)^2(X - \beta_2)(X - \beta_1) + \eta_1\eta_2 X^3 \lambda_k\right), \tag{4.105}$$

where $(\lambda_k)_{1 \leq k \leq r}$ are the positive eigenvalues of $\boldsymbol{A}^\top \boldsymbol{A}$ of rank r. Particularly, when $\beta_1 = \beta_2 = 0$ we have that,

$$\chi(X) = X^m(X - 1)^{m-2r} \prod_{k=1}^{r} \left((X - 1)^2 + \eta_1\eta_2 X\lambda_k\right) \tag{4.106}$$

□

We report the maximum magnitudes of the eigenvalues of the polynomial from Prop. 2 in Fig. C.1. We observe that they are smaller than 1 for a large choice of step-size and momentum values. This is a satisfying numerical result but we want analytical convergence rates. This is what we prove in Thm. 6.

Theorem 6. *If we set* $\eta \leq \frac{1}{\sigma_{\max}(A)}$, $\beta_1 = -\frac{1}{2}$ *and* $\beta_2 = 0$ *then we have*

$$\Delta_{t+1} \in O\left(\max\{\tfrac{1}{2}, 1 - \tfrac{\eta^2\sigma^2_{\min}(A)}{16}\}^t \Delta_0\right) \tag{4.107}$$

If we set $\beta_1 = 0$ *and* $\beta_2 = 0$, *then there exists* $M > 1$ *such that for any* $\eta_1, \eta_2 \geq 0$, $\Delta_t = \Theta(\Delta_0)$.

Proof. In Lemma 12 we showed of the affine transformations $\boldsymbol{\theta}_t \to \boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)$ and $\boldsymbol{\varphi}_t \to \boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*)$ allow us to work on the span of a diagonal matrix $\boldsymbol{D}$. Then in that case the eigenspace of $\boldsymbol{D}$ do not interact with each other. In the sense that for

each coordinate of $[\boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)]_i$ and $[\boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*)]_i$ $(1 \leq i \leq r)$ we have from (3.5) that

$$\begin{cases} [\boldsymbol{U}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*)]_i = [\boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)]_i - \eta_1\sigma_i[\boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*)]_i + \beta_1[\boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}_{t-1})]_i \\ [\boldsymbol{V}(\boldsymbol{\varphi}_{t+1} - \boldsymbol{\varphi}^*)]_i = [\boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*)]_i + \eta_2\sigma_i[\boldsymbol{U}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*)]_i + \beta_2[\boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}_{t-1})]_i \end{cases} \tag{4.108}$$

Consequently we only need to study the 4 dimensional linear operators

$$\begin{bmatrix} (1+\beta_1) & -\eta_1\sigma_i & -\beta_1 & 0 \\ (1+\beta_1)\eta_2\sigma_i & (1+\beta_2) - \eta_1\eta_2\sigma_i^2 & -\beta_1\eta_2\sigma_i & -\beta_2 \\ & \boldsymbol{I}_2 & & \boldsymbol{0}_2 \end{bmatrix} \tag{4.109}$$

for $\sigma_1 \leq \cdots \leq \sigma_r > 0$ the positive singular values of $\boldsymbol{A}$. These equations are a particular case of (4.96). Using the proof of Proposition 2 the eigenvalues of these matrices are the solution of

$$P_i(X) = (X-1)^2(X-\beta_1)(X-\beta_2) + \eta_1\eta_2 X^3\sigma_i^2\,, \quad 1 \leq i \leq r\,. \tag{4.110}$$

We will now consider two case:

- When, $\beta_1 = \beta_2 = 0$ we have that,

$$P_i(X) = X^2((X-1)^2 + \eta_1\eta_2 X\sigma_i^2)\,, \quad 1 \leq i \leq r\,. \tag{4.111}$$

  Then the roots of $P_i(X)$ are 0 and two complex conjugate value with a magnitude equal to the constant term of $(X-1)^2 + \eta_1\eta_2 X\sigma_i^2$ which is 1. Since these two eigenvalues are different, the matrix (4.109) is diagonalizable (for $\beta_1 = \beta_2 = 0$ we can remove the state augmentation to only work with these two eigenvector). Consequently our linear operator is diagonalizable and has all its eigenvalues larger than 1 in magnitude, we can then apply Lemma 13 to conclude that $\Delta_t = \Omega(\Delta_0)$.

- When, $\beta_1 = -\frac{1}{2}$ and $\beta_2 = 0$, we have that $P_i(X) = XQ_i(X)$ where

$$Q_i(X) := (X-1)^2(X+\tfrac{1}{2}) + \eta_1\eta_2 X^2\sigma_i^2\,, \quad 1 \leq i \leq r\,. \tag{4.112}$$

  Then $P_i(-1/2) = +\frac{\eta_1\eta_2\sigma_i^2}{4} > 0$ and $P_i(-1) = -2 + \eta_1\eta_2\sigma_i^2$. If $\eta_1\eta_2 < \frac{2}{\sigma_i^2}$ we have $P_i(-1) < 0$. Consequently, this polynomial has a negative root λ_- such that $-1 < \lambda_- < -\frac{1}{2} < 0$. Moreover the derivative of $Q_i(X)$ is

$$Q_i'(X) = (X-1)(2X+1) + (X-1)^2 + 2\eta_1\eta_2 X\sigma_i^2 = (3X - 3 - \eta_1\eta_2\sigma_i^2)X\,. \tag{4.113}$$

  If $\eta_1\eta_2 < \frac{3}{2\sigma_i^2}$, then $Q_i'(x) > 0\,, \forall x > 0$. Since $Q_i(0) = 1/2 > 0$ then $Q_i(x) > 0\,, \forall x \geq 0$ and consequently all the real roots of Q_i are negative.

Since by the root coefficient relationship the sum of the roots of Q_i has to be equal to $\frac{3}{2} - \eta_1\eta_2\sigma_i^2 > 0$, all the roots of Q_i cannot be real (because the real roots of Q_i are negative). Hence Q_i has two conjugate roots λ_c and $\bar{\lambda}_c$ and one real negative root λ_r. Let us consider $-1 < \lambda_r < -1/2$, we have,

$$4(\lambda_r + \tfrac{1}{2}) + \alpha\lambda_r^2 < (\lambda_r - 1)^2(\lambda_r + \tfrac{1}{2}) + \alpha\lambda_r^2 = 0\,, \tag{4.114}$$

where we called $\alpha = \eta_1\eta_2\sigma_i^2$. Thus we have,

$$-\frac{2+\sqrt{4-2\alpha}}{\alpha} < \lambda_r < \frac{\sqrt{4-2\alpha}-2}{\alpha} \leq \frac{2-\alpha/2-\alpha^2/16-2}{\alpha} = -\frac{1}{2} - \frac{\alpha}{16} \tag{4.115}$$

where we used $1 - \frac{x}{2} - \frac{x^2}{8} \geq \sqrt{1-x}\,,\ 1 > x \geq 0$. Moreover the roots coefficient relationship are

$$\frac{3}{2} - \alpha = 2\Re(\lambda_c) + \lambda_r \tag{4.116}$$

$$0 = |\lambda_c|^2 + 2\lambda_r\Re(\lambda_c) \tag{4.117}$$

$$-\frac{1}{2} = \lambda_r|\lambda_c|^2 \tag{4.118}$$

where we called $\alpha = \eta_1\eta_2\sigma_i^2$. Plugging (4.116) into (4.117) we get

$$0 = |\lambda_c|^2 + (\tfrac{3}{2} - \alpha - \lambda_r)\lambda_r \tag{4.119}$$

Multiplying by λ_r and plugging (4.118) in we get

$$\lambda_r^2 = \frac{1}{3 - 2\alpha - 2\lambda_r} \leq \frac{1}{4-2\alpha} \tag{4.120}$$

where we used that $\lambda_r < -\frac{1}{2}$. Consequently, since in the theorem we assumed that $\eta_1\eta_2 \leq \frac{1}{\sigma_{\max}(\boldsymbol{A})^2}$, we have that $\alpha \leq 1$, we have

$$\lambda_r^2 \leq \frac{1}{4-2\alpha} \leq \frac{1}{2} \quad \text{and} \quad |\lambda_c|^2 = \frac{-1}{2\lambda_r} \leq \frac{1}{1+\frac{\alpha}{8}} \leq 1 - \frac{\alpha}{16} \tag{4.121}$$

where for the last inequality we used $\sqrt{1+x} \leq 1 + \frac{x}{2}\,, \forall x \in \mathbb{R}$ and $(1+x)^{-1} \leq 1 - x/2\,, \forall\, 0 \leq x \leq 1$.

One last thing to say is that the four roots of P_i which are the four eigenvalues of the matrix in (4.109) are different and consequently this matrix is diagonalizable.

We can then apply Lemma 13 in a case of a spectral radius strictly smaller that 1 to conclude that,

$$\Delta_{t+1} \leq \max\{1/2, 1 - \eta_1\eta_2\tfrac{\sigma_{\min}^2(\boldsymbol{A})}{16}\}\Delta_t \tag{4.122}$$

where,

$$\Delta_t := \|\boldsymbol{U}(\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*)\|_2^2 + \|\boldsymbol{U}(\boldsymbol{\theta}_t - \boldsymbol{\theta}^*)\|_2^2 \tag{4.123}$$

$$+ \|\boldsymbol{V}(\boldsymbol{\varphi}_{t+1} - \boldsymbol{\varphi}^*)\|_2^2 + \|\boldsymbol{V}(\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*)\|_2^2 \tag{4.124}$$

$$= \|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\theta}_t - \boldsymbol{\theta}^*\|_2^2 + \|\boldsymbol{\varphi}_{t+1} - \boldsymbol{\varphi}^*\|_2^2 + \|\boldsymbol{\varphi}_t - \boldsymbol{\varphi}^*\|_2^2 \tag{4.125}$$

because $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthogonal.

This concludes the proof. □

D A Closer Look at the Optimization Landscapes of Generative Adversarial Networks

1 Proof of theorems and propositions

1.1 Proof of Theorem 1

Let us recall the theorem of interest:

Proposition 1. *Let us assume that (3.6) is an equality and that $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ is diagonalizable, then there exists a basis $\boldsymbol{P}$ such that the coordinates $\tilde{\boldsymbol{\omega}}(t) := \boldsymbol{P}(\boldsymbol{\omega}(t) - \boldsymbol{\omega}^*)$ have the following behavior,*

1. *For $\lambda_j \in \operatorname{Sp} \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$, $\lambda_j \in \mathbb{R}$, we observe pure attraction:* $\tilde{\boldsymbol{\omega}}_j(t) = e^{-\lambda_j t}[\tilde{\boldsymbol{\omega}}_j(0)$.
2. *For $\lambda_j \in \operatorname{Sp} \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$, $\Re(\lambda_j) = 0$, we observe pure rotation:* $\begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(t) \\ \tilde{\boldsymbol{\omega}}_{j+1}(t) \end{bmatrix} = R_{|\lambda_j| t} \begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(0) \\ \tilde{\boldsymbol{\omega}}_{j+1}(0) \end{bmatrix}$.
3. *Otherwise, we observe both:* $\begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(t) \\ \tilde{\boldsymbol{\omega}}_{j+1}(t) \end{bmatrix} = e^{-\operatorname{Re}(\lambda_j)t} R_{\operatorname{Im}(\lambda_j)t} \begin{bmatrix} \tilde{\boldsymbol{\omega}}_j(0) \\ \tilde{\boldsymbol{\omega}}_{j+1}(0) \end{bmatrix}$.

The matrix R_φ corresponds to a rotation of angle φ. Note that, we re-ordered the eigenvalues such that the complex conjugate eigenvalues form pairs: if $\lambda_j \notin \mathbb{R}$ then $\lambda_{j+1} = \bar{\lambda}_j$.

Proof. The ODE we consider is,

$$\frac{d\boldsymbol{\omega}(t)}{dt} = \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)(\boldsymbol{\omega}(t) - \boldsymbol{\omega}^*) \tag{1.1}$$

The solution of this ODE is

$$\boldsymbol{\omega}(t) = e^{-(t-t_0)\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)}(\boldsymbol{\omega}(t_0) - \boldsymbol{\omega}^*) + \boldsymbol{\omega}^* \tag{1.2}$$

Let us now consider λ an eigenvalue of $\operatorname{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$ such that $\operatorname{Re}(\lambda) > 0$ and $\operatorname{Im}(\lambda) \neq 0$. Since $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ is a real matrix and $\operatorname{Im}(\lambda) \neq 0$ we know that the complex conjugate $\bar{\lambda}$ of λ belongs to $\operatorname{Sp}(\nabla \boldsymbol{v}(\boldsymbol{\omega}^*))$. Let $\boldsymbol{u}_0$ be a complex eigenvector of λ, then we have that,

$$\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)\boldsymbol{u}_0 = \lambda \boldsymbol{u}_0 \quad \Rightarrow \quad \nabla \boldsymbol{v}(\boldsymbol{\omega}^*)\bar{\boldsymbol{u}}_0 = \bar{\lambda}\bar{\boldsymbol{u}}_0 \tag{1.3}$$

and thus $\bar{\boldsymbol{u}}_0$ is a eigenvector of $\bar{\lambda}$. Now if we set $\boldsymbol{u}_1 := \boldsymbol{u}_0 + \bar{\boldsymbol{u}}_0$ and $i\boldsymbol{u}_2 := \boldsymbol{u}_0 - \bar{\boldsymbol{u}}_0$, we have that

$$e^{-t\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)}\boldsymbol{u}_1 = e^{-t\lambda}\boldsymbol{u}_0 + e^{-t\bar{\lambda}}\bar{\boldsymbol{u}}_0 = \mathrm{Re}(e^{-t\lambda})\boldsymbol{u}_1 + \mathrm{Im}(e^{-t\lambda})\boldsymbol{u}_2 \tag{1.4}$$

$$e^{-t\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)}i\boldsymbol{u}_2 = e^{-t\lambda}\boldsymbol{u}_0 - e^{-t\bar{\lambda}}\bar{\boldsymbol{u}}_0 = i(\mathrm{Re}(e^{-t\lambda})\boldsymbol{u}_2 - \mathrm{Im}(e^{-t\lambda})\boldsymbol{u}_1) \tag{1.5}$$

Thus if we consider the basis that diagonalizes $\nabla \boldsymbol{v}(\boldsymbol{\omega}^*)$ and modify the complex conjugate eigenvalues in the way we described right after 1.3 we get the expected diagonal form in a real basis. Thus there exists $\boldsymbol{P}$ such that

$$\nabla \boldsymbol{v}(\boldsymbol{\omega}^*) = \boldsymbol{P}\boldsymbol{D}\boldsymbol{P}^{-1} \tag{1.6}$$

where $\boldsymbol{D}$ is the block diagonal matrix with the block described in Theorem 1. □

1.2 Being a DNE is neither necessary or sufficient for being a LSSP

Let us first recall Example 4.

Example 4. *Let us consider $\mathcal{L}_G$ as a hyperbolic paraboloid (a.k.a., saddle point function) centered in $(1,1)$ where $(1,\varphi)$ is the principal descent direction and $(-\varphi,1)$ is the principal ascent direction, while $\mathcal{L}_D$ is a simple bilinear objective.*

$$\mathcal{L}_G(\theta_1,\theta_2,\varphi) = (\theta_2 - \varphi\theta_1 - 1)^2 - \tfrac{1}{2}(\theta_1 + \varphi\theta_2 - 1)^2\,, \quad \mathcal{L}_D(\theta_1,\theta_2,\varphi) = \varphi(5\theta_1 + 4\theta_2 - 9)$$

We want to show that $(1,1,0)$ is a locally stable stationary point.

Proof. The game vector field has the following form,

$$\boldsymbol{v}(\theta_1,\theta_2,\varphi) = \begin{pmatrix} (2\varphi^2 - 1)\theta_1 - 3\varphi\theta_2 + 2\varphi + 1 \\ (2 - \varphi^2)\theta_2 - 3\varphi\theta_1 - 2 + \varphi \\ 5\theta_1 + 4\theta_2 - 9 \end{pmatrix} \tag{1.7}$$

Thus, $(\theta_1^*,\theta_2^*,\varphi^*) := (1,1,0)$ is a stationary point (i.e., $\boldsymbol{v}(\theta_1^*,\theta_2^*,\varphi^*) = 0$). The Jacobian of the game vector field is

$$\nabla \boldsymbol{v}(\theta_1,\theta_2,\varphi) = \begin{pmatrix} 2\varphi^2 - 1 & -3\varphi & 2 - 3\theta_2 \\ -3\varphi & 2 - \varphi^2 & 1 - 3\theta_1 \\ 5 & 4 & 0 \end{pmatrix}, \tag{1.8}$$

and thus,

$$\nabla \boldsymbol{v}(\theta_1^*,\theta_2^*,\varphi^*) = \begin{pmatrix} -1 & 0 & -1 \\ 0 & 2 & -2 \\ 5 & 4 & 0 \end{pmatrix}. \tag{1.9}$$

We can verify that the eigenvalues of this matrix have a positive real part with any solver (the eigenvalues of a 3×3 always have a closed form) . For completeness we

provide a proof without using the closed form of the eigenvalues. The eigenvalues $\nabla \boldsymbol{v}(\theta_1^*, \theta_2^*, \varphi^*)$ are given by the roots of its characteristic polynomial,

$$\chi(X) := \begin{vmatrix} X+1 & 0 & 1 \\ 0 & X-2 & 2 \\ -5 & -4 & 0 \end{vmatrix} = X^3 - X^2 + 11X - 2\,. \tag{1.10}$$

This polynomial has a real root in $(0,1)$ because $\chi(0) = -2 < 0 < 9 = \chi(1)$. Thus we know that, there exists $\alpha \in (0,1)$ such that,

$$X^3 - X^2 + 11X - 2 = (X-\alpha)(X-\lambda_1)(X-\lambda_2)\,. \tag{1.11}$$

Then we have the equalities,

$$\alpha\lambda_1\lambda_2 = 2 \tag{1.12}$$
$$\alpha + \lambda_1 + \lambda_2 = 1\,. \tag{1.13}$$

Thus, since $0 < \alpha < 1$, we have that,

- If λ_1 and λ_2 are real, they have the same sign $\lambda_1\lambda_2 = 2/\alpha > 0$) and thus are positive ($\lambda_1 + \lambda_2 = 1 - \alpha > 0$).
- If λ_1 is complex then $\lambda_2 = \bar{\lambda}_1$ and thus, $2\Re(\lambda_1) = \lambda_1 + \lambda_2 = 1 - \alpha > 0$.

□

Example 4 showed that LSSP did not imply DNE. Let us construct an example where a game have a DNE which is not locally stable.

Example 5. *Consider the non-zero-sum game with the following respective losses for each player,*

$$\mathcal{L}_1(\theta,\phi) = 4\theta^2 + (\tfrac{1}{2}\phi^2 - 1)\cdot\theta \quad \textit{and} \quad \mathcal{L}_2(\theta,\phi) = (4\theta-1)\phi + \tfrac{1}{6}\theta^3 \tag{1.14}$$

This game has two stationary points for $\theta = 0$ and $\phi = \pm 1$. The Jacobian of the dynamics at these two points are

$$\nabla\boldsymbol{v}(0,1) = \begin{pmatrix} 1 & 1/2 \\ 2 & 1/2 \end{pmatrix} \quad \text{and} \quad \nabla\boldsymbol{v}(0,-1) = \begin{pmatrix} 1 & -1/2 \\ 2 & -1/2 \end{pmatrix} \tag{1.15}$$

Thus,

- The stationary point $(0,1)$ is a DNE but $\mathrm{Sp}(\nabla\boldsymbol{v}(0,1)) = \{\frac{3\pm\sqrt{17}}{4}\}$ contains an eigenvalue with negative real part and so is *not* a LSSP.
- The statioanry point $(0,-1)$ is *not* a DNE but $\mathrm{Sp}(\nabla\boldsymbol{v}(0,1)) = \{\frac{1\pm i\sqrt{7}}{4}\}$ contains only eigenvalue with positive real part and so is a LSSP.

2 Computation of the top-k Eigenvalues of the Jacobian

Neural networks usually have a large number of parameters, this usually makes the storing of the full Jacobian matrix impossible. However the Jacobian vector product can be efficiently computed by using the trick from [Pearlmutter, 1994]. Indeed it's easy to show that $\nabla \boldsymbol{v}(\boldsymbol{\omega})\boldsymbol{u} = \nabla(\boldsymbol{v}(\boldsymbol{\omega})^T \boldsymbol{u})$.

To compute the eigenvalues of the Jacobian of the Game, we first compute the gradient $\boldsymbol{v}(\boldsymbol{\omega})$ over a subset of the dataset. We then define a function that computes the Jacobian vector product using automatic differentiation. We can then use this function to compute the top-k eigenvalues of the Jacobian using the `sparse.linalg.eigs` functions of the Scipy library.

3 Experimental Details

3.1 Mixture of Gaussian Experiment

Dataset. The Mixture of Gaussian dataset is composed of 10,000 points sampled independently from the following distribution $p_{\mathcal{D}}(x) = \frac{1}{2}\mathcal{N}(2, 0.5) + \frac{1}{2}\mathcal{N}(-2, 1)$ where $\mathcal{N}(\mu, \sigma^2)$ is the probability density function of a 1D-Gaussian distribution with mean μ and variance σ^2. The latent variables $z \in \mathbb{R}^d$ are sampled from a standard Normal distribution $\mathcal{N}(0, I_d)$. Because we want to use full-batch methods, we sample 10,000 points that we re-use for each iteration during training.

Neural Networks Architecture. Both the generator and discriminator are one hidden layer neural networks with 100 hidden units and ReLU activations.

WGAN Clipping. Because of the clipping of the discriminator parameters some components of the gradient of the discriminator's gradient should no be taken into account. In order to compute the relevant path angle we apply the following filter to the gradient:

$$\mathbf{1}\left\{(|\boldsymbol{\varphi}| = \mathbf{c}) \ \texttt{and} \ (\text{sign}\nabla_{\boldsymbol{\varphi}}\mathcal{L}_{\mathbf{D}}(\boldsymbol{\omega}) = -\text{sign}\boldsymbol{\varphi})\right\} \tag{3.1}$$

where $\boldsymbol{\varphi}$ is clipped between $-c$ and c. If this condition holds for a coordinate of the gradient then it mean that after a gradient step followed by a clipping the value of the coordinate will not change.

3.2 MNIST Experiment

Dataset We use the training part of MNIST dataset LeCun et al. [2010] (50K examples) for training our models, and scale each image to the range $[-1, 1]$.

Hyperparameters for WGAN-GP on MoG	
Batch size	$= 10,000$ (Full-Batch)
Number of iterations	$= 30,000$
Learning rate for generator	$= 1 \times 10^{-2}$
Learning rate for discriminator	$= 1 \times 10^{-1}$
Gradient Penalty coefficient	$= 1 \times 10^{-3}$

Hyperparameters for NSGAN on MoG	
Batch size	$= 10,000$ (Full-Batch)
Number of iterations	$= 30,000$
Learning rate for generator	$= 1 \times 10^{-1}$
Learning rate for discriminator	$= 1 \times 10^{-1}$

Architecture We use the DCGAN architecture Radford et al. [2016] for our generator and discriminator, with both the NSGAN and WGAN-GP objectives. The only change we make is that we replace the Batch-norm layer in the discriminator with a Spectral-norm layer Miyato et al. [2018], which we find to stabilize training.

Training Details

Hyperparameters for NSGAN with Adam	
Batch size	$= 100$
Number of iterations	$= 100,000$
Learning rate for generator	$= 2 \times 10^{-4}$
Learning rate for discriminator	$= 5 \times 10^{-5}$
β_1	$= 0.5$

Hyperparameters for NSGAN with ExtraAdam	
Batch size	$= 100$
Number of iterations	$= 100,000$
Learning rate for generator	$= 2 \times 10^{-4}$
Learning rate for discriminator	$= 5 \times 10^{-5}$
β_1	$= 0.9$

Hyperparameters for WGAN-GP with Adam	
Batch size	$= 100$
Number of iterations	$= 200,000$
Learning rate for generator	$= 8.6 \times 10^{-5}$
Learning rate for discriminator	$= 8.6 \times 10^{-5}$
β_1	$= 0.5$
Gradient penalty λ	$= 10$
Critic per Gen. iterations λ	$= 5$

Hyperparameters for WGAN-GP with ExtraAdam	
Batch size	$= 100$
Number of iterations	$= 200,000$
Learning rate for generator	$= 8.6 \times 10^{-5}$
Learning rate for discriminator	$= 8.6 \times 10^{-5}$
β_1	$= 0.9$
Gradient penalty λ	$= 10$
Critic per Gen. iterations λ	$= 5$

Computing Inception Score on MNIST. We compute the inception score (IS) for our models using a LeNet classifier pretrained on MNIST. The average IS score of real MNIST data is 9.9.

3.3 Path-Angle Plot

We use the path-angle plot to illustrate the dynamics close to a LSSP. To compute this plot, we need to choose an initial point $\boldsymbol{\omega}$ and an end point $\boldsymbol{\omega}'$. We choose the $\boldsymbol{\omega}$ to be the parameters at initialization, but $\boldsymbol{\omega}'$ can more subtle to choose. In practice, when we use stochastic gradient methods we typically reach a neighborhood of a LSSP where the norm of the gradient is small. However, due to the stochastic noise, we keep moving around the LSSP. In order to be robust to the choice of the end point $\boldsymbol{\omega}'$, we take multiple close-by points during training that have good performance (e.g., high IS in MNIST). In all of figures, we compute the path-angle (and path-norm) for all these end points (with the same start point), and we plot the median path-angle (middle line) and interquartile range (shaded area).

3.4 Instability of Gradient Descent

For the MoG dataset we tried both the extragradient method [Korpelevich, 1976, Gidel et al., 2019b] and the standard gradient descent. We observed that

gradient descent leads to unstable results. In particular the norm of the gradient has very large variance compared to extragradient this is shown in Fig. D.1.

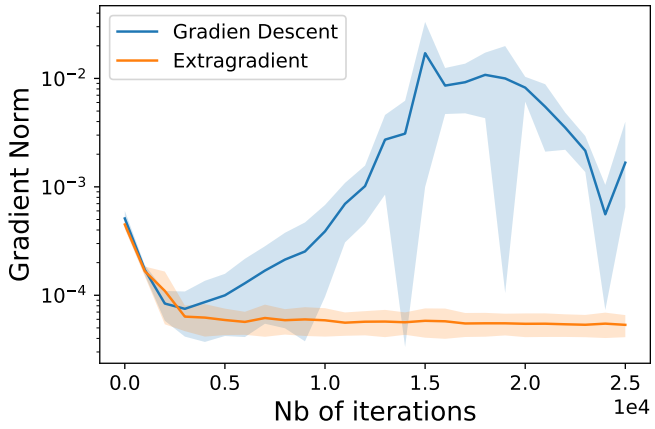
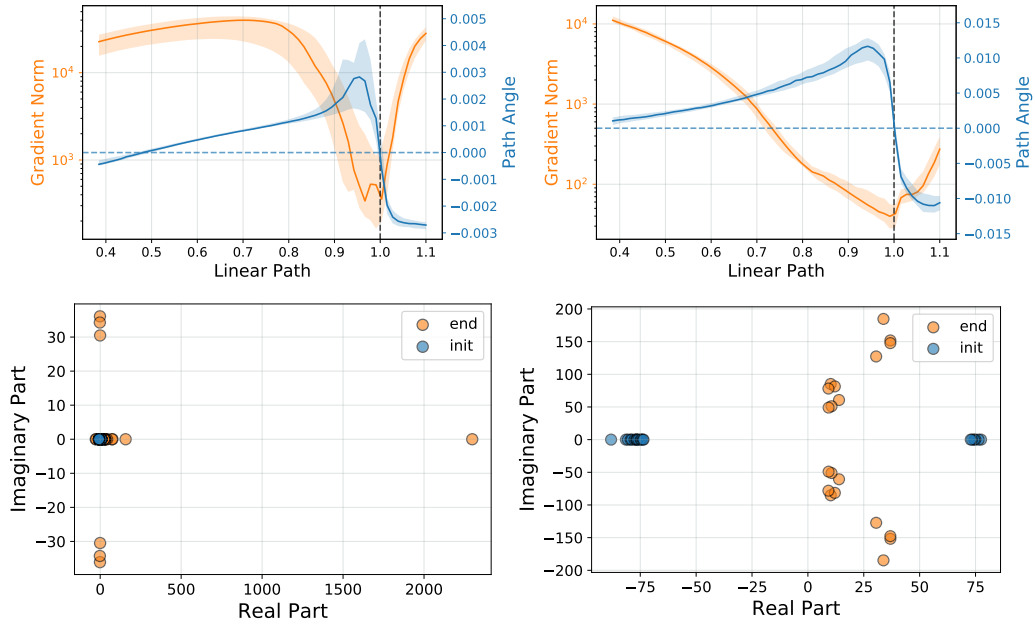


Figure D.1: The norm of gradient during training for the standard GAN objective. We observe that while extra-gradient reaches low norm which indicates that it has converged, the gradient descent on the contrary doesn't seem to converge.

3.5 Additional Results with Adam



(a) NSGAN on MNIST, IS: 8.95

(b) WGAN-GP on MNIST, IS: 9.30

Figure D.2: Path-angle and Eigenvalues computed on MNIST with Adam.

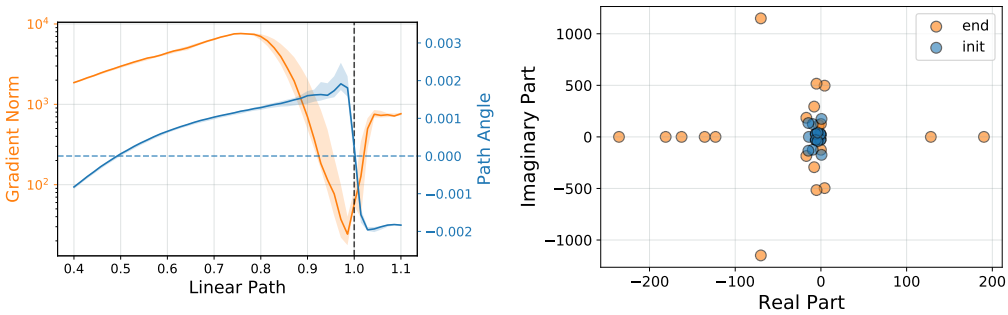


Figure D.3: Path-angle and Eigenvalues for NSGAN on CIFAR10 computed on CIFAR10 with Adam. We can see that the model has eigenvalues with negative real part, this means that we've actually reached an unstable point.