





Université de Montréal

Analyse de la réduction du chatoiement sur les images  
radar polarimétrique à l'aide des réseaux neuronaux à  
convolutions

par

**Mario Beaulieu**

Département de géographie  
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures et postdoctorales  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.)  
en Géographie

avril 2020



# Université de Montréal

Faculté des études supérieures et postdoctorales

Ce mémoire intitulé

## Analyse de la réduction du chatoiement sur les images radar polarimétrique à l'aide des réseaux neuronaux à convolutions

présenté par

**Mario Beaulieu**

a été évalué par un jury composé des personnes suivantes :

*Liliana Perez*

---

(président-rapporteur, UdeM)

*François Cavayas*

---

(directeur de recherche, UdeM)

*Samuel Foucher*

---

(co-directeur, CRIM)

*Saeid Homayouni*

---

(examineur externe, INRS)

Mémoire accepté le :

*Le 22 avril 2020*

---



# Sommaire

---

En raison de la nature cohérente du signal RADAR à synthèse d'ouverture (**RSO**), les images **RSO** polarimétriques (**RSOPOL**) sont affectées par le bruit de chatolement. L'effet du chatolement peut être sévère au point de rendre inutilisable la donnée **RSOPOL**. Ceci est particulièrement vrai pour les données à une vue qui souffrent d'un chatolement très intense. Un filtrage du bruit est nécessaire pour améliorer l'estimation des paramètres polarimétriques pouvant être calculés à partir de ce type de données. Cette opération constitue une étape importante dans le traitement et l'analyse des images **RSOPOL**.

Récemment une nouvelle approche est apparue en traitement de données visant la solution d'une multitude de problèmes dont le filtrage, la restauration d'images, la reconnaissance de la parole, la classification ou la segmentation d'images. Cette approche est l'apprentissage profond et les réseaux de neurones à convolution (**RNC**). Des travaux récents montrent que les **RNC** sont une alternative prometteuse pour le filtrage des images **RSO**. En effet par leur capacité d'apprendre un modèle optimal de filtrage, ils tendent à surpasser les approches classiques du filtrage sur les images **RSO**.

L'objectif de cette présente étude est d'analyser et d'évaluer l'efficacité du filtrage par **RNC** sur des données **RSOPOL** simulées et sur des images satellitaires **RSOPOL** RADARSAT-2, ALOS/PalSAR et GaoFen-3 acquises sur la région urbaine de San Francisco (Californie). Des modèles inspirés de l'architecture d'un **RNC** utilisé notamment en Super-résolution ont été adaptés pour le filtrage de la matrice de cohérence polarimétrique. L'effet de différents paramètres structuraux de l'architecture des **RNC** sur le filtrage ont été analysés, parmi ceux-ci on retrouve entre autres la profondeur du réseau (le nombre de couches empilées), la largeur du réseau (le nombre de filtres par couches convolutives) et la taille des filtres de la première couche convolutive.

L'apprentissage des modèles a été effectué par la rétropropagation du gradient de l'erreur en utilisant 3 ensembles de données qui simulent la polarimétrie une vue des diffuseurs selon les classes de Cloude-Pottier. Le premier ensemble ne comporte que des zones homogènes. Les deux derniers ensembles sont composés de simulations en patchwork dont l'intensité locale est simulée par des images de texture et de cibles ponctuelles ajoutées au patchwork dans le cas du dernier ensemble. Les performances des différents filtres par **RNC** ont été mesurées par des indicateurs comprenant l'erreur relative sur l'estimation de signatures polarimétriques et des paramètres de décomposition ainsi que des mesures de distorsion sur la récupération des détails importants et sur la conservation des cibles ponctuelles.

Les résultats montrent que le filtrage par **RNC** des données polarimétriques est soit équivalent ou nettement supérieur aux filtres conventionnellement utilisés en polarimétrie. Les résultats des modèles les plus profonds obtiennent les meilleures performances pour tous les indicateurs sur l'ensemble des données homogènes simulées. Dans le cas des données en patchwork, les résultats pour la restauration des détails sont nettement favorables au filtrage par **RNC** les plus profonds.

L'application du filtrage par **RNC** sur les images satellitaires RADARSAT-2, ALOS/PalSAR ainsi GaoFen-3 montre des résultats comparables ou supérieurs aux filtres conventionnels. Les meilleurs résultats ont été obtenus par le modèle à 5 couches cachées (si on ne compte pas la couche d'entrée et de sortie), avec 8 filtres  $3 \times 3$  par couche convolutive, sauf pour la couche d'entrée où la taille des filtres étaient de  $9 \times 9$ . Par contre, les données d'apprentissage doivent être bien ajustées à l'étendue des statistiques des images polarimétriques réelles pour obtenir de bon résultats. Ceci est surtout vrai au niveau de la modélisation des cibles ponctuelles dont la restauration semblent plus difficiles.

**Mots clés:** Apprentissage automatique, réseau de neurones à convolution, polarimétrie, RADAR à synthèse d'ouvertures, filtrage, chatouement, estimation polarimétrique, San Francisco.



## Summary

---

Due to the coherent nature of the Synthetic Aperture Radar (**SAR**) signal, polarimetric **SAR** (**POLSAR**) images are affected by speckle noise. The effect of speckle can be so severe as to render the **POLSAR** data unusable. This is especially true for single-look data that suffer from very intense speckle. Noise filtering is necessary to improve the estimation of polarimetric parameters that can be computed from this type of data. This is an important step in the processing and analysis of **POLSAR** images.

Recently, a new approach has emerged in data processing aimed at solving a multitude of problems including filtering, image restoration, speech recognition, classification or image segmentation. This approach is deep learning and convolutional neural networks (**CONVNET**). Recent works show that **CONVNET** are a promising alternative for filtering **SAR** images. Indeed, by their ability to learn an optimal filtering model only from the data, they tend to outperform classical approaches to filtering on **SAR** images.

The objective of this study is to analyze and evaluate the effectiveness of **CONVNET** filtering on simulated **POLSAR** data and on RADARSAT-2, ALOS/PalSAR and GaoFen-3 satellite images acquired over the San Francisco urban area (California). Models inspired by the architecture of a **CONVNET** used in particular in super-resolution have been adapted for the filtering of the polarimetric coherency matrix. The effect of different structural parameters of the **CONVNET** architecture on filtering were analyzed, among which are the depth of the neural network (the number of stacked layers), the width of the neural network (the number of filters per convoluted layer) and the size of the filters of the first convolution layer.

The models were learned by back-propagating the error gradient using 3 datasets that simulate single-look polarimetry of the scatterers according to Cloude-Pottier classes. The first dataset contains only homogeneous areas. The last two datasets consist of patchwork

simulations where local intensity is simulated by texture images and point target are added to the patchwork in the case of the last dataset. The performance of the different filters by **CONVNET** was measured by indicators including relative error on the estimation of polarimetric signatures and decomposition parameters as well as distortion measurements on the recovery of important details and on the conservation of point targets.

The results show that **CONVNET** filtering of polarimetric data is either equivalent or significantly superior to conventional polarimetric filters. The results of the deepest models obtain the best performance for all indicators over the simulated homogeneous dataset. In the case of patchwork dataset, the results for detail restoration are clearly favourable to the deepest **CONVNET** filtering.

The application of **CONVNET** filtering on RADARSAT-2, ALOS/PalSAR and GaoFen-3 satellite images shows results comparable or superior to conventional filters. The best results were obtained by the 5 hidden layers model (not counting the input and output layers), with 8 filters  $3 \times 3$  per convolutional layer, except for the input layer where the filter size was  $9 \times 9$ . On the other hand, the training data must be well adjusted to the statistical range of the real polarimetric images to obtain good results. This is especially true when modeling point targets that appear to be more difficult to restore.

**Key words:** Deep Learning, convolution neural network, polarimetry, synthetic aperture radar, filtering, speckle, polarimetric estimation, San Francisco.

# Table des matières

---

<b>Sommaire</b> .....	v
<b>Summary</b> .....	vii
<b>Liste des tableaux</b> .....	xv
<b>Liste des figures</b> .....	xix
<b>Remerciements</b> .....	xxxix
<b>Chapitre 1. INTRODUCTION</b> .....	1
1.1. Contexte .....	1
1.2. Le problème de la recherche .....	2
1.2.1. Les objectifs de l'étude .....	3
1.2.2. L'hypothèse de la recherche .....	4
1.2.3. Le plan de la thèse .....	5
<b>Chapitre 2. ÉTAT DE L'ART</b> .....	7
2.1. Les images polarimétriques .....	7
2.1.1. La polarimétrie RSO .....	8
2.1.2. Le chatoiement des images RSO et RSOPOL .....	15
2.1.3. La décomposition polarimétrique .....	18
2.2. Les réseaux neuronaux artificiel et l'apprentissage automatique .....	22
2.2.1. Les neurones artificiels .....	22
2.2.2. Les réseaux de neurones artificiels .....	23
2.2.3. Les réseaux de neurones à convolution .....	24

2.2.4.	Les réseaux de neurones pleinement convolutifs.....	31
2.3.	L'apprentissage des réseaux.....	33
2.3.1.	Optimisation par descente du gradient.....	34
2.3.2.	Optimisation par descente du gradient stochastique par mini-lots.....	34
2.3.3.	Optimisation par rétropropagation du gradient de l'erreur.....	35
2.3.4.	Exemples de Réseaux Neurons à convolution appliqués à l'imagerie RSO .	37
2.3.4.1.	Le filtre Dn-CNN.....	37
2.3.4.2.	Le filtre FFDNet.....	39
2.3.4.3.	Le filtre SAR-CNN.....	41
2.3.4.4.	Le filtre ID-CNN.....	43
2.3.4.5.	Le filtre SAR-DRN.....	45
2.3.5.	Exemples de Réseaux Neurons à convolution appliqués à l'imagerie <b>RSOPOL</b> .....	46
2.4.	Conclusion.....	48
<b>Chapitre 3. MÉTHODOLOGIE ET DONNÉES EXPÉRIMENTALES.....</b>		<b>49</b>
3.1.	Données synthétiques.....	49
3.1.1.	Génération de données polarimétriques.....	49
3.1.2.	Signatures polarimétriques tests.....	50
3.1.3.	Simulation de régions homogènes.....	51
3.1.4.	Simulation de régions hétérogènes (patchwork).....	55
3.1.4.1.	Étape 1 - Génération aléatoire d'une carte de régions homogènes.....	55
3.1.4.2.	Étape 2 - Génération de l'image étiquette polarimétrique.....	56
3.1.4.3.	Étape 3 - Génération de l'image polarimétrique une vue.....	57
3.2.	Données réelles.....	58
3.2.1.	Image du capteur AIRSAR.....	60
3.2.2.	Image du capteur RADARSAT-2.....	61
3.2.3.	Image du capteur ALOS-2/PALSAR-2.....	61

3.2.4.	Image du capteur GAOFEN-3.....	64
3.3.	Apprentissage du réseau .....	65
3.3.1.	Description du réseau expérimental .....	65
3.3.2.	Les paramètres fixes .....	66
3.3.3.	Les paramètres testés .....	69
3.3.4.	L'estimation des paramètres polarimétriques .....	69
3.3.5.	L'évaluation des filtrages .....	71
3.3.6.	La comparaison des filtrages .....	71
3.3.6.1.	Le filtre Boxcar .....	71
3.3.6.2.	Le filtre adaptatif de Lee affiné.....	72
3.4.	Le matériel .....	72
3.5.	Conclusion.....	74
<b>Chapitre 4. PRÉSENTATION ET DISCUSSION DES RÉSULTATS SUR LES APPRENTISSAGES.....</b>		<b>75</b>
4.1.	L'analyse de l'apprentissage.....	75
4.2.	L'analyse d'un modèle .....	83
4.2.1.	Les sorties des couches d'activation.....	83
4.2.2.	Les poids des filtres des convolutions.....	83
4.3.	L'estimation des paramètres de la matrice de cohérence sur les échantillons homogènes .....	87
4.4.	L'estimation du nombre de vue équivalent sur les échantillons homogènes.....	96
4.5.	L'estimation des paramètres de la décomposition H-A-Alpha sur les données homogènes.....	106
4.6.	La qualité de la reconstruction des détails.....	111
4.7.	La qualité de la restitution des cibles ponctuelles.....	118

4.8.	L'effet de la complexité de l'apprentissage sur l'estimation des paramètres polarimétriques .....	121
4.9.	Évaluation qualitative sur les images polarimétriques réelles .....	129
4.10.	Évaluation du nombre équivalent de vues sur les images polarimétriques réelles	139
4.11.	Conclusion.....	140
<b>Chapitre 5. CONCLUSION .....</b>		<b>143</b>
<b>Bibliographie.....</b>		<b>147</b>
<b>Annexe A. Données simulées homogènes .....</b>		<b>A-i</b>
A.1.	Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées homogènes des classes de diffuseurs polarimétriques.....	A-i
A.2.	Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes. ....	A-xviii
A.3.	Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes avec inclusion de cibles ponctuelles.....	A-xxxv
A.4.	Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées homogènes des classes de diffuseurs polarimétriques.....	A-lii
A.5.	Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes. ....	A-lxi

A.6. Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes avec inclusion de cibles ponctuelles..... A-lxx





## Liste des tableaux

---

2.1	Résultats comparatifs des deux apprentissages du DnCNN avec des méthodes conventionnelles de pointe. De manière globale, les méthodes par apprentissage profond performant mieux que les méthodes conventionnelles . . . . .	38
2.2	Résultats comparatifs du FFDNet avec des méthodes conventionnelles de pointe. De manière générale FFDNET surpasse les autres approches pour des niveaux de bruit élevés ou obtient des résultats comparables à DnCNN, avec lequel il partage une architecture semblable . . . . .	40
2.3	Résultats comparatifs du PSNR et du SSIM sur les images <b>RSO</b> simulées . . . . .	42
2.4	Résultats comparatifs du ENL et de l'index $\alpha\beta$ sur les images <b>RSO</b> réelles . . . . .	43
2.5	Configuration du réseau ID-CNN . . . . .	44
2.6	Résultats comparatifs de ID-CNN avec des méthodes traditionnelles en radar (Lee, Kuan) et conventionnelles de pointe (PPB, SAR-BM3D) et des approches en apprentissage machine profond (CNN=Dn-CNN, SAR-CNN. . . . .	44
2.7	Description de réseau SAR-DRN . . . . .	46
3.2	Nombre de vues équivalent ( <b>NEV</b> ) calculé sur quelques simulations . . . . .	55
3.3	Paramètres qui demeurent fixes . . . . .	68
3.4	Paramètres testés . . . . .	69
4.1	Ensemble des apprentissages produits à partir des <b>données simulées homogènes</b> en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par $m$ , le nombre de filtres par convolution est donné par $p$ et la taille de l'entrée de la première convolution est donnée par $n$ . . . . .	76

4.2	Ensemble des apprentissages produits à partir des <b>données simulées hétérogènes</b> en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par $m$ , le nombre de filtres par convolution est donné par $p$ et la taille de l'entrée de la première convolution est donnée par $n$ . . . . .	77
4.3	Ensemble des apprentissages produits à partir des <b>données simulées hétérogènes avec l'inclusion de cibles ponctuelles</b> en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par $m$ , le nombre de filtres par convolution est donné par $p$ et la taille de l'entrée de la première convolution est donnée par $n$ . . . . .	77
4.4	Les nombres équivalents de vue calculés sur les échantillons des trois diffuseurs typiques filtrés par le filtre <b>Refined LEE</b> ( $n = 11$ ) comparées aux valeurs du filtre <b>Boxcar</b> ( $n = 11$ ). Les <b>NEV</b> sont calculés sur les termes en puissance de la matrice de cohérence. . . . .	97
4.5	Les nombres équivalents de vue calculés sur les échantillons des trois diffuseurs typiques filtrés par le filtre de <b>VDPolSARF</b> ( $m = 5, p = 8, n = 11$ ) comparées aux valeurs du filtre <b>Boxcar</b> de même taille. Les <b>NEV</b> sont calculés sur les termes en puissance de la matrice de cohérence. . . . .	97
4.6	Biais relatifs sur l'estimé de l'entropie. Comparaison entre le modèle <b>VDPLSARF</b> ( $m = 5, p = 8, n = 15$ ) et le filtre <b>Boxcar</b> $15 \times 15$ . . . . .	106
4.7	Biais relatifs sur l'estimé de l'anisotropie. Comparaison entre le modèle <b>VDPLSARF</b> ( $m = 5, p = 8, n = 15$ ) et le filtre <b>Boxcar</b> $15 \times 15$ . . . . .	107
4.8	Biais relatifs sur l'estimé de l'angle $\bar{\alpha}$ . Comparaison entre le modèle <b>VDPLSARF</b> ( $m = 5, p = 8, n = 15$ ) et le filtre <b>Boxcar</b> $15 \times 15$ . . . . .	107
4.9	Mesure du nombre équivalent de vues sur une zone de type surfacique sur les termes en puissance de l'image RADARSAT2. La moyenne, l'écart-type et le <b>NEV</b> est calculé pour chaque filtre. . . . .	139

4.10	Mesure du nombre équivalent de vues équivalent sur une zone de type surfacique sur les termes en puissance de l'image ALOS2. La moyenne, l'écart-type et le <b>NEV</b> est calculé pour chaque filtre.....	140
4.11	Mesure du nombre de vues équivalent sur une zone de type surfacique sur les termes en puissance de l'image GAF3. La moyenne, l'écart-type et le <b>NEV</b> est calculé pour chaque filtre. ....	140



# Liste des figures

---

2.1	Géométrie de l'acquisition des images <b>RSO</b> et <b>RSOPOL</b> . . . . .	8
2.2	Le spectre électromagnétique et la transmittance de l'atmosphère entre l'espace et la terre. L'atmosphère est transparente dans le domaine des hyperfréquences (3GHz à 300 MHz). [30] . . . . .	9
2.3	Les bandes utilisées par les capteurs <b>RSO</b> et <b>RSOPOL</b> . [30] . . . . .	9
2.4	Résumé du procédé de la formation des images <b>RSO</b> à partir de la donnée brute (adapté de [27]). . . . .	10
2.5	La séquence des impulsions transmise et reçue par l'antenne d'un capteur <b>RSOPOL</b> . Dans le cas des plateformes satellitaires, la séquence transmission/réception n'est pas nécessairement immédiate. . . . .	10
2.6	Les quatre bandes polarimétriques d'une image en QuadPol RADARSAT-2 (San Francisco, Californie) . . . . .	11
2.7	Illustration du mécanisme de formation du chatolement. . . . .	16
2.8	Distribution des données <b>RSO</b> en fonction du nombre de vues $L_n$ . . . . .	17
2.9	Diagramme $H/\bar{\alpha}$ avec les régions qui définissent les limites des classes de diffuseurs de Cloude-Pottier . . . . .	20
2.10	Structure d'un neurone artificiel. Le neurone calcule la somme de ses entrées puis cette valeur passe par la fonction d'activation pour produire sa sortie. . . . .	23
2.11	Exemple d'un perceptron multicouches avec 5 couches. . . . .	24
2.12	Architecture du Neocognitron [10] . . . . .	25
2.13	Un exemple des réponses de la sortie de chaque couche du Neocognitron. L'entrée est détectée correctement comme le chiffre 8. [10] . . . . .	25

2.14	Architecture d'un réseau <b>LeNet</b> démontrant les structures de base des <b>RNC</b> .....	28
2.15	Illustration des opérations d'un bloc convolutif <i>conv + ReLU</i> . .....	29
2.16	Illustration de certaines fonctions d'activation non-linéaires. (a) Rectifier Linear Unit (ReLU), (b) tangente hyperbolique, (c) sigmoïde .....	30
2.17	Illustration de l'opération d'échantillonnage <i>max-pooling</i> .....	30
2.18	Illustration d'un bloc résiduel .....	30
2.19	Illustration de l'opération convolution versus l'opération déconvolution .....	32
2.20	Exemple d'un <b>RNPC</b> sans couche de ré-échantillonnage utilisé en superrésolution: VDSR ( <i>Very Deep Super Resolution</i> [16]). .....	33
2.21	L'architecture du réseau Dn-CNN. ....	37
2.22	L'architecture du réseau FFDNet. ....	40
2.23	Comparaison des vitesses d'exécution (en secondes) sur CPU (Single Thread). ...	41
2.24	L'architecture du réseau SAR-CNN. ....	42
2.25	L'architecture du réseau ID-CNN. ....	43
2.26	L'architecture du réseau SAR-DRN .....	46
2.27	L'entropie (rangée supérieure) et l'anisotropie (rangée inférieure) pour un filtre de Lee $7 \times 7$ et pour l'approche de (Foucher et al. [7]) .....	47
3.1	Diagramme $H/\bar{\alpha}$ de l'image AIRSAR de la région de San Francisco. (a) Les marqueurs numérotés de 1 à 14 représente la localisation sur la plan $H/\bar{\alpha}$ de chacune des matrices de cohérence sélectionnées pour les tests (voir Tab. 3.1). (b) Légende des couleurs associées à chacune des classes de diffuseur. ....	51
3.2	Exemples des simulations de chatoiement une vue ( $N = 1$ ) de données polarimétriques sur des régions homogènes. Le figures de (a) à (h) représentent le composé de Pauli de chacune des signatures polarimétriques sur le diagramme $H/\bar{\alpha}$ . ....	53
3.3	Schéma des étapes de production des simulations. ....	54

3.4	Exemples de texture de l'ensemble d'images <i>Describable Textures Dataset</i> [3] utilisés pour simuler la variation de la puissance de la rétrodiffusion sur les image <b>RSOPOL</b> .	57
3.5	Trois exemples de simulation polarimétrique avec des puissances déterminées par des images de texture et avec des cibles ponctuelles incrustées. La colonne (a) montre les étiquettes des classes de diffuseurs ( $C_z^+$ ). La classification $H/\bar{\alpha}$ des régions est donnée par la légende à droite; La colonne (b) présente le composé de Pauli des images de signatures polarimétriques ( $S_p^+$ ) sans chatolement; La colonne (c) présente le composé de Pauli des images une vue ( $\bar{S}_p^+$ ).	58
3.6	Région de la ville de San Francisco, Californie, (ÉU). (a) Plan de la ville (source Google). (b) Vue satellite de la région en couleur visible (source Google). (c) Le rectangle représente la région d'intérêt qui est approximativement de la taille de l'image acquise en 1999 par le capteur aéroporté AIRSAR.	59
3.7	Image AIRSAR, Bande L, ( $S_{HH} - S_{VV}$ , $\sqrt{2}S_{HV}$ , $S_{HH} + S_{VV}$ )	60
3.8	Image RADARSAT-2, Bande C, ( $S_{HH} - S_{VV}$ , $\sqrt{2}S_{HV}$ , $S_{HH} + S_{VV}$ )	62
3.9	Image ALOS-2, Bande L, ( $S_{HH} - S_{VV}$ , $\sqrt{2}S_{HV}$ , $S_{HH} + S_{VV}$ )	63
3.10	Image GAF-3, Bande C, ( $S_{HH} - S_{VV}$ , $\sqrt{2}S_{HV}$ , $S_{HH} + S_{VV}$ )	64
3.11	Schéma de l'architecture du réseau à convolution utilisé pour les expérimentations, où $p$ est le nombre de filtres par couches, $n$ est la taille des filtres de la première couche et $m$ est le nombre de répétition du bloc pour le calcul du résidu. Le symbole $\oplus$ est l'opération de calcul du résidu.	65
3.12	Distribution $H/\bar{\alpha}$ des matrices de cohérence pour la génération des données d'apprentissage. Les couleurs représentent les différentes classes de diffuseur.	67
3.13	Distribution $H/A$ des matrices de cohérence pour la génération des données d'apprentissage. Les couleurs représentent les différentes classes de diffuseur.	68
4.1	Temps d'exécution des apprentissages de l'ensemble des $m = \{0,1,3,5\}$ et $p = 8$ . La figure de gauche présente les temps pour l'apprentissage sur les données homogènes. La figure centrale montre le temps pour l'apprentissage sur les données hétérogènes et la figure de	

	gauche montre celui pour les données hétérogènes avec inclusion de cibles ponctuelles. Nombre totale d'apprentissage = 84. Nombre d'heures total = 1003.4 heures. Temps moyen d'un apprentissage = 11.9 heures $\pm$ 4.4 heures. ....	78
4.2	Exemple de courbes d'apprentissage sur la famille $m = 5, p = 8$ montrant l'erreur d'apprentissage en fonction du nombre d'époques ( <b>vert</b> = entraînement, <b>rouge</b> = validation, <b>bleu</b> = meilleurs modèles). La figure de gauche présente les résultats relatif à l'apprentissage des modèles sur l'ensemble de données homogènes. La figure centrale montre les résultats pour l'apprentissage sur les données hétérogènes et la figure de gauche montre celui pour les données hétérogènes avec inclusion de cibles ponctuelles.	79
4.3	L'erreur de l'apprentissage des meilleurs modèles sur l'ensemble de validation pour les trois expériences: homogène ( <b>rouge</b> ), hétérogène ( <b>cyan</b> ) et hétérogène avec inclusion de cibles ponctuelles ( <b>vert</b> ) .....	80
4.4	Exemple de résultats sur un mini-lot de 16 échantillons extraits de l'ensemble de validation. Les trois figures présentent pour chaque type d'apprentissage en colonne: 1) l'image caractéristique simulée avec chatoiement $N = 1$ , 2) les images étiquettes, 3) les prédictions et 4) le filtre <b>Boxcar</b> $5 \times 5$ . ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ) .....	82
4.5	Exemple des activations du réseau à trois couche $m = 1, p = 8, n = 15$ . La Figure (a) est l'image d'entrée des matrices de cohérence visualisée par le <i>span</i> . La Figure (b) montre les 8 activations de la première couche. La Figure (c) montre les 8 activations du bloc convolutif. La Figure (d) présente la carte des résidus et la Figure (e) montre le résultat de l'opération résiduel $\oplus$ , ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). .....	84
4.6	Visualisation des filtres $15 \times 15$ de la première couche convolutive. ....	86
4.7	Exemple de la légende utilisée pour identifier les différents algorithmes de filtrage. ....	88
4.8	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . ....	90



4.9	<b>Diffuseurs double bond.</b> Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	91
4.10	<b>Diffuseurs volumétriques.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	92
4.11	<b>Diffuseurs volumétriques.</b> Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	93
4.12	<b>Diffuseurs surfaciques.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	94
4.13	<b>Diffuseurs surfaciques.</b> Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	95
4.14	<b>Diffuseurs double bond.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ entraînés sur les données d'apprentissage homogènes. . . . .	98
4.15	<b>Diffuseurs volumétriques.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ entraînés sur les données d'apprentissage homogènes. . . . .	99
4.16	<b>Diffuseurs surfaciques.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ entraînés sur les données d'apprentissage homogènes. . . . .	99

4.17	<b>Diffuseurs double bond.</b> Comparaison entre les différents filtrages - Partie 1. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	100
4.18	<b>Diffuseurs double bond.</b> Comparaison entre les différents filtrages - Partie 2. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	101
4.19	<b>Diffuseurs volumétriques.</b> Comparaison entre les différents filtrages - Partie 1. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	102
4.20	<b>Diffuseurs volumétriques.</b> Comparaison entre les différents filtrages - Partie 2. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	103
4.21	<b>Diffuseurs surfaciques.</b> Comparaison entre les différents filtrages - Partie 1. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	104
4.22	<b>Diffuseurs surfaciques.</b> Comparaison entre les différents filtrages - Partie 2. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .....	105
4.23	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . .....	107
4.24	<b>Diffuseurs double bond.</b> Écart-type des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . .....	108
4.25	<b>Diffuseurs volumétriques.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . .....	108
4.26	<b>Diffuseurs volumétriques.</b> Écart-type des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . .....	109
4.27	<b>Diffuseurs surfaciques.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . .....	109

4.28	<b>Diffuseurs surfaciques.</b> Écart-type des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ . . . . .	110
4.29	<b>Image simulée 0.</b> Comparaison entre les différents filtrages - Partie 1. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ . . . . .	112
4.30	<b>Image simulée 0.</b> Comparaison entre les différents filtrages - Partie 2. (Improved Lee = Sigma Lee) $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ . . . . .	113
4.31	<b>Image simulée 3.</b> Comparaison entre les différents filtrages - Partie 1. $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ . . . . .	114
4.32	<b>Image simulée 3.</b> Comparaison entre les différents filtrages - Partie 2. (Improved Lee = Sigma Lee) $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ . . . . .	115
4.33	<b>Image simulée 0).</b> Moyenne du <b>PSNR</b> et du <b>SSIM</b> sur l'image du <b>span</b> et sur l'image de Sobel du <b>span</b> . Les résultats sont présentés en fonction de la taille $n$ de l'entrée des filtres. . . . .	116
4.34	<b>Image simulée 3.</b> Moyenne du <b>PSNR</b> et du <b>SSIM</b> sur l'image du <b>span</b> et sur l'image de Sobel du <b>span</b> . Les résultats sont présentés en fonction de la taille $n$ des filtres. . . . .	117
4.35	Schéma représentant les changements possibles de la géométrie de la cible. . . . .	118
4.36	<b>Image simulée 0.</b> Résultats sur la préservation des cibles ponctuelles . . . . .	119
4.37	<b>Image simulée 3.</b> Résultats sur la préservation des cibles ponctuelles . . . . .	120
4.38	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données homogènes). . . . .	122
4.39	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes). . . . .	123

4.40	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles) . . . . .	124
4.41	<b>Diffuseurs double bond.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données homogènes). . . . .	125
4.42	<b>Diffuseurs double bond.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes). . . . .	125
4.43	<b>Diffuseurs double bond.</b> Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles). . . . .	126
4.44	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données homogènes). . . . .	126
4.45	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes). . . . .	127
4.46	<b>Diffuseurs double bond.</b> Moyenne des erreurs relatives (%) des estimés de la décomposition $H/A/\bar{\alpha}$ en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles $m = \{0,1,3,5\}$ et $p = 8$ (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles). . . . .	127

4.47	<b>Diffuseurs double bond.</b> Comparaison entre les différents apprentissages pour le modèle <b>VDPolSARF</b> ( $m = 5, p = 8$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ) . . . . .	128
4.48	<b>Comparaison entre les filtrages de l'image RADARSAT2 (1):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	130
4.49	<b>Comparaison entre les filtrages de l'image RADARSAT2 (2):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	131
4.50	<b>Comparaison entre les filtrages de l'image ALOS2 (1):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	132
4.51	<b>Comparaison entre les filtrages de l'image ALOS2 (2):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	133
4.52	<b>Comparaison entre les filtrages de l'image GAF3 (1):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	134
4.53	<b>Comparaison entre les filtrages de l'image GAF3 (2):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	135
4.54	<b>Comparaison entre les filtrages de l'image RADARSAT2 (1):</b> (a) image originale, (b) modèle <b>VDPolSARF</b> ( $m=5, p=8, n=9$ ) apprentissage sur les données	

	hétérogènes, (c) modèle <b>VDPolSARF</b> (m=5, p=8, n=9) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> (9 × 9). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	136
4.55	<b>Comparaison entre les filtrages de l'image ALOS2 (1)</b> : (a) image originale, (b) modèle <b>VDPolSARF</b> (m=5, p=8, n=9) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> (m=5, p=8, n=9) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> (9 × 9). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	137
4.56	<b>Comparaison entre les filtrages de l'image GAF3 (1)</b> : (a) image originale, (b) modèle <b>VDPolSARF</b> (m=5, p=8, n=9) apprentissage sur les données hétérogènes, (c) modèle <b>VDPolSARF</b> (m=5, p=8, n=9) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre <b>Sigma Lee</b> (9 × 9). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ). . . . .	138
5.1	Exemple d'une architecture permettant de filtrer indépendamment les termes copolaires et les termes croisés de la matrice de cohérence. La branche du haut modélise les 3 termes copolaires et la branche du bas modélise chacune des parties des nombres complexes des termes croisés. .	144
5.2	Résultats comparatifs en les modèles antécédents et le résultat de la nouvelle architecture. En-haut à gauche, l'image ALOS2 une vue. En-haut à droite, le résultat du modèle <b>VDPolSARF</b> entraîné sur les données hétérogènes. En-bas à gauche, le résultat du modèle <b>VDPolSARF</b> entraîné sur les données hétérogènes avec inclusion de cibles ponctuelles. Et finalement, en-bas à droite, le résultat du modèle bi-céphale <b>VDPolSARF-Bi</b> entraîné sur les données hétérogènes avec inclusion de cibles ponctuelles. ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ) . . . . .	145







# Remerciements

---

Ténacité et Persévérance sont filles de  
Réussite.

---

*Les adages français - XVIe siècle*

Je tiens avant tout à exprimer ma gratitude à mon directeur de recherche François Cavayas, pour m'avoir fait découvrir la télédétection il y a déjà plus de vingt années. Au travers de très nombreux projets, tu m'as permis d'explorer un domaine fascinant et façonné mes premières expériences de travail qui ont influencé mon parcours professionnel. Voilà 3 années, nous nous sommes revus et je t'ai exprimé le désir de reprendre le flambeau et de recommencer les études graduées! Tu as accueilli ma proposition avec une grande ouverture. Je t'en serai toujours reconnaissant.

Je tiens à remercier tout autant mon codirecteur de recherche Samuel Foucher qui m'a transmis son enthousiasme et une partie de son expertise de la polarimétrie radar. Sans ton savoir-faire et tes conseils, ce projet aurait été nettement plus laborieux. Merci de ton temps offert gratuitement, malgré tous les aléas du travail et tous les projets qu'on se devait d'accomplir en surplus.

Je remercie Liliana Perez et Saeid Homayouni pour m'avoir fait l'honneur d'accepter la charge de président-rapporteur et d'examineur externe de mon jury. Je vous exprime toute ma gratitude pour votre temps et les corrections qui m'ont permis d'améliorer la qualité du mémoire.

Je remercie mes collègues Mohamed Dahmane, Pierre-Luc Saint-Charles qui m'ont donné de bons conseils ou aider indirectement par leurs travaux à l'accomplissement de ce mémoire et à Justine Boulent pour ses encouragements et son aide pour la correction du français. Je remercie mon employeur, le Centre de recherche Informatique de Montréal pour m'avoir permis d'utiliser des équipements de pointe pour effectuer ces travaux.

Je remercie aussi tous ceux qui ont pensé que je ne terminerais pas, vous m'avez donné de la ténacité dans l'adversité.

En dernier lieu, mais non les moindres, je remercie ma conjointe Corine de son soutien indéfectible pour ce plan fou que j'ai eu, de sa confiance en sa réussite, de sa patience et de son endurance. Et merci à mes deux enfants, Mattéo et Gabriel, pour tous vos sacrifices et votre patience.

Mario Beaulieu (avril, 2020)



# Chapitre 1

---

## INTRODUCTION

### 1.1. Contexte

Les systèmes de **RADAR** à synthèse d'ouverture (**RSO**) sont des capteurs actifs qui opèrent dans le domaine des micro-ondes du spectre électromagnétique. Ils fournissent leur propre source d'illumination qui leur permet d'acquérir des images de la surface de la Terre jour et nuit, ainsi que sous presque toutes les conditions météorologiques. Ils émettent du rayonnement électromagnétique vers le sol à partir de leur antenne et mesurent la réflectivité du signal rétrodiffusé. En comparaison avec les données optiques en télédétection, les données **RSO** présentent des avantages uniques en termes d'acquisition de données à haute résolution spatiale et temporelle de la surface terrestre.

Au cours de la dernière décennie, un nombre croissant de satellites **RSO** opérant avec une capacité polarimétrique à différentes fréquences ont été placés en orbite. Ces **RSO** polarimétriques (**RSOPOL**) sont des systèmes **RSO** qui mesurent l'intensité et la phase du signal rétrodiffusé dans toutes les combinaisons possibles de polarisation linéaire horizontale et verticale en émission et réception (HH, HV, VH, VV). Ces mesures permettent de mieux caractériser le type de rétrodiffusion des cibles observées sur la surface terrestre: diffusion surfacique, diffusion volumétrique, diffusion en double ou multiple bond, etc. Ces différents systèmes **RSOPOL** multicanaux (polarisation quadruple ou QuadPol) tels que le **RSO** du satellite japonais ALOS/PaLSAR opérant dans la bande L, celui du satellite canadien RADARSAT-2 opérant en bande C et celui du satellite allemand TerraSAR-X fonctionnant sur la bande X ont engendré un foisonnement de la recherche en télédétection sur les images **RSOPOL** et les façons dont les mêmes objets réagissent face à des ondes de fréquence et

de polarisation différentes. Leur utilisation est maintenant bien ancrée dans de nombreuses applications reliées à l’observation de la Terre, y compris la cartographie de la glace de mer et le routage des navires, la détection des icebergs, la surveillance marine de la pollution et des navires, la surveillance de la défense terrestre et l’identification des cibles, le suivi des cultures agricoles, le suivi des milieux humides, la cartographie géologique, la surveillance des mines, la mise à jour de la cartographie de l’utilisation des sols et des cartes topographiques.

## 1.2. Le problème de la recherche

En raison de la nature cohérente du signal **RSO**, les données **RSOPOL** sont affectées par le bruit de chatoiement (*speckle*) qui donne un aspect sel et poivre aux images. L’effet du bruit peut être sévère au point de rendre inutilisable la donnée polarimétrique. Ceci est vrai en particulier pour les données à une vue (*single look*) qui souffrent d’un chatoiement très intense. Par conséquent, les résultats des algorithmes de discrimination de cibles, de classification, de segmentation d’images ainsi que de détection des contours sont à leur tour affectés par le chatoiement. Une réduction du bruit est nécessaire pour améliorer l’estimation des paramètres polarimétriques pouvant être calculés à partir des images. Cette opération constitue, dans la plupart des applications, une étape obligatoire dans les pipelines de traitement et d’analyse des images **RSOPOL**.

Cependant plusieurs chercheurs ont mis en évidence que le filtrage des données en Quad-Pol n’est pas une entreprise facile. Quelles sont les données à filtrer (brutes ou transformées) [43], quelles seraient les propriétés des filtres pour préserver les relations entre les canaux de polarisation et du coup les propriétés polarimétriques des objets visés [42], [25], [26], [29]. Les filtres polarimétriques proposés jusqu’à présent malgré leur sophistication donnent des résultats contradictoires. Foucher et López-Martinez [9] ont démontré que le choix de l’algorithme de filtrage des images **RSOPOL** peut influencer les résultats des applications telles que les classifications et les décompositions polarimétriques qui dépendent étroitement de la conservation de l’information polarimétrique. Le filtrage peut aussi influencer la conservation de détails importants comme les cibles ponctuelles.

Récemment une nouvelle approche est apparue en traitement de données (images, parole, textes) visant la solution d’une multitude de problèmes dont le filtrage, la restauration d’images, la reconnaissance de la parole, la classification ou la segmentation d’images. Cette

approche est l'apprentissage profond et les réseaux de neurones à convolution (**RNC**) [21]. Les **RNC** sont un type spécialisé de réseau neuronal pour le traitement des données qui ont une corrélation spatiale sur une grille entre les points autour d'un voisinage, par exemple les images sont considérées comme une grille 2-D de pixels. De plus, ils utilisent la convolution au lieu de la multiplication matricielle générale dans au moins une de leurs couches neuronales.

Des travaux récents montrent que les **RNC** sont une alternative prometteuse pour le filtrage des images **RSO**. En effet par leur capacité d'apprendre un modèle optimal de filtrage, ils tendent à surpasser les approches classiques sur les images **RSO** en intensité [2], [44]. Quelques travaux pointent dans la même direction sur le filtrage par **RNC** des images **RSOPOL** [7]. Malgré les bons résultats obtenus par ces approches de filtrage par **RNC**, des questions fondamentales demeurent ouvertes sur leurs application sur les images en QuadPol:

- (1) Peut-on restaurer à la fois l'information polarimétrique et spatiale d'une image **RSOPOL** une vue?
- (2) Quels sont les biais introduits par le filtrage sur les paramètres polarimétriques?
- (3) Quelle est la performance du filtrage sur les cibles ponctuelles?
- (4) Quel est l'impact de l'architecture du **RNC** sur le filtrage par rapport aux interrogations précédentes (1), (2) et (3)?
- (5) Comment se comparent les **RNC** aux approches classiques pour le filtrage des images polarimétriques?

### 1.2.1. Les objectifs de l'étude

Le premier objectif de cette étude est d'analyser et d'évaluer l'efficacité du filtrage par **RNC** des images en QuadPol une vue. Les performances seront mesurées par différents indicateurs comprenant l'erreur relative sur l'estimation de signatures polarimétriques et des paramètres de décomposition ainsi que des mesures de distorsion sur la récupération des détails importants et la conservation des cibles ponctuelles.

Pour atteindre cet objectif nous devons poursuivre plusieurs sous-objectifs d'ordre technique:

- (1) L'apprentissage d'un **RNC** demande tout à bord de construire un ensemble de données polarimétriques à partir d'images simulées une vue afin de créer des paires d'images pour l'apprentissage.
- (2) Les approches **RNC** et le pipeline de l'apprentissage doivent être programmés à l'aide d'une librairie spécialement dédiée au développement d'algorithmes d'apprentissage profond afin de fonctionner rapidement sur les **GPU** . Dans notre cas nous utiliserons la librairie Pytorch.
- (3) Les **RNC** programmés seront simples afin de pouvoir mesurer facilement l'effet de différents paramètres structuraux de l'architecture sur le filtrage: la profondeur du **RNC** (c'est-à-dire le nombre de couches), la largeur (le nombre de filtres convolutifs par couches) et la taille des filtres.

Le second objectif est d'analyser et d'évaluer le filtrage par **RNC** entraîné sur des données polarimétriques simulées, sur des images réelles en QuadPol RADARSAT-2, ALOS/PalSAR et GaoFen-3.

### 1.2.2. L'hypothèse de la recherche

Notre hypothèse principale de travail suppose que les filtrages polarimétriques par **RNC** permettent d'atténuer nettement le chatoiement tout en préservant adéquatement les détails pertinents à la condition que le **RNC** soit correctement optimisé.

On peut décomposer l'hypothèse principale en points secondaires:

- Les **RNC** peuvent apprendre la représentation d'un modèle du filtrage du chatoiement des données polarimétriques sans connaissance a priori;
- Les modèles de filtrage appris sont conservateurs du point de vue radiométrique. Ils se comportent comme des filtres avec des fenêtres d'estimation larges afin de préserver les propriétés polarimétriques sur les cibles homogènes étendues;
- Les modèles de filtrage appris sont adaptatifs du point de vue spatial. Ils se comportent comme des filtres avec des fenêtres d'estimation étroites afin de préserver les informations hautes fréquences importantes comme les contours, les textures et les cibles ponctuelles;
- Les **RNC** sont des estimateurs consistants, c'est-à-dire que l'erreur faite sur les estimations diminue lorsque la taille des l'échantillons augmente.

### 1.2.3. Le plan de la thèse

Le mémoire se présente en 4 chapitres excluant l'introduction.

Le chapitre 2 présente une revue de la théorie de la polarimétrie appliquée à l'image satellite. Il aborde le problème fondamental du chatoiement et les notions de décompositions polarimétriques. En seconde partie, il jette les bases théoriques pour la compréhension des réseaux de neurones à convolution appliqués à l'analyse des images.

Le chapitre 3 présente la méthodologie pour la production des données nécessaires à l'apprentissage. Il décrit aussi le plan pour générer les différentes configurations des **RNC** à tester. Il présente la méthodologie utilisée pour l'évaluation des modèles générés par les apprentissages. Il donne une description des données polarimétriques réelles employées et finalement, il décrit les outils matériels nécessaires à la réalisation du projet.

Le chapitre 4 présente les résultats en fonction du plan des expérimentations. Il présente la comparaison du filtrage par **RNC** avec d'autres filtres couramment utilisés en polarimétrie. Cette comparaison est basée sur des critères de performances propres à la réduction de chatoiement d'images **RSOPOL**. Elle se fait à la fois sur les images simulées et réelles. Le chapitre aborde aussi les problèmes d'apprentissage rencontrés.

Le dernier chapitre conclut le mémoire en discutant des limites de l'approche proposée et en suggérant des pistes pour améliorer les résultats.





# Chapitre 2

---

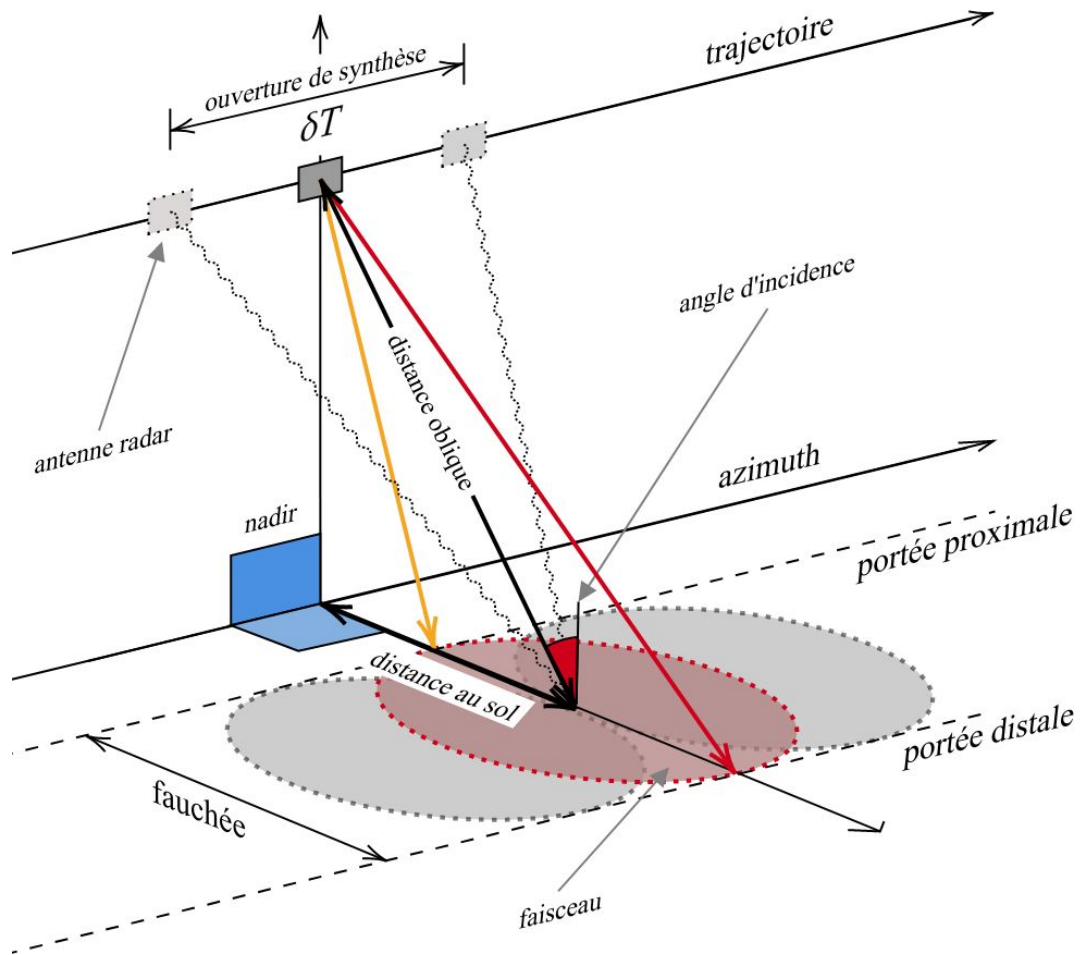
## ÉTAT DE L'ART

Ce chapitre présente les fondements théoriques sur lesquelles les objectifs du projet sont établis. Il présente en premier lieu une revue de la théorie de la polarimétrie appliquée à l'image satellite. Il aborde le problème fondamental du chatoiement et les notions de décompositions polarimétriques. Et en deuxième lieu il jette les bases théoriques pour la compréhension des réseaux de neurones à convolution appliqués à l'analyse des images.

### 2.1. Les images polarimétriques

Les capteurs **RSOPOL** sont des systèmes radar à balayage latéral. Ils sont capables d'obtenir des résolutions spatiales métriques à partir des satellites selon l'axe azimutal, malgré la dimension limitée de leur antenne physique, en synthétisant une longue antenne virtuelle combinant de multiples retours des cibles le long de la trajectoire azimutale (Fig. 2.1). La combinaison des retours s'effectue en exploitant l'effet Doppler dû au déplacement relatif du capteur par rapport aux cibles observées. Ces capteurs sont indépendants de l'illumination solaire et peuvent acquérir des images de jour et de nuit. Ils opèrent dans la région des micro-ondes du spectre électromagnétique à laquelle l'atmosphère est transparente, évitant ainsi les effets des nuages, des brouillards et de la pluie (Fig. 2.2 et Fig. 2.3). Ils observent la surface de la Terre avec un rayonnement électromagnétique cohérent sous la forme d'impulsions et reçoivent sa rétrodiffusion du sol. L'image **RSO** est synthétisée à l'aide de procédés de traitement d'image pour recréer une représentation 2-D de la réflectivité de la surface du sol (Fig. 2.4).

En mode transmission, les capteurs **RSOPOL** émettent des ondes polarisées linéairement selon des axes orthogonaux: soit H pour la polarisation horizontale et V pour la polarisation

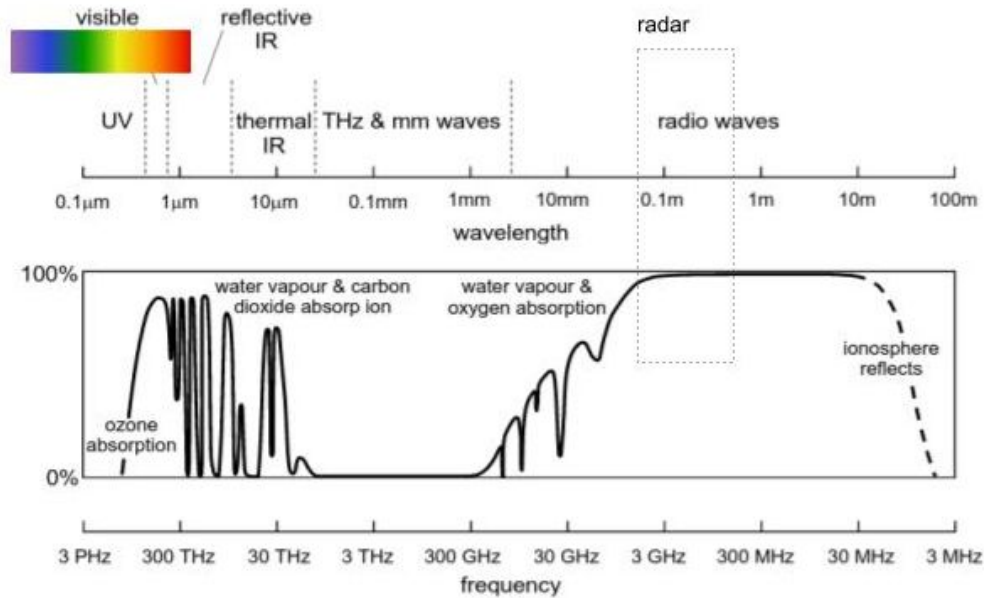


**Fig. 2.1.** Géométrie de l'acquisition des images **RSO** et **RSOPOL**.

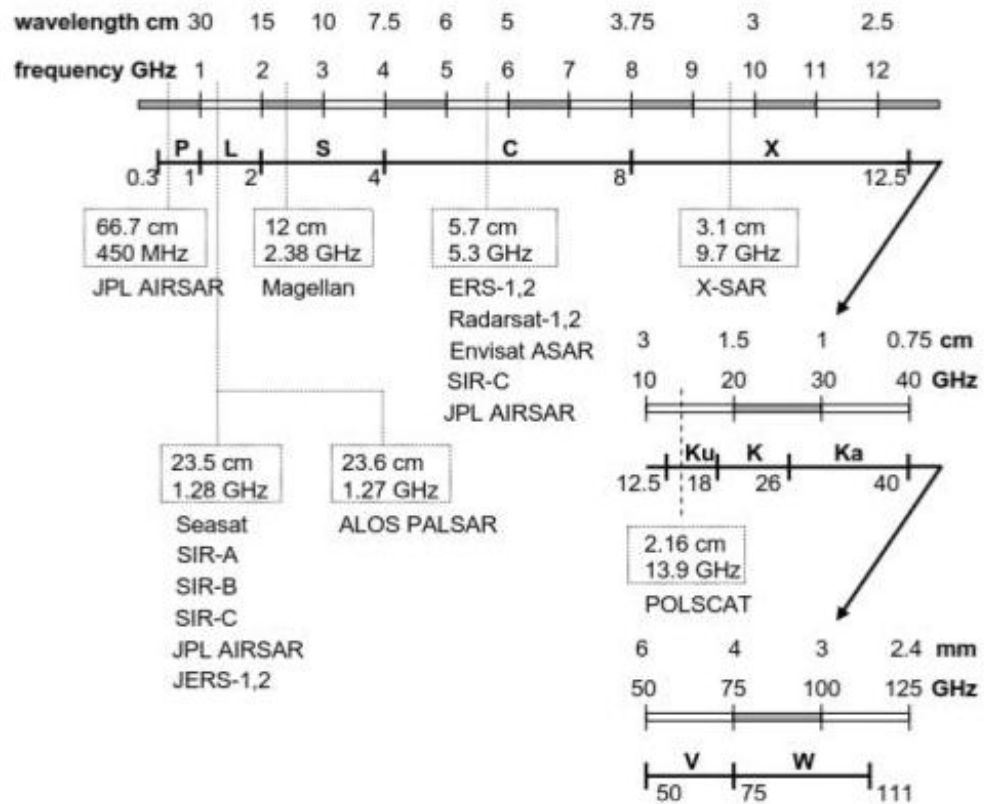
verticale. En mode réception, ils mesurent la rétrodiffusion polarisée horizontalement (H) et verticalement (V) (Fig. 2.5). Une image **RSOPOL** est formée en mesurant l'énergie rétrodiffusée selon les 4 combinaisons réception-transmission possibles, soient  $HH$ ,  $HV$ ,  $HV$ ,  $VV$  (Fig. 2.6).

### 2.1.1. La polarimétrie **RSO**

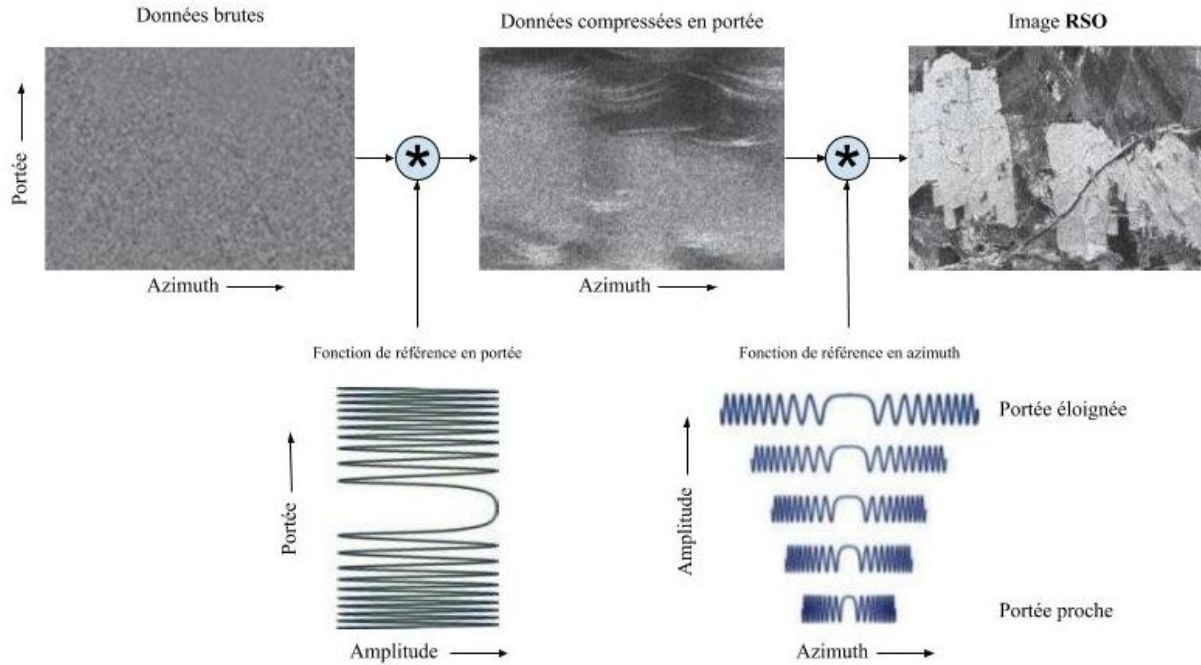
La polarimétrie **RSO** étudie les propriétés d'une cible en fonction des caractéristiques du comportement de l'onde électromagnétique (**OE**) rétrodiffusée par celle-ci. Les facteurs qui affectent la mesure de l'énergie rétrodiffusée au niveau du capteur sont: la fréquence de l'onde, la polarisation, l'angle d'incidence et l'orientation de la visée. Les facteurs qui influencent la mesure au niveau de la cible sont: les propriétés géométriques (structure, orientation) et géophysiques (rugosité, humidité du sol, conductivité du médium).



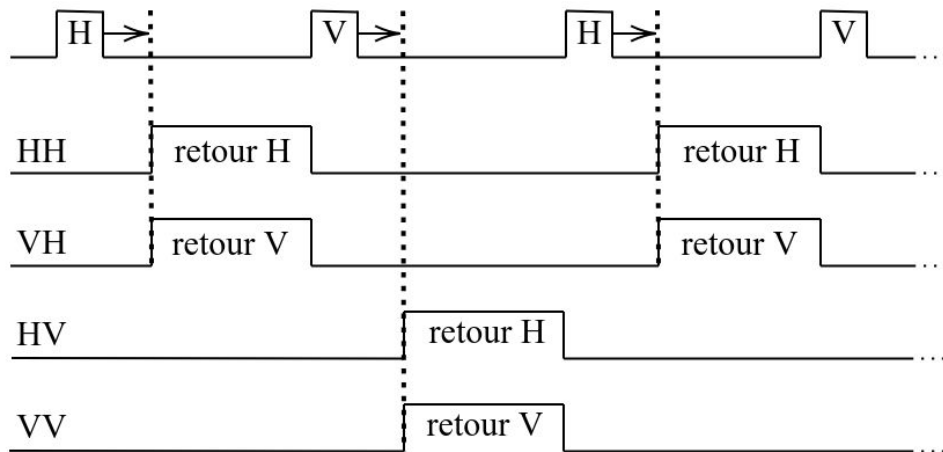
**Fig. 2.2.** Le spectre électromagnétique et la transmittance de l'atmosphère entre l'espace et la terre. L'atmosphère est transparente dans le domaine des hyperfréquences (3GHz à 300 MHz). [30]



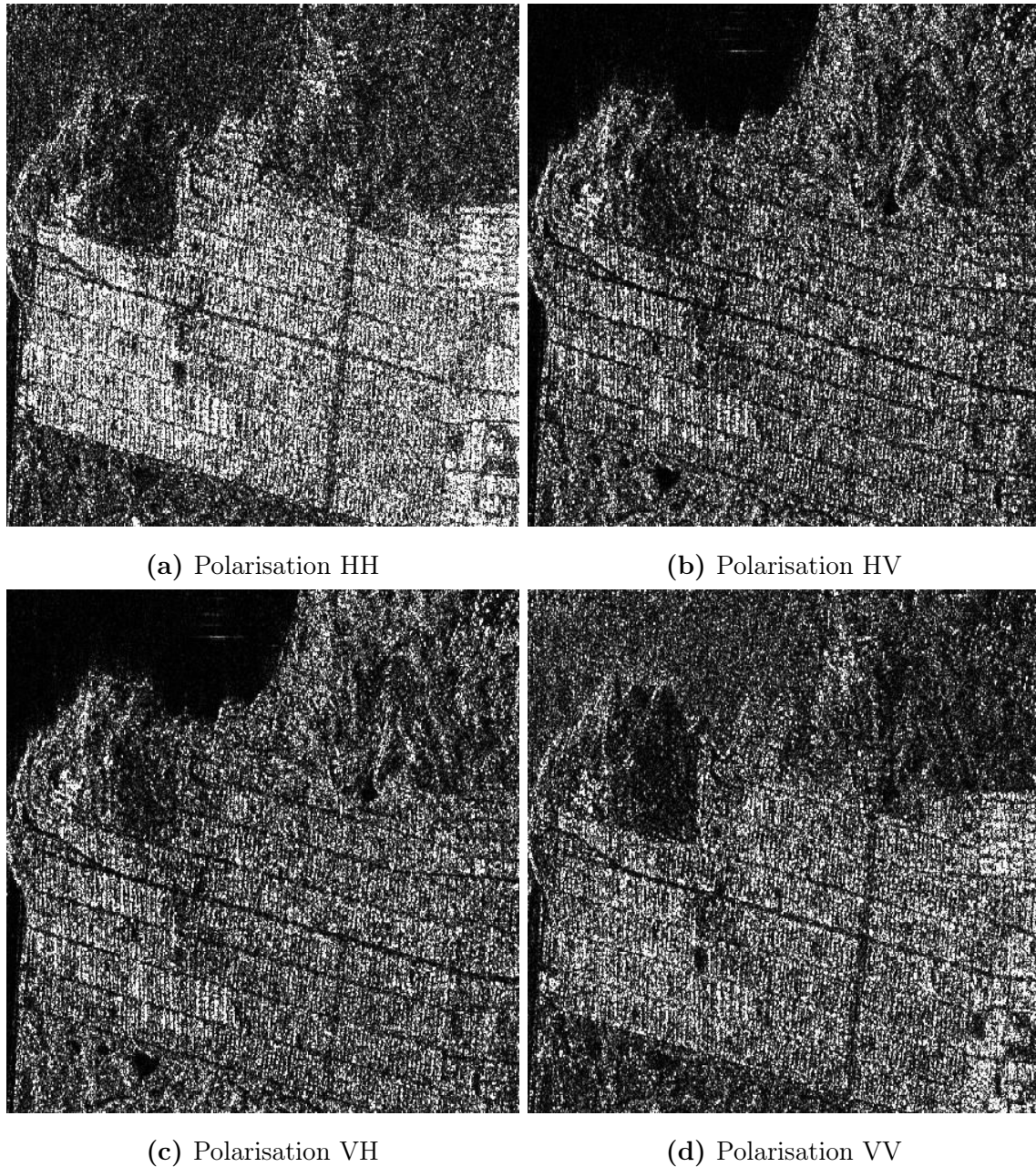
**Fig. 2.3.** Les bandes utilisées par les capteurs RSO et RSOPOL. [30]



**Fig. 2.4.** Résumé du procédé de la formation des images **RSO** à partir de la donnée brute (adapté de [27]).



**Fig. 2.5.** La séquence des impulsions transmise et reçue par l'antenne d'un capteur **RSOPOL**. Dans le cas des plateformes satellitaires, la séquence transmission/réception n'est pas nécessairement immédiate.



**Fig. 2.6.** Les quatre bandes polarimétriques d'une image en QuadPol RADARSAT-2 (San Francisco, Californie)

De manière générale, la modélisation des interactions de l'OE polarisée avec un milieu peut être décrite par la projection du champ électrique  $\vec{E}$  sur un plan perpendiculaire à la direction de propagation de celle-ci. Le vecteur du champ  $\vec{E}$  dessine ainsi une ellipse en fonction du temps: l'ellipse de polarisation. Cette dernière peut à son tour être représentée

par une combinaison linéaire de deux états de polarisation d'amplitude  $(E_x, E_y)$  avec la base des vecteurs d'états orthogonaux  $\{\vec{e}_x, \vec{e}_y\}$

$$\vec{E} = E_x \vec{e}_x + E_y \vec{e}_y \quad (2.1.1)$$

Cette formulation (Eq. 2.1.1) peut se réécrire selon le vecteur de Jones en coordonnées polaires

$$E_{xy} = \begin{bmatrix} E_x \\ E_y \end{bmatrix} = \begin{bmatrix} |E_x| e^{i\phi_x} \\ |E_y| e^{i\phi_y} \end{bmatrix} \quad (2.1.2)$$

Dans notre cas, les états représentés par la base  $\{\vec{e}_x, \vec{e}_y\}$  sont assimilés aux états de polarisation linéaire (H, V). Nous ne décrivons pas le cas de la polarisation circulaire. Nous pouvons définir la matrice  $[S_2]$  comme la matrice de rétrodiffusion qui décrit la façon dont l'OE transmise a été dépolarisée par les cibles au sol et mesurée par le capteur:

$$[S_2] = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} \quad (2.1.3)$$

Le vecteur de Jones se formule en tenant compte de la matrice de rétrodiffusion comme suit:

$$\begin{bmatrix} E_x \\ E_y \end{bmatrix}_{reception} = \frac{e^{ikr}}{kr} [S_2] = \begin{bmatrix} S_{HH} & S_{HV} \\ S_{VH} & S_{VV} \end{bmatrix} = \begin{bmatrix} E_x \\ E_y \end{bmatrix}_{transmission} \quad (2.1.4)$$

Les quatre coefficients de la matrice  $[S_2]$  sont des nombres qui enregistrent l'amplitude et la phase de l'onde sous forme complexe. Les éléments  $S_{HH}$  et  $S_{VV}$  mesurent la puissance du retour de la polarisation dans le même axe, tandis que les éléments  $S_{HV}$  et  $S_{VH}$  mesurent cette puissance dans les axes croisés. Dans le cas mono-statique, où l'antenne de réception et de transmission sont les mêmes, la matrice est aussi nommée matrice de Sinclair et les coefficients  $S_{HV}$  et  $S_{VH}$  sont égaux.

Les cibles d'origine anthropique sont stationnaires et leurs propriétés polarimétriques sont souvent caractérisées par un mécanisme de rétrodiffusion unique. Dans ce cas, l'observation d'une seule matrice de Sinclair permet de caractériser la cible. Cependant dans la plupart

des observations, les cibles naturelles sont des cibles étendues et leurs propriétés polarimétriques sont généralement hétérogènes. Dans ce cas, la matrice de Sinclair décrit plutôt la superposition d'un ensemble de matrices de diffusion représentant les différents diffuseurs dans la cellule de résolution. Afin de bien décrire les caractéristiques de rétrodiffusion polarimétrique des cibles étendues on utilise plutôt une analyse statistique de second ordre. Le formalisme le plus courant pour caractériser complètement les diffuseurs distribués est l'emploi de la matrice  $3 \times 3$  de cohérence  $[T_3]$  (ou de covariance  $[C_3]$ ). Les matrices  $[T_3]$  ou  $[C_3]$  permettent de calculer une moyenne d'ensemble sur plusieurs cellules de résolution (pixels) contiguës pour la réduction du chatolement des images en QuadPol.

On obtient la matrice de covariance  $[C_3]$  à partir de la matrice  $[S_2]$  en représentant cette dernière sous la forme du vecteur lexicographique  $\vec{k}_C$ :

$$\vec{k}_C = \begin{bmatrix} S_{HH} \\ \sqrt{2}S_{HV} \\ S_{HH} \end{bmatrix} \quad (2.1.5)$$

et la matrice de covariance  $[C_3]$  nous est donnée par la moyenne des produits:

$$[C_3] = \frac{1}{L_n} \sum_{i=1}^L (\vec{k}_{iC} \cdot \vec{k}_{iC}^H) \quad (2.1.6)$$

où l'opération Hermitienne H est définie comme le conjugué transposé du vecteur complexe  $\vec{k}_C$  et  $L_n$  exprime le nombre de vues sur lequel la matrice est moyennée et l'opérateur  $\langle \cdot \rangle$  résume l'opération de moyennage sur un voisinage.

$$[C_3] = \begin{bmatrix} \langle S_{hh}S_{hh}^* \rangle & \langle \sqrt{2}S_{hh}S_{hv}^* \rangle & \langle S_{hh}S_{vv}^* \rangle \\ \langle \sqrt{2}S_{hv}S_{hh}^* \rangle & \langle 2S_{hv}S_{hv}^* \rangle & \langle \sqrt{2}S_{hv}S_{vv}^* \rangle \\ \langle S_{vv}S_{hh}^* \rangle & \langle \sqrt{2}S_{vv}S_{hv}^* \rangle & \langle S_{vv}S_{vv}^* \rangle \end{bmatrix} \quad (2.1.7)$$

De manière similaire la matrice de cohérence  $[T_3]$  est obtenue à partir de la représentation de la matrice  $[S_2]$  sous la forme du vecteur de Pauli :

$$\vec{k}_T = \frac{1}{\sqrt{2}} \begin{bmatrix} S_{HH} + S_{VV} \\ S_{VV} - S_{HH} \\ 2S_{HV} \end{bmatrix} \quad (2.1.8)$$

et par le produit suivant:

$$[T_3] = \frac{1}{L_n} \sum_{i=1}^L (\vec{k}_{iT} \cdot \vec{k}_{iT}^H) \quad (2.1.9)$$

$$[T_3] = \frac{1}{2} \begin{bmatrix} \langle |S_{hh} + S_{vv}|^2 \rangle & \langle (S_{hh} + S_{vv})(S_{hh} - S_{vv})^* \rangle & \langle 2(S_{hh} + S_{vv})S_{hv}^* \rangle \\ \langle (S_{hh} - S_{vv})(S_{hh} + S_{vv})^* \rangle & \langle |S_{hh} - S_{vv}|^2 \rangle & \langle 2(S_{hh} - S_{vv})S_{hv}^* \rangle \\ \langle 2S_{hv}(S_{hh} + S_{vv})^* \rangle & \langle 2S_{hv}(S_{hh} - S_{vv})^* \rangle & 4 \langle |S_{hv}|^2 \rangle \end{bmatrix} \quad (2.1.10)$$

Les matrices  $[T_3]$  et  $[C_3]$  sont des matrices Hermitiennes définies positives de rang 3. Elles partagent les mêmes valeurs propres positives, ce qui implique que leurs traces sont égales:

$$Tr([T_3]) = Tr([C_3]) = span \quad (2.1.11)$$

où le *span* est la mesure de la puissance totale de la rétrodiffusion. Par contre leurs vecteurs propres orthogonaux respectifs sont différents.

Les matrices  $[T_3]$  et  $[C_3]$  sont similaires et interchangeable sous une transformation linéaire  $[U_3]$ .

$$\vec{k}_T = \frac{1}{\sqrt{2}} \begin{bmatrix} +1 & 0 & +1 \\ +1 & 0 & -1 \\ 0 & \sqrt{2} & 0 \end{bmatrix} \vec{k}_C \iff \vec{k}_T = [U_3] \vec{k}_C \quad (2.1.12)$$

Nous avons donc:

$$[T_3] = [U_3] [C_3] [U_3]^{-1} \quad (2.1.13)$$

Elles génèrent des résultats équivalents et contiennent la même information polarimétrique. Ainsi le choix de la représentation ( $[T_3]$  ou  $[C_3]$ ) des données est arbitraire. Cependant pour la suite des travaux, nous utiliserons seulement la représentation en matrice de cohérence  $[T_3]$  des images en QuadPol car elle présente l'avantage de facilement visualiser la décomposition de Pauli à partir des termes diagonaux de la matrice.

A noter que la conséquence du moyennage sur un voisinage est la dégradation de l'information spatiale lorsque l'on utilise les matrices  $[T_3]$  et  $[C_3]$  comme outils d'analyse.



### 2.1.2. Le chatoisement des images RSO et RSOPOL

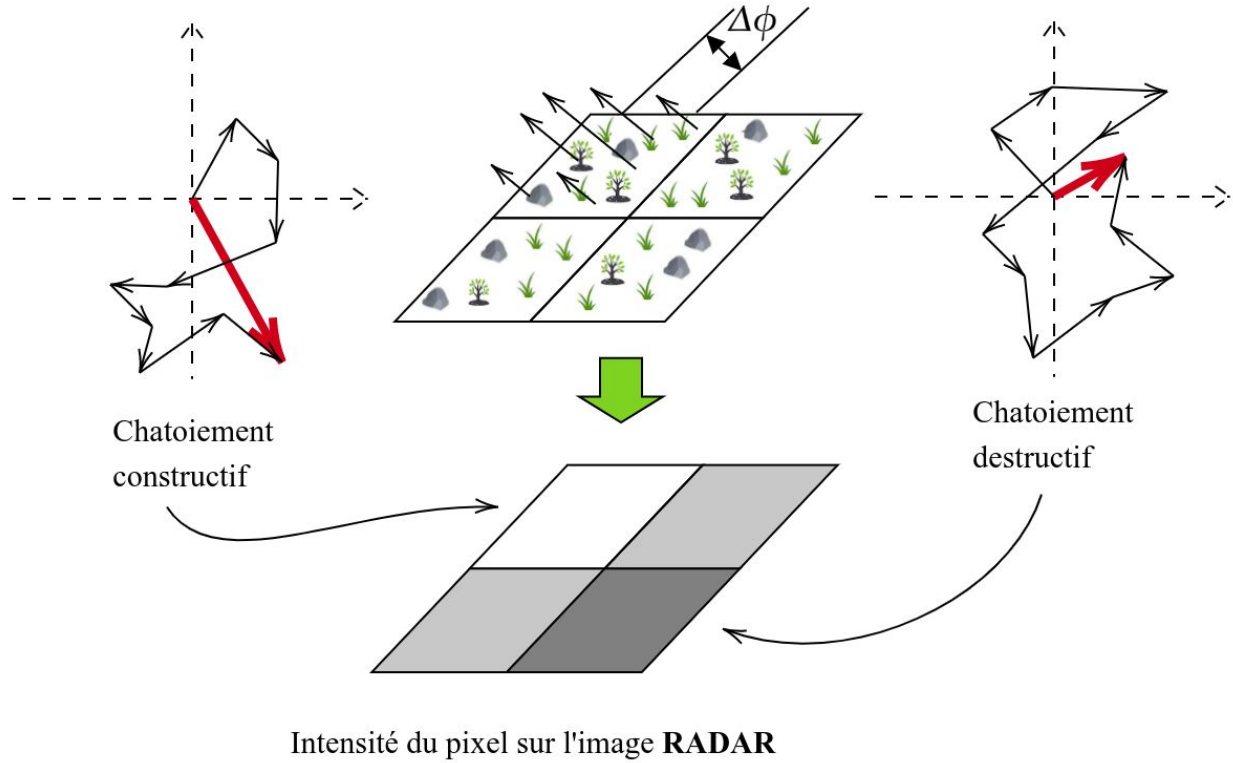
A cause de la nature cohérente du signal **RSO**, les mesures de la matrice de Sinclair sont affectées par le bruit de chatoisement (*speckle*) et par la même occurrence les matrices  $[T_3]$  et  $[C_3]$ . C'est ce qui donne un aspect sel et poivre sur chacun des canaux des images en QuadPol (Fig. 2.6). Le chatoisement est le résultat de la sommation cohérente de plusieurs échos radars en provenance des diffuseurs élémentaires d'une cellule de résolution. La Figure 2.7 donne une illustration du phénomène. Si nous supposons que la cellule contient  $N$  diffuseurs, chacun diffusant un signal complexe  $a_i e^{j\phi_i}$  alors la mise "bout à bout" de ces contributions dans le plan complexe, mesure la réponse totale du pixel, en amplitude et en phase.

$$Ae^{j\Phi} = \sum_{i=1}^N (a_i e^{j\phi_i}) = \sum_{i=1}^N (a_i(x_i + jy_i)) = A(X + jY) \quad (2.1.14)$$

La distribution aléatoire des diffuseurs fait en sorte que les ondes rétrodiffusées ne sont plus nécessairement en phase. Il arrive donc que la sommation des échos produise une forte rétrodiffusion qui n'est pas nécessairement en relation avec la vraie réflectivité. On parlera d'interférences constructives. Dans le cas contraire, le cas d'interférences destructives, le signal sera moindre que le vrai signal.

Le chatoisement est modélisé mathématiquement comme un bruit aléatoire multiplicatif sous l'hypothèse du "chatoisement pleinement développé" [13]. Cette hypothèse impose que le chatoisement soit indépendant du type de surface observée, que les diffuseurs contenus dans une cellule de résolution soient en grand nombre et statistiquement indépendants les uns des autres et que leurs phases soient distribuées uniformément dans l'intervalle  $[-\pi, \pi]$  sur les régions homogènes. Sous ces conditions, en vertu du théorème de la limite centrale, les composantes réelles  $X$  et imaginaires  $Y$  du signal peuvent être représentées comme des variables aléatoires indépendantes suivant une loi normale centrée sur zéro et de même écart-type  $\sigma$ :

$$\begin{cases} p(X = x) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(-\frac{x^2}{2 \cdot \sigma^2}\right) \\ p(Y = y) = \frac{1}{\sigma \cdot \sqrt{2 \cdot \pi}} \cdot \exp\left(-\frac{y^2}{2 \cdot \sigma^2}\right) \end{cases} \quad (2.1.15)$$



**Fig. 2.7.** Illustration du mécanisme de formation du chatoiement.

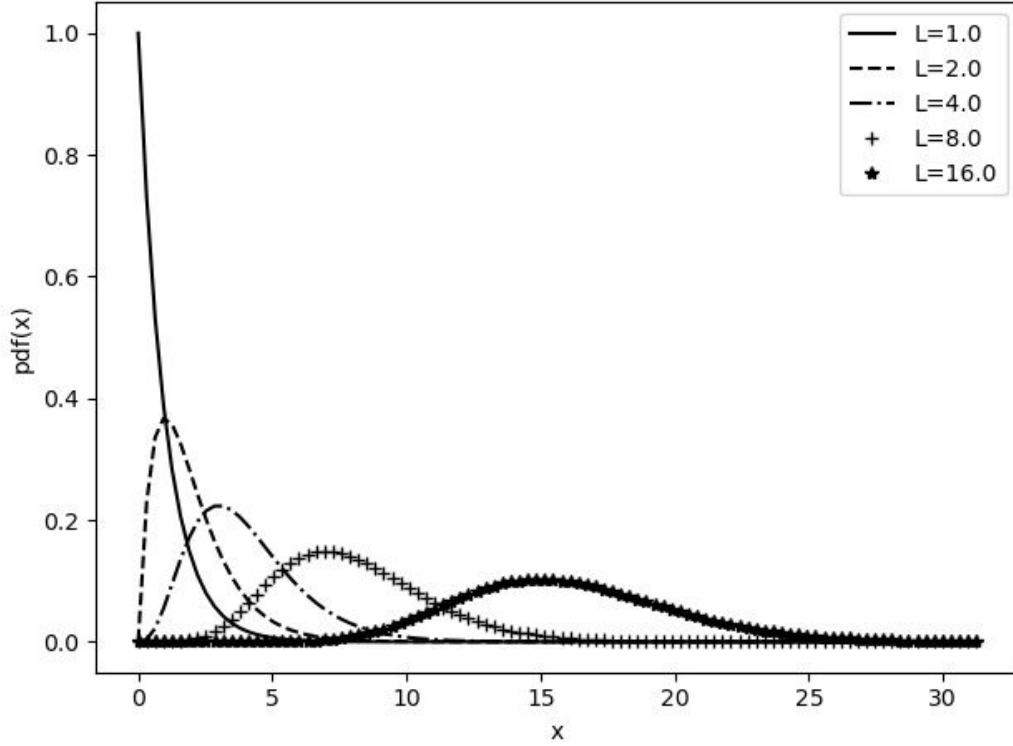
La densité de probabilité de l'intensité ( $I = X^2 + Y^2$ ) du chatoiement dans les régions homogènes est décrite par une loi exponentielle dans le cas des données une vue et d'une loi Chi-carré dans le cas des données multivues donnée par:

$$P(I) = \frac{L_n^{L_n} \cdot I^{L_n-1}}{(L_n - 1)! \cdot \sigma^{2L_n}} \exp\left(-\frac{L_n \cdot I}{\sigma^2}\right) \quad (2.1.16)$$

où  $L_n$  exprime le nombre de vues. La Figure 2.8 présente la distribution Chi-carré en fonction de différentes valeurs de  $L_n$ . Dans le cas des images une vue,  $L_n$  est égal à 1 et la distribution est exponentielle.

Dans le cas de l'imagerie polarimétrique, le vecteur de rétrodiffusion  $\vec{k}_C$  peut être décrit comme une variable aléatoire suivant une loi normale complexe circulaire multidimensionnelle centrée sur zéro:

$$p(\vec{K}_C = \vec{k}_C) = \frac{1}{\pi^3 \cdot |\mathbf{C}|} \exp(-\vec{k}_C^H \cdot \mathbf{C}^{-1} \cdot \vec{k}_C) \quad (2.1.17)$$



**Fig. 2.8.** Distribution des données **RSO** en fonction du nombre de vues  $L_n$

où  $|\mathbf{C}|$  est le déterminant de la matrice de covariance polarimétrique. L'estimé de la matrice de covariance ( $\mathbf{Z}$ ) dans le cas multivues est donné par le moyenne des matrices de covariance dans un voisinage:

$$\mathbf{Z} = \frac{1}{L_n} \sum_{i=1}^L \mathbf{C}_i \quad (2.1.18)$$

Dans ce cas, la matrice  $\mathbf{Z}$  peut être décrite comme une variable aléatoire suivant une distribution de Wishart avec la densité de probabilité:

$$P(\mathbf{Z}) = \frac{L^{3L} |\mathbf{Z}^{L-3}|}{|\mathbf{C}|^L \pi^3 \prod_{i=1}^3 \Gamma(L - i + 1)} \exp(-L \cdot \text{Tr}(\mathbf{C}^{-1} \cdot \mathbf{Z})) \quad (2.1.19)$$

Nous pourrons par la suite utiliser cette connaissance sur la distribution statistique du chatoisement pour simuler des données polarimétriques une-vue. Par contre, notons que dans la pratique, les conditions du "chatoisement pleinement développé" ne sont pas toujours

respectées si bien que la description du chatoiement se rapproche plus d'un modèle de texture. Dans ce cas, le chatoiement sera dépendant du type de surface et peut même aider à l'interprétation de la scène.

### 2.1.3. La décomposition polarimétrique

L'objectif principal d'une décomposition polarimétrique est d'exprimer la matrice de diffusion  $[S_2]$  selon des mécanismes polarimétriques canoniques dans le cas d'une décomposition cohérente ou selon des descripteurs plus simples de la matrice  $[C_3]$  ou de la matrice  $[T_3]$  dans le cas d'une décomposition incohérente. Dans le cas des cibles cohérentes, on privilégie les méthodes de décomposition cohérente, par exemples les approches de Pauli, Krogager [20], Cameron [1]. Dans le cas des cibles étendues, on utilisera plutôt les approches de décomposition incohérente selon Freeman [9], Huynen ou Cloude-Pottier [4].

L'approche que nous utiliserons au cours de l'évaluation des résultats est la décomposition polarimétrique basée sur les valeurs propres de la matrice de cohérence  $[T_3]$  de Cloude et Pottier [5]. Elle consiste à exprimer la matrice  $[C_3]$  sous la forme:

$$[T_3] = [U_3] [\Lambda_3] [U_3]^H = \sum_{k=1}^3 \lambda_k \vec{u}_k \cdot \vec{u}_k^H \quad (2.1.20)$$

où  $[U_3]$  représente la matrice des vecteurs propres  $\vec{u}_k$  et  $[\Lambda_3]$  représente la matrice diagonale des valeurs propres  $\lambda_k$ . Les vecteurs propres de la décomposition sont donnés par

$$\vec{u}_k = [\cos \alpha_k, \sin \alpha_k \sin \beta_k e^{j\delta_k}, \sin \alpha_k \sin \beta_k e^{j\gamma_k}]^T \quad (2.1.21)$$

Cette décomposition peut être interprétée comme la somme de trois matrices de cohérence  $[T_3]_k = \vec{u}_k \cdot \vec{u}_k^H$  (avec  $k = 1,2,3$ ) non corrélées décrivant les caractéristiques de la cible. A partir des paramètres principaux  $\lambda_k$  et  $\alpha_k$ , on peut calculer les trois paramètres de la décomposition  $H/A/\bar{\alpha}$ :

L'entropie  $H$  permet de quantifier le caractère aléatoire du processus de diffusion lié à la réponse polarimétrique. La valeur de l'entropie est égale à la somme logarithmique des valeurs propres normalisées de  $[T_3]$ :

$$H = \sum_{k=1}^3 p_k \log_3 p_k \quad (2.1.22)$$

avec

$$p_k = \frac{\lambda_k}{\sum_{k=1}^3 \lambda_k} \quad (2.1.23)$$

$H$  est comprise entre 0 et 1. Les cas extrêmes rencontrés sont les mécanismes de rétrodiffusion non dépolarisants où  $H = 0$ , indiquant la présence d'un diffuseur dominant, et les mécanismes de rétrodiffusion totalement dépolarisants, où  $H = 1$ , suggérant la présence de plusieurs types de diffuseur de force égale.

L'anisotropie  $A$  varie aussi entre 0 et 1. Elle permet de mesurer l'importance relative des diffuseurs secondaires et apporte une information complémentaire à  $H$  dans le cas où l'entropie polarimétrique posséderait une valeur moyenne. L'anisotropie est calculée comme suit:

$$A = \frac{\lambda_2 - \lambda_3}{\lambda_2 + \lambda_3} \quad (2.1.24)$$

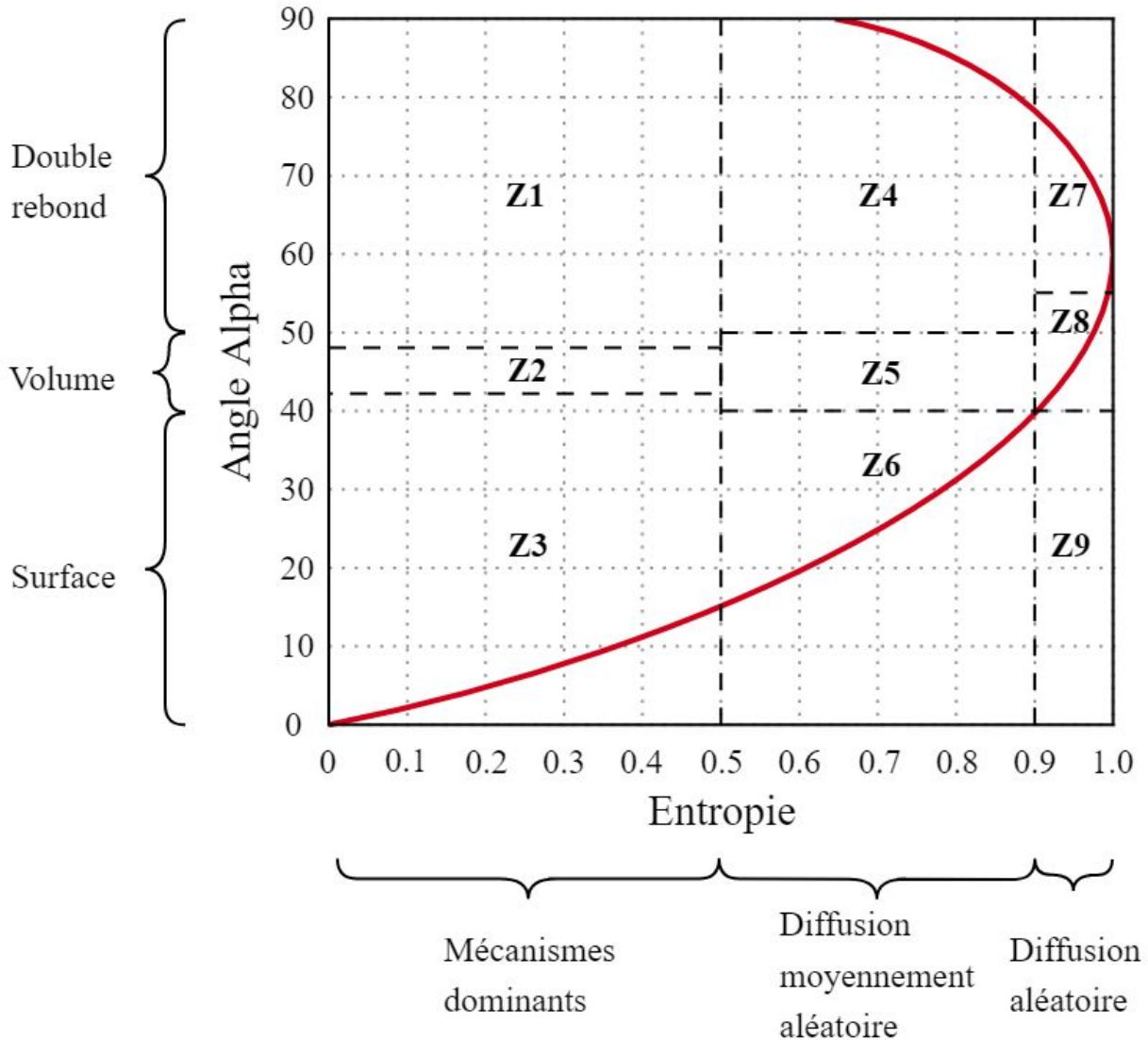
L'angle  $\bar{\alpha}$  décrit physiquement le mécanisme de diffusion et varie entre  $0^\circ$  et  $90^\circ$ .

$$\bar{\alpha} = \sum_{k=1}^3 p_k \cdot \alpha_k \quad (2.1.25)$$

Pour des valeurs proches de  $0^\circ$ , l'angle  $\bar{\alpha}$  indique la présence de diffuseurs surfaciques. Si les valeurs sont proches de  $45^\circ$  alors elle indique une diffusion volumétrique. Et pour des valeurs qui tendent vers  $90^\circ$ , le mécanisme de rétrodiffusion s'apparente à celui d'un dièdre, à double rebond.

La Figure 2.9 montre le diagramme  $H/\bar{\alpha}$ . La courbe (rouge) donne pour chaque valeur de l'entropie  $H$  les bornes minimales et maximales pour  $\bar{\alpha}$ . En fonction de l'entropie  $H$ , on peut séparer le diagramme en trois grandes zones qui à leur tour peuvent être raffinées en fonction de l'angle  $\bar{\alpha}$ :

- (1) les zones de diffusion de faible entropie pour lesquelles un seul mécanisme de diffusion prédomine:
  - (a) **Z3** (Surfaces de Bragg):  $\bar{\alpha}$  est inférieur à  $\pi/4$ , ce qui correspond à une simple diffusion par une surface (sans mécanisme introduisant une rotation de phase de



**Fig. 2.9.** Diagramme  $H/\bar{\alpha}$  avec les régions qui définissent les limites des classes de diffuseurs de Cloude-Pottier

- $\pi$  entre HH et VV). En pratique, tous les mécanismes décrits par un nombre impair de diffusions appartiennent à cette classe: par exemple les surfaces d'eau.
- (b) **Z1** (Réflecteurs diédraux):  $\bar{\alpha}$  est supérieur à  $\pi/4$  et inférieur à  $\pi/2$ , ce qui correspond à une diffusion double. En pratique, tous les mécanismes décrits par un nombre pair de diffusions appartiennent à cette classe.
- (c) **Z2** (Dipôles):  $\bar{\alpha}$  est proche de  $\pi/4$ , le mécanisme proposé est une diffusion par des dipôles. Ce mécanisme est fortement corrélé avec un déséquilibre prononcé en amplitude entre HH et VV

- (2) les zones de diffusion d'entropie moyenne (entre 0.5 et 0.9)
  - (a) **Z6** (Surfaces aléatoires):  $\bar{\alpha}$  est faible, on a affaire à des mécanismes de diffusion simple avec des effets liés à la rugosité de la surface.
  - (b) **Z4** (Double réflexions):  $\bar{\alpha}$  est supérieur à  $\pi/4$ , on a des effets de diffusion multiple (par exemple, diffusion volumique dans la canopée).
  - (c) **Z5** (Particules anisotropiques):  $\bar{\alpha}$  est proche de  $\pi/4$ , ce qui peut s'expliquer par une certaine corrélation entre l'orientation des diffuseurs.
- (3) les zones de diffusion de forte entropie
  - (a) **Z9** (Zone impossible):  $\bar{\alpha}$  est faible, ce qui correspond à des valeurs de  $\bar{\alpha}$  supérieures à la borne supérieure, ne peut se caractériser par cette approche.
  - (b) **Z7** (Structures complexes): cette zone correspond à des diffusion multiples, telles que l'on peut les observer en forêts.
  - (c) **Z8** (Diffuseurs anisotropiques aléatoires): cette zone correspond donc à de la diffusion volumique par un nuage de particules de type aiguille.

À noter que les limites sont arbitraires et dépendent de l'étalonnage du radar, du bruit minimum des observations et de la variance des estimations des paramètres.

## 2.2. Les réseaux neuronaux artificiel et l'apprentissage automatique

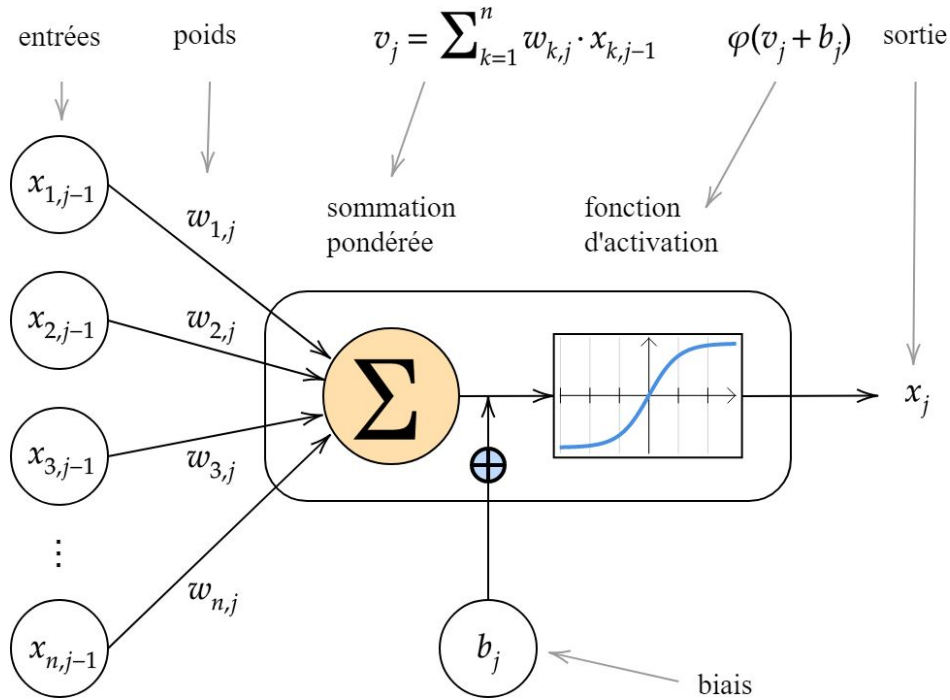
Les réseaux de neurones sont des constructions abstraites qui simulent l'activité des réseaux de neurones biologiques élémentaires. Ils sont utilisés en apprentissage automatique pour construire des modèles à partir de données existantes dans le but d'effectuer des prédictions sur de nouvelles données soit à l'aide de classification, dans le cas discret, ou de régression, dans le cas continu. La topologie des réseaux neuronaux est diversifiée et il est difficile de répertorier toutes les variétés existantes et leurs applications. Liu et al. [24] propose une revue des architectures les plus utilisées en apprentissage machine profond (comme le **RNC**). Van Veen [37] fait une revue exhaustive des topologies et des différents types de neurones que l'on retrouve actuellement dans la littérature. On y dénombre pas moins d'une trentaine d'architectures neuronales, mais quelques architectures importantes sont manquantes comme le UNet (Ronnenberg [31]). Dans le cadre de ce mémoire nous concentrons notre attention sur les **RNC**.

### 2.2.1. Les neurones artificiels

Le premier modèle du neurone artificiel a été présenté par McCulloch et Pits [26]. Un neurone artificiel est un modèle mathématique d'un neurone biologique. Il possède généralement plusieurs entrées et une sortie qui correspondent par analogie aux dendrites et au cône d'émergence du neurone biologique. Les actions excitatrices et inhibitrices des synapses sont modélisées par les poids (paramètres) associés aux entrées. Les valeurs numériques de ces poids sont ajustées lors de la phase d'apprentissage. Dans sa version la plus simple, un neurone artificiel calcule la somme pondérée des entrées reçues, puis applique à cette valeur une fonction d'activation, généralement non linéaire. La valeur finale obtenue est la sortie du neurone.

La Figure 2.10 illustre un exemple de base d'un modèle non-linéaire d'un neurone, où  $x_{1,j-1}, x_{2,j-1}, x_{3,j-1}, \dots, x_{n,j-1}$  sont les signaux d'entrée de la couche  $j - 1$ ;  $w_{1,j}, w_{2,j}, w_{3,j}, \dots, w_{n,j}$  sont les poids synaptiques de la couche  $j$ ;  $v_j$  est la valeur de pré-activation qui correspond à la combinaison linéaire des signaux d'entrées;  $\varphi(\cdot)$  est une fonction d'activation non-linéaire telle que la fonction tangente hyperbolique, et  $x_j$  est la sortie de la couche  $j$ . Le biais  $b_j$  est sommé avec la valeur de pré-activation ce qui vaut à l'application d'une transformation affine pour calculer la sortie.





**Fig. 2.10.** Structure d'un neurone artificiel. Le neurone calcule la somme de ses entrées puis cette valeur passe par la fonction d'activation pour produire sa sortie.

Considérant que les poids peuvent être condensés en une matrice  $\mathbf{W}$  où chaque colonne contient les paramètres menant à un neurone, que les entrées et les biais peuvent être représentés par des vecteurs, le calcul sur une couche donnée peut s'exprimer simplement par une transformation affine suivie d'une fonction non linéaire:

$$h = \varphi_i(W^\top x + b) \quad (2.2.1)$$

### 2.2.2. Les réseaux de neurones artificiels

Un réseau de neurones peut être vu comme un assemblage de neurones connectés ensembles suivant une loi de connectivité que l'on définit. Le réseau peut être décrit par un graphe où les noeuds sont les neurones et les arêtes sont les connexions entre les neurones. Dans le cas des réseaux neuronaux acycliques (*feed-forward*), ils propagent l'entrée du réseau vers les couches suivantes sans jamais revenir en arrière. Ils sont assemblés comme une composition de fonctions.

$$y = f(x) = \varphi(w_j \dots \varphi(w_2 \cdot \varphi(w_1 \cdot x + b_1) + b_2 \dots + b_j) \quad (2.2.2)$$

Un exemple classique est le réseau perceptron multicouche (**RPM**). Les neurones sont combinés par couche tel que présenté par la Figure 2.11. Il n'y a pas de connexion entre neurones d'une même couche et les connexions sont établis qu'avec les neurones des couches en aval. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante. Par extension, on appelle "couche d'entrée" l'ensemble des neurones d'entrée, "couche de sortie" l'ensemble des neurones de sortie. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées "couches cachées".

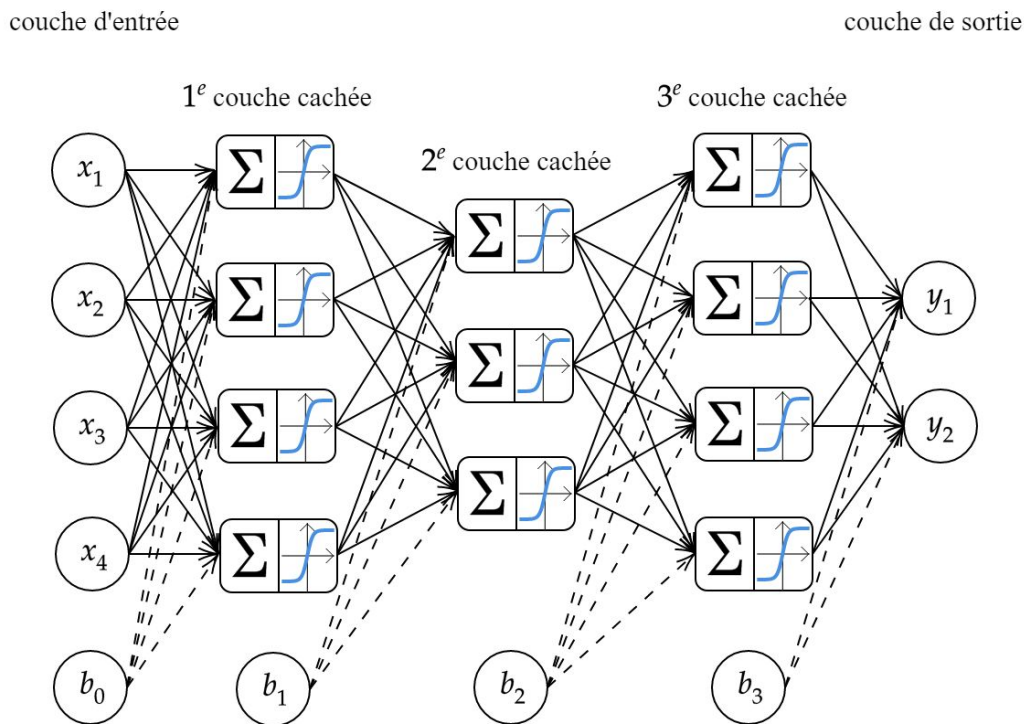
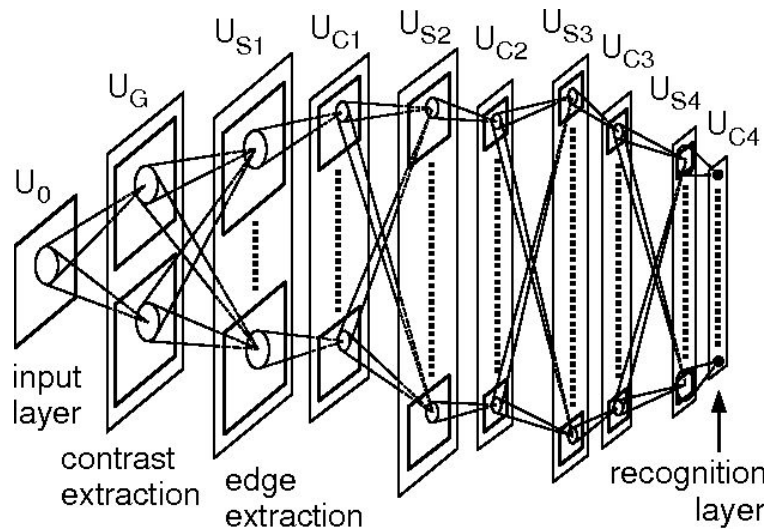


Fig. 2.11. Exemple d'un perceptron multicouches avec 5 couches.

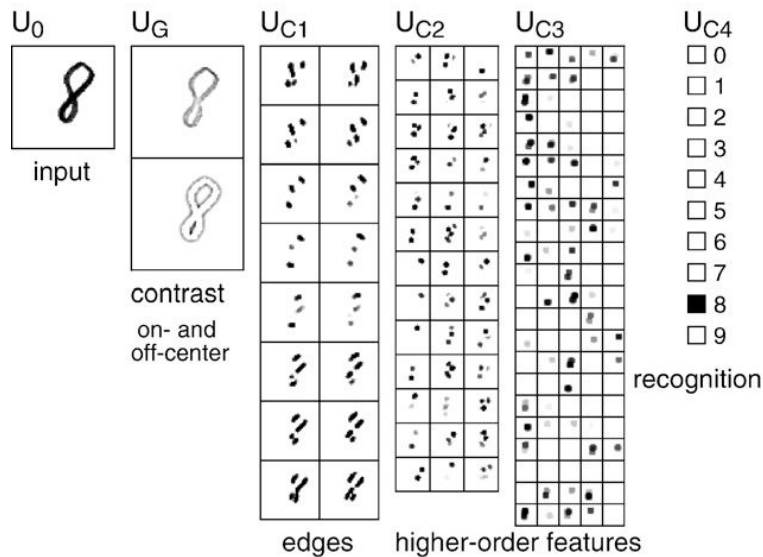
### 2.2.3. Les réseaux de neurones à convolution

L'architecture des **RNC** a été inspirée par les travaux de Hubel et Wiesel [15] sur le cortex visuel des mammifères. Ils introduisirent la notion de champ récepteur d'un neurone du système visuel qui est la portion du champ visuel qui, lorsqu'on lui présente un stimulus lumineux, modifie la réponse de ce neurone. Ils ont démontré que les neurones des premières couches du cortex visuel s'activaient en fonction de différents types de stimuli. Les neurones fonctionnaient comme des détecteurs simples, certains étaient spécialisés pour la détection

de motifs linéaires dans un sens et certains autres groupes de neurones détectaient les motifs linéaires dans un autre sens. Les premiers algorithmes se rapprochant des **RNC** remontent aux années 1980 avec les travaux de Fukushima [10] et le Neocognitron (Fig. 2.12). Le modèle introduit la notion d'une série de transformations hiérarchiques qui extraient des caractéristiques visuelles de plus en plus complexes pour déterminer la classe de sortie.



**Fig. 2.12.** Architecture du Neocognitron [10]



**Fig. 2.13.** Un exemple des réponses de la sortie de chaque couche du Neocognitron. L'entrée est détectée correctement comme le chiffre 8. [10]

Ce n'est que dans les années 1990, grâce aux travaux sur la reconnaissance des caractères manuscrits par LeCun et al [21], que les **RNC** seront popularisés. Les auteurs ont proposé une série d'architectures (**LeNet**) qui influenceront les recherches suivantes. L'apprentissage des réseaux se font par rétropropagation du gradient de l'erreur.

Plus formellement, dans le cas des **RNC**, l'Équation 2.2.1 transforme la multiplication matricielle des poids par une opération de convolution (\*):

$$h = \varphi_i(W^\top * x + b) \quad (2.2.3)$$

La convolution est une opération mathématique sur deux fonctions (f et g) qui produit une troisième fonction exprimant la façon dont la forme de l'une est modifiée par l'autre:

$$\begin{aligned} s(t) &= (f * g)(t) \\ &= \int_{-\infty}^{\infty} f(a) g(t - a) da \\ &= \sum_{a=-\infty}^{\infty} f(a) g(t - a) \\ &= (g * f)(t) \\ &= \sum_{a=-\infty}^{\infty} f(t - a) g(a) \end{aligned} \quad (2.2.4)$$

Dans le cas des images, l'opération de convolution sur un signal en 2 dimensions est donnée par la double somme suivante:

$$S(i,j) = (K * I)(i,j) = \sum_m \sum_n I(i + m, j + n) K(m,n) \quad (2.2.5)$$

où  $I$  représente notre image d'entrée et  $K$  représente le noyau (filtre) de l'opération de convolution. Lors de l'apprentissage d'un **RNC** les poids  $\mathbf{W}$  de l'Équation 2.2.3 correspondent aux poids du filtre  $K(m,n)$  de l'Équation 2.2.5. La sortie  $S(i,j)$  correspond à la carte de caractéristiques avant l'application de la fonction d'activation.

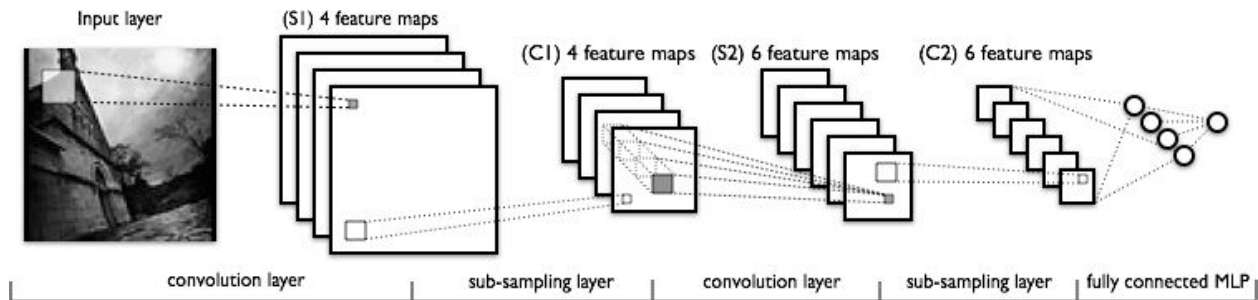
L'architecture des **RNC** s'appuie sur trois principes fondamentaux:

- (1) Des champs récepteurs locaux associés à des convolutions (filtres, noyaux) qui permettent de détecter des caractéristiques visuelles élémentaires sur l'image, formant ainsi des cartes de caractéristiques;
- (2) Le partage des poids, qui consiste à apprendre les mêmes poids d'une convolution et par le fait même d'extraire les mêmes types de caractéristiques visuelles pour toutes les positions sur l'image;
- (3) Des opérations de ré-échantillonnage (*pooling*) qui permettent de réduire la sensibilité aux translations, ainsi que de réduire le coût du traitement;

Le deuxième principe présente la pierre angulaire des **RNC**, car il permet de réduire considérablement la taille du réseau en mémoire et en diminuant le nombre de paramètres à apprendre. Cela permet à son tour de construire des architectures multicouches qui opèrent sur des entrées de grande dimension tout en demeurant de taille réaliste (ce qui est irréalisable avec les **RPM**).

L'architecture typique des **RNC** (Fig. 2.14) est composée par une séquence de plusieurs couches comprenant la couche d'entrée, quelques couches cachées et la couche de sortie. Les couches cachées consistent en une ou plusieurs couches convolutives avec en sortie une fonction d'activation qui utilise la somme pondérée de toutes les entrées de la couche précédente pour générer une valeur de sortie transformée par une fonction d'activation non-linéaire. Cette valeur est transmise à la couche suivante. La fonction d'activation est quelquefois suivie d'une couche de re-échantillonnage (*pooling*) qui effectue une réduction de la sortie antérieure en prenant généralement la valeur maximale ou la valeur moyenne de l'ensemble d'une zone regroupée. Le réseau peut se terminer par une ou plusieurs couches entièrement connectées intercalées avec des couches d'activation non-linéaire. Les couches convolutives effectuent des calculs locaux définis par la taille des champs réceptifs et les couches entièrement connectées apprennent les représentations globales de l'image sur toutes les activations des couches précédentes.

Pendant l'opération de convolution, les filtres de la taille du champ récepteur local sont convolués avec l'image pour produire une valeur de sortie pour chaque pixel, formant des cartes de caractéristiques pour chaque filtre. Ces cartes deviennent à leur tour l'entrée de la prochaine couche convolutive et produisent d'autres cartes de caractéristiques. Ce procédé hiérarchique est répété en séquence pour toute la pile de couches convolutives (Fig. 2.15).



**Fig. 2.14.** Architecture d'un réseau **LeNet** démontrant les structures de base des **RNC**

La fonction d'activation est appliquée aux cartes de caractéristiques. Cette fonction peut être une des très nombreuses fonctions non-linéaires existantes, soient entre autres l'unité de rectification linéaire (**ReLU**) [11], la fonction tangente hyperbolique ou la fonction sigmoïde (Fig. 2.16).

La couche d'échantillonnage de type *max* sélectionne la valeur maximale des éléments des cartes de caractéristiques dans un voisinage de taille définie, tandis la couche d'échantillonnage de type *ave* sélectionne la valeur moyenne. Dans le cas où le ré-échantillonnage est appliqué avec un pas égal ou supérieur à la taille du voisinage, les cartes de caractéristiques seront sous-échantillonnées, de sorte que les prochaines étapes d'analyse seront effectuées à plus large échelle avec une résolution diminuée. L'application de la couche de ré-échantillonnage a pour objectif premier d'augmenter la robustesse du **RNC** au bruit et aux distorsions spatiales. De plus, en réduisant la taille du réseau, on bénéficie d'une réduction du nombre de paramètres à estimer et du risque de surapprentissage.

La sortie de la dernière couche convolutive est aplatie pour être injectée dans les couches entièrement connectées. Ces couches sont complètement connectées en amont comme en aval, impliquant un nombre élevé de paramètres à estimer si on les compare aux couches convolutives. Ce genre de configuration a été couramment utilisé avec succès pour la classification en vision par ordinateur. Les modèles de classification les plus reconnus mettant en œuvre cette configuration sont les gagnants successifs du concours ILSVRC: AlexNet (2012) [19], ZFNet (2013) [40], VGGNet (2014, 2<sup>e</sup> position) [35], GoogLeNet (2014) [36] et ResNet (2015) [14].

Les réseaux à résidus (ResNet) (He et al. [14]) ont introduit une composante importante dans l'architecture des **RNC**. Ces réseaux ont des connexions de saut qui permettent au signal original  $x$  de traverser comme suit les couches:  $G(x) = F(x) + x$ . Ils forment des blocs résiduels (Fig. 2.18). La fonction  $F(x)$  apprend le résiduel, qui est la différence entre

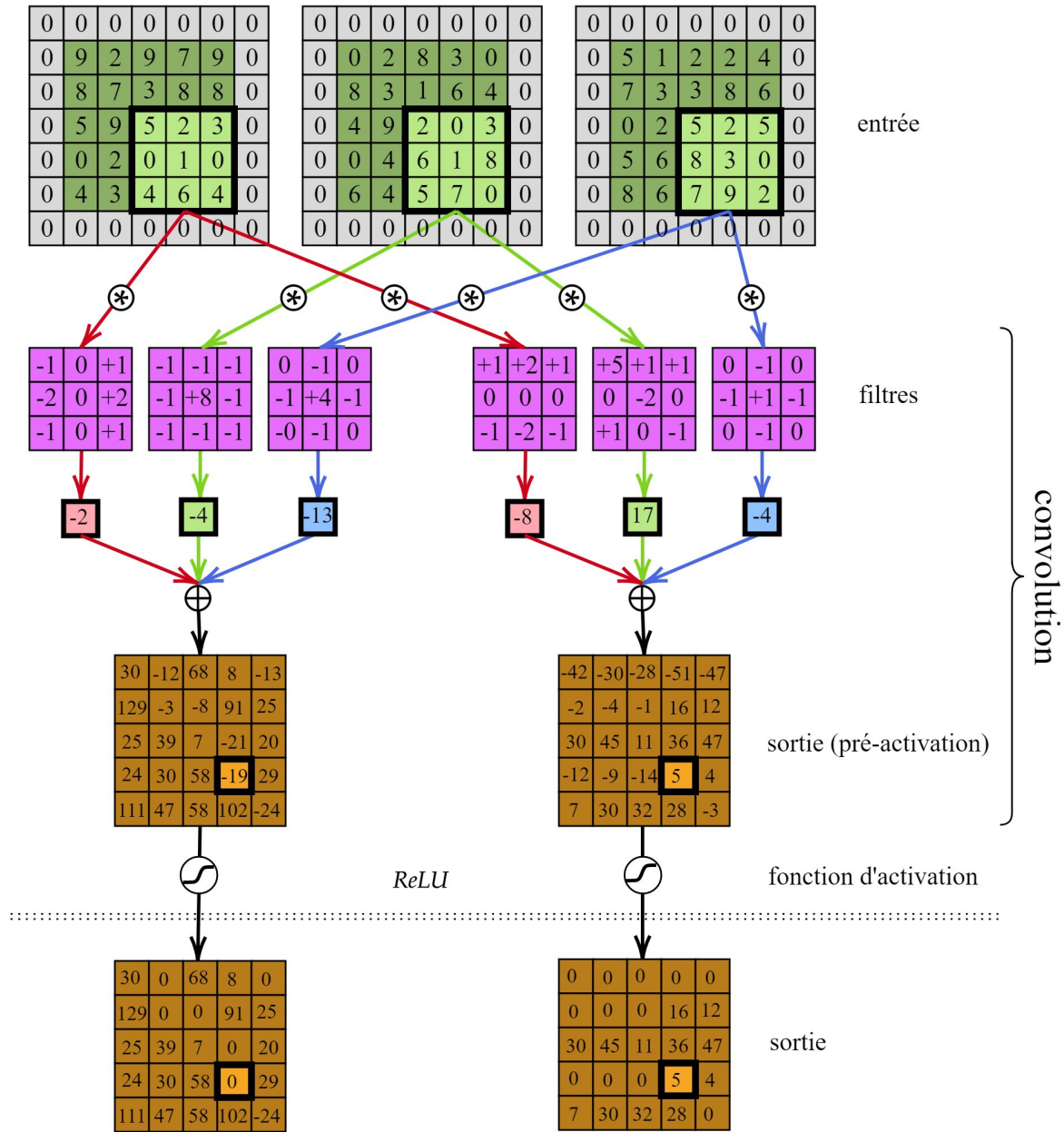
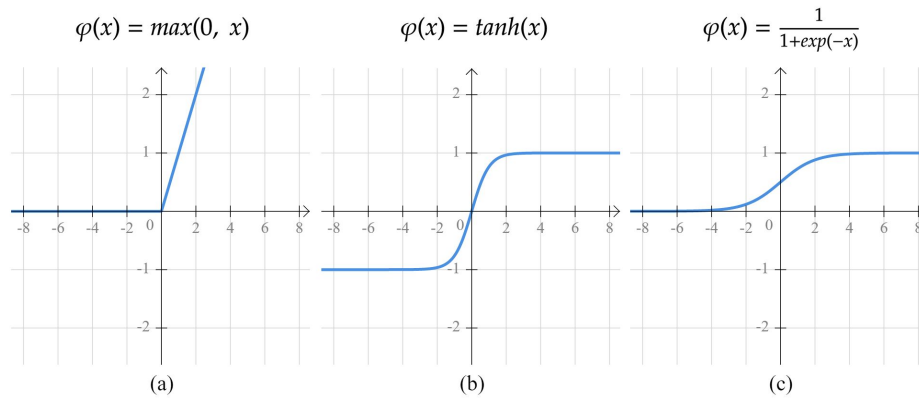
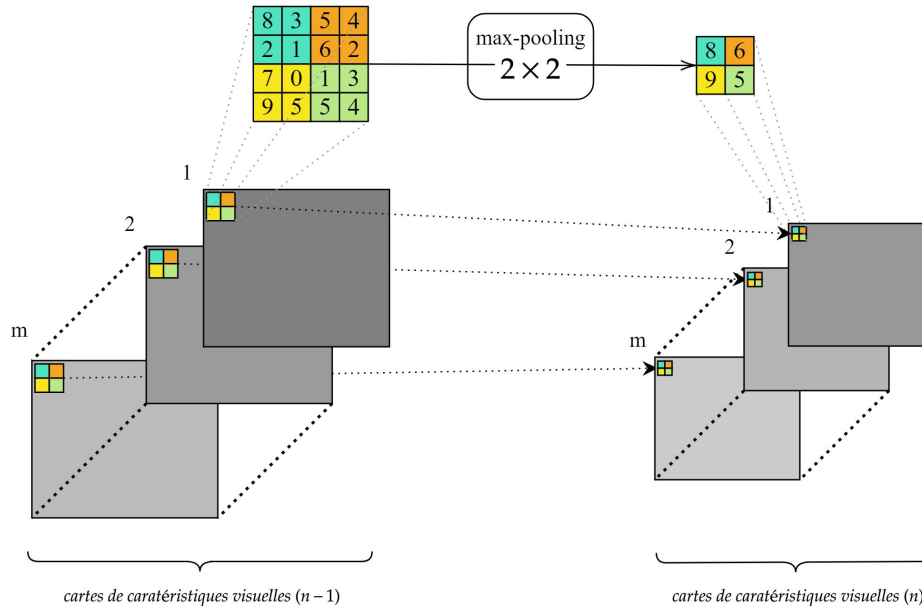


Fig. 2.15. Illustration des opérations d'un bloc convolutif *conv* + *ReLU*.

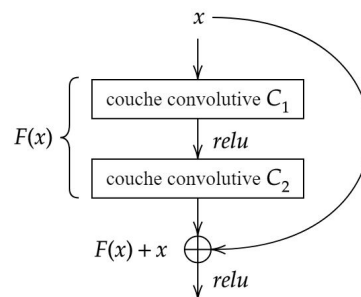
le résultat final et le signal d'entrée  $x$ . L'effet de ce saut de connections est de limiter le problème du gradient évanescent que l'on rencontre parfois lors de l'apprentissage de réseaux neuronaux avec la méthode de rétropropagation du gradient de l'erreur.



**Fig. 2.16.** Illustration de certaines fonctions d'activation non-linéaires. (a) Rectifier Linear Unit (ReLU), (b) tangente hyperbolique, (c) sigmoïde



**Fig. 2.17.** Illustration de l'opération d'échantillonnage *max-pooling*



**Fig. 2.18.** Illustration d'un bloc résiduel



#### 2.2.4. Les réseaux de neurones pleinement convolutifs

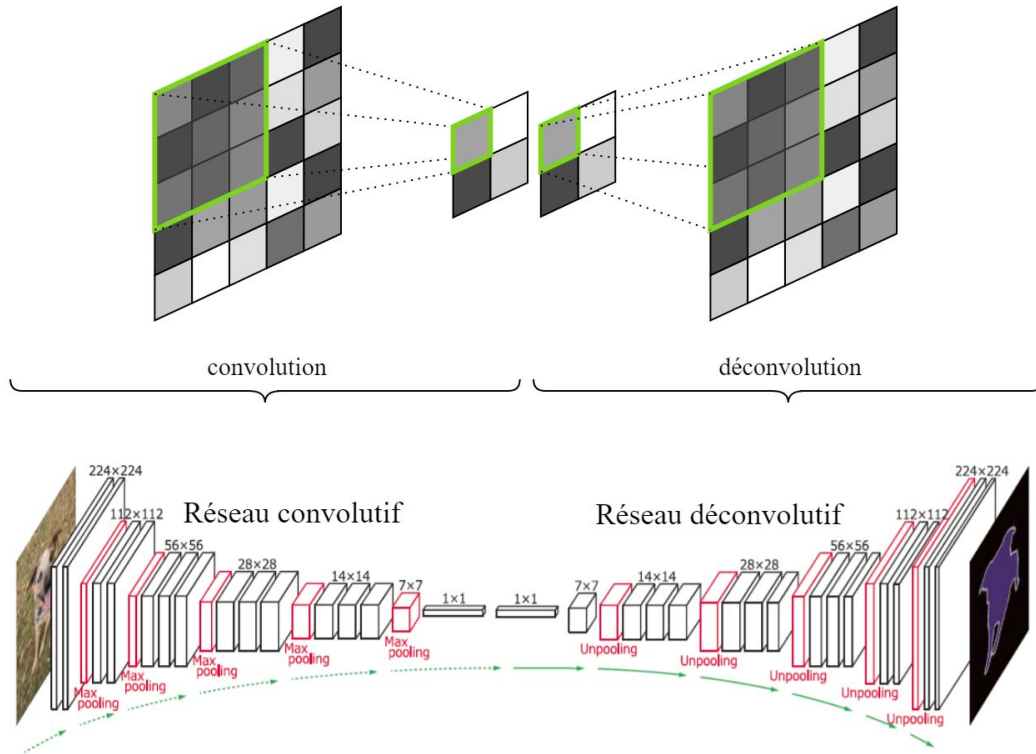
Les réseaux de neurones pleinement convolutifs (**RNPC**) (*fully convolutional network*) sont utilisés pour la segmentation sémantique (Shelhamer et al.[**33**]), l'étiquetage de scène (Pinheiro et al. [**28**]) et pour la détection de caractéristiques (Sermanet et al. [**32**]). En télédétection, Sherrah [**34**] a obtenu des résultats de pointe avec les **RNPC** pour l'étiquetage sémantique des images aériennes.

L'architecture (**RNPC**) est semblable à celle des **RNC** à la différence que les couches entièrement connectées sont omises. La dernière couche est directement connectée à la fonction de perte pour l'apprentissage. Cette omission de la partie **RPM** change le fonctionnement du réseau de trois façons:

- (1) La couche **RPM** permet au réseau d'apprendre des représentations globales dans lesquelles les relations spatiales des cartes de caractéristiques sont ignorées. Ce n'est plus le cas avec les **RNPC** où le réseau tente d'apprendre des représentations qui tiennent compte des relations spatiales locales;
- (2) Les **RNPC** peuvent recevoir des entrées de taille arbitraire, contrairement aux **RNC** qui n'acceptent que des entrées de taille fixe;
- (3) Les **RNC** retournent une seule valeur, les **RNPC** produisent des cartes de sortie grossières lorsqu'un ré-échantillonnage est appliqué;

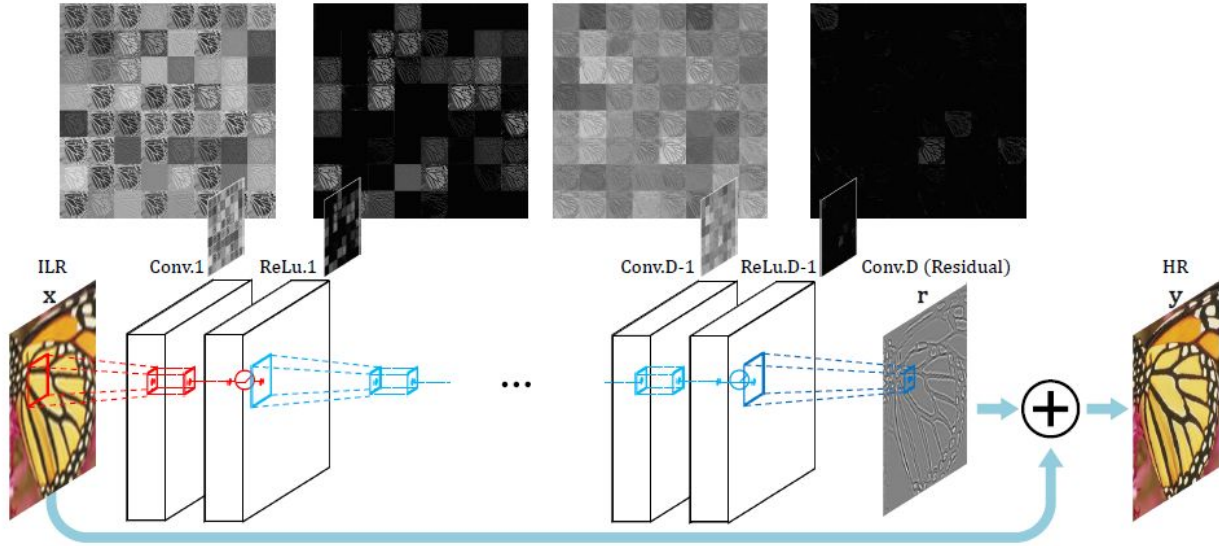
Pour reconstruire la résolution originale de l'image deux approches sont proposées. Sherrah [**34**] suggère une approche "*shift and stitch*" où les cartes grossières calculées par translation de la position sur l'image d'entrée sont fusionnées pour former une carte dense. La seconde approche consiste à utiliser des couches déconvolutives (Zeiler et al. [**41**]) après toutes les couches convolutives pour suréchantillonner la sortie sous-échantillonnée afin d'obtenir la couche de sortie finale (Fig. 2.19).

Une autre architecture **RNPC** que l'on retrouve surtout en restauration d'image est un réseau formé purement de couches convolutives et de couches d'activation (Fig. 2.20). Ces réseaux produisent en sortie une carte de caractéristiques (*features map*) de la même dimension que la taille 2D de l'entrée. Le nombre de canaux formant la sortie dépend du nombre de convolutions de la dernière couche. Le champ réceptif de la couche d'entrée est généralement plus petit, car sans cela le nombre de paramètres à apprendre serait trop grand. Le réseau se termine habituellement par une fonction de perte qui calcule une régression. Ce genre



**Fig. 2.19.** Illustration de l'opération convolution versus l'opération déconvolution

de configuration a été couramment utilisé avec succès pour les problèmes de superrésolution (Dong et al. [6]; Kim et al. [16]), débruitage en général (Zhang et al. [42], [43]). Dans le cadre de ce mémoire, nous utiliserons cette approche pour calculer nos modèles pour le filtrage du chatoisement.



**Fig. 2.20.** Exemple d'un RNPC sans couche de ré-échantillonnage utilisé en superrésolution: VDSR (*Very Deep Super Resolution* [16] ).

### 2.3. L'apprentissage des réseaux

Les approches en apprentissage machine profond utilisent plusieurs techniques issues de l'apprentissage machine en générale. Nous décrivons dans cette section certaines techniques que nous utiliserons. Les grandes étapes du fonctionnement d'un apprentissage sont décrites par l'Algorithme 1. Considérons un ensemble de données  $(X, y)$  où  $X$  est la donnée caractéristique et  $y$  est son étiquette. Un modèle  $F(\Theta, X)$  permet de calculer un estimé  $\bar{y} = F(x)$ . Les valeurs  $\Theta$  sont les paramètres du modèle. L'objectif de l'apprentissage est de trouver les paramètres qui minimisent l'erreur  $\epsilon$  entre l'estimé  $\bar{y}$  et l'étiquette  $y$ . La minimisation se fait en trouvant les minimums de la fonction  $\epsilon$ , ce qui implique que cette dernière doit être convexe.

---

#### **Algorithme 1 :** Étapes générales de l'apprentissage d'un réseau de neurone

---

Initialiser le modèle avec des poids aléatoires;

**pour chaque** *exemple de la liste des données d'entraînement* **faire**

- (1) Donner l'entrée au modèle pour obtenir la sortie;
- (2) Calculer l'erreur en comparant la sortie avec le résultat cible
- (3) Propager l'erreur de couche en couche vers l'arrière
- (4) Mettre à jour tous les poids du réseau

**fin**

---

### 2.3.1. Optimisation par descente du gradient

L'optimisation par descente du gradient est un algorithme d'optimisation de premier ordre. Il permet de calculer les minimum locaux d'une fonction de coût convexe à partir des dérivées premières. La descente de gradient ajuste de manière itérative les paramètres du modèle afin de trouver progressivement la meilleure combinaison de poids et de biais pour minimiser la fonction de coût en fonction des données d'apprentissage. La mise à jour des paramètres est fait selon:

$$\Theta = \Theta - \lambda \cdot \frac{1}{N} \sum_{n=1}^N \frac{\partial \epsilon(y, \bar{y})}{\partial \Theta} \quad (2.3.1)$$

où  $\lambda$  est le taux d'apprentissage. La valeur  $\lambda$  règle la vitesse de la mise à jour des paramètres. L'Algorithme 2 présente le pseudo-code de la méthode.

---

**Algorithme 2** : La descente du gradient

---

**Sorties** : Paramètres  $\Theta$  optimaux qui minimisent  $\epsilon$

**Entrées** : Fonction de coût  $\epsilon$ ; Taux d'apprentissage  $\lambda$ ; Ensemble de données  $(X, y)$ ; Modèle

$$F(\Theta, x)$$

initialisation aléatoire des paramètres  $\Theta$ ;

**tant que** *La minimisation converge?* **faire**

    (1) Calculer la prédiction  $\bar{y}$ ;

$$\bar{y} = F(\Theta, x)$$

    (2) Ajuster les paramètres  $\Theta$ ;

$$\Theta = \Theta - \lambda \cdot \frac{1}{N} \sum_{n=1}^N \frac{\partial \epsilon(y, \bar{y})}{\partial \Theta}$$

**fin**

---

### 2.3.2. Optimisation par descente du gradient stochastique par mini-lots

L'algorithme de descente de gradient par mini-lots estime le gradient à partir d'un petit sous-ensemble des données d'apprentissage  $x_i, y_i \in (X, y)$ . Il permet d'accélérer l'apprentissage. La version *Vanilla* utilise un mini-lot de taille 1. Dans la plupart des cas, l'algorithme de la descente de gradient peut trouver un point proche du minimum d'une fonction strictement convexe. Dans le cas de la descente de gradient stochastique par mini-lots, il n'est pas

garanti de trouver ce point mais une forte probabilité. A noter que les modèles profonds ne sont jamais des fonctions convexes. Cependant les algorithmes conçus pour l'optimisation convexe trouvent généralement des solutions suffisamment satisfaisantes pour les réseaux profonds, mais ils ne garantissent pas que les solutions soient des minimums globaux. L'Algorithme 3 présente le pseudo-code de la méthode.

---

**Algorithme 3 :** La descente du gradient par mini-lots

---

**Sorties :** Paramètres  $\Theta$  optimaux qui minimisent  $\epsilon$

**Entrées :** Fonction de coût  $\epsilon$ ; Taux d'apprentissage  $\lambda$ ; Ensemble de données  $(X, y)$ ; Modèle

$$F(\Theta, x)$$

. initialisation aléatoire des paramètres  $\Theta$ ;

**tant que** *La minimisation converge?* **faire**

Mélanger l'ensemble  $(X, y)$ ;

**pour chaque** *mini-lot*  $x_i, y_i \in (X, y)$  **faire**

(1) Calculer la prédiction  $\bar{y}$ ;

$$\bar{y}_i = F(\Theta, x_i)$$

(2) Ajuster les paramètres  $\Theta$ ;

$$\Theta = \Theta - \lambda \cdot \frac{1}{N} \sum_{n=1}^N \frac{\partial \epsilon(y_i, \bar{y}_i)}{\partial \Theta}$$

**fin**

**fin**

---

### 2.3.3. Optimisation par rétropropagation du gradient de l'erreur

La rétropropagation du gradient de l'erreur est employée pour calculer la descente de gradient sur des réseaux de neurones. La méthode calcule le gradient de l'erreur en fonction des paramètres du réseaux de manière récursive de la couche de sortie jusqu'à la première couche afin d'optimiser les paramètres. L'Algorithme 4 présente le pseudo-code de la méthode.

---

**Algorithme 4** : La rétropropagation du gradient de l'erreur

---

**Entrées** : Un réseau de neurones avec  $j$  couches; La fonction d'activation  $\varphi_j(\cdot)$ ; Les sorties des couches cachées  $h_j = \varphi_j(W_j^T \cdot h_{j-1} + b_j)$ ; et la sortie du réseau  $\bar{y} = h_j$ .

Calculer le gradient de la couche de sortie:  $\delta \leftarrow \frac{\partial \epsilon(y, \bar{y})}{\partial y}$

**pour**  $i \leftarrow j$  à 1 **faire**

(1) Calculer le gradient de la couche courante pour les poids et les biais:

$$\frac{\partial \epsilon(y, \bar{y})}{\partial W_j} = \frac{\partial \epsilon(y, \bar{y})}{\partial h_j} \frac{\partial h_j}{\partial W_j} = \delta \frac{\partial h_j}{\partial W_j}$$

$$\frac{\partial \epsilon(y, \bar{y})}{\partial b_j} = \frac{\partial \epsilon(y, \bar{y})}{\partial h_j} \frac{\partial h_j}{\partial b_j} = \delta \frac{\partial h_j}{\partial b_j}$$

(2) Appliquer l'algorithme de la descente du gradient:

$$W_j \leftarrow W_j - \lambda \frac{\partial \epsilon(y, \bar{y})}{\partial W_j}$$

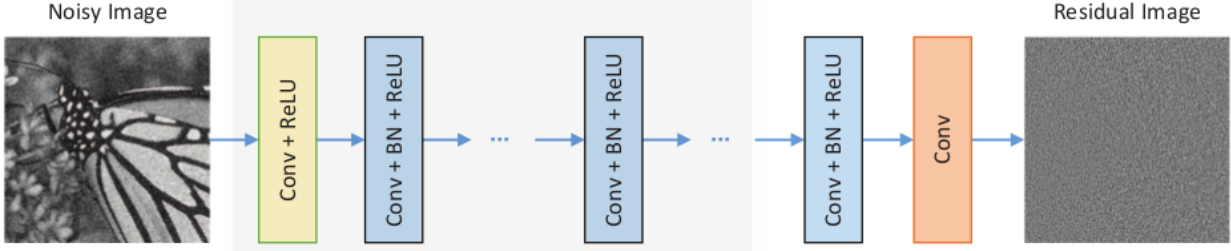
$$b_j \leftarrow b_j - \lambda \frac{\partial \epsilon(y, \bar{y})}{\partial b_j}$$

(3) Rétropropager le gradient à la couche supérieure  $j - 1$ :

$$\delta \leftarrow \frac{\partial \epsilon(y, \bar{y})}{\partial h_j} \frac{\partial h_j}{\partial h_{j-1}} = \delta \frac{\partial h_j}{\partial h_{j-1}}$$

**fin**

---



**Fig. 2.21.** L'architecture du réseau Dn-CNN.

### 2.3.4. Exemples de Réseaux Neurons à convolution appliqués à l'imagerie RSO

Dans cette section nous présentons quelques réseaux proposés dans la littérature qui nous semblent des approches intéressantes pour le filtrage des images **RSO**.

#### 2.3.4.1. Le filtre Dn-CNN

Le réseau Dn-CNN (*Denoising convolutional neural network*) proposé par Zhang et al. [42] permet l'apprentissage intégral d'un filtre de débruitage des images affectées par un bruit additif de type gaussien ( $y = x + v$ ). À la différence des réseaux qui font l'apprentissage d'un modèle direct pour la prédiction de l'image débruitée de la forme  $x = F(y)$ . Ce réseau adopte plutôt la modélisation des résidus  $R(y) \approx v$ , ce qui se traduit par l'équation de prédiction suivante  $x = y - R(y)$ . L'apprentissage est simplement obtenu par la minimisation de l'erreur quadratique entre les résidus désirés ( $y_i - x_i$ ) et estimés  $R(y_i)$ . Il en découle la fonction de perte suivante:

$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^N \| R(y_i; \theta) - (y_i - x_i) \|^2 \quad (2.3.2)$$

où  $\Theta$  représente les paramètres du réseau,  $\{(y_i, x_i)\}_{i=1}^N$  forme l'ensemble des paires d'image corrompue et propre.

La Fig. 2.21 présente l'architecture du réseau pour réaliser la modélisation des résidus. Le réseau de profondeur  $D$  est composé de trois types de blocs de traitement. 1) L'entrée du réseau (*Conv-ReLU*) est formée par une couche de 64 filtres convolutifs de taille  $3 \times 3 \times c$ , où  $c$  est le nombre de canaux d'entrée. Elle génère 64 cartes de caractéristiques qui sont ingérées par une couche d'activation de rectification linéaire (ReLU) pour introduire la non-linéarité. 2) Les couches cachées sont composées par  $D - 2$  blocs de la forme (*Conv-ReLU-BN*): une couche de 64 filtres convolutifs  $3 \times 3 \times 64$ , suivi de la fonction non-linéaire ReLU et se finalisant

Methods	BM3D	WNNM	EPLL	MLP	CSF	TNRD	DnCNN-S	DnCNN-B
$\sigma = 15$	31.07	31.37	31.21	-	31.24	31.42	<b>31.73</b>	31.61
$\sigma = 25$	28.57	28.83	28.68	28.96	28.74	28.92	<b>29.23</b>	29.16
$\sigma = 50$	25.62	25.87	25.67	26.03	-	25.97	<b>26.23</b>	<b>26.23</b>

**Tab. 2.1.** Résultats comparatifs des deux apprentissages du DnCNN avec des méthodes conventionnelles de pointe. De manière globale, les méthodes par apprentissage profond performant mieux que les méthodes conventionnelles

par une couche de normalisation par lot (BN, pour Batch Normalisation) afin d’améliorer et d’accélérer l’apprentissage. Le réseau se termine en sortie par une couche de  $c \times 3 \times 3 \times 64$  filtres qui apprend le résidu pour débruiter l’image.

Deux apprentissages sont effectuées à partir d’échantillons produits à partir de 400 images en ton de gris. Le premier (Dn-CNN-S) consiste à faire l’apprentissage du réseau sur des niveaux de bruit spécifiés d’avance:  $\sigma = 15, 25$  et  $50$ , avec une taille de *patch* de  $40 \times 40$  et 1600 échantillons pour l’entraînement. Le second (DnCNN-B) est fait à l’aveugle, i.e que le niveau de bruit est aléatoire avec 3000 échantillons. Suite à plusieurs expérimentations, les auteurs proposent une profondeur optimale  $D = 17$ . L’optimisation est faite par l’algorithme du gradient stochastique (SGD, stochastic gradient descent) avec une valeur du *momentum* égale 0.9 et une taille de 128 échantillons par mini-lots. Un *weight decay* égale 0.0001 est appliqué pour pénaliser le surapprentissage. Le paramètre du taux d’apprentissage est fixé à  $10^{-1}$  avec une décroissance exponentielle avec une valeur planché de  $10^{-4}$ . Le réseau est entraîné sur 50 époques.

Les résultats des deux apprentissages se comparent favorablement par rapport aux méthodes conventionnelles de pointe comme BMD3, WNNM, EPLL, TNRD. Les deux réseaux obtiennent des pointages supérieurs en terme de PSNR sur tous les niveaux de bruits testés sur l’ensemble de données BSD68 (voir Tab. 2.1). Sur d’autres ensembles d’images de référence, DnCNN-B se démarque particulièrement pour le niveau de bruit le plus élevé et DnCNN-S pour les autres. Les auteurs démontrent aussi que l’architecture du réseau s’utilise pareillement pour résoudre d’autres tâches de restauration d’images, comme la super-résolution et le filtrage anti-bloc pour les artéfacts sur les images au format JPEG.



### 2.3.4.2. Le filtre FFDNet

Le réseaux FFDNet (*Fast and Flexible Denoising Net*) proposé par Zhang et al. [43] permet l'apprentissage intégral d'un modèle de débruitage des images affectées par un bruit additif de type gaussien ( $y = x+v$ ). A la différence de la plupart des méthodes qui apprennent un modèle pour un certain niveau de bruit en particulier, FFDNet considère celui-ci comme une variable supplémentaire au modèle, ce qui lui permet une plus grande flexibilité et de mieux généraliser dans les cas de bruits fluctuants dans les images. Un canal  $M$  est ajouté en entrée au réseau avec le niveau de bruit  $\sigma$  associé à  $y$ ,  $\sigma$  peut être constant ou variable. L'apprentissage ne se fait pas sur les résidus mais sur le modèle directe  $x = F(y, M)$ . De plus, FFDNet travaille sur des sous-images sous-échantillonnées des images originales, réalisant un bon compromis entre la vitesse d'inférence et la performance du débruitage. L'apprentissage est fait par la minimisation de l'erreur quadratique entre les images désirées ( $x$ ) et estimées  $F(y_i, M_i)$ . Il en découle la fonction de perte suivante:

$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^N \| F(y_i; M_i; \theta) - x_i \|^2 \quad (2.3.3)$$

où  $\Theta$  représente les paramètres du réseau,  $\{((y_i, M_i), x_i)\}_{i=1}^N$  forme l'ensemble des paires (image cible + image  $\sigma$ ) et image étiquette.

La Fig. 2.22 présente l'architecture du réseau. L'image d'entrée subit une réduction d'échelle d'un facteur 2 et les 4 sous-images décalées sont réorganisées pour former une nouvelle pile de taille  $\frac{W}{2} \times \frac{H}{2} \times 4 \times c$ , où  $c$  est le nombre de canaux de l'image. À cette pile est enchaînée le canal  $M$  représentant le bruit. Dans le cas d'un bruit blanc gaussien  $\sigma$ , le canal  $M$  est uniforme. Le réseau compte au total  $4 \times c + 1$  canaux d'entrée. La partie convolutive de l'architecture est construite de la même manière que celle du DC-CNN. Après la dernière convolution, les  $4 \times c$  sous-images de sortie sont remises à l'échelle originale ( $W \times H \times c$ ) pour former l'images  $x$  débruitée. Les auteurs notent qu'il n'y a pas d'avantage à utiliser une architecture résiduelle car le réseau n'est pas assez profond.

L'apprentissage est fait par l'optimiseur ADAM (Kingmam [18]). L'hyperparamètre du taux d'apprentissage est initialisé à  $10^{-3}$  et est réduit à  $10^{-4}$  lorsque l'erreur de la régression cesse de diminuer. Après 5 époques de stagnation, une opération de fusion entre les couches BN et les couches convolutives qui leurs sont adjacentes est appliquée afin de réduire le nombre de paramètres à apprendre et en même temps le taux d'apprentissage est relaxé à  $10^{-6}$  pour

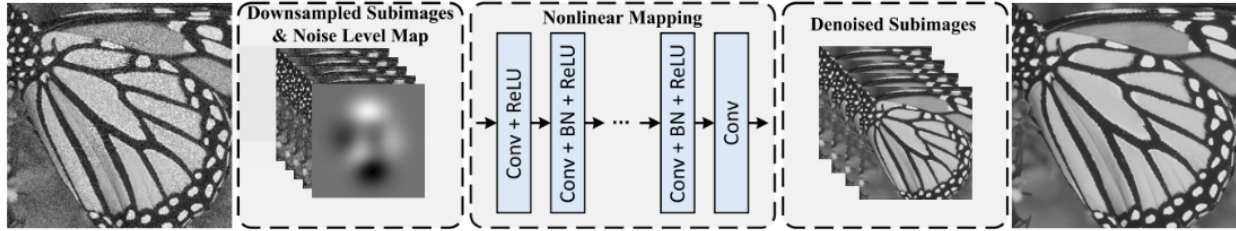


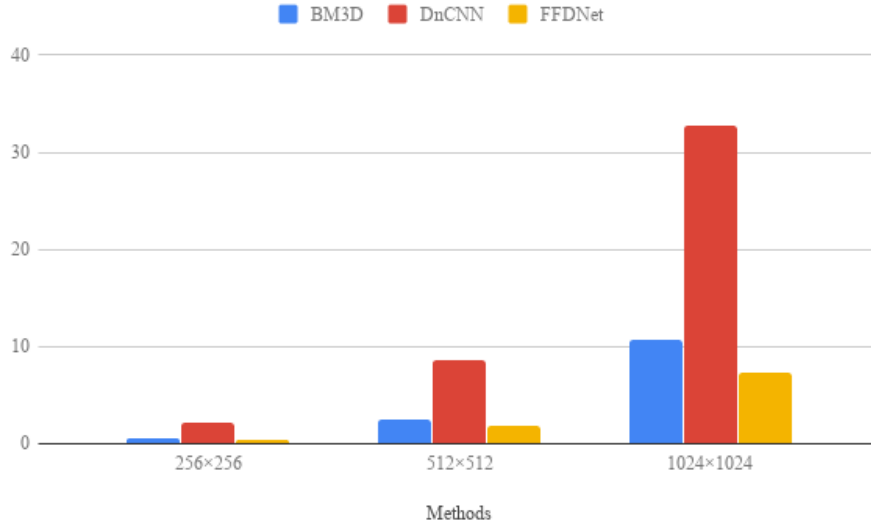
Fig. 2.22. L'architecture du réseau FFDNet.

Methods	BM3D	WNNM	MLP	TNRD	DnCNN	FFDNet
$\sigma = 15$	31.07	31.37	–	31.42	31.72	31.63
$\sigma = 25$	28.57	28.83	28.96	28.92	29.23	29.19
$\sigma = 35$	27.08	27.30	27.50	–	27.69	27.73
$\sigma = 50$	25.62	25.87	26.03	25.97	26.23	26.29
$\sigma = 75$	24.21	24.40	24.59	–	24.64	24.79

**Tab. 2.2.** Résultats comparatifs du FFDNet avec des méthodes conventionnelles de pointe. De manière générale FFDNET surpasse les autres approches pour des niveaux de bruit élevés ou obtient des résultats comparables à DnCNN, avec lequel il partage une architecture semblable

les 50 dernières époques. Les images utilisées pour l'entraînement proviennent de différents ensembles de référence (BSD, ImageNet, Waterloo Exploration Database). A partir des images en ton gris, des *patch* avec une taille de  $50 \times 50$  pixels sont extraites pour former les entrées et un bruit blanc gaussien est ajouté avec un niveau de bruit aléatoire ( $\sigma \in [0, 75]$ ) pour simuler la corruption. Une augmentation des données par des transformations de rotation et de *flip* horizontal et vertical est appliquée pour simuler un plus grand ensemble d'entraînement.

Les résultats sur l'ensemble de données de référence BSD68 montrent que FFDNET se compare favorablement par rapport aux compétiteurs (BMD3, WNNM, MLP, TNRD et DN-CNN) ou également à DN-CNN. Il performe mieux que les autres pour des niveaux de bruits élevés ( $\sigma \geq 35$ ) (voir Tab. 2.2). Une comparaison des temps d'exécution sur CPU (Single Thread) pour le débruitage d'images de tailles différentes démontre que FFDnet est plus rapide que BM3D et nettement plus rapide de DnCNN (voir Fig. 2.23).



**Fig. 2.23.** Comparaison des vitesses d’exécution (en secondes) sur CPU (Single Thread).

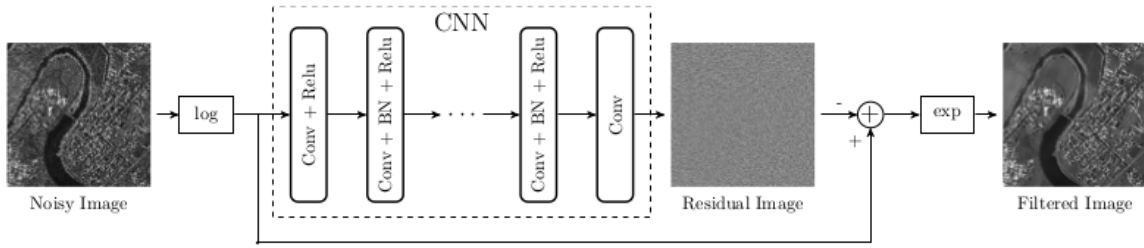
### 2.3.4.3. Le filtre SAR-CNN

Sur les traces du DN-CNN, Cherchia et al. [2] propose un réseau résiduel SAR-CNN (*SAR Despeckling Convolutional Neural Networks*) pour le filtrage du chatolement sur les images en intensité. Le réseau est composé de 17 couches convolutives. Chaque couches calculent 64 cartes de caractéristiques par des convolutions  $3 \times 3 \times 64$ . La différence fondamentale entre les deux approches est le rejet de la fonction de perte euclidienne, qui selon les auteurs n’est pas adaptée à la statistique du chatolement. Le chatolement est un bruit multiplicatif qui peut être modéliser comme  $y = x \cdot v$ , où  $v$  suit une loi gamma. L’approche SAR-CNN convertit le bruit multiplicatif en bruit additif par une transformation holomorphique ( $\log(y) = \log(x \cdot v) = \log(x) + w$ ). Après la dernière convolution la sortie est reconvertie par l’exponentielle (voir Fig. 2.24). La fonction de perte est adaptée pour tenir compte de la distribution du bruit du chatolement (par contre l’article n’explique pas très bien l’origine de cette métrique):

$$l(\Theta) = \sum_{i=1}^N \log(\cosh(R(y_i; \theta) + c - \log \frac{y_i}{x_i})) \quad (2.3.4)$$

où  $c$  est la valeur positive ( $c > 0$ ) de la moyenne de la distribution logarithmique du chatolement pour corriger le biais introduit par la transformation holomorphique.

Afin de résoudre le problème symptomatique de l’inexistence de données terrain sans bruit, les auteurs ont synthétisé des échantillons propres sans chatolement à partir de la



**Fig. 2.24.** L'architecture du réseau SAR-CNN.

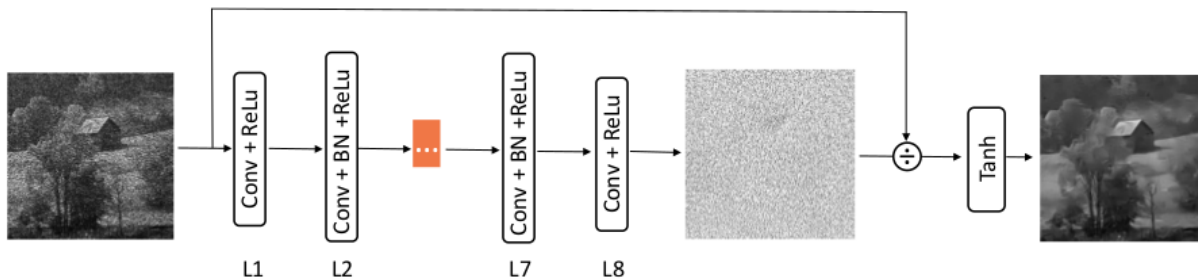
moyenne de 26 images **RSO** multi-temporelles coregistrées. Les  $2000 \times 128$  paires (propres, bruitées) sont formées par l'extraction de *patch*  $40 \times 40$  pixels, à la fois de l'image moyenne et des images bruitées. L'apprentissage du réseau s'est fait sur 50 époques avec la méthode d'optimisation ADAM et des mini-lots de taille 128. Le taux d'apprentissage est demeuré fixe sur 30 époques à  $10^{-3}$  et abaissé à  $10^{-4}$  sur les 20 dernières époques. La validation durant l'apprentissage est faite à partir des mesures PSNR et SSIM sur des image **RSO** simulées.

Les résultats du SAR-CNN sont comparés avec les 3 techniques de pointe couramment utilisées PPB, NL-SAR, SAR-BM3D. La comparaison est effectuée avec le meilleur modèle SAR-CNN sur des images **RSO** synthétiques. Comme le démontre le tableau 2.3, le SAR-CNN obtient de meilleurs résultats à la fois pour la réduction du chatoiement et la conservation des détails.

	PPN	NL-SAR	SAR-BM3D	SAR-CNN
PSNR moyen	23.43	23.98	24.99	<b>25.95</b>
SIMM moyen	0.639	0.677	0.729	<b>0.764</b>

**Tab. 2.3.** Résultats comparatifs du PSNR et du SSIM sur les images **RSO** simulées

Un autre apprentissage a été effectué à partir d'une pile de 25 images COSMO-SkyMed à une vue. La moitié de la pile a été utilisée pour faire l'entraînement en suivant la procédure décrite auparavant et l'autre moitié a servi à la validation. L'évaluation est faite en mesurant le nombre équivalent de vues (ENL, *equivalent number of looks*) et l'index  $\alpha\beta$  (Gomez, [12]) qui permet d'évaluer la performance des filtres sur les images **RSO**. SAR-CNN et NL-SAR donnent des résultats similaires en terme de mesure du ENL et de l'index  $\alpha\beta$ . Ces deux



**Fig. 2.25.** L'architecture du réseau ID-CNN.

méthodes déclasses les deux autres facilement (voir 2.4). En comparaison avec la première expérimentation faite à partir d'images simulées, l'apprentissage sur des images réelles semble moins performant. La perte semble provenir de la différence entre les images sans bruit et les images bien filtrées qui servent de référence pour l'optimisation du réseau.

regions	mesure	PPN	NL-SAR	SAR-BM3D	SAR-CNN
1	ENL	47.61	154.10	4.87	129.10
2	ENL	25.28	52.12	4.71	56.32
1	$\alpha\beta$	0.162	0.076	0.530	0.187
2	$\alpha\beta$	0.171	0.065	0.511	0.182

**Tab. 2.4.** Résultats comparatifs du ENL et de l'index  $\alpha\beta$  sur les images **RSO** réelles

#### 2.3.4.4. Le filtre ID-CNN

Comme ces compatriotes précédents, le filtre ID-CNN (*Image Despeckling Convolutional Neural Network*) proposé par Wang et al. [38] modélise les résidus afin de réduire le chatoiement des images **RSO**. Cependant, il n'applique pas de transformation holomorphe au signal d'entrée mais estime directement le chatoiement en utilisant le modèle du bruit multiplicatif ( $y = x \cdot v$ ). Assez inusité, l'opération résiduelle utilise la division pour calculer la valeur des résidus en divisant l'image d'entrée par l'estimation du chatoiement apprise par le réseau. Celui-ci est formé de 8 couches du même type que celles de l'architecture Dn-CNN. Avec seulement 8 couches, ce réseau est nettement moins profond que les autres (voir Fig. 2.25). Le Tab. 2.5 présente les détails de la configuration du réseau.

	couche	taille du filtre	nombre de filtre
L1	Conv-ReLU	$3 \times 3 \times 1$	64
L2-L7	Conv-BN-ReLU	$3 \times 3 \times 1$	64
L8	Conv-ReLU	$3 \times 3 \times 1$	1

**Tab. 2.5.** Configuration du réseau ID-CNN

	Metric	Noisy	Lee	Kuan	PPB	SAR-BM3D	CNN	SAR-CNN	ID-CNN
$L = 1$	PSNR	14.53	21.48	21.95	21.74	22.99	21.04	23.59	<b>24.74</b>
	SSIM	0.369	0.511	0.592	0.619	0.692	0.630	0.640	<b>0.727</b>
	UQI	0.374	0.450	0.543	0.488	0.591	0.560	0.561	<b>0.621</b>
	DG	-	16.01	17.08	14.30	17.17	14.98	20.87	<b>23.51</b>
$L = 4$	PSNR	18.49	22.12	22.84	23.72	24.96	22.60	26.20	<b>26.89</b>
	SSIM	0.525	0.555	0.650	0.725	0.782	0.722	0.771	<b>0.818</b>
	UQI	0.527	0.485	0.594	0.605	0.679	0.648	0.688	<b>0.723</b>
	DG	-	8.35	10.00	10.52	14.89	13.76	17.74	<b>19.33</b>
$L = 10$	PSNR	20.54	22.30	23.11	24.92	26.45	23.52	27.63	<b>28.07</b>
	SSIM	0.602	0.571	0.671	0.779	0.834	0.741	0.825	<b>0.853</b>
	UQI	0.599	0.498	0.613	0.678	0.745	0.683	0.741	<b>0.765</b>
	DG	-	4.06	5.93	7.75	13.61	7.82	16.32	<b>17.35</b>

**Tab. 2.6.** Résultats comparatifs de ID-CNN avec des méthodes traditionnelles en radar (Lee, Kuan) et conventionnelles de pointe (PPB, SAR-BM3D) et des approches en apprentissage machine profond (CNN=Dn-CNN, SAR-CNN).

L'apprentissage du réseau est fait en minimisant une composition de fonctions de pertes:

$$l(\Theta)_{total} = l(\Theta)_E + \lambda_{TV} \cdot l(\Theta)_{TV} \quad (2.3.5)$$

où  $l(\Theta)_E$  représente la fonction de perte euclidienne:

$$l(\Theta) = \frac{1}{2N} \sum_{i=1}^N \| R(y_i; \Theta) - (y_i - x_i) \|^2 \quad (2.3.6)$$

où  $\Theta$  correspond aux paramètres du réseau,  $\{(y_i, x_i)\}_{i=1}^N$  forme l'ensemble des paires d'image cibles et étiquettes et  $l(\Theta)_{TV}$  est la fonction de perte qui calcule la variation totale :

$$l_{TV}(\Theta) = \sum_{w=1}^W \sum_{h=1}^H \sqrt{((X^{w+1,h} - X^{w,h})^2 + (X^{w,h+1} - X^{w,h})^2)} \quad (2.3.7)$$

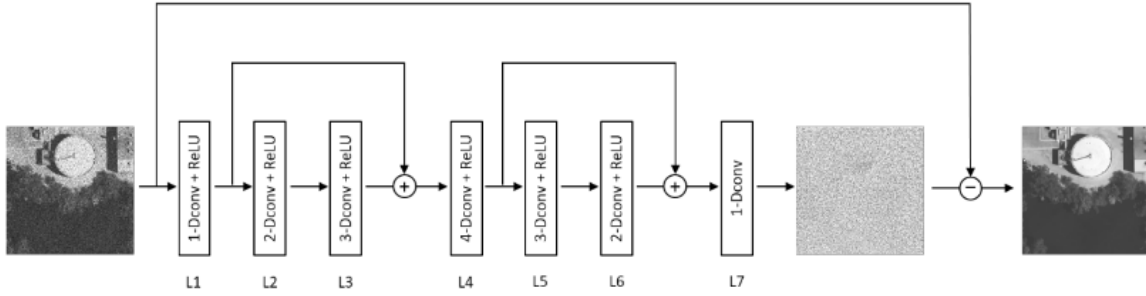
où  $w$  et  $h$  sont les index des pixels d'une image  $W \times H$  et  $X$  représente la valeur du pixel. Selon les auteurs, la fonction de perte euclidienne est responsable du filtrage du chatoiment

tandis que la fonction de perte de la variation totale est responsable du lissage. Le paramètre  $\lambda_{TV}$  pondère l'intensité du lissage. Ce paramètre est fixé à une valeur plus petite que 1 afin de ne pas perdre trop de détails au dépend du lissage. Le réseau est entraîné sur 3580 images de taille  $256 \times 256$  avec un bruit multiplicatif simulé en fonction du nombre de vues ( $L$ ). L'apprentissage des paramètres est calculé par l'optimiseur ADAM, avec des mini-lots de 16 et un taux d'apprentissage fixé à 0.0002. Le paramètre  $\lambda_{TV}$  est fixe à 0.002. Trois apprentissages ont été effectués en fonction du nombre de vues ( $L = 1, L = 4, L = 10$ ) et les performances ont été calculées sur 85 images indépendantes de l'ensemble d'entraînement.

L'évaluation de la performance est calculée à partir des 4 métriques suivantes: PSNR, SSIM, l'index de qualité universel (UQI) et le gain de suppression du chatolement (DG, *despeckling gain*). Les résultats obtenus sont étonnements bons. ID-CNN obtient les meilleurs scores toutes catégories de mesures confondues. De plus, grâce au fait que l'architecture soit peu profonde, la vitesse d'exécution sur gpu (0.56 sec.) est environ 5 fois plus rapide que son plus proche compétiteur Dn-CNN (2.42 sec.). Malheureusement les auteurs ne donnent pas d'explications satisfaisantes sur le choix de leur architecture et de la fonction de perte. A noter que Dn-CNN correspond à l'approche CNN dans l'article.

#### 2.3.4.5. *Le filtre SAR-DRN*

Les auteurs (Zhang et al. [44], un autre que ceux des articles précédents) propose un filtre SAR-DRN (*SAR Dilated Residus Network*) qui s'entraîne bout en bout avec des convolution dilatées (ou à trou). Ces convolutions ont un paramètre appelé le niveau de dilatation qui définit l'espace entre les éléments du filtre convolutif. Augmenter le niveau de dilatation permet d'augmenter le champs réceptif du ConvNet sans augmenter le nombre de paramètres. Élargir le champs réceptif permet à son tour de capturer plus de contexte autour d'un point et possiblement améliorer la capacité discriminatoire du réseau. Le réseau est composé de 7 couches de convolutions dilatées plus deux sauts de connections (voir Fig. 2.26). L'ajout des sauts de connections aide à la convergence de l'apprentissage. Celui-ci se fait sans utiliser de transformation logarithmique (SAR-CNN) et sans modification de la fonction de perte (ID-CNN). La dilatation des convolutions suit le schème telle que décrit par le Tab .2.7. L'apprentissage suit le même procédé que dans le cas de Dn-CNN avec la minimisation de la fonction de perte euclidienne sur les résidus (voir 2.3.2). L'optimiseur utilisé est toujours



**Fig. 2.26.** L'architecture du réseau SAR-DRN

ADAM et le taux d'apprentissage est initialisé à 0.01 et est diminué à toute les 10 époques sur 50. Plusieurs modèles ont été entraînés pour les quatre niveaux de bruit radar (L=1, 2, 4 et 8).

couche	niveau de dilatation	bloc	padding des convolutions	filtres
L1	1	1	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L2	2	2	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L3	3	3	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L4	4	4	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L5	3	3	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L6	2	2	Convolution Dilatée + ReLU	$64 \times 3 \times 3$
L7	1	1	Convolution Dilatée	$64 \times 3 \times 3$

**Tab. 2.7.** Description de réseau SAR-DRN

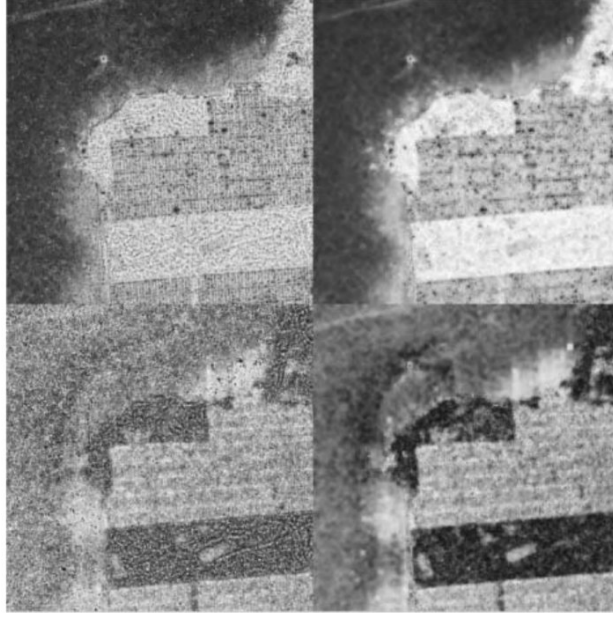
Afin de tester leur algorithme, les auteurs ont sélectionné les 4 approches suivantes: PPB, SAR-BM3D, SAR-POTDF et SAR-CNN. L'évaluation de la performance est calculée à partir des 2 métriques suivantes: PSNR, SSIM sur trois images provenant de différents milieux.

### 2.3.5. Exemples de Réseaux Neurons à convolution appliqués à l'imagerie RSO-POL

La littérature sur le filtrage des images polarimétriques à l'aide de réseaux à convolution est plus rare. Nous ne présentons que quelques exemples.

Quelques articles semblent directement s'intéresser à la question (Foucher et al. [7]). Les auteurs explorent en particulier l'impact du filtrage sur les mesures polarimétriques de la





**Fig. 2.27.** L'entropie (rangée supérieure) et l'anisotropie (rangée inférieure) pour un filtre de Lee  $7 \times 7$  et pour l'approche de (Foucher et al. [7])

décomposition H-A-Alpha. L'apprentissage du modèle est calculé sur les valeurs du **span** de la matrice de cohérence  $[T_3]$  sur des données simulées 1 look. Le réseau entraîné est adapté de l'architecture de (Dong, [6]) utilisé en super-résolution. Le modèle est par la suite employé dans un filtre itératif. Les tests sur l'image AIRSAR de San-Francisco avec une décomposition polarimétrique de Cloud-Pottier sont encourageants. Ils montrent une bonne préservation des paramètres tout en filtrant le chatoiement et sans perdre trop de détails (voir Fig. 2.27). Les auteurs concluent que malgré de bon résultats, le choix d'une bonne architecture neuronale pour le filtrage polarimétrique demeure un problème ouvert. De plus la préservation des cibles ponctuelles posent toujours un problème à cause de l'aspect convolutionnel du réseaux. Ce qui soulève d'autres questions du point de vue de l'ensemble de données pour l'apprentissage et de sa représentativité du point de vue polarimétrique.

Par contre, on retrouve certaines études en classifications polarimétriques qui indirectement font du filtrage. Les auteurs Li et al [23] proposent un réseau de type FCN pour la classification des images polarimétriques basé sur les paramètres de Cloud-Pottier. La matrices de covariance  $[T_3]$  est fournie en entrée du réseau pour estimer les paramètres

$(H, A, \bar{\alpha}, \lambda_1, \lambda_2\lambda_3)$ . Le filtrage est indirecte dans le sens que pour obtenir des valeurs débruitées en sortie, le réseau a appris comment filtrer le chatoiement. Un véritable filtrage serait d'estimer la matrice  $[T_3]$  débruitée directement.

## 2.4. Conclusion

Nous avons présenté un survol de la théorie de la polarimétrie appliquée aux images satellites **RSOPOL** ainsi qu'un les grandes lignes de l'apprentissage machine profond et l'utilisation des **RNC** pour l'analyse des images. Avec ces connaissances nous pouvons entreprendre la mise en place des différents aspects des expérimentations et tests que nous voulons effectuer.

# Chapitre 3

---

## MÉTHODOLOGIE ET DONNÉES EXPÉRIMENTALES

Ce chapitre présente la méthodologie pour produire les données nécessaires à l'apprentissage ainsi que pour l'évaluation des modèles générés. Il présente aussi une description des données en QuadPol réelles que nous utilisons pour évaluer les modèles issus des nombreux apprentissages et finalement, il décrit les outils matériels nécessaires à la réalisation du projet.

### 3.1. Données synthétiques

L'image de référence ou image étiquette (sans chatolement) n'est pas disponible en **RA-DAR**, ce qui résulte en une incapacité de réaliser une évaluation précise des effets des filtrages à l'aide d'une vérité terrain "réelle". Afin de générer les ensembles d'entraînement et de test composés de paires d'images caractéristiques et de leurs étiquettes, nous avons simulé le chatolement sur des images synthétiques. La simulation du chatolement des images **RSO** en amplitude ou en intensité est assez simple à produire. La partie réelle et imaginaire du retour du signal **RSO** sur les images une vue suivent chacune une distribution normale centrée en zéro et de variance  $\sigma/2$  (voir Sec. 2.1.2).

#### 3.1.1. Génération de données polarimétriques

Dans le cas des données polarimétriques, nous devons tenir compte de la corrélation entre les trois polarisations du vecteur  $\vec{k}_C$  (Eq. 2.17) . Pipia et Fàbregas [29] proposent une procédure qui permet de produire à partir de mesures réelles de matrices de covariance, des échantillons avec le chatolement corrélé simulé. On procède selon les étapes suivantes:

(1) Calculer la matrice triangulaire  $[F]$  tel que

$$[C_3] = [F] [F]^H \quad (3.1.1)$$

par la méthode de factorisation de Cholesky. La matrice  $[F]$  est en quelque sorte une «racine carrée» de  $[C_3]$ ;

(2) Générer un vecteur aléatoire  $\vec{v}$  suivant une loi normal complexe centrée en zéro. La partie réelle et imaginaire du vecteur est générée indépendamment à partir d'une distribution normale centrée en zéro et de variance égale 0.5;

(3) Calculer le vecteur de polarisation  $\vec{k}_C$  tel que

$$\vec{k}_C = [F] \vec{v} \quad (3.1.2)$$

(4) Calculer la matrice de covariance multivue simulée

$$[C_3]' = \frac{1}{L_n} \sum_{i=1}^{L_n} (\vec{k}_{iC} \cdot \vec{k}_{iC}^H) \quad (3.1.3)$$

où  $L_n$  est le nombre de vues. Dans notre cas, le nombre de vues est égale à 1 afin de simuler des données une vue.

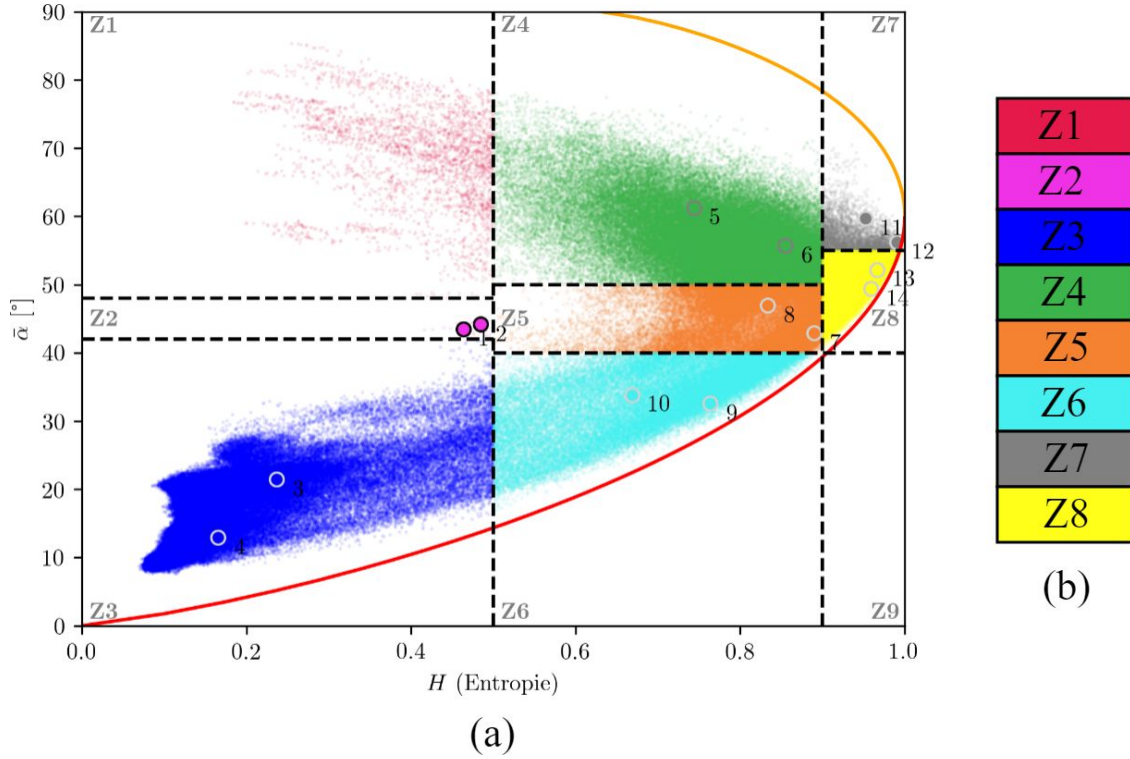
### 3.1.2. Signatures polarimétriques tests

Les signatures polarimétriques ( $[T_3]$ ) ont été calculées à partir de l'image de démonstration NASA/JPL AIRSAR en Bande L et un nombre de vues égal à 4. Les signatures on été produites en sélectionnant initialement deux mesures représentatives pour chaque classe de diffuseur sur le plan  $H/\bar{\alpha}$  (Fig. 3.1).

La sélection des signatures a été faite automatiquement par la procédure suivante:

- (1) Effectuer une décomposition  $H/Alpha/\bar{\alpha}$  de l'image en QuadPol AIRSAR;
- (2) Effectuer une partitionnement par k-moyennes à N classes pour chacune des classes  $H/\bar{\alpha}$  de diffuseur;

Les paramètres de l'algorithme de k-moyennes ont été fixés à N=2 et le nombre d'itérations à 10. Le clustering a été appliqué sur les 9 termes uniques de la matrice  $[T_3]$ . Les signatures de test correspondent aux centroïdes de chacun des clusters résultants. Le Tableau 3.1 présente l'ensemble des matrices  $[T_3]$  ainsi calculées et normalisées par le *span*. Dans le cas de la classe des diffuseurs ponctuels purs (Z1) nous utilisons la mesure donnée



**Fig. 3.1.** Diagramme  $H/\bar{\alpha}$  de l'image AIRSAR de la région de San Francisco. (a) Les marqueurs numérotés de 1 à 14 représente la localisation sur la plan  $H/\bar{\alpha}$  de chacune des matrices de cohérence sélectionnées pour les tests (voir Tab. 3.1). (b) Légende des couleurs associées à chacune des classes de diffuseur.

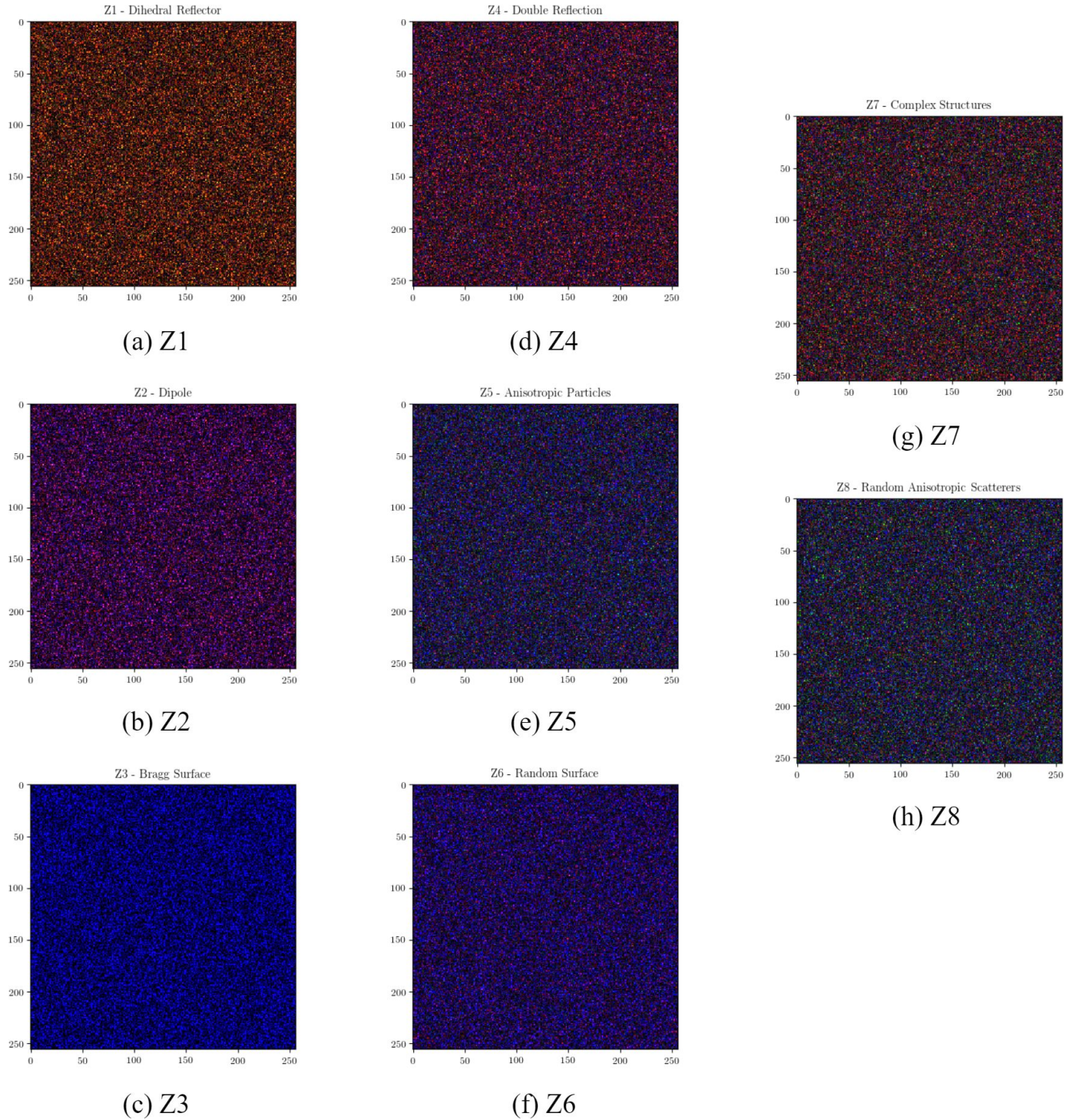
par [8]. Cette matrice correspond à une matrice de rang 1 représentant la signature d'un double-bond. Elle est non-définie sur le plan  $H/\bar{\alpha}$ .

### 3.1.3. Simulation de régions homogènes

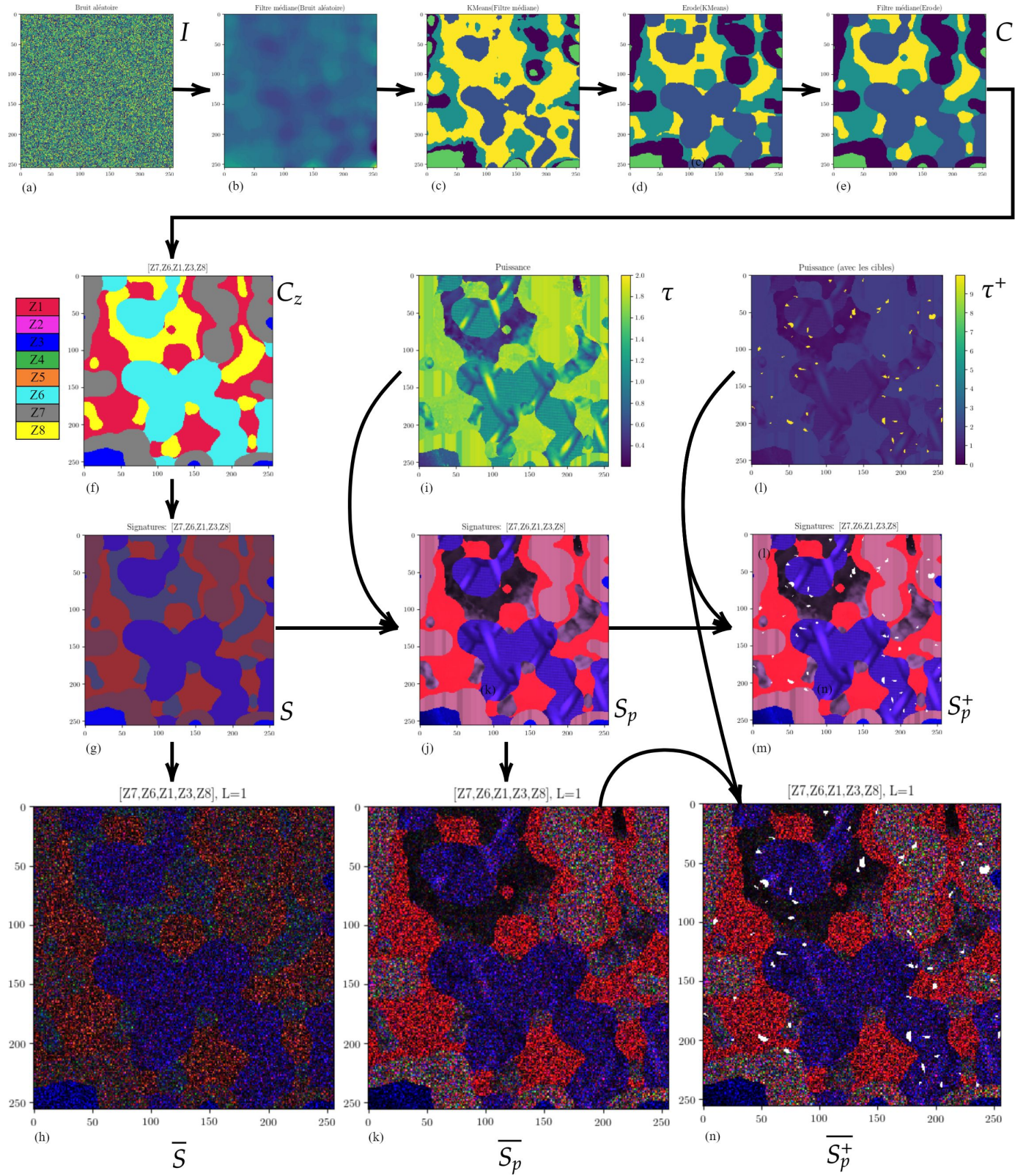
Des images caractéristiques et des images étiquettes de taille  $256 \times 256$  pour chaque signature polarimétrique ont été générées. La Figure 3.2 présente quelques exemples de simulation sur des régions homogènes produites à partir des matrices de cohérence calculées. Chaque imagerie présente le composé de Pauli normalisé. Le nombre équivalent de vues (**NEV**) a été vérifié sur les trois bandes de puissance ( $T_{11}, T_{22}, T_{33}$ ) de la matrice de cohérence  $[T_3]$ . Ils sont consistants avec la valeur théorique pour les images une vue: **NEV** = 1 (voir Tab. 3.2).

<b>Z1</b>	$\begin{bmatrix} 0.529 & -0.310 - 0.137i & -0.005 - 0.105i \\ -0.310 + 0.137i & 0.400 & 0.092 + 0.070i \\ -0.005 + 0.105i & 0.092 - 0.070i & 0.071 \end{bmatrix}$ $\mathbf{H} = NA, \mathbf{A} = NA, \bar{\alpha} = NA$	—————
<b>Z2</b>	$\begin{bmatrix} 0.529 & -0.310 - 0.137i & -0.005 - 0.105i \\ -0.310 + 0.137i & 0.400 & 0.092 + 0.070i \\ -0.005 + 0.105i & 0.092 - 0.070i & 0.071 \end{bmatrix}$ $\mathbf{H} = 0.46, \mathbf{A} = 0.70, \bar{\alpha} = 43.5^\circ$	$\begin{bmatrix} 0.517 & -0.275 - 0.170i & 0.006 - 0.114i \\ -0.275 + 0.170i & 0.403 & 0.102 + 0.081i \\ 0.006 + 0.114i & 0.102 - 0.081i & 0.079 \end{bmatrix}$ $\mathbf{H} = 0.49, \mathbf{A} = 0.71, \bar{\alpha} = 44.2^\circ$
<b>Z3</b>	$\begin{bmatrix} 0.855 & -0.255 - 0.038i & 0.033 - 0.062i \\ -0.255 + 0.038i & 0.117 & -0.005 + 0.020i \\ 0.033 + 0.062i & -0.005 - 0.020i & 0.028 \end{bmatrix}$ $\mathbf{H} = 0.24, \mathbf{A} = 0.26, \bar{\alpha} = 21.5^\circ$	$\begin{bmatrix} 0.933 & -0.134 - 0.051i & 0.059 - 0.059i \\ -0.134 + 0.051i & 0.046 & -0.004 + 0.012i \\ 0.059 + 0.059i & -0.004 - 0.012i & 0.021 \end{bmatrix}$ $\mathbf{H} = 0.16, \mathbf{A} = 0.28, \bar{\alpha} = 12.9^\circ$
<b>Z4</b>	$\begin{bmatrix} 0.361 & 0.014 - 0.011i & 0.056 - 0.018i \\ 0.014 + 0.011i & 0.480 & 0.127 + 0.017i \\ 0.056 + 0.018i & 0.127 - 0.017i & 0.159 \end{bmatrix}$ $\mathbf{H} = 0.86, \mathbf{A} = 0.55, \bar{\alpha} = 55.7^\circ$	$\begin{bmatrix} 0.283 & 0.062 + 0.017i & 0.056 - 0.009i \\ 0.062 - 0.017i & 0.587 & 0.165 + 0.017i \\ 0.056 + 0.009i & 0.165 - 0.017i & 0.131 \end{bmatrix}$ $\mathbf{H} = 0.74, \mathbf{A} = 0.59, \bar{\alpha} = 61.3^\circ$
<b>Z5</b>	$\begin{bmatrix} 0.548 & -0.047 - 0.016i & 0.061 - 0.027i \\ -0.047 + 0.016i & 0.257 & 0.003 - 0.013i \\ 0.061 + 0.027i & 0.003 + 0.013i & 0.196 \end{bmatrix}$ $\mathbf{H} = 0.89, \mathbf{A} = 0.18, \bar{\alpha} = 42.9^\circ$	$\begin{bmatrix} 0.479 & 0.015 - 0.024i & 0.057 - 0.035i \\ 0.015 + 0.024i & 0.411 & 0.076 + 0.009i \\ 0.057 + 0.035i & 0.076 - 0.009i & 0.110 \end{bmatrix}$ $\mathbf{H} = 0.83, \mathbf{A} = 0.66, \bar{\alpha} = 47.0^\circ$
<b>Z6</b>	$\begin{bmatrix} 0.672 & -0.011 - 0.025i & 0.060 - 0.033i \\ -0.011 + 0.025i & 0.191 & 0.006 - 0.009i \\ 0.060 + 0.033i & 0.006 + 0.009i & 0.137 \end{bmatrix}$ $\mathbf{H} = 0.76, \mathbf{A} = 0.21, \bar{\alpha} = 32.6^\circ$	$\begin{bmatrix} 0.677 & -0.164 - 0.010i & 0.049 - 0.045i \\ -0.164 + 0.010i & 0.236 & -0.002 + 0.007i \\ 0.049 + 0.045i & -0.002 - 0.007i & 0.087 \end{bmatrix}$ $\mathbf{H} = 0.67, \mathbf{A} = 0.40, \bar{\alpha} = 33.9^\circ$
<b>Z7</b>	$\begin{bmatrix} 0.330 & -0.004 - 0.005i & 0.033 - 0.013i \\ -0.004 + 0.005i & 0.440 & 0.068 + 0.003i \\ 0.033 + 0.013i & 0.068 - 0.003i & 0.230 \end{bmatrix}$ $\mathbf{H} = 0.95, \mathbf{A} = 0.25, \bar{\alpha} = 59.7^\circ$	$\begin{bmatrix} 0.341 & -0.037 - 0.023i & 0.030 - 0.021i \\ -0.037 + 0.023i & 0.325 & 0.014 - 0.000i \\ 0.030 + 0.021i & 0.014 + 0.000i & 0.334 \end{bmatrix}$ $\mathbf{H} = 0.99, \mathbf{A} = 0.10, \bar{\alpha} = 56.2^\circ$
<b>Z8</b>	$\begin{bmatrix} 0.404 & 0.001 - 0.024i & 0.035 - 0.017i \\ 0.001 + 0.024i & 0.367 & 0.032 - 0.002i \\ 0.035 + 0.017i & 0.032 + 0.002i & 0.229 \end{bmatrix}$ $\mathbf{H} = 0.97, \mathbf{A} = 0.25, \bar{\alpha} = 52.2^\circ$	$\begin{bmatrix} 0.464 & -0.016 - 0.025i & 0.037 - 0.020i \\ -0.016 + 0.025i & 0.278 & 0.005 - 0.008i \\ 0.037 + 0.020i & 0.005 + 0.008i & 0.257 \end{bmatrix}$ $\mathbf{H} = 0.96, \mathbf{A} = 0.07, \bar{\alpha} = 49.3^\circ$

**Tab. 3.1.** Les matrices de cohérence  $[T_3]$  simulées et leurs paramètres  $H/A/\bar{\alpha}$  pour chacune des classes de diffusion. La classe **Z1** représente la classe des cibles ponctuelles. Elle est non définie selon les paramètres de Cloude-Pottier.



**Fig. 3.2.** Exemples des simulations de chatoiement une vue ( $N = 1$ ) de données polarimétriques sur des régions homogènes. Les figures de (a) à (h) représentent le composé de Pauli de chacune des signatures polarimétriques sur le diagramme  $H/\bar{\alpha}$ .



**Fig. 3.3.** Schéma des étapes de production des simulations.



	<b>Z1</b>	<b>Z2</b>	<b>Z3</b>	<b>Z4</b>	<b>Z5</b>	<b>Z6</b>	<b>Z7</b>	<b>Z8</b>
$T_{11}$	1.006	0.991	0.994	1.006	0.999	0.996	1.007	0.993
$T_{22}$	1.001	0.995	0.997	1.006	1.008	1.001	0.998	1.006
$T_{33}$	1.001	0.993	1.005	1.004	1.002	1.005	1.000	0.998

**Tab. 3.2.** Nombre de vues équivalent (**NEV**) calculé sur quelques simulations

### 3.1.4. Simulation de régions hétérogènes (patchwork)

Trois types de simulation sont générés de manière aléatoire à partir des signatures polarimétriques du tableau 3.1, soient:

- (1) Des simulations où la puissance demeure uniforme, ce qui génère des images avec des régions variées mais homogènes (Fig. 3.3-(h));
- (2) Des simulations où la puissance est donnée par la variation de l'intensité d'images de texture, ce qui génère des images avec des régions variées avec des intensités variables (Fig. 3.3-(k));
- (3) Des simulations avec inclusions de cibles ponctuelles, ce qui génère des images parsemées de points dont la puissance est supérieure de plusieurs ordres de magnitude que la puissance de leur voisinage (Fig. 3.3-(n));

La méthode de création des simulations polarimétriques se divise en 3 étapes principales (Fig. 3.3).

#### 3.1.4.1. Étape 1 - Génération aléatoire d'une carte de régions homogènes

Cette étape consiste à créer à partir d'une image de valeurs aléatoires, une carte partitionnée formée de régions homogènes. Ces régions sont obtenues à partir d'un clustering et de transformations morphologiques appliquées à l'image. Le résultat de chacune des sous-étapes est illustré par les sous-figures (a)-(e) sur la figure principale 3.3.

- (1) Générer une image  $I$  aléatoire de taille  $m \times m$  à partir d'une distribution uniforme sur l'intervalle  $[0, 255]$ . La dimension  $m$  est fixée à  $m = 256$  (Fig. 3.3-(a));
- (2) Filtrer  $I$  par un filtre de la médiane large ( $31 \times 31$ ). Le filtrage aide l'étape suivante (Fig. 3.3-(b));

- (3) Partitionner  $I$  en régions en effectuant un apprentissage  $k$  – moyennes non-supervisé à  $N$  classes. Le nombre de classes est inférieur ou égal au nombre de classes de diffuseur de la décomposition  $H/A/\bar{\alpha}$ :  $N \leq 8$ . L'image  $C$  résultante est composée de régions appartenant à  $K$  classes numérotées de 0 à  $K - 1$ ,  $K$  pouvant être égal ou inférieur à  $N$  (Fig. 3.3-(c));
- (4) Appliquer un filtre morphologique avec l'opérateur d'érosion sur l'image  $C$  pour éliminer les régions trop petites (Fig. 3.3-(d));
- (5) Appliquer un second filtre de la médiane sur l'image  $C$  pour adoucir les contours (Fig. 3.3-(e));

#### 3.1.4.2. Étape 2 - Génération de l'image étiquette polarimétrique

Cette étape consiste à produire l'image étiquette  $S$  de dimension  $9 \times m \times m$ . Les neuf canaux correspondent aux neuf termes uniques de la matrice  $[T_3]$  des signatures polarimétriques. La forme générale de la matrice  $[T_3]$  est donnée par:

$$[T_3] = \begin{bmatrix} A & B + jC & D + jE \\ B - jC & F & G + jH \\ D - jE & G - jH & I \end{bmatrix} \quad (3.1.4)$$

où  $j$  est le nombre complexe  $\sqrt{-1}$  et les coefficients  $\{A, B, C, D, E, F, G, H, I\}$  sont les neuf termes uniques que nous noterons par l'ensemble  $\{T_9\}$ .

- (1) Assigner aux  $K$  classes de l'image  $C$  une seule des huit classes de diffuseur polarimétrique  $\{Z1, Z2, Z3, Z4, Z5, Z6, Z7, Z8\}$  de façon aléatoire. Cette nouvelle image  $C_z$  donne pour chacun des pixels l'index de la classe polarimétrique assignée. Elle est présentée par la sous-figure 3.3-(f). La couleur de chacune des classes polarimétriques correspond aux couleurs de la Figure 3.1.
- (2) Générer l'image  $S$  des étiquettes de signatures. Les valeurs  $\{T_9\}$  de la classe polarimétrique déterminée par l'image  $C_z$  sont assignées aux pixels de l'image  $S$ . La sous-figure 3.3-(g) présente un exemple du composé de Pauli de l'image  $S$ .



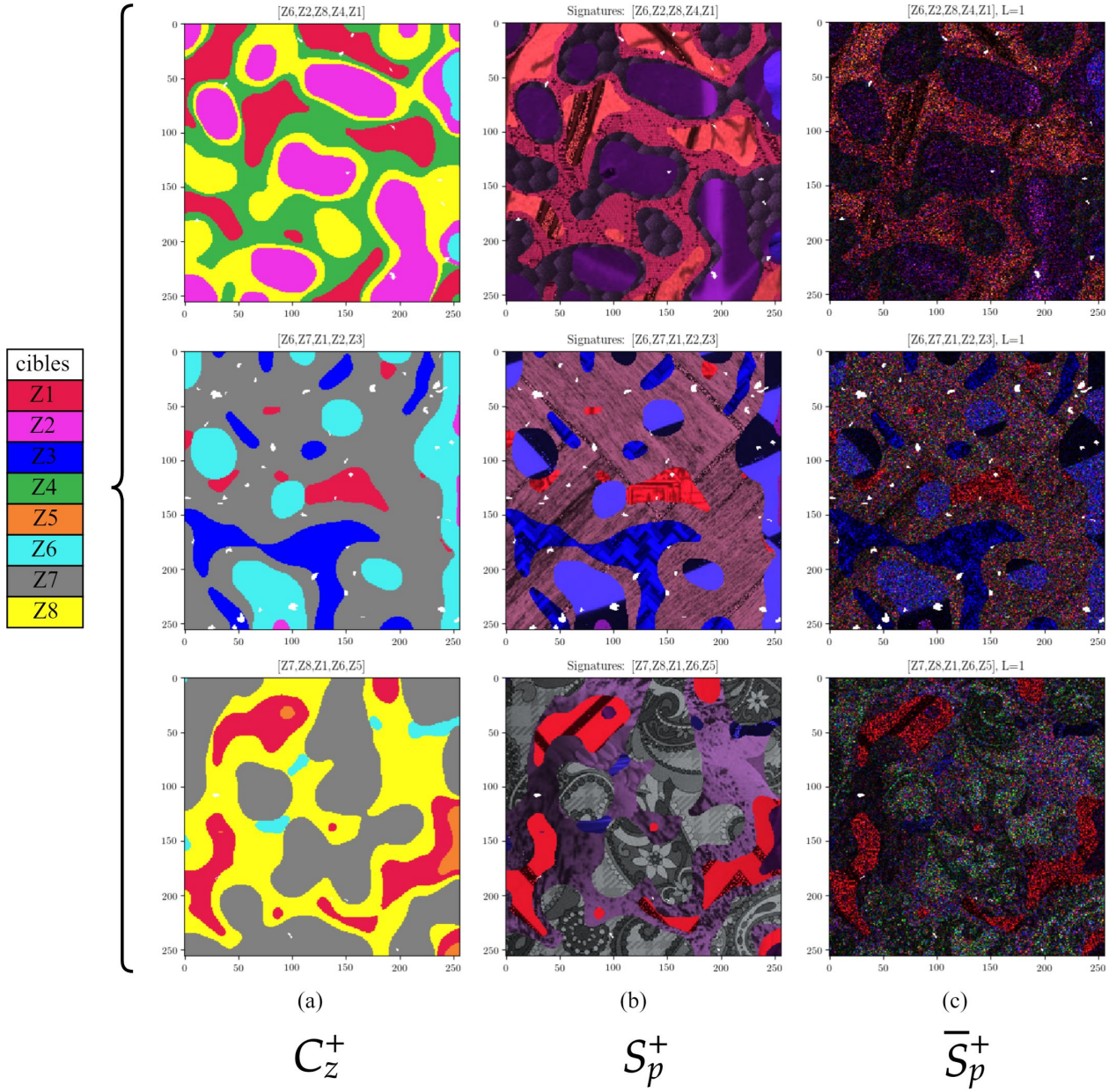
**Fig. 3.4.** Exemples de texture de l'ensemble d'images *Describable Textures Dataset* [3] utilisés pour simuler la variation de la puissance de la rétrodiffusion sur les image **RSOPOL** .

### 3.1.4.3. Étape 3 - Génération de l'image polarimétrique une vue

Chaque pixel de l'image  $S$  des étiquettes est bruité pour former l'image polarimétrique une vue  $\bar{S}$ . Dans le cas où la puissance demeure constante, la synthèse du chatolement polarimétrique est faite pixel par pixel par l'algorithme de Pipia et Fàbregas [29] que nous avons décrit à la Section 3.1.1. La sous-figure 3.3-(h) présente un exemple du composé de Pauli de l'image une vue  $\bar{S}$  synthétisée.

La simulation des images  $S_p$  des signatures polarimétriques à puissance variable est produite en multipliant terme par terme l'image originale  $S$  des signatures polarimétriques avec une image  $\tau$  de texture (Fig. 3.3-(i)). Cette image  $\tau$  est composée aléatoirement à partir de la banque de texture *DTD* [3] (Fig. 3.4). La nouvelle image  $S_p$  des signatures polarimétriques est introduite dans le même procédé pour générer les images une vue pour synthétiser l'image une vue  $\bar{S}_p$  à puissance variable (Fig. 3.3-(j) pour  $S_p$  et Fig. 3.3-(k) pour  $\bar{S}_p$ ).

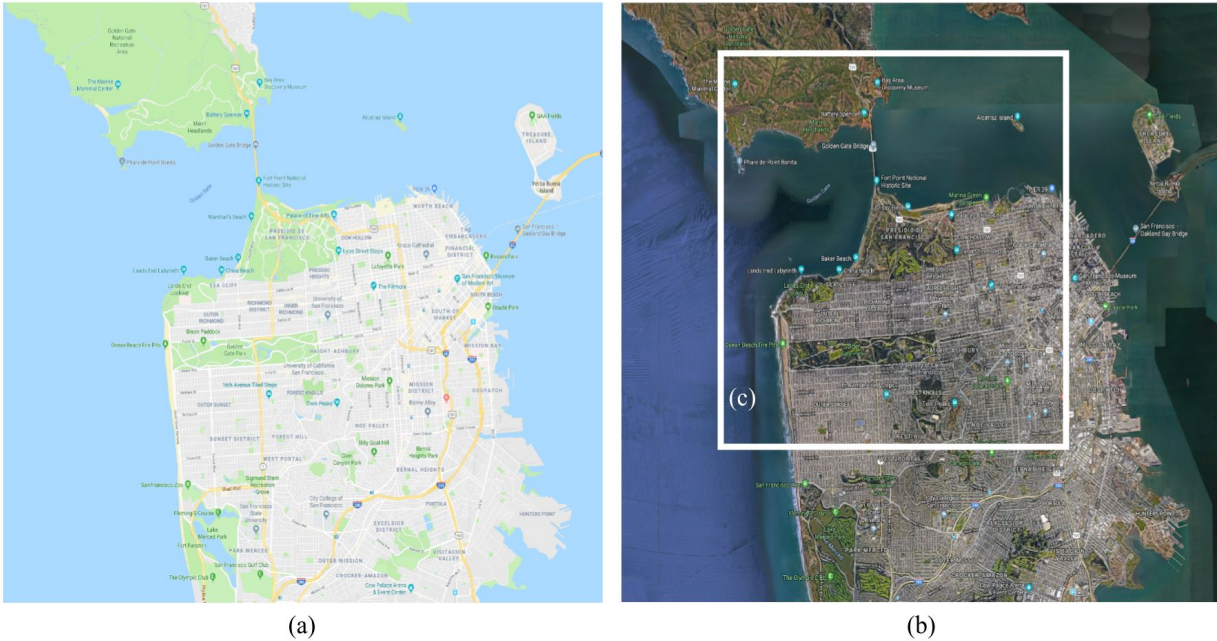
Les images une vue  $S_p^+$  et  $\bar{S}_p^+$  à puissance variable avec cibles ponctuelles sont générées en distribuant des cibles de quelques pixels de façon aléatoire sur les images. La Figure 3.3-(l) présente un exemple de distribution de cibles. La puissance des cibles demeure la même pour les deux images  $S_p^+$  et  $\bar{S}_p^+$ , car elle n'est pas influencée par l'effet du chatolement. Cette puissance est de l'ordre de 3 à 10 fois supérieur à la puissance moyenne des images. La Figure 3.3-(m) et la Figure 3.3-(n) présentent respectivement un exemple de  $S_p^+$  et  $\bar{S}_p^+$ . La Figure 3.5 illustre quelques exemples de synthèses polarimétriques qui montrent la diversité des résultats issus du procédé.



**Fig. 3.5.** Trois exemples de simulation polarimétrique avec des puissances déterminées par des images de texture et avec des cibles ponctuelles incrustées. La colonne (a) montre les étiquettes des classes de diffuseurs ( $C_z^+$ ). La classification  $H/\bar{\alpha}$  des régions est donnée par la légende à droite; La colonne (b) présente le composé de Pauli des images de signatures polarimétriques ( $S_p^+$ ) sans chatolement; La colonne (c) présente le composé de Pauli des images une vue ( $\bar{S}_p^+$ ).

### 3.2. Données réelles

Les données que nous utilisons sont des images en QuadPol acquises au-dessus de la ville de San Francisco aux États-Unis ( $37^\circ 46' 26'' N$ ,  $122^\circ 25' 52'' W$ ). La ville occupe la pointe



**Fig. 3.6.** Région de la ville de San Francisco, Californie, (ÉU). (a) Plan de la ville (source Google). (b) Vue satellite de la région en couleur visible (source Google). (c) Le rectangle représente la région d'intérêt qui est approximativement de la taille de l'image acquise en 1999 par le capteur aéroporté AIRSAR.

d'une péninsule à mi-hauteur de la côte nord de la Californie, entourée sur trois côtés par des étendues d'eau: l'océan Pacifique, le détroit du Golden Gate et la baie de San Francisco. Elle est disposée dans une grille sur quelque 40 collines, pouvant atteindre des hauteurs de près de 300 mètres (Fig. 3.6-a et Fig. 3.6-b).

Cette région est particulièrement bien étudiée dans le domaine de la recherche en **RSO-POL**. Elle sert souvent pour l'étalonnage des nouveaux capteurs. Des images sont disponibles avec le logiciel *polsapro-bio* et peuvent être téléchargées<sup>1</sup> sans contraintes à des fins de recherche. Nous avons à notre disposition trois images en QuadPol provenant de plateformes satellites: RADARSAT-2, ALOS-2/PALSAR-2, GAOFEN-3. Nous avons également l'image aéroportée AIRSAR qui définit notre région commune de recherche pour toutes les images (Fig. 3.6-c pour la région d'intérêt).

<sup>1</sup><https://www.ietr.fr/polsarpro-bio/san-francisco/>



Fig. 3.7. Image AIRSAR, Bande L,  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$

### 3.2.1. Image du capteur AIRSAR

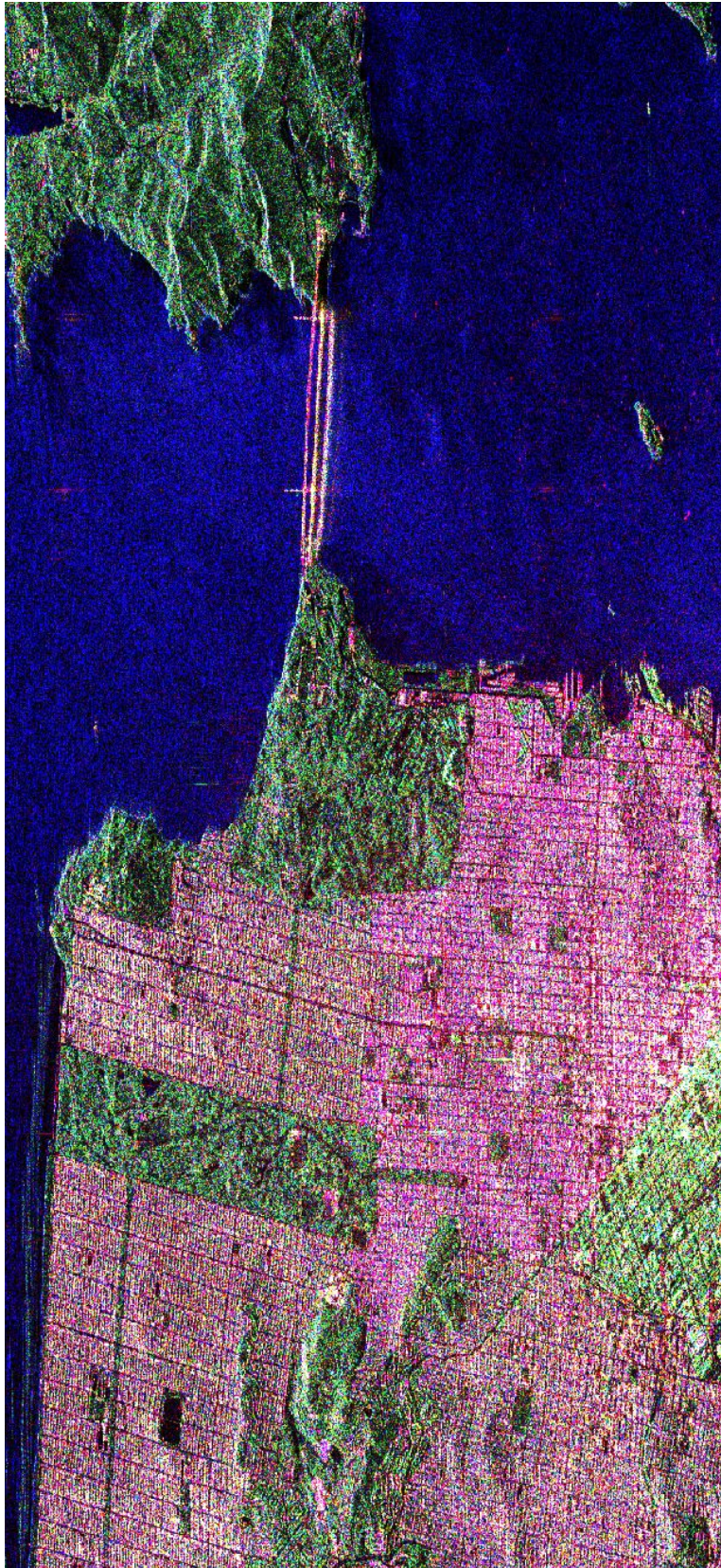
Le capteur aéroporté AIRSAR (*Airborne Synthetic Aperture Radar*) était la plateforme à visée latérale **RSOPOL** de recherche du Jet Propulsion Laboratory (JPL). Son premier vol eu lieu en 1988 et sa dernière mission se termina en 2004. Il pouvait acquérir des données entièrement polarimétriques à trois longueurs d'onde radar: Bande C (5.3 GHz), Bande L (1.2 GHz) et Bande P (0.44 GHz). L'image multivue (nombre de vues = 4) acquise en 1998 est utilisée pour la synthèse des matrices  $\begin{bmatrix} T_3 \end{bmatrix}$  des signatures polarimétriques (Fig. 3.7).

### 3.2.2. Image du capteur RADARSAT-2

Le satellite RADARSAT-2 est un satellite canadien qui fait suite à la mission de RADARSAT-1 qui s'est terminée en avril 2013. Il a été lancé le 14 décembre 2007 et fonctionne sur la même Bande C (5.40 GHz) que son prédécesseur (5.30 GHz). Il partage aussi les mêmes caractéristiques orbitales. Le satellite est situé à une altitude 798 km et son orbite est hélio-synchrone avec une inclinaison de  $98.16^\circ$ . L'orbite a un nœud ascendant à 18 h et un nœud descendant à 6 h avec des cycles répétés de 24 jours (343 orbites). Le satellite fait 14.29 fois le tour de la terre par jour et chaque orbite a une durée de 100.75 minutes. Il donne accès aux modes de RADARSAT-1 en permettant un choix sélectif de polarisation (HH, VH, HV, VV). A la différence de RADARSAT-1, il a des modes d'acquisition à double polarisation (HH+HV ou VV+VH) ou quadruple polarisation (HH+VH+HV+VV). Il permet d'acquérir des images avec une résolution spatiale plus grande pouvant atteindre jusqu'à 3 mètres par pixel. L'image une vue que nous utilisons a été acquise le 9 avril 2008, en mode fin (FQ9,  $28.9^\circ$ ) avec une résolution spatiale de près de 5 mètres. La Figure 3.8 montre la région extraite de l'image originale.

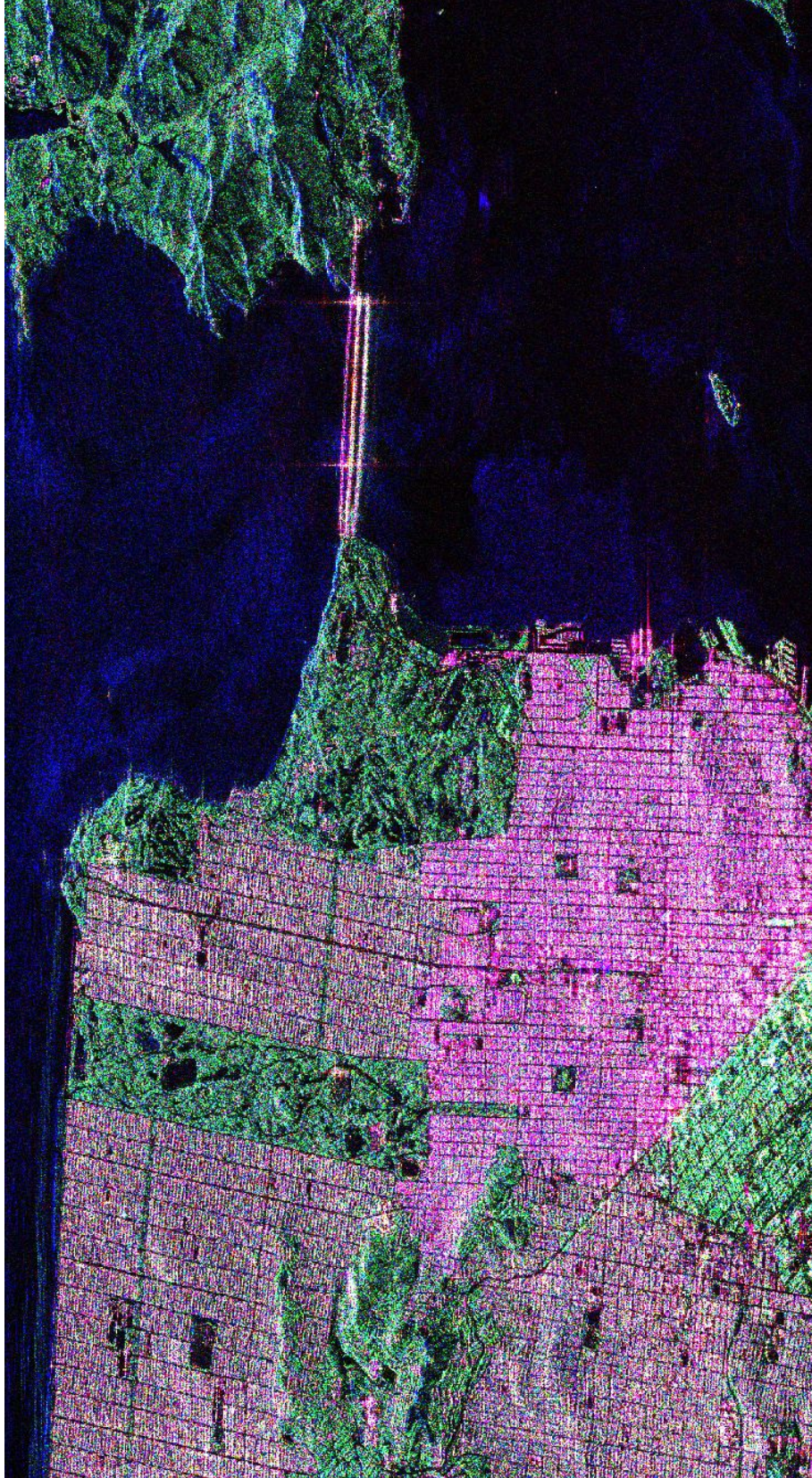
### 3.2.3. Image du capteur ALOS-2/PALSAR-2

Le satellite ALOS-2 (Advanced Land Observing Satellite 2), également appelé Daichi-2, est un satellite japonais de 2 tonnes lancé en 2014. Il est le successeur du satellite ALOS et fonctionne sur la Bande L (1.2 GHz). Il est sur une orbite hélio-synchrone à une altitude de 628 km avec des cycles répétées de 14 jours. Il donne accès aux mêmes modes de surveillance de ALOS-1/PALSAR-1 en permettant un choix sélectif de polarisation (HH, VH, HV, VV) et des modes d'acquisition à double polarisation (HH+HV ou VV+VH) ou quadruple polarisation (HH+VH+HV+VV). Le radar PALSAR-2 constitue une mise à niveau importante du radar PALSAR, permettant des modes à plus haute résolution spatiale (1 x 3m par pixel) comparativement à 10 mètres pour son prédécesseur. L'image une vue que nous utilisons a été acquise le 6 juin 2015, avec un angle d'incidence  $33.9^\circ$  et avec une résolution spatiale de 3 mètres. La Figure 3.9 montre la région extraite de l'image originale.



**Fig. 3.8.** Image RADARSAT-2, Bande C, ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )





**Fig. 3.9.** Image ALOS-2, Bande L,  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$



Fig. 3.10. Image GAF-3, Bande C,  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$

#### 3.2.4. Image du capteur GAOFEN-3

Le satellite GAOFEN-3 (Gāofēn, haute résolution) est l'un des satellites d'observation de la terre du programme chinois High-Definition Earth Observation Satellite (HDEOS). Ce

satellite a été lancé en 2016, sur une orbite héliosynchrone ( $98.4^\circ$ ) à 758 km d'altitude avec des cycles répétés de 29 jours. Il fonctionne comme RADARSAT-2 sur la Bande C (5.4 GHz) et présente des modes d'acquisition similaires. La résolution maximale du capteur est de 1 mètre par pixel. L'image une vue que nous utilisons a été acquise 15 septembre 2017, avec un angle d'incidence de  $21.22^\circ$  et une résolution spatiale de près de 3 mètres. La Figure 3.10 montre la région extraite de l'image originale.

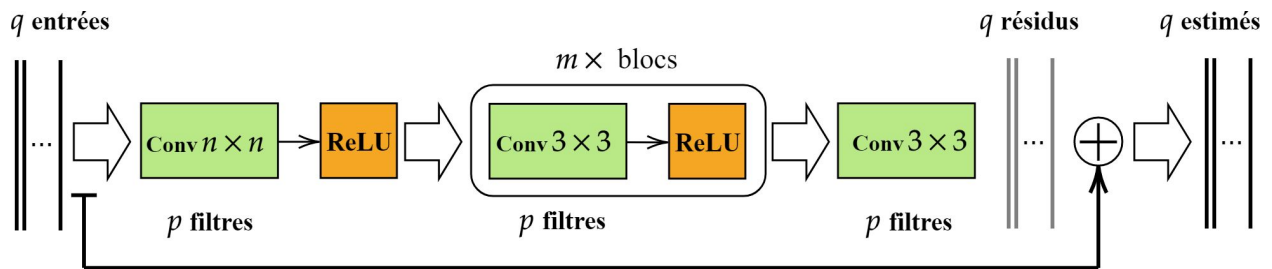
Ces trois images en QuadPol seront utilisées dans la dernière partie du mémoire pour analyser et évaluer qualitativement la performance des filtrages par **RNC** sur des données réelles.

### 3.3. Apprentissage du réseau

L'objectif de l'apprentissage est d'apprendre un modèle permettant d'estimer la matrice de cohérence à partir de données polarimétriques une vue.

#### 3.3.1. Description du réseau expérimental

Nous avons choisi d'utiliser une architecture neuronale issue du domaine de la superrésolution [39]. Ces architectures sont facilement convertibles pour effectuer d'autres types de restauration d'image. Et parmi celles-ci, l'architecture du VDSR[17] nous semblait tout indiquée par sa simplicité et par la démonstration qu'elle pouvait être modifiée et ajustée au filtrage des images naturelles ainsi qu'aux images **RSO**. L'architecture originale est inspirée des réseaux VGG [35] qui se distinguent par l'utilisation unique de filtres de convolution  $3 \times 3$  pour toutes les couches convolutives.



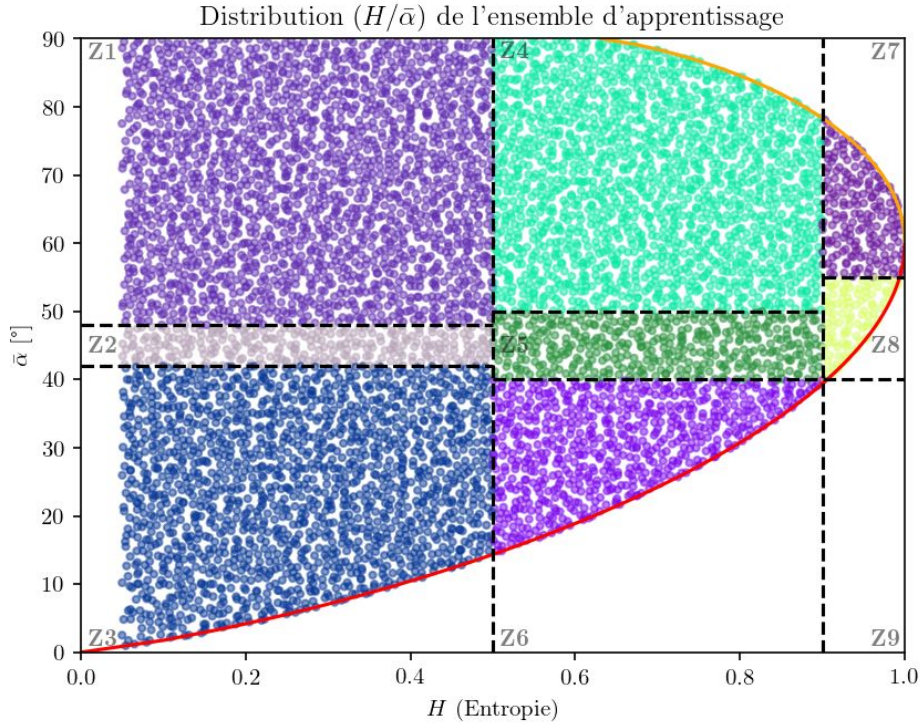
**Fig. 3.11.** Schéma de l'architecture du réseau à convolution utilisé pour les expérimentations, où  $p$  est le nombre de filtres par couches,  $n$  est la taille des filtres de la première couche et  $m$  est le nombre de répétition du bloc pour le calcul du résidu. Le symbole  $\oplus$  est l'opération de calcul du résidu.

La Figure 3.11 présente le schéma général du réseau que nous avons adapté pour nos besoins. Il est formé de  $m$  blocs convolutifs en couches cachées. Un bloc convolutif est composé d’une couche convolutive de  $p$  filtres de taille  $3 \times 3$  suivie de la fonction d’activation **ReLU**. Un filtre opère sur une région spatiale  $3 \times 3$  sur les  $p$  cartes de caractéristiques visuelles. La première couche est formée aussi par un bloc convolutif et opère sur les  $q$  canaux de l’image d’entrée. La dernière couche est la couche résiduelle utilisée pour la reconstruction. Elle consiste en une seule couche de  $p$  filtres  $3 \times 3$  avec  $q$  sorties. La couche résiduelle est calculée grâce au saut de connections ( $\oplus$ ) qui additionne les canaux d’entrée aux canaux de sortie. Les cartes de caractéristiques visuelles en sortie représentent donc les images résiduelles à additionner aux images d’entrée pour les restaurer.

### 3.3.2. Les paramètres fixes

A la suite de certains tests préliminaires, nous gardons des paramètres constants pour l’entraînement afin que les conditions expérimentales demeurent les plus stables possibles et afin de limiter les variations (voir Tab. 3.3). Les entraînements se font sur **150** époques avec un ensemble de données de taille fixe de **5000** échantillons pour l’apprentissage et de **256** échantillons pour l’ensemble de validation. Une époque correspond à un cycle d’apprentissage complet sur l’intégralité de l’ensemble de données de manière à ce que chaque exemple ait été vu une fois. La taille des échantillons est de  $9 \times 64 \times 64$ . Ils sont créés à la volée par la méthode décrite à la Section 3.1.4.2. L’ensemble de données est par conséquent renouvelé et unique pour chaque époque. La taille du mini-lot d’apprentissage est de  $9 \times 1 \times 64 \times 64$ . Nous présentons les échantillons un à la fois au réseau. Les 9 termes des matrices de cohérence sont vues comme des entrées indépendantes et dans le même ordre. La mise à jour des paramètres est faite en calculant le gradient de l’erreur moyenné sur les 9 entrées du mini-lot. Ce qui correspond à optimiser un seul modèle pour tous termes de la matrice de cohérence. Ceci est en accord avec les principes émis par Lee et al. [22] dont celui qui stipule que tous les termes de la matrice de cohérence doivent être filtrés de manière identique pour conserver la corrélation entre les différents canaux de polarisation.

Le taux d’apprentissage initial est fixé à  $\lambda = 10^{-3}$ . Il est réduit d’un facteur de 0.1 aux époques suivantes: 40, 80, 100, 120, 130 et 140. L’optimisation du modèle est calculé par la



**Fig. 3.12.** Distribution  $H/\bar{\alpha}$  des matrices de cohérence pour la génération des données d'apprentissage. Les couleurs représentent les différentes classes de diffuseur.

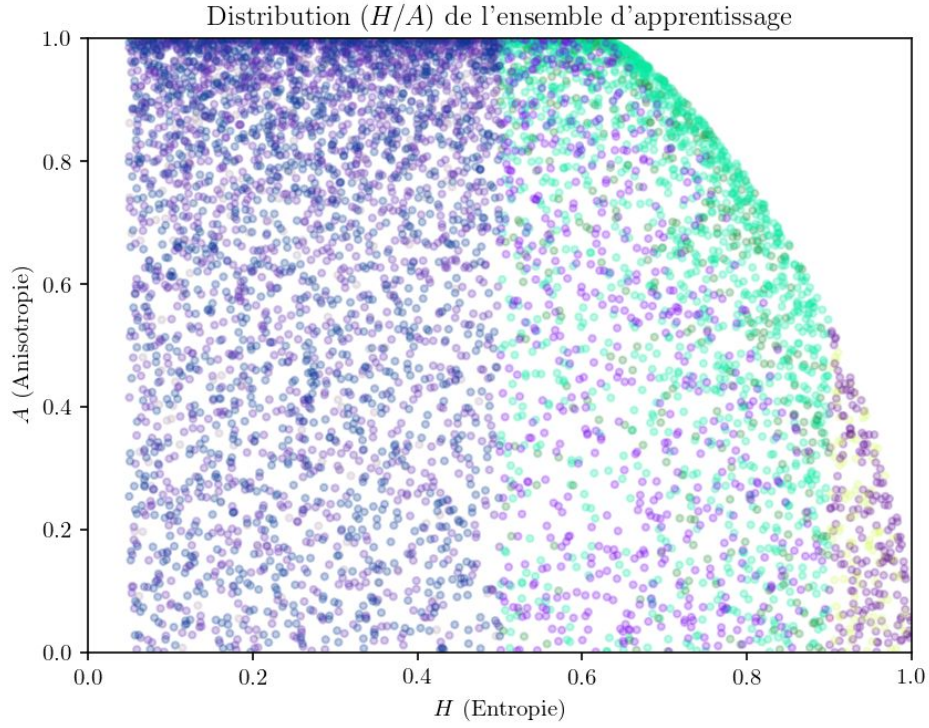
méthode *ADAM* [18]. Cette méthode permet une convergence plus rapide de l'apprentissage si on la compare à la méthode classique *Stochastic Gradient Descent* (SGD).

Les matrices de cohérence  $[T_3]$  utilisées pour produire les données d'entraînement sont sélectionnées à partir d'un ensemble de **6528** signatures générées aléatoirement et couvrant l'ensemble du plan  $H/\bar{\alpha}$ . Les Figures 3.12 et 3.13 montrent la distribution des signatures calculées. Les matrices de cohérence  $[T_3]$  utilisées pour produire les données de validation sont fournies par les signatures présentées au Tableau 3.1. Ces signatures ont été calculées préalablement à partir de l'image AIRSAR de San Francisco.

La fonction de coût  $\epsilon$  à minimiser est l'erreur absolue entre l'estimé du modèle  $\bar{y}$  et l'échantillon étiquette  $y$ :

$$\epsilon(y, \bar{y}) = |y - \bar{y}| \quad (3.3.1)$$

Cette fonction permet de faire l'apprentissage d'un modèle dont la tâche est de minimiser le biais absolu sur les 9 paramètres uniques de la matrice  $[T_3]$ .



**Fig. 3.13.** Distribution  $H/A$  des matrices de cohérence pour la génération des données d'apprentissage. Les couleurs représentent les différentes classes de diffuseur.

Nombre d'époques	150
Taux d'apprentissage $\lambda$	$10^{-3}$
Méthode de planification de réduction de $\lambda$	Par saut de 0.1 à certaines époques précises
Méthode d'optimisation	ADAM
Taille de l'ensemble d'entraînement	5000
Taille de l'ensemble de validation	250
Taille des échantillons	$9 \times 64 \times 64$
Taille du mini-lot	$9 \times 1 \times 64 \times 64$ ( <i>Vanilla</i> )
La fonction de coût $\epsilon$	$L_1 =  y - \bar{y} $
<i>Seeds</i> des générateurs aléatoires	0

**Tab. 3.3.** Paramètres qui demeurent fixes

Taille $n$ des filtres de la première couche convolutive	3, 5, 7, 9, 11, 13, 15
Nombre $p$ de filtres par couche convolutive	8, 16, 32, 64
Nombre $m$ de blocs convolutifs	0, 1, 3, 5

**Tab. 3.4.** Paramètres testés

### 3.3.3. Les paramètres testés

Les paramètres d'apprentissage que nous faisons varier sont tous reliés à la configuration du réseau convolutif (voir Tab. 3.4). La taille de la convolution ( $n$ ) de la première couche varie entre 3 et 15 à titre de comparaison avec un filtre Boxcar de même taille. Le nombre de filtres ( $p$ ) varie de 64 à 8 pour tester le nombre minimal de paramètres par bloc convolutif. L'impact de la hiérarchisation des caractéristiques visuelles et leur complexification est testé en faisant varier le nombre de blocs convolutifs ( $m$ ) de 0 (aucun bloc) à 5. Nous obtenons au total  $5 \times 4 \times 5 = 120$  apprentissages distincts.

### 3.3.4. L'estimation des paramètres polarimétriques

La qualité des modèles comme estimateur de la matrice  $[T_3]$  est évaluée en calculant les biais sur l'estimation  $\bar{\Theta}_p$  de certains paramètres polarimétriques  $\Theta_p$ . L'erreur  $\epsilon$  de l'estimation  $\bar{\Theta}_p$  sur un échantillon  $\bar{y}$  de la sortie du modèle est donnée par:

$$\epsilon(\bar{y}) = \bar{\Theta}_p(\bar{y}) - \Theta_p(y) \quad (3.3.2)$$

où  $\Theta(y)_p$  est la valeur réelle du paramètre polarimétrique calculée à partir de la valeur étiquette ( $y$ ) de la matrice  $[T_3]$ . Le biais sur le paramètre  $\bar{\Theta}_p$  est défini par:

$$\beta(\bar{\Theta}_p) = E[\bar{\Theta}_p(\bar{y})] - \Theta_p(y) \quad (3.3.3)$$

où  $E[\bar{\Theta}_p(\bar{y})]$  est l'espérance de l'estimation sur un échantillon. Le biais absolu relatif est donné par

$$\beta_r(\bar{\Theta}_p) = \left| \frac{E[\bar{\Theta}_p(\bar{y})] - \Theta_p(y)}{\Theta_p(y)} \right| \quad (3.3.4)$$

où  $\Theta(y)_p \neq 0$ . Dans le cas où  $\beta = 0$ , l'estimateur  $\bar{\Theta}_p$  est dit non-biaisé. L'erreur quadratique moyenne absolue (**EQM**) de  $\bar{\Theta}_p$  est définie par le calcul de l'espérance  $E[\cdot]$  de la différence au carré entre l'estimation et la valeur estimée:

$$\mathbf{EQM} = E[(\bar{\Theta}_p - \Theta_p)^2] \quad (3.3.5)$$

de manière similaire, l'erreur quadratique moyenne absolue relative (**EQM<sub>r</sub>**) est définie comme suit:

$$\mathbf{EQM}_r = E\left[\left(\frac{\bar{\Theta}_p - \Theta_p}{\Theta_p}\right)^2\right] \quad (3.3.6)$$

Ces erreurs mesurent la divergence moyenne d'un ensemble d'estimations par rapport à la valeur réelle du paramètre. Nous pouvons aussi construire les mesures de l'erreur absolue moyenne (**EAM<sub>a</sub>**) et de l'erreur absolue relative moyenne (**EAM<sub>r</sub>**) en calculant la moyenne des biais comme suit:

$$\mathbf{EAM}_a = E[|\bar{\Theta}_p - \Theta_p|] \quad (3.3.7)$$

et

$$\mathbf{EAM}_r = E\left[\left|\frac{\bar{\Theta}_p - \Theta_p}{\Theta_p}\right|\right] \quad (3.3.8)$$

La variance de l'estimateur  $\bar{\Theta}_p$  est donnée par:

$$\mathbf{VAR}(\bar{\Theta}_p) = E[(\bar{\Theta}_p - E[\bar{\Theta}_p])^2] \quad (3.3.9)$$

Elle mesure la divergence d'un ensemble d'estimation par rapport à la valeur moyenne de l'estimateur. A partir des Équations (3.3.3, 3.3.5 et 3.3.9), nous pouvons établir la relation suivante:

$$\mathbf{EQM}(\bar{\Theta}_p) = \mathbf{VAR}(\bar{\Theta}_p) + (\beta(\bar{\Theta}_p))^2 \quad (3.3.10)$$

Elle montre que l'absence de biais ( $\beta = 0$ ), à elle toute seule, ne garantit pas que l'estimateur soit bon. En effet, un estimateur sans biais mais à variance élevée peut fournir des estimations très éloignées de la vraie valeur du paramètre, i.e. que l'erreur quadratique entre



la valeur de l'estimation et de la valeur estimée pourrait être grande. Dans le cas où nous avons plusieurs estimateurs non-biaisés, la sélection de l'estimateur le plus efficace se fait en fonction de la plus petite valeur de variance.

Une propriété importante d'un bon estimateur est sa cohérence. Elle implique que si la taille  $N$  de l'échantillon grandit alors le biais tendra vers 0 et la variance aussi. La valeur de l'estimation converge asymptotiquement vers la valeur estimée.

### 3.3.5. L'évaluation des filtrages

Les paramètres polarimétriques suivants sont testés par l'évaluation des biais telle que présentée à la section précédente.

- (1) Les 9 valeurs uniques de la matrices de cohérence  $[T_3]$ ;
- (2) Les valeurs de la décomposition  $H/A/\bar{\alpha}$ ;

La valeur du nombre de vues équivalent (**NEV**) est évaluée à partir des termes diagonaux de la matrice  $[T_3]$  estimée sur des zones homogènes.

La reconstruction des détails est évaluée par des mesures de distorsion entre l'image estimée et ceux de l'image étiquette.

La conservation des cibles ponctuelles est évaluée en mesurant le changement de la forme des cibles par rapport à la vérité terrain.

### 3.3.6. La comparaison des filtrages

Les résultats des estimations de la matrice  $[T_3]$  des modèles produits sont comparés principalement avec deux filtres polarimétriques:

#### 3.3.6.1. Le filtre *Boxcar*

Le filtre multilook **Boxcar** pose comme hypothèse que le signal est homogène autour d'un pixel donné. Il fait la moyenne des  $n$  pixels appartenant à son voisinage défini par une fenêtre rectangulaire. C'est le filtre de chatoiement le plus simple et le plus rapide. Il correspond aussi à l'estimation du maximum de vraisemblance de la matrice de covariance et ce sans biais ni distorsion. Par contre, l'hypothèse du signal stationnaire n'est pas valable pour tous les pixels, en particulier dans les cas des cibles ponctuelles et des contours. Dans ces cas le **Boxcar** peut donner de mauvais résultats en raison du mélange de pixels hétérogènes. Le filtre se comporte comme un filtre passe-bas. Il réduit la résolution spatiale et génère une

image floue. Cet effet indésirable est particulièrement perceptible pour les cibles ponctuelles fortes en énergie car elles apparaissent étendues proportionnellement à la taille de la fenêtre de filtrage.

### 3.3.6.2. Le filtre adaptatif de Lee affiné

Le filtre Lee affiné (**Refined Lee**) est un filtre d'erreur minimale moyenne (MMSE) développé sur la base du modèle de bruit multiplicatif [22]. Un inconvénient majeur du filtre MMSE est que le bruit de chatoiement près des arêtes fortes n'est pas filtré de manière adéquate. Pour compenser ce problème, le filtre de Lee affiné utilise une fenêtre non carrée pour faire correspondre la direction des arêtes. Le filtre utilise une fenêtre glissantes  $7 \times 7$  (ou  $9 \times 9$ ,  $11 \times 11$ ). L'une des huit fenêtres alignées sur les arêtes est sélectionnée pour filtrer le pixel central. Seuls les pixels de la zone non contenue dans la fenêtre alignée sur les contours sont utilisés dans le calcul du filtrage. Le filtre se déroule sur trois étapes de traitement:

- (1) **Sélection de la fenêtre alignée sur les arêtes:** Pour chaque pixel, l'image de *span* est utilisée pour sélectionner la fenêtre alignée sur les arêtes;
- (2) **Calcul du poids du filtrage:** Un filtre évaluant les statistiques local sur l'image de *span* est appliqué pour calculer le poids du filtrage  $b$ ;
- (3) **Filtrage de la matrice de covariance:** Le même poids  $b$  et la même fenêtre sélectionnée sont utilisés pour filtrer indépendamment et également chacun des éléments de la matrice  $Z$ . La matrice filtrée est donnée par:

$$\hat{Z} = \bar{Z} + b(Z - \bar{Z}) \quad (3.3.11)$$

où  $\bar{Z}$  est la moyenne locale de la matrice calculée à partir des pixels sous la fenêtre alignée.

Nous comparons aussi avec d'autres filtres disponibles dans le software ESA-SNAP utilisé pour faire le traitement d'images en QuadPol.

## 3.4. Le matériel

Tout au long du projet nous avons utilisé plusieurs logiciels, nous donnons une courte description de chacun.

- (1) **PYTHON** <sup>2</sup>

---

<sup>2</sup><https://www.python.org>

Python est un langage de programmation *open source* créé par Guido van Rossum en 1991. Il s'agit d'un langage interprété, qui ne nécessite donc pas d'être compilé pour fonctionner. Un programme interpréteur permet d'exécuter le code Python sur n'importe quel ordinateur. Il permet de facilement importer de nouvelles bibliothèques logicielles de traitement et d'analyse. L'ensemble de nos traitements sont réalisées sous forme de scripts python. La version de python que nous utilisons est la version 3.6;

(2) **PYTORCH** <sup>3</sup>

PyTorch est une bibliothèque logicielle Python *open source* d'apprentissage machine qui s'appuie sur Torch (en) développée par Facebook. PyTorch permet d'effectuer les calculs tensoriels nécessaires pour l'apprentissage profond. Les calculs sont optimisés et effectués soit par le processeur (CPU) soit, lorsque c'est possible, par un processeur graphique (**GPU**) supportant CUDA. Cette bibliothèque est au coeur du développement de nos scripts d'entraînement. La version de PyTorch que nous utilisons est la version 1.1;

(3) **ESA SNAP Toolbox** <sup>4</sup>

SNAP est un logiciel *open source* en JAVA fourni par l'Agence Spatial Européenne (ESA) pour l'exploitation des données d'observation de la Terre. Le logiciel est pourvue d'une boîte à outils spécialisée pour la manipulation des images **RSOPOL**. La boîte à outils Sentinel-1 (S1TBX) comprend un ensemble d'outils de traitement, des lecteurs et des rédacteurs de produits de données, ainsi qu'une application d'affichage et d'analyse permettant de gérer l'archive des données des missions **RSO** de l'ESA, notamment SENTINEL-1, ERS-1 et 2 et ENVISAT, ainsi que des données **RSO** tierces d'ALOS PALSAR, TerraSAR-X, COSMO-SkyMed et RADARSAT-2. Les différents outils de traitement peuvent être exécutés indépendamment à ligne de commande et où à partir de l'interface graphique. La boîte à outils comprend aussi des outils pour l'étalonnage, le filtrage du chatoiement, la coregistration, l'orthorectification, le mosaïquage, la conversion de données, l'analyse polarimétrique et

---

<sup>3</sup><https://pytorch.org>

<sup>4</sup><https://step.esa.int/main/toolboxes/snap>

l'interférométrie des données. La version de SNAP que nous utilisons est la version 6.0.4;

#### (4) **SNAPPY Python Toolbox**

SNAPPY est une bibliothèque logicielle python qui donne accès à l'API JAVA de SNAP. Elle permet à même un script python l'utilisation des opérateurs de traitement des boites à outils de SNAP. La version de SNAPPY que nous utilisons est la version 6.0.4;

L'apprentissage profond demande en général un matériel spécialisé pour être calculé efficacement. En particulier l'ordinateur doit être muni d'une carte graphique avec un **GPU** compatible pour l'utilisation de PyTorch. Dans le cadre de ce projet nous avons utilisé deux systèmes pour les traitements:

##### (1) **Un système de prototypage:**

Ordinateur de bureau;

Carte graphique: GeForce GTX 1080 (8GB);

Mémoire: 32G;

CPU: 8 × Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz;

Il a servi à la conception et au débogage des scripts python;

##### (2) **Un système de production:**

Grappe de **GPU** du CRIM<sup>5</sup> (7 **GPU** utilisés)

Cartes graphiques par machine: 3 × GeForce GTX 1080-TI (11GB)

Mémoire par machine: 125G

CPU par machine: 18 × Intel(R) Core(TM) i9-7980XE CPU @ 2.60GHz

La grappe a servi à la production en parallèle des modèles des différentes expériences;

### 3.5. Conclusion

Nous avons présenté dans ce chapitre l'ensemble des outils mathématiques pour produire les données et évaluer les résultats. Ainsi que le outils matériels nécessaires pour la réalisation du projet. Nous avons décrit la génération des données simulées et les données réelles que nous utilisons pour évaluer les différents modèles produits au cours des nombreuses expérimentations.

---

<sup>5</sup>Centre de Recherche Informatique de Montréal ([www.crim.ca](http://www.crim.ca))

# Chapitre 4

---

## PRÉSENTATION ET DISCUSSION DES RÉSULTATS SUR LES APPRENTISSAGES

Ce chapitre présente l'ensemble des résultats des expérimentations. Il se divise en trois volets principaux:

- (1) La présentation et l'analyse de l'apprentissage;
- (2) La présentation et l'analyse des résultats de l'application des modèles pour l'estimation des paramètres polarimétriques sur les données synthétiques.
- (3) La présentation des résultats de l'application des modèles sur les données réelles.

Nous discuterons des résultats en familles de modèles ou ensembles de modèles pour un même nombre de blocs résiduels  $m$  et un même nombre de filtres par convolution  $p$ . Chacune de ces familles comporte 7 modèles en lien avec la taille  $n$  de la première convolution. Nous ne présenterons pas tous les résultats individuels, car ils seraient trop nombreux. Nous pourrions les consulter dans les annexes. Nous donnons un condensé permettant de tirer des conclusions sur l'ensemble des résultats.

### 4.1. L'analyse de l'apprentissage

Les apprentissages ont été produits selon le protocole expérimental expliqué à la Section 3.3. L'objectif n'était pas de trouver les paramètres susceptibles de produire des modèles optimaux mais de minimiser les effets susceptibles d'influencer les résultats, par exemple nous avons toujours utilisé le même taux d'apprentissage  $\lambda$  pour toutes les expérimentations. Les tableaux suivants (Tab. 4.1, Tab. 4.2 et Tab. 4.3) comptabilisent le nombre des apprentissages produits pour chacun des ensembles de données simulées décrits aux Sections

3.1.3 et 3.1.4. Au total nous avons produit 154 modèles en faisant varier les paramètres de l’architecture **RNC** décrite à la Section 3.3.1. Nous avons modulé sur certaines plages, le nombre  $m$  de blocs résiduels, le nombre  $p$  de filtres par convolution et la taille  $n$  des filtres de la première couche convolutive du **RNC**.

Le temps d’exécution d’un apprentissage a été en moyenne de 11.9 heures  $\pm$  4.4 heures et le nombre total d’heures machine est estimé à 1833 heures, soit environ 76.3 jours. L’utilisation d’un cluster **GPU** au CRIM<sup>1</sup> a donc été primordiale pour la réussite du projet. Il a permis de réduire le temps machine en exécutant plusieurs apprentissages en parallèle. La Figure 4.1 présente les temps d’exécution pour la série d’apprentissages correspondant à l’ensemble  $m = 0,1,3,5$  et  $p = 8$ . De manière générale le temps dépend de la taille du modèle et du type de carte graphique. Les temps au-dessous de 5 heures comptabilisent seulement le temps suite à une reprise pour finaliser un entraînement qui avait échoué. Dans ces cas d’échec, l’entraînement du modèle est repris à l’époque  $t$  du dernier modèle valide et est poursuivi jusqu’à l’époque finale  $t = 150$ . Les temps supérieurs à 20 heures montrent que le cluster a été surchargé, c’est-à-dire que la charge au niveau des CPU était excessive parce qu’il y avait trop de tâches par CPU. Ce qui provoque une dégradation au niveau de la performance car le transfert des données entre le CPU et le **GPU** est limité par la vitesse du CPU.

m/p	8	16	32	64	total
0	n={3,5,7,9,11,13,15}	n={3,5,7,9,11,13,15}	n={3,5,7,9,11,13,15}	n={3,5,7,9,11,13,15}	28 modèles
1	n={3,5,7,9,11,13,15}				7 modèles
3	n={3,5,7,9,11,13,15}				7 modèles
5	n={3,5,7,9,11,13,15}				7 modèles
total	28 modèles	7 modèles	7 modèles	7 modèles	49 modèles

**Tab. 4.1.** Ensemble des apprentissages produits à partir des **données simulées homogènes** en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par  $m$ , le nombre de filtres par convolution est donné par  $p$  et la taille de l’entrée de la première convolution est donnée par  $n$ .

<sup>1</sup>Centre de Recherche Informatique de Montréal, ww.crim.ca

m/p	8	16	32	64	total
0	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	28 modèles
1	$\mathbf{n}=\{3,5,7,9,11,13,15\}$				7 modèles
3	$\mathbf{n}=\{3,5,7,9,11,13,15\}$				7 modèles
5	$\mathbf{n}=\{3,5,7,9,11,13,15\}$				7 modèles
total	28 modèles	7 modèles	7 modèles	7 modèles	49 modèles

**Tab. 4.2.** Ensemble des apprentissages produits à partir des **données simulées hétérogènes** en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par  $m$ , le nombre de filtres par convolution est donné par  $p$  et la taille de l'entrée de la première convolution est donnée par  $n$ .

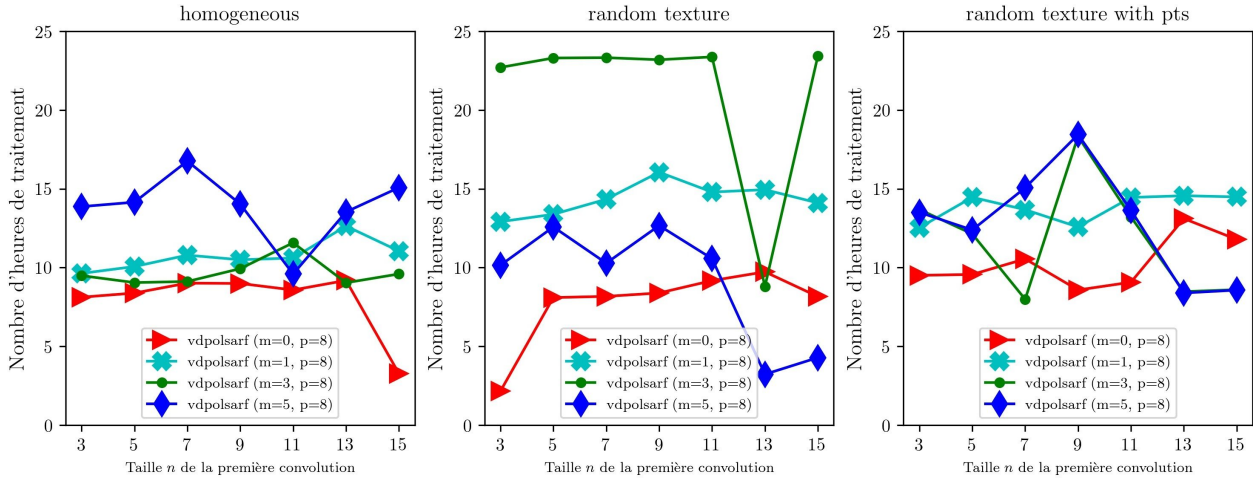
m/p	8	16	32	64	total
0	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	$\mathbf{n}=\{3,5,7,9,11,13,15\}$	28 modèles
1	$\mathbf{n}=\{3,5,7,9,11,13,15\}$				7 modèles
3	$\mathbf{n}=\{3,5,7,9,11,13,15\}$				7 modèles
5	$\mathbf{n}=\{3,5,7,9,11,13,15\}$			$\mathbf{n}=\{3,5,7,9,11,13,15\}$	14 modèles
total	28 modèles	7 modèles	7 modèles	14 modèles	56 modèles

**Tab. 4.3.** Ensemble des apprentissages produits à partir des **données simulées hétérogènes avec l'inclusion de cibles ponctuelles** en fonction de la paramétrisation de la structure du réseau de neurones. Le nombre de blocs résiduels est donné par  $m$ , le nombre de filtres par convolution est donné par  $p$  et la taille de l'entrée de la première convolution est donnée par  $n$ .

La Figure 4.2 présente les courbes de l'erreur de l'apprentissage sur l'ensemble des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  en fonction du nombre d'époques pour nos trois types d'entraînement. Il est à noter que l'erreur absolue  $|\epsilon|$  est représentée par son inverse sous une échelle logarithmique en base 10 pour faciliter l'interprétation des résultats en fonction des époques finales. L'erreur sur les courbes subséquentes prend la forme suivante:

$$ERR = 10 \times \log_{10} \left( \frac{1}{|\epsilon|} \right) \quad (4.1.1)$$

Les courbes vertes représentent l'erreur sur l'ensemble de données d'entraînement et les courbes en rouge présentent l'erreur sur les données de validation. Une première observation générale est le changement de rythme dans les courbes autour de l'époque 40. L'apprentissage est tout d'un coup plus souple après que le taux d'apprentissage eut été réduit par un facteur 10 à  $\lambda = 10^{-4}$ . Ce qui nous donne un indice que le taux original était légèrement trop grand.

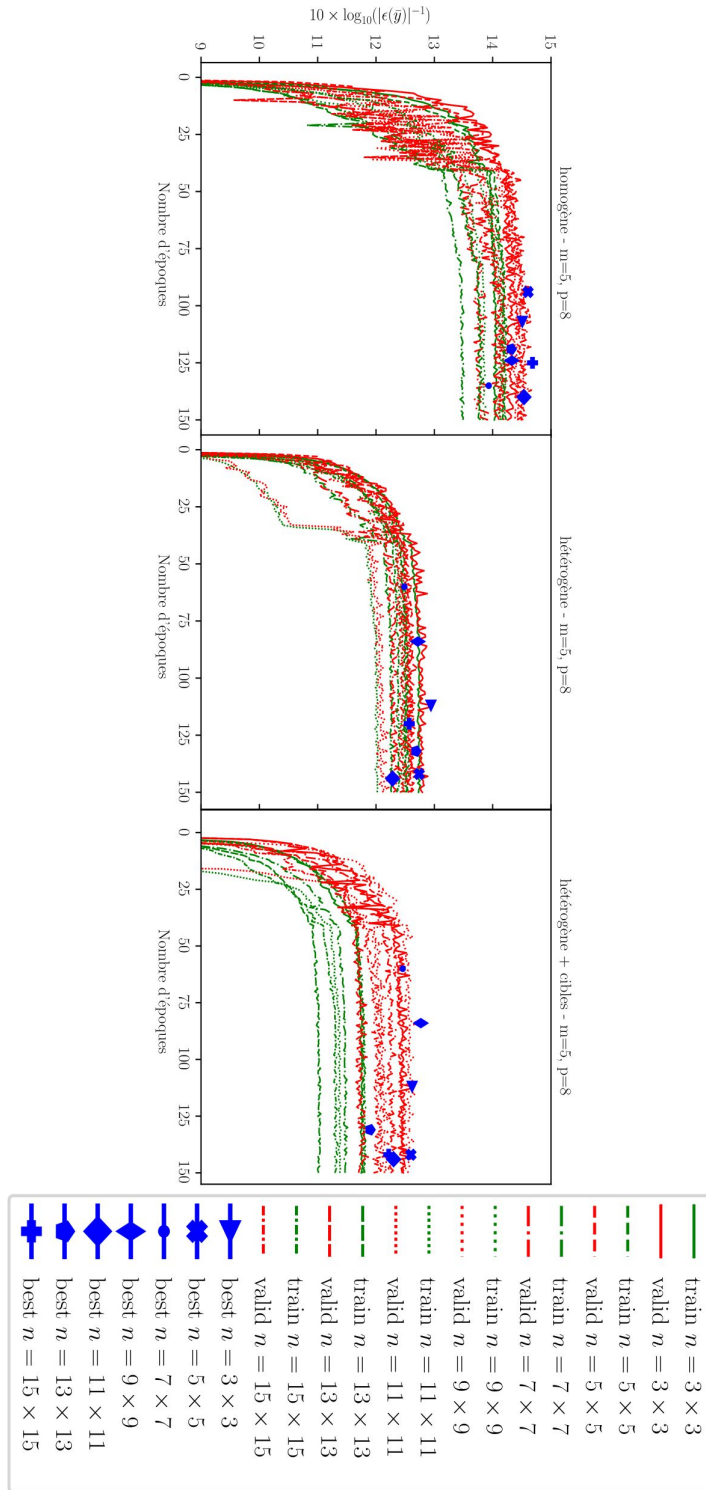


**Fig. 4.1.** Temps d'exécution des apprentissages de l'ensemble des  $m = \{0,1,3,5\}$  et  $p = 8$ . La figure de gauche présente les temps pour l'apprentissage sur les données homogènes. La figure centrale montre le temps pour l'apprentissage sur les données hétérogènes et la figure de droite montre celui pour les données hétérogènes avec inclusion de cibles ponctuelles. Nombre totale d'apprentissage = 84. Nombre d'heures total = 1003.4 heures. Temps moyen d'un apprentissage = 11.9 heures  $\pm$  4.4 heures.

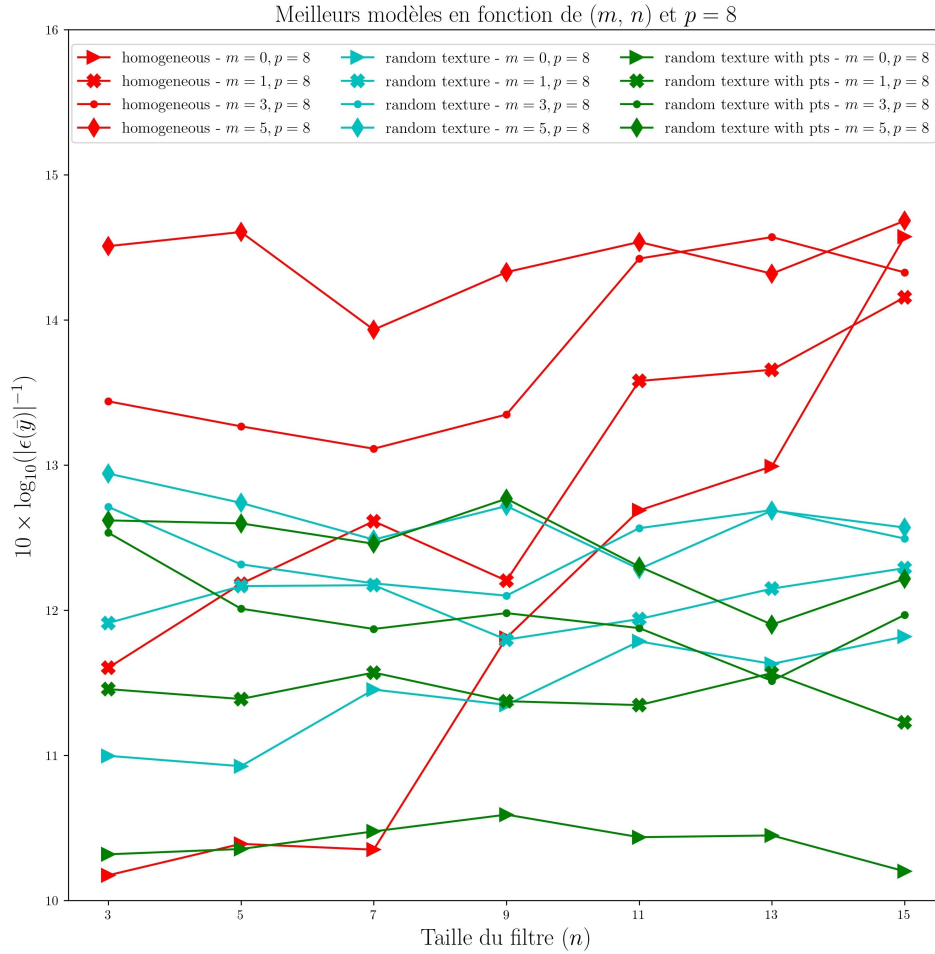
On observe aussi qu'à partir de l'époque 75, l'apprentissage semble stagné et aurait pu être raccourci de moitié sans trop de perte de précision. Une troisième observation générale est que l'on obtient de meilleurs pointages sur les données de validation. A priori ceci paraît contre intuitif mais pourrait s'expliquer par le fait que l'ensemble de validation contient moins d'échantillons difficiles pouvant faire chuter le pointage si on le compare à l'ensemble d'entraînement beaucoup plus diversifié. Ce dernier contient beaucoup plus de signatures qui sont plus compliquées à estimer correctement. Une dernière observation générale est la dégradation des résultats en fonction du type d'entraînement. Dans le cas de l'entraînement sur les données homogènes le pointage moyen est de 13.5 sur l'échelle logarithmique et décroît sur les données hétérogènes à 12.3 et chute sur les entraînement avec les données hétérogènes avec inclusion de cibles ponctuelles à 11.0. Le pointage moyen décroît donc en fonction de la complexité de l'ensemble de données utilisé pour l'apprentissage.

La Figure 4.3 présente les pointages des modèles ayant obtenu les meilleurs performances pour chacun des apprentissages. Les 4 courbes en rouge présentent les performances calculées sur les données d'entraînement homogènes. On observe que pour ces modèles la performance s'améliore à la fois en fonction de la taille  $n$  de la couche d'entrée et du nombre  $m$  de blocs





**Fig. 4.2.** Exemple de courbes d'apprentissage sur la famille  $m = 5, p = 8$  montrant l'erreur d'apprentissage en fonction du nombre d'époques (vert = entraînement, rouge = validation, bleu = meilleurs modèles). La figure de gauche présente les résultats relatif à l'apprentissage des modèles sur l'ensemble de données homogènes. La figure centrale montre les résultats pour l'apprentissage sur les données hétérogènes et la figure de gauche montre celui pour les données hétérogènes avec inclusion de cibles ponctuelles.



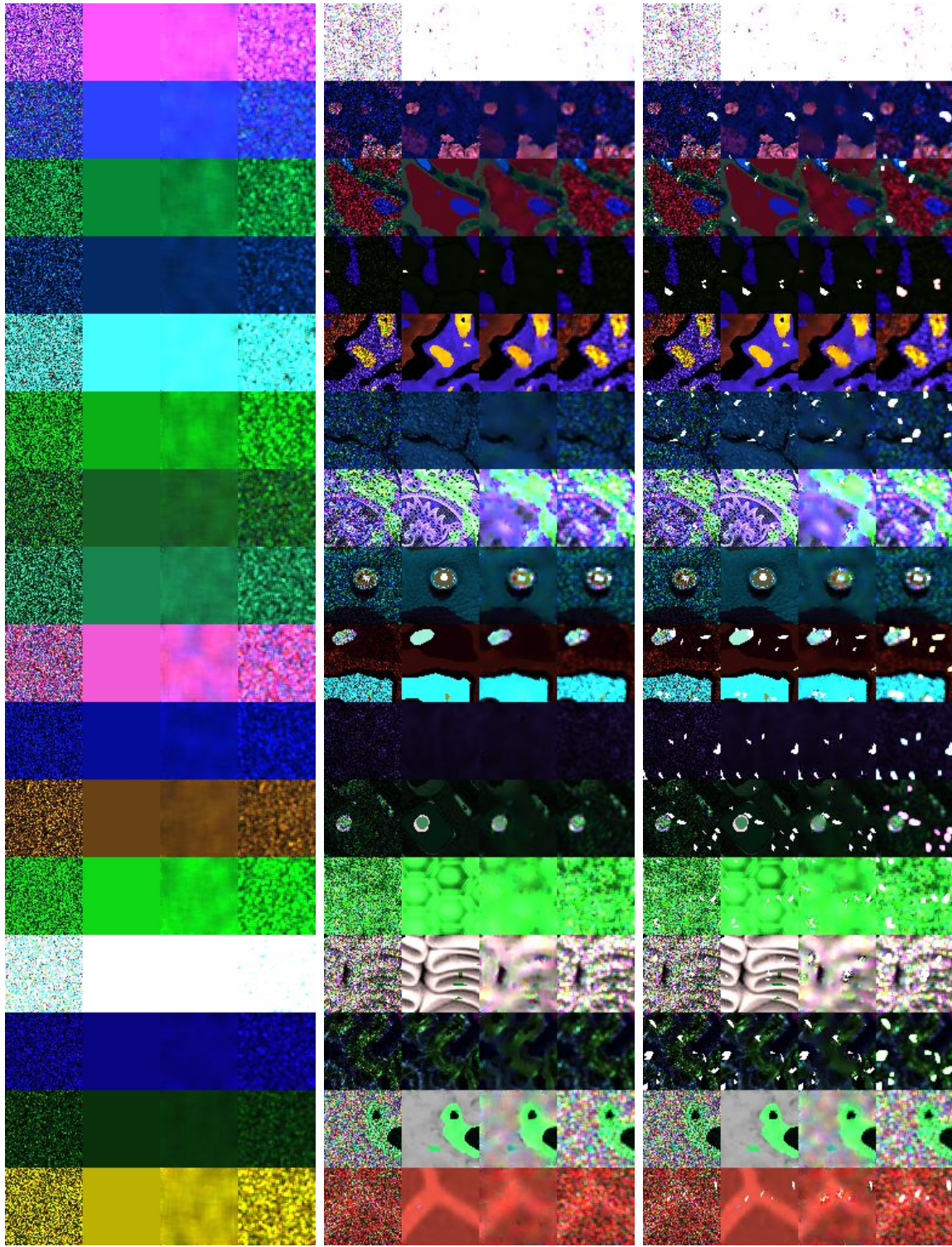
**Fig. 4.3.** L'erreur de l'apprentissage des meilleurs modèles sur l'ensemble de validation pour les trois expériences: homogène (rouge), hétérogène (cyan) et hétérogène avec inclusion de cibles ponctuelles (vert)

convolutifs. Les 4 courbes de couleur cyan présentent les performances calculées sur les données d'entraînement hétérogènes. La performance est dégradée et semble n'être sensible qu'au nombre  $m$  de blocs convolutifs. La taille  $n$  n'améliore que légèrement les résultats que pour la valeur de  $m$  égale à 0. Les 4 courbes vertes présentent les performances calculées sur les données d'entraînement hétérogènes avec inclusion de cibles ponctuelles. Les résultats sont légèrement dégradés comparativement aux résultats hétérogènes. La taille  $n$  de la couche d'entrée ne semble pas avoir d'influence sur la performance et semble même dégrader les résultats pour les grandes tailles de  $n$ .

Nous concluons cette section avec la Figure 4.4 qui présente 16 exemples sortis de l'ensemble de validation ayant servi aux différents apprentissages. Pour les trois ensembles nous montrons:

- (1) L'image caractéristique simulée avec un chatolement une vue qui constitue la donnée à filtrer;
- (2) L'image étiquette qui est la signature unique ou l'ensemble des signatures à cibler selon le cas;
- (3) Un exemple de prédiction obtenu au courant des apprentissages;
- (4) Une comparaison avec le filtre **Boxcar**  $5 \times 5$ ;

A remarquer que les données pour l'apprentissage des modèles hétérogènes et hétérogènes avec inclusion de cibles ponctuelles sont présentées dans le même ordre car les expériences sont déterministes afin de garantir la reproductibilité des expériences.



(a) Imagettes simulées homogènes (b) Imagettes simulées hétérogènes (c) Imagettes simulées hétérogènes + cibles ponctuelles

**Fig. 4.4.** Exemple de résultats sur un mini-lot de 16 échantillons extraits de l'ensemble de validation. Les trois figures présentent pour chaque type d'apprentissage en colonne: 1) l'image caractéristique simulée avec chatoisement  $N = 1$ , 2) les images étiquettes, 3) les prédictions et 4) le filtre **Boxcar**  $5 \times 5$ . ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

## 4.2. L'analyse d'un modèle

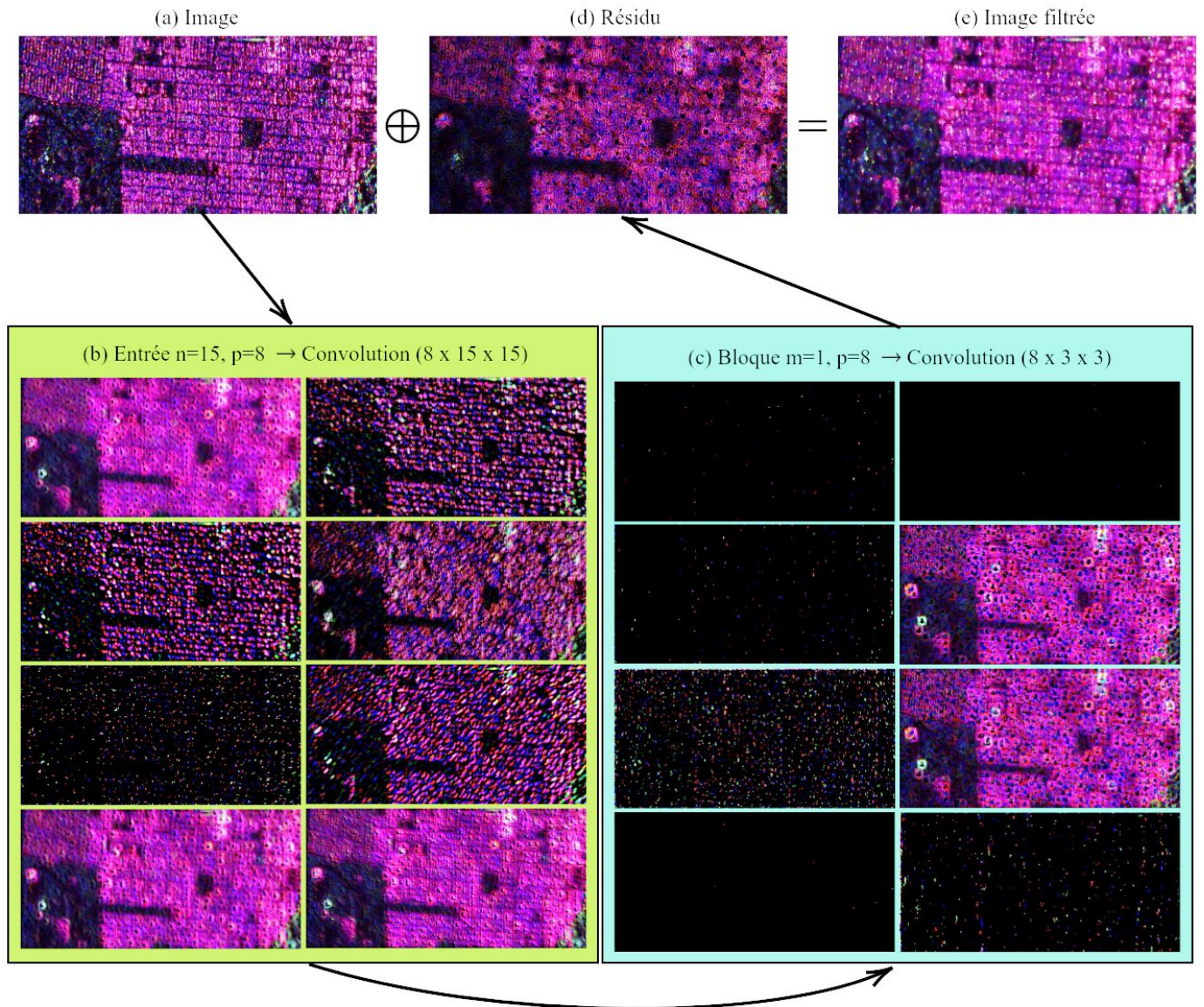
Cette section présente une analyse visuelle des paramètres appris par les réseaux. Nous prenons comme exemple de base le modèle simple à trois couches  $m = 1$ ,  $p = 8$  et  $n = 15$ . Ce modèle est appliqué sur une portion de l'image RADARSAT-2 en matrice de cohérence pour illustrer notre analyse.

### 4.2.1. Les sorties des couches d'activation

La technique de visualisation la plus simple est de montrer les activations du réseau pendant la passe en avant. La Figure 4.5 montre chacune des activations pour toutes les sorties. La première couche convolutive est représentée sur la figure par la boîte verte annotée *(b)*. Cette couche est formée de 8 convolutions de taille  $15 \times 15$  et les cartes d'activation sont les 8 images respectives associées aux filtres à l'intérieur de la boîte de la figure. Ces cartes servent par la suite d'intrant à la couche suivante *(c)* composée aussi de 8 convolutions mais de taille  $3 \times 3$ . Elle produisent à leur tour les cartes d'activation secondaires. Et finalement la dernière carte présente le résidu *(d)* qui additionné à l'image bruitée *(a)* donne l'image filtrée *(e)*. Pour les réseaux avec des couches d'activation de type **ReLU**, les résultats commencent à paraître relativement *blobby* et denses, mais à mesure que l'apprentissage progresse, les activations deviennent généralement plus rares et localisées. Un piège dangereux que l'on peut facilement remarquer avec cette visualisation est que certaines cartes d'activation peuvent être toutes nulles pour de nombreuses entrées. Ceci peut indiquer des filtres morts et ou bien un symptôme de sur-apprentissage.

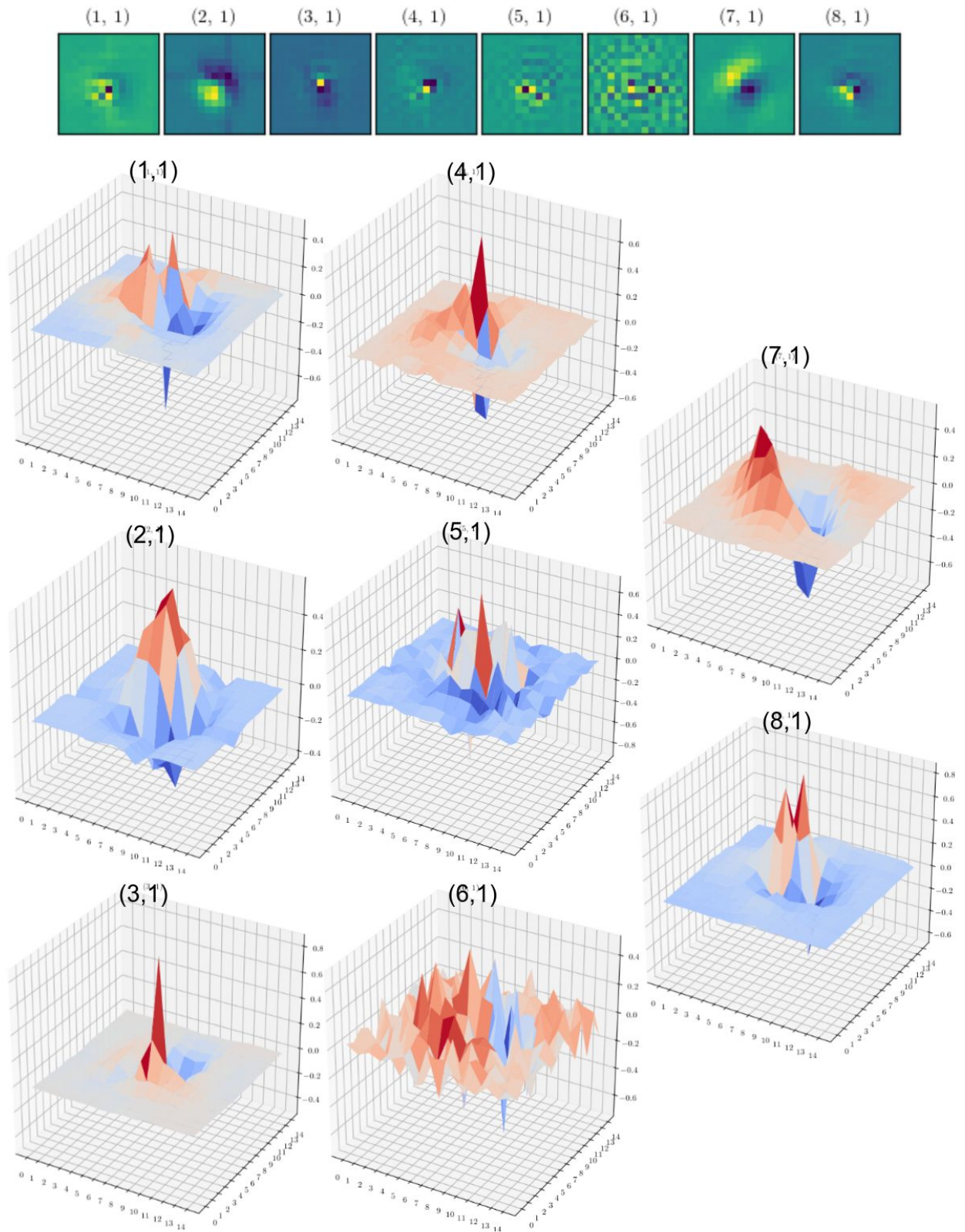
### 4.2.2. Les poids des filtres des convolutions

La deuxième stratégie communément utilisée consiste à visualiser les poids des filtres. Les filtres de la première couche convolutive qui traite directement les données d'entrée sont généralement plus faciles à interpréter et la visualisation de ceux-ci permet de vérifier si le réseau est bien entraîné. Un réseau bien entraîné montre généralement des filtres lisses sans aucun motif bruyant ou aléatoire. Des schémas bruyants peuvent être un indicateur d'un réseau qui n'a pas été entraîné assez longtemps, ou peut-être d'une régularisation insuffisante qui peut avoir conduit à un sur-apprentissage. La Figure 4.6-b donne une représentation intéressante des 8 filtres de taille  $15 \times 15$  de notre réseau exemple. Leur fonction principale



**Fig. 4.5.** Exemple des activations du réseau à trois couche  $m = 1$   $p = 8$   $n = 15$ . La Figure (a) est l'image d'entrée des matrices de cohérence visualisée par le *span*. La Figure (b) montre les 8 activations de la première couche. La Figure (c) montre les 8 activations du bloc convolutif. La Figure (d) présente la carte des résidus et la Figure (e) montre le résultat de l'opération résiduel  $\oplus$ ,  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .

est d'extraire des primitives de base comme des lignes ou des points. Et comme observé par de nombreux travaux en apprentissage machine profond, les filtres de la première couche s'approchent naturellement de la représentation des filtres de Gabor. La représentation en trois dimensions permet de mieux interpréter la fonction des filtres si nous la comparons à la représentation 2-D en intensité (Fig. 4.6-a). L'appréciation générale est bonne et l'on peut discerner que les filtres sont plutôt du type passe-hauts. Les filtres numérotés (7,1) et (2,1) sont de bons exemples de filtres de détection d'arêtes selon une orientation donnée.



**Fig. 4.6.** Visualisation des filtres  $15 \times 15$  de la première couche convolutive.



### 4.3. L'estimation des paramètres de la matrice de cohérence sur les échantillons homogènes

Cette section présente les résultats de l'estimation des matrices de cohérence pour certaines des classes de diffuseurs polarimétriques de la décomposition  $H/\bar{\alpha}$ . À la Section 3.1.3 nous avons décrit le procédé par lequel nos simulations de signatures sur des surfaces homogènes de tailles  $256 \times 256$  ont été produites à partir des matrices de cohérences du tableau 3.1. Ces échantillons ont été filtrés par les modèles produits sur les données d'entraînement homogènes. À titre de comparaison, nous avons appliqué les filtres "standard" polarimétriques suivants aux échantillons:

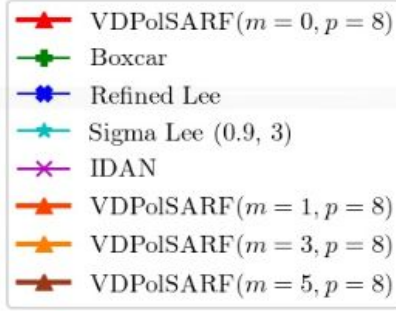
- (1) Le filtre de la moyenne (**Boxcar**);
- (2) Le filtre de Lee affiné (**Refined Lee**);
- (3) Le filtre de Lee Sigma (**Sigma Lee**);
- (4) Le filtre **IDAN**;

Nous présentons les résultats pour les trois grandes classes de diffuseurs selon les angles  $\bar{\alpha}$  du diagramme  $H/\bar{\alpha}$  suivants:

- (1) Les diffuseurs de type **double bond** représentés par la classe **Z4 (Double réflexion)**;
- (2) Les diffuseurs de type **volumétrique** représentés par la classe **Z5 (Particules anisotropiques)**;
- (3) Les diffuseurs de type **surfaccique** représentés par la classe **Z3 (Surfaces de Bragg)**;

La plupart des résultats sont présentés sous la forme de graphiques qui illustrent l'estimé d'un paramètre polarimétrique en fonction de la taille  $n$  de la fenêtre d'entrée et de l'algorithme utilisé pour filtrer les images. Chaque courbe des graphiques correspond à un filtre en particulier et est identifiée avec une ligne de couleur unique. Un marqueur identifie tous les types de filtres. Les filtres que nous avons entraînés portent le même marqueur  $\Delta$  mais de couleur différente. La Figure 4.7 montre la légende qui sera affichée sur chaque graphique pour départager les résultats.

Les Figures 4.8, 4.10, 4.12 montrent les résultats réalisés sur les données homogènes en appliquant les modèles entraînés sur les données d'apprentissage homogènes. Elles présentent



**Fig. 4.7.** Exemple de la légende utilisée pour identifier les différents algorithmes de filtrage.

la moyenne de l'erreur relative des 9 éléments de la matrices de cohérence pour chacun de nos diffuseurs. Les Figures 4.9, 4.11, 4.13 présentent l'écart-type sur la mesures des moyennes.

Les figures se séparent en deux parties. La première colonne montre la moyenne des erreurs calculées sur les termes diagonaux en puissance ( $T_{11}, T_{22}, T_{33}$ ). Les deux colonnes suivantes donnent les résultats respectifs pour les parties réels et imaginaire des termes croisés ( $T_{12}, T_{13}, T_{23}$ ) de la matrice de cohérence. Par exemple, pour la valeur  $T_{11}$  du **Boxcar** de taille  $n = 5$  l'erreur relative  $|\epsilon|$  est égale à  $15.2 \pm 12.4\%$  pour la donnée de type *double bond*. Nous remarquons les grandes amplitudes au niveau de certains termes croisés. Ces derniers peuvent diverger de plus de 400 %. Ceci s'explique par le fait que les valeurs sont extrêmement petites et proches de zéro mais non-nulle. Un petit changement est susceptible de générer une grande erreur relative à cause de la petite valeur au dénominateur du calcul. Il est à noter que nous avons expressément généré des matrices de cohérence telles qu'aucun élément de la matrice est nul afin d'éviter le piège de la division par zéro.

Le filtre **Boxcar** sépare les résultats en deux ensembles en fonction des filtres. Les filtres "standards" obtiennent généralement des résultats inférieurs au **Boxcar**. Le filtre **IDAN** présente des résultats plus biaisés par rapport aux termes en puissance si on le compare aux autres filtres. Il atteint rapidement une asymptote de l'ordre de 23 % indépendamment du nombre de pixels utilisés pour effectuer ses estimés. Par contre il donne souvent de meilleurs résultats pour le terme  $im\{T_{13}\}$  que les autres filtres "standard". L'explication valable vient que l'estimé par la médiane est moins biaisé sur la distribution presque symétrique des partie réelles et imaginaires des termes croisés comparée à l'estimation calculées sur les distributions non-symétriques des termes diagonaux de la matrice de cohérence. De manière générale le filtre **Refined LEE** donne des résultats égaux ou légèrement supérieurs au filtre **Sigma**

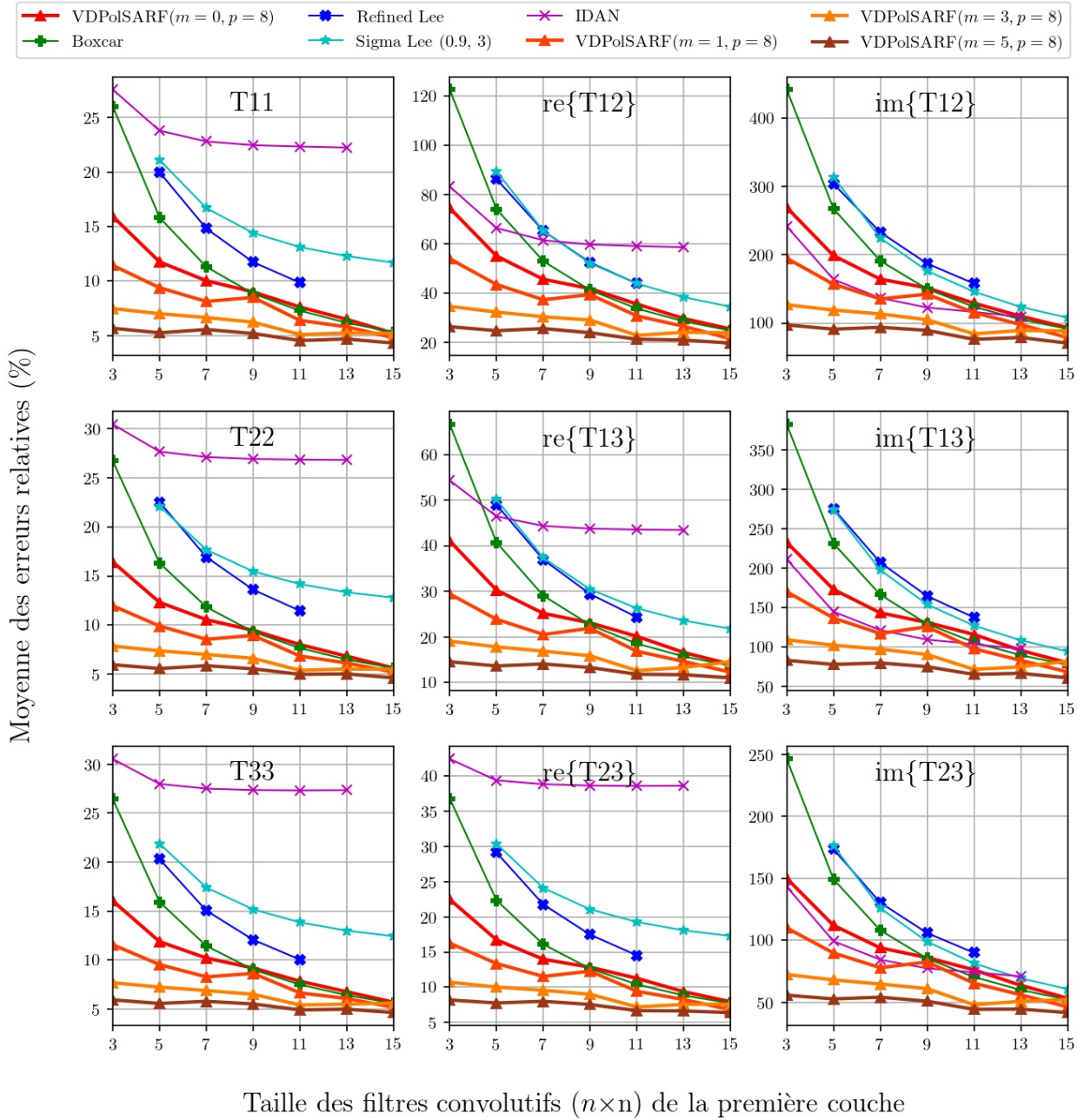
**Lee.** Ceci est un peut étonnant compte-tenu du fait que le filtre **Refined LEE** génère une forme de texture dans les images filtrées tandis que le filtre **Sigma Lee** semble donner des résultats plus homogènes (Figs. 4.18, 4.20, et 4.22 pour une comparaison visuelle).

Les filtres que nous avons entraînés (**VDPolSARF**) ont des résultats supérieurs comparés aux filtres "standards". Ils ont une asymptote qui tend vers un biais égal ou meilleur que le filtre **Boxcar** avec une large fenêtre ( $n = 15$ ). On remarque que les résultats tendent à devenir indépendants de la taille de la fenêtre d'entrée lorsque le nombre de couches augmentent. Le filtre **VDPolSARF** ( $m = 0, p = 8, n = 3$ ) qui ne contient que deux couches se comporte comme un filtre **Boxcar** mais amélioré. Par exemple pour l'estimé  $T_{11}$  du diffuseur volumétrique (Fig. 4.10 et Fig. 4.11), le **Boxcar** tend vers un biais de  $5.0 \pm 3.7\%$  et pour notre filtre le biais tend vers  $4.9 \pm 3.7\%$ . Tous deux ont des biais et des écart-types qui décroissent en fonction de la taille de l'échantillon. Le biais de départ ( $n = 3$ ) pour le filtre **Boxcar** est de  $26.2 \pm 20.4\%$  et pour notre filtre nous obtenons un biais de  $15.7 \pm 11.9\%$ . Pour le filtre **VDPolSARF**, l'amélioration du biais et de son écart-type en fonction du nombre de blocs convolutifs ( $m$ ) est intéressante car elle permet à première vue d'éliminer la variable de la taille de la fenêtre d'entrée. Les biais et leur écart-types décroissent de 29% de  $m = 0$  ( $15.7 \pm 11.9\%$ ) à  $m = 1$  ( $11.1 \pm 8.4\%$ ), de 36% de  $m = 1$  à  $m = 3$  ( $7.0 \pm 5.3\%$ ) et de 25% de  $m = 3$  à  $m = 5$  ( $5.3 \pm 3.9\%$ ). Au total entre  $m = 0$  et  $m = 5$  nous avons une amélioration du biais de l'ordre de 66 % en empilant 5 couches de blocs convolutifs avec des filtres  $3 \times 3$ .

Nous remarquons que l'estimation semble meilleure sur les termes en puissance de la matrice de cohérence et ceci pour chaque diffuseur. Les termes complexes sont plus variables et semblent plus difficiles à estimer pour les filtres courts ( $m = 0, 1$ ). Nous avons de moins bons résultats pour les tailles de fenêtres  $n = 9$  et  $n = 15$ . Ils sortent de la courbe attendue: monotone et décroissante. Est-ce un effet du hasard de l'apprentissage? La question demeure ouverte mais l'apprentissage des filtres plus courts semble être plus imprécis pour les grandes tailles de fenêtre.

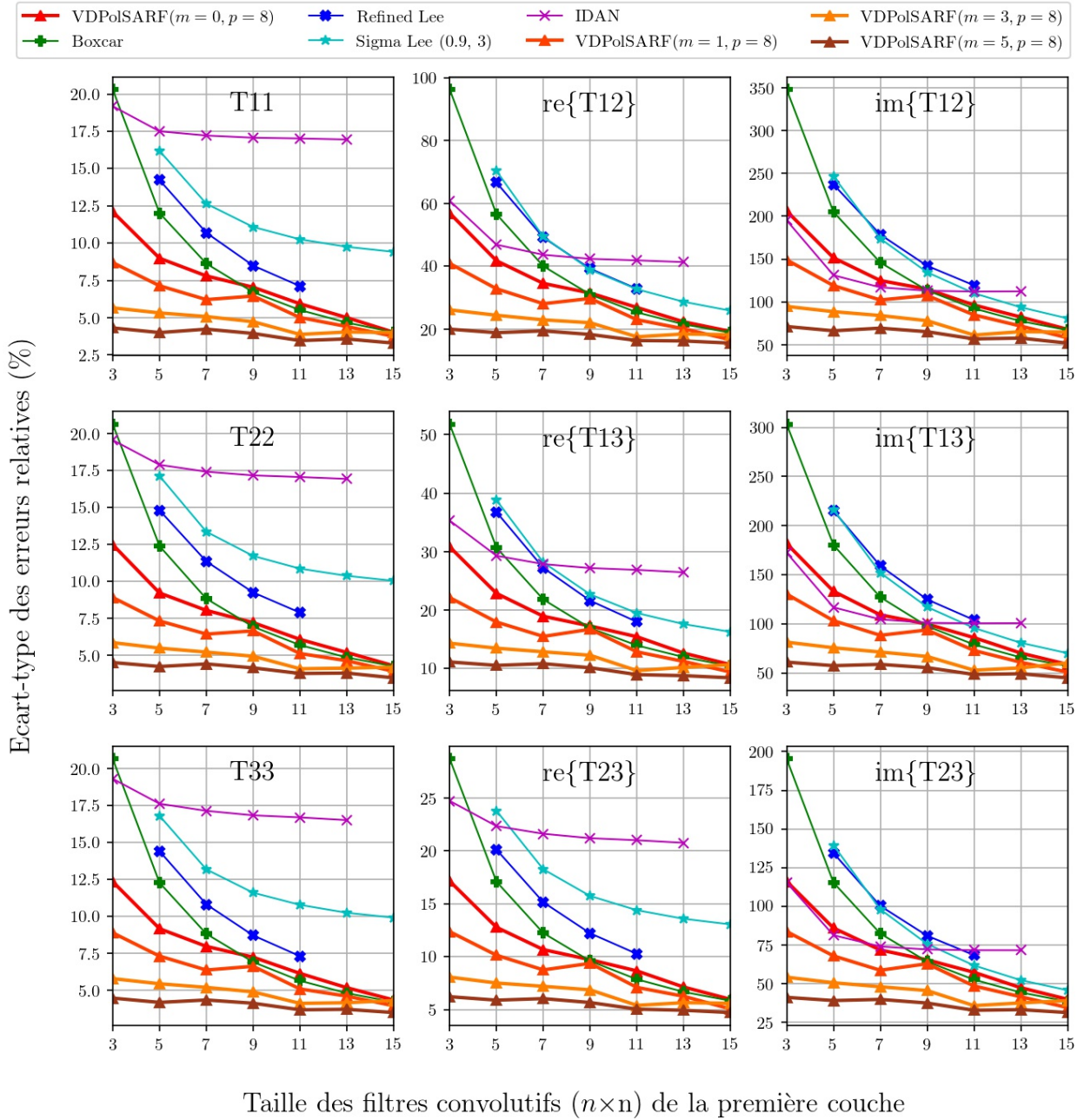
En conclusion de cette section nous pouvons considérer que les filtres appris sont de bons estimateurs de la matrice de cohérence pour les données une vue sur les surfaces homogènes. En particulier la famille des filtres plus profond  $m = 3, 5, p = 8$  donnent les meilleures performances en terme de biais. Les biais et leurs écart-types demeurent constants en fonction de la taille de la fenêtre.

### Double Réflexion (Diffuseurs double bond)



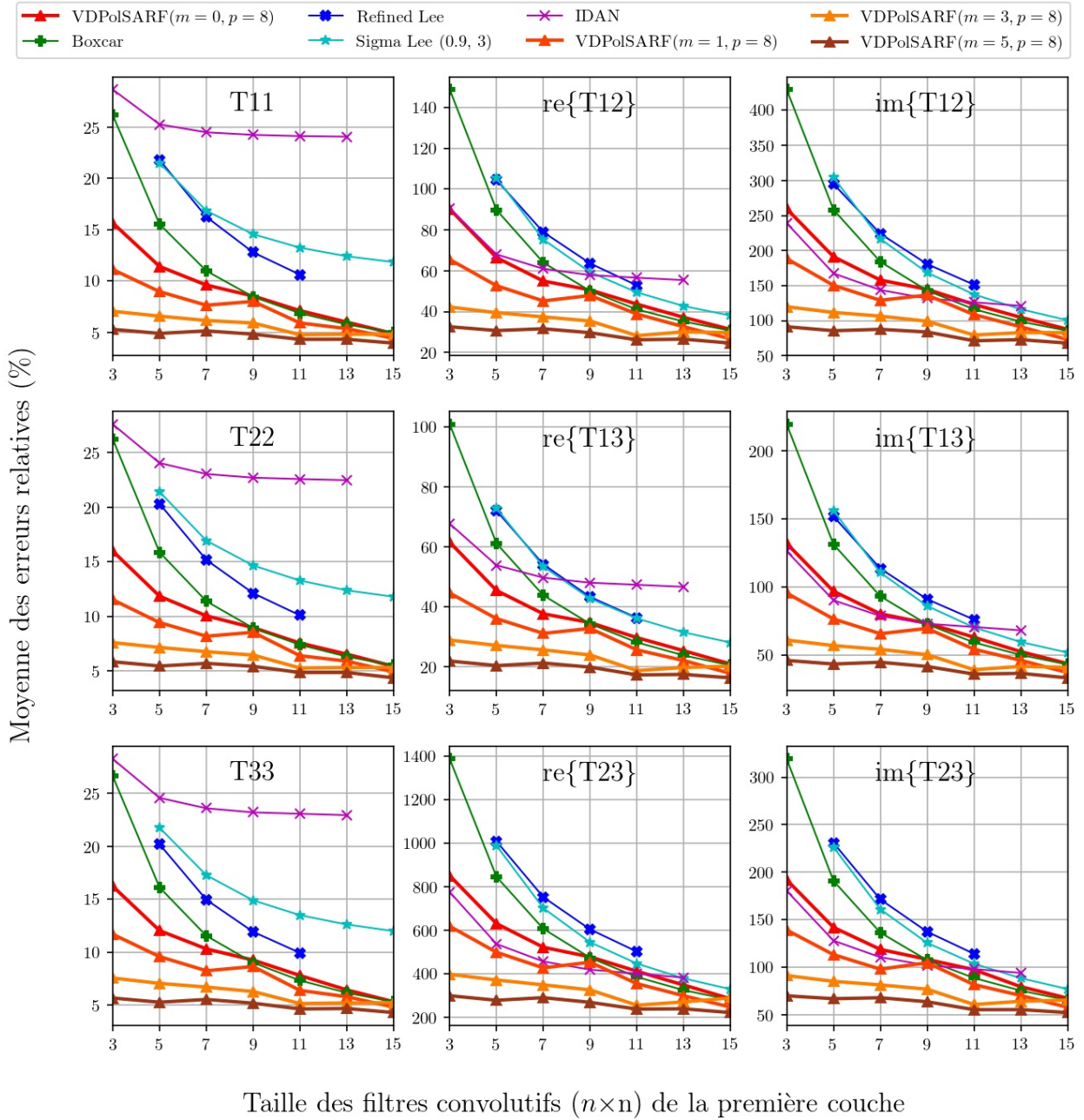
**Fig. 4.8. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

## Double Réflexion (Diffuseurs double bond)



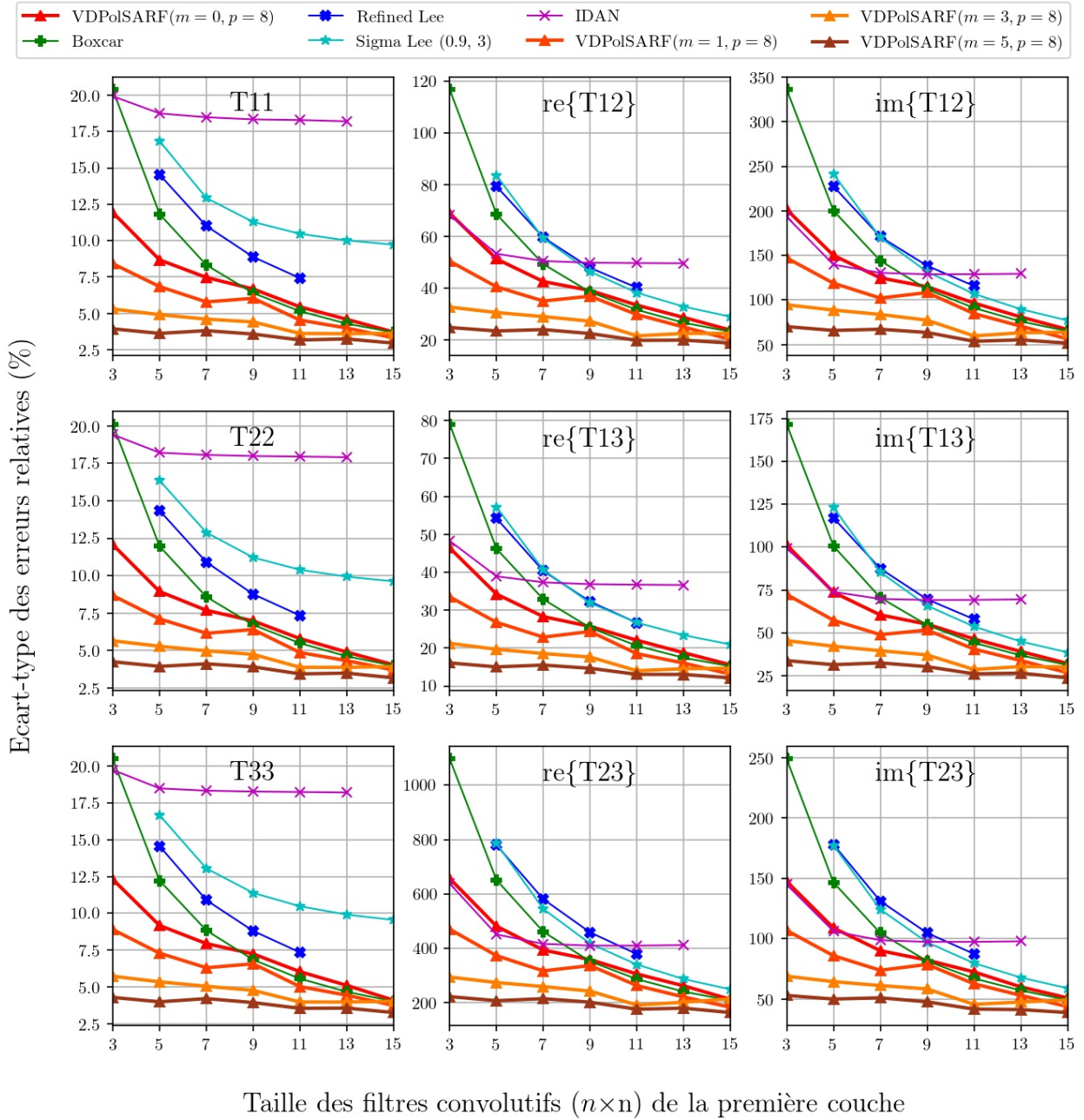
**Fig. 4.9. Diffuseurs double bond.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

Particules Anisotropiques (Diffuseurs volumétriques)



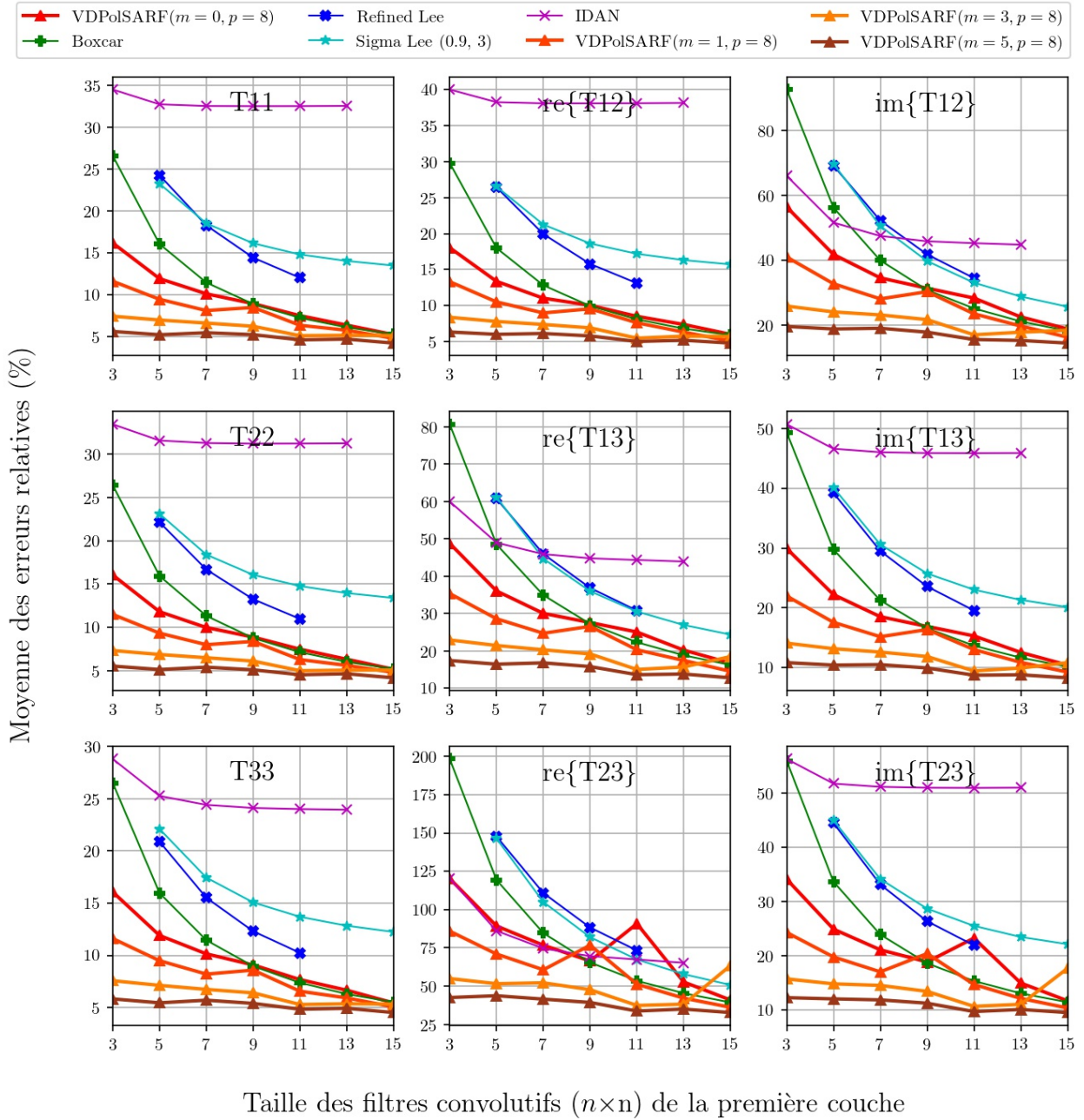
**Fig. 4.10. Diffuseurs volumétriques.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

Particules Anisotropiques (Diffuseurs volumétriques)



**Fig. 4.11. Diffuseurs volumétriques.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

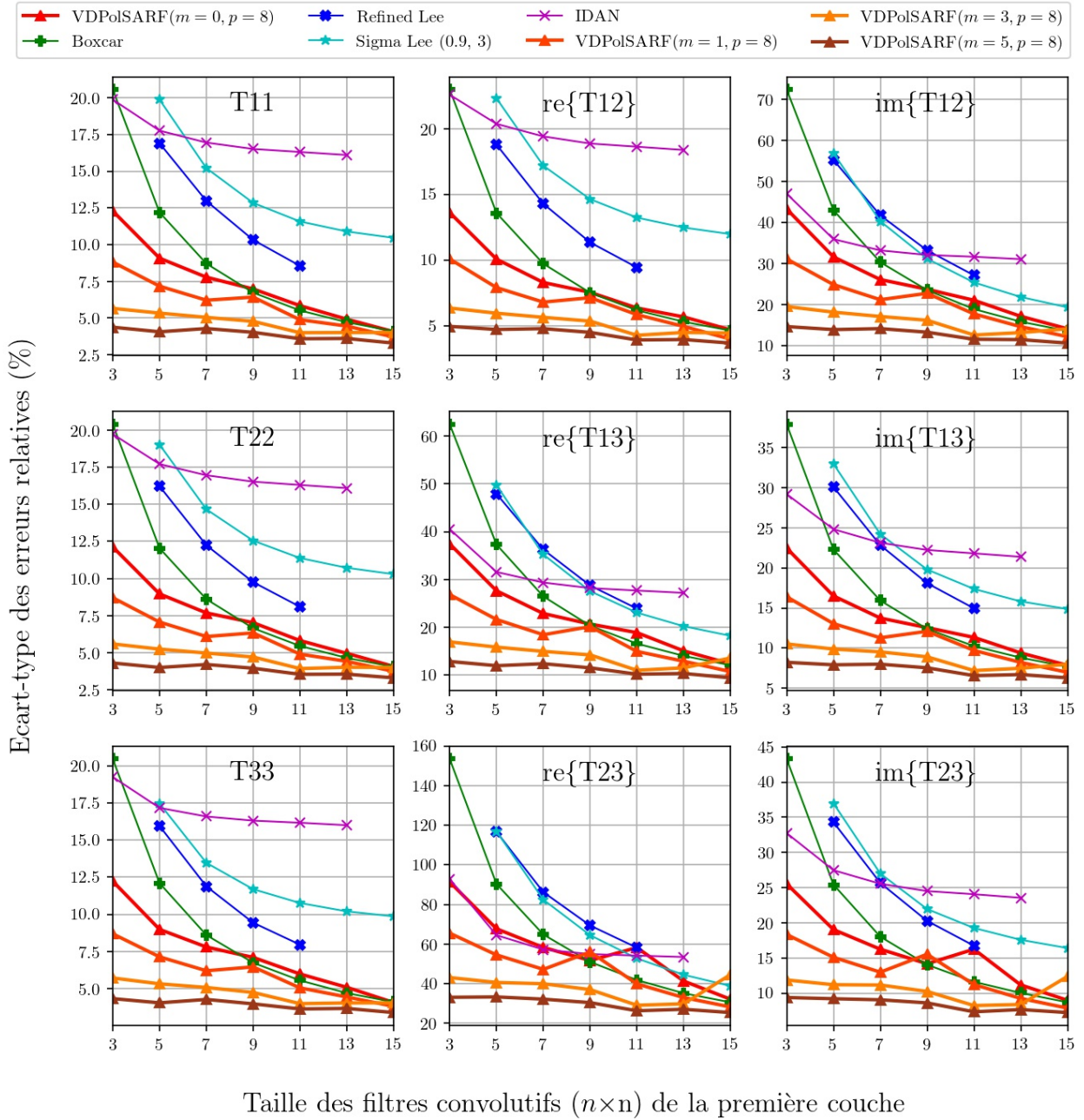
### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. 4.12. Diffuseurs surfaciques.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .



### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. 4.13. Diffuseurs surfaciques.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

#### 4.4. L'estimation du nombre de vue équivalent sur les échantillons homogènes

Cette section présente les résultats sur l'estimation du nombre équivalent de vues (**NEV**) sur les échantillons homogènes de nos trois diffuseurs. Cette mesure indique l'intensité de réduction du chatolement et nous la calculons sur les trois termes en puissance ( $T11, T22, T33$ ) de la matrice de cohérence. Nous rappelons que les échantillons non filtrés ont un **NEV**  $\approx 1$  (voir Tab. 3.2). Sur les images en intensité, le **NEV** est calculé comme suit:

$$NEV = \frac{1}{\beta^2} \quad (4.4.1)$$

où  $\beta$  est défini comme le ratio de l'écart-type sur la moyenne des intensités  $I$ :

$$\beta = \frac{\sqrt{E[(I - E[I])^2]}}{E[I]} \quad (4.4.2)$$

Les Figures 4.14, 4.15 et 4.16 présentent le calcul du **NEV** pour chacun des filtres appliqués aux échantillons homogènes de nos trois types de diffuseurs. Les différentes courbes illustrent le **NEV** en fonction de la taille  $n$  de la fenêtre utilisée pour filtrer les données polarimétriques. La courbe du filtre **Boxcar** sépare les résultats en deux groupes. Les filtres "standards" présentent une réduction du chatolement plus faible que le **Boxcar**. Le filtre **IDAN** ne semble pas aussi efficace pour la réduction du chatolement que les autres filtres. Les **NEV** demeurent faibles peu importe la taille en pixels utilisés pour faire les calculs. Le filtre **Refined LEE** donne de meilleurs résultats en fonction de la taille de la fenêtre que son homologue **Sigma Lee**. Le **NEV** du filtre **Refined LEE** semble croître de manière exponentielle comme le **Boxcar** tandis que le filtre **Sigma Lee** semble croître de manière logarithmique et tend à plafonner en fonction de la taille de la fenêtre. Le tableau 4.4 donne les valeurs maximales du **NEV** calculés sur les termes diagonaux des diffuseurs pour le filtre **Refined LEE** ( $11 \times 11$ ) et du filtre **Boxcar** de même taille. Grossièrement le **NEV** du filtrage par **RNC** est doublé par rapport à celui du le filtre **Boxcar**.

Dans le cas des filtres appris, la performance est soit égale ou nettement supérieure au **Boxcar**. La famille des filtres les plus courts ( $m = 0, p = 8$ ) se comporte sensiblement comme ce dernier en fonction de la taille de la fenêtre. Tandis que les filtres les plus profonds ( $m = 5, p = 8$ ) montrent des performances supérieures de plusieurs ordres de magnitude

	<b>T11</b>	<b>T22</b>	<b>T33</b>
<b>Diffuseurs double bond</b>	81.3 / 119.8	66.8 / 110.3	65.2 / 115.4
<b>Diffuseurs volumétriques</b>	61.7 / 135.2	61.4 / 117.1	58.5 / 119.7
<b>Diffuseurs surfaciques</b>	69.6 / 119.8	68.8 / 122.9	67.2 / 117.0

**Tab. 4.4.** Les nombres équivalents de vue calculés sur les échantillons des trois diffuseurs typiques filtrés par le filtre **Refined LEE** ( $n = 11$ ) comparées aux valeurs du filtre **Boxcar** ( $n = 11$ ). Les **NEV** sont calculés sur les termes en puissance de la matrice de cohérence.

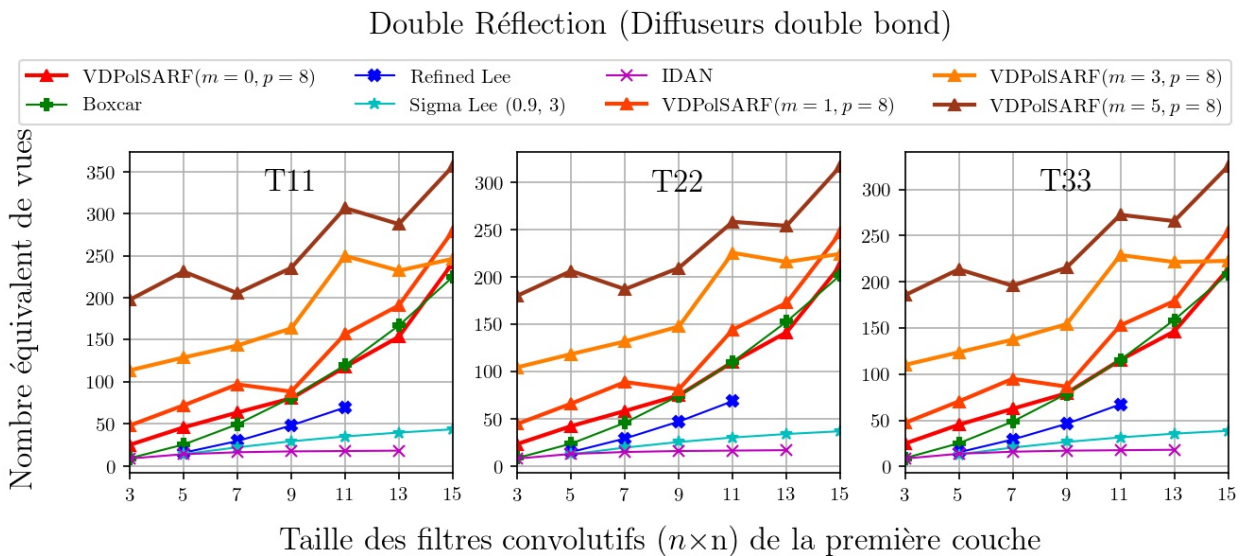
pour des petites tailles de fenêtres comparés au **Boxcar** (voir tab 4.5). Comme nous l’avons remarqué à la section 4.3, les filtres **VDPolSARF** profonds sont moins sensibles à la taille  $n$  de la fenêtre d’entrée pour l’estimation de la matrice de cohérence. Par contre dans le cas du calcul du **NEV**, la taille du filtre a une influence marquée sur le calcul de celui-ci et ceci pour tous les diffuseurs. Le tableau 4.5 met en relief la différence entre les filtres **VDPolSARF** et **Boxcar** de taille  $n = 11$ . La réduction du chatolement est approximativement triplée en faveur du premier. Nous remarquons que pour les filtres de la famille ( $m = 3, p = 8$ ), l’augmentation de la performance s’arrête après le point culminant à  $n = 11$ . Nous sommes sans explications par rapport à cette observation.

	<b>T11</b>	<b>T22</b>	<b>T33</b>
<b>Diffuseurs double bond</b>	306.7 / 119.8	258.2 / 110.3	272.6 / 115.4
<b>Diffuseurs volumétriques</b>	347.6 / 135.2	281.9 / 117.1	300.8 / 119.7
<b>Diffuseurs surfaciques</b>	289.7 / 119.8	299.0 / 122.9	272.8 / 117.0

**Tab. 4.5.** Les nombres équivalents de vue calculés sur les échantillons des trois diffuseurs typiques filtrés par le filtre de **VDPolSARF** ( $m = 5, p = 8, n = 11$ ) comparées aux valeurs du filtre **Boxcar** de même taille. Les **NEV** sont calculés sur les termes en puissance de la matrice de cohérence.

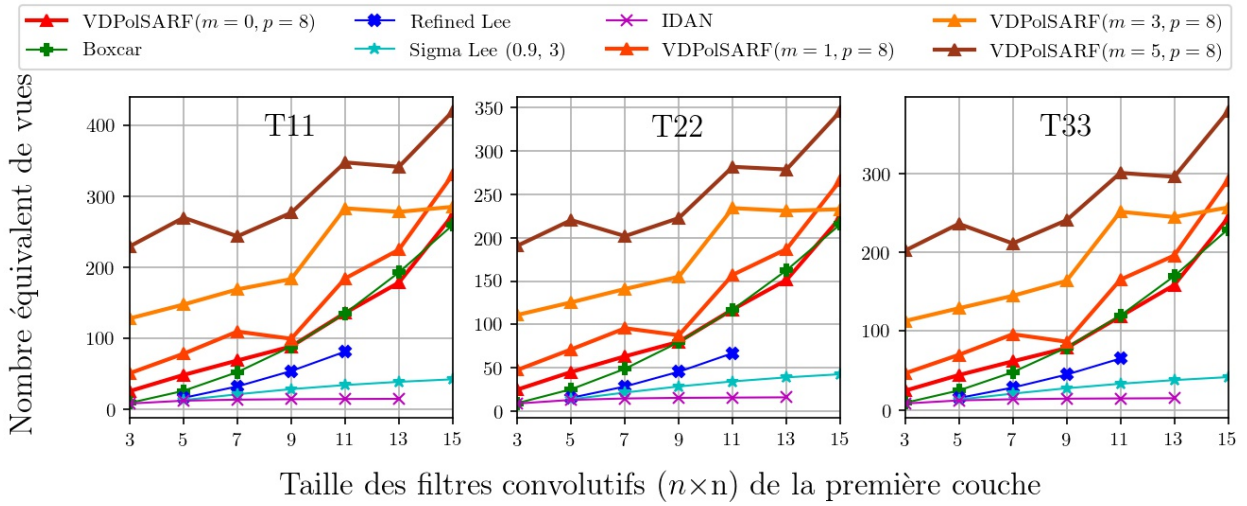
Pour conclure cette section nous présentons un aperçu visuel des résultats du filtrage. Les figures sont présentées en deux parties par diffuseur. Les Figures 4.17 et 4.18 montrent les résultats des filtrages sur le diffuseur *double bond*, les Figures 4.19 et 4.20 montrent les résultats des filtrages sur le diffuseur volumétrique, et finalement les Figures 4.21 et 4.22 montrent les résultats des filtrages sur le diffuseur surfacique. Dans la première partie de chaque paire des figures et sur la première ligne nous affichons l’image caractéristique et

l'image étiquette associée au diffuseur. Ensuite, les filtrages pour les deux tailles de fenêtres  $n = 5$  et  $n = 13$  du filtre **VDPolSARF** et **Boxcar** sont affichés. Dans la seconde partie nous continuons l'affichage du filtrage obtenu à partir du filtre **Refined LEE**, du filtre **Sigma Lee** et du filtre **IDAN**. Toutes les images présentent les valeurs du *span* selon l'ordre  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ : ce qui correspond aux termes de la matrice de cohérence (T22, T33, T11). La visualisation met particulièrement en valeur le lissage soutenu du filtre **VDPolSARF** pour les deux tailles de filtres  $n = 5$  et  $n = 13$ .



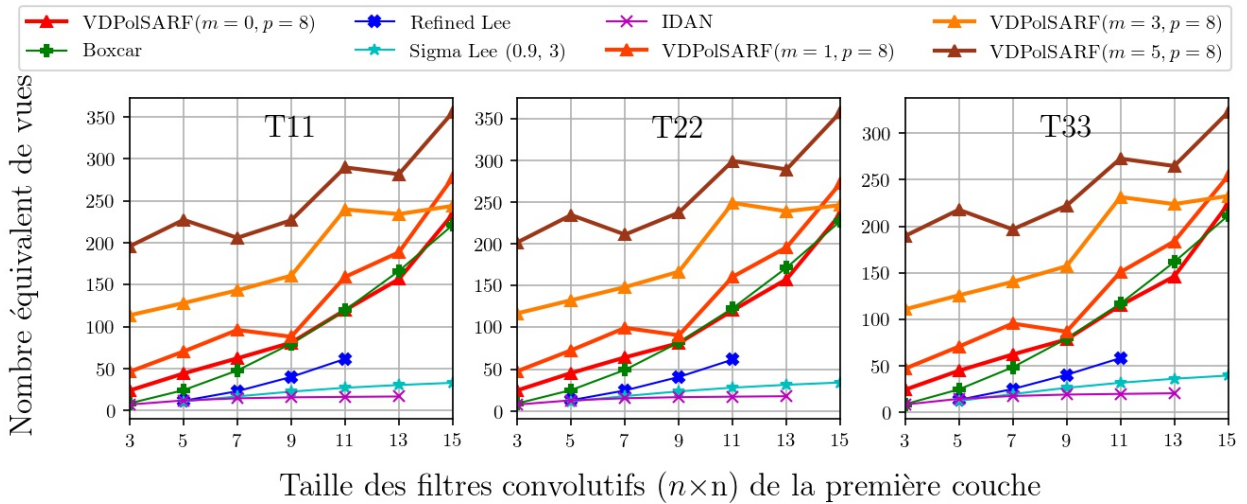
**Fig. 4.14. Diffuseurs double bond.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  entraînés sur les données d'apprentissage homogènes.

### Particules Anisotropiques (Diffuseurs volumétriques)

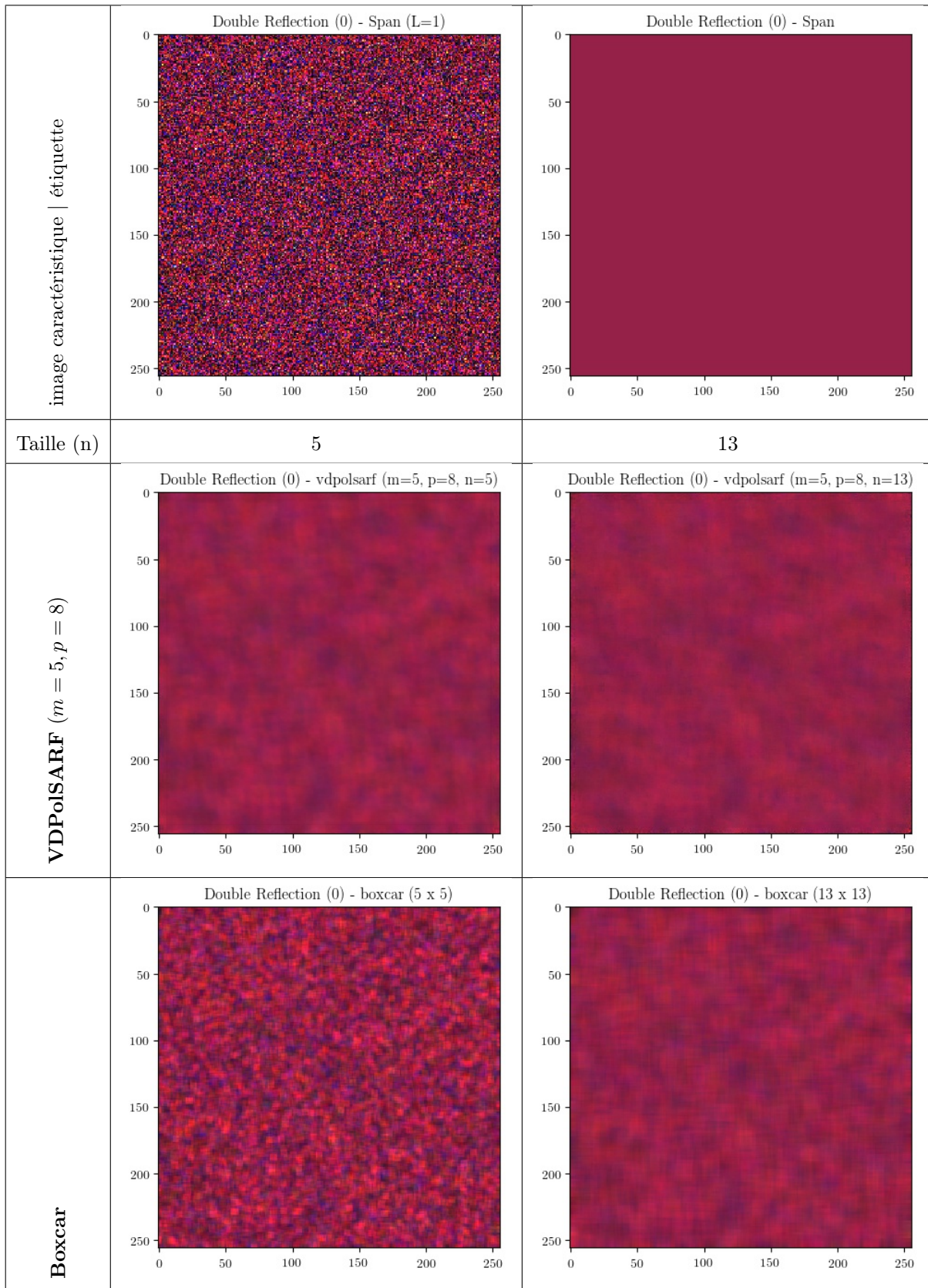


**Fig. 4.15. Diffuseurs volumétriques.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  entraînés sur les données d'apprentissage homogènes.

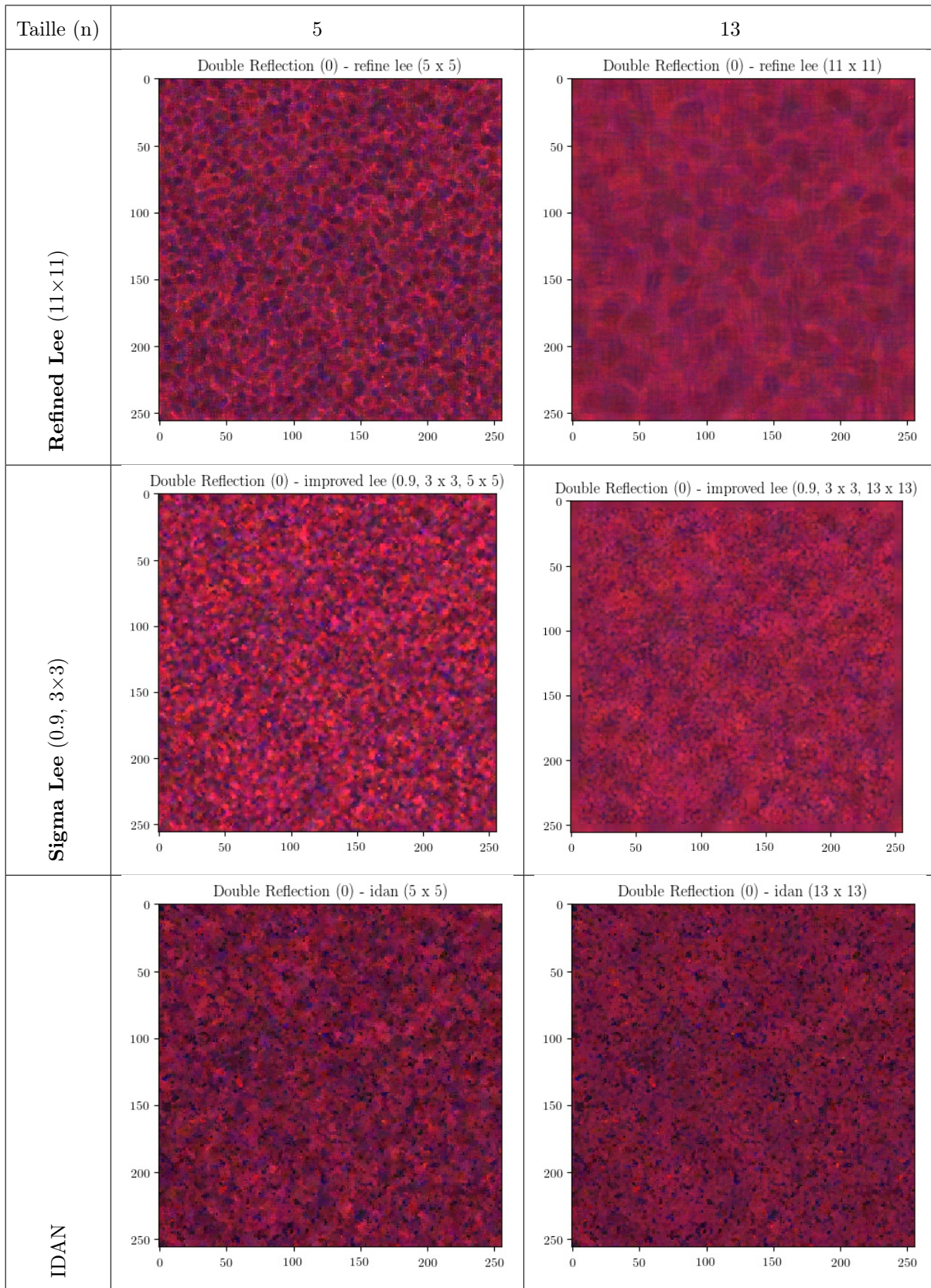
### Surfaces de Bragg (Diffuseurs surfaciques)



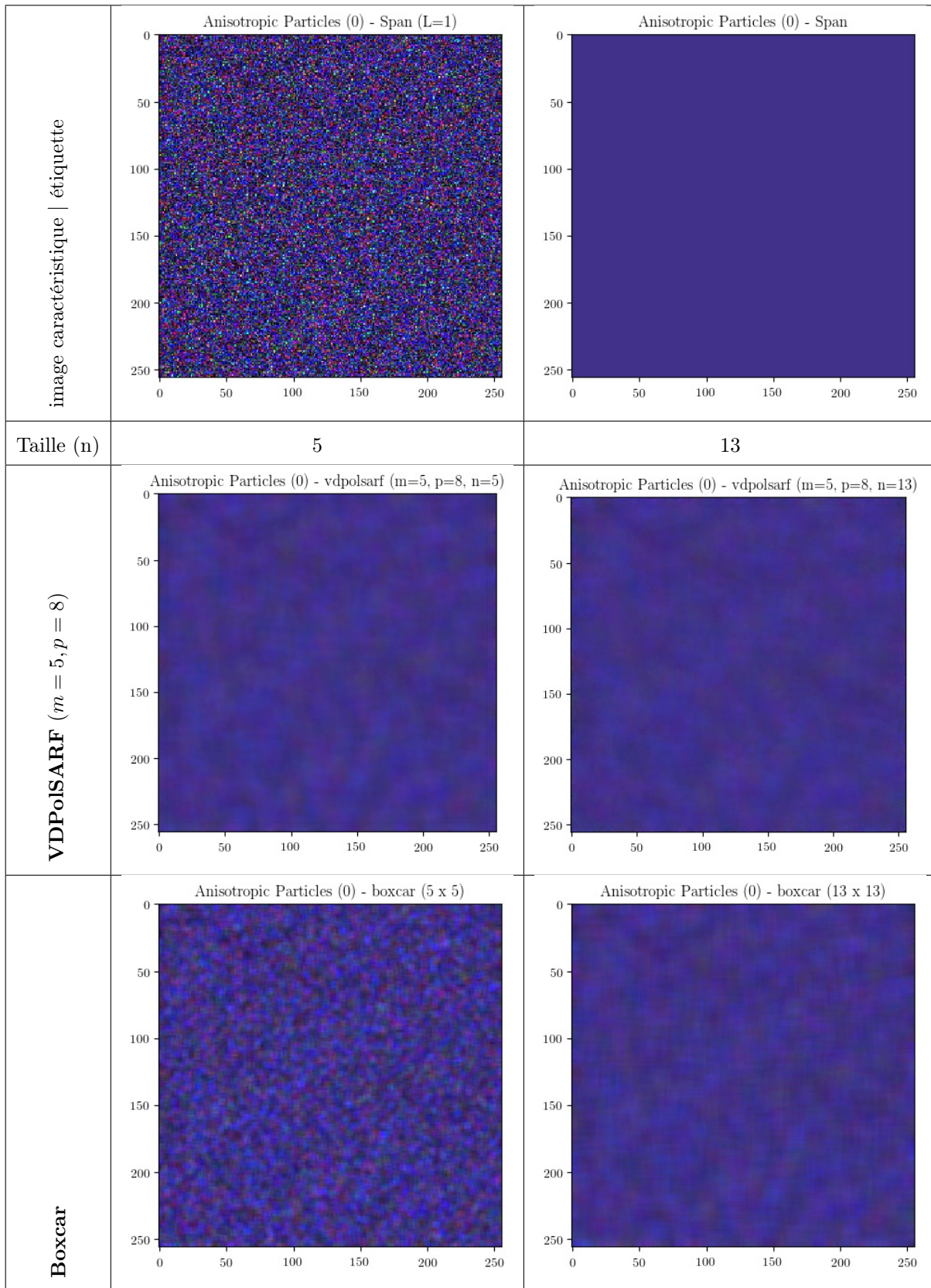
**Fig. 4.16. Diffuseurs surfaciques.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  entraînés sur les données d'apprentissage homogènes.



**Fig. 4.17. Diffuseurs double bond.** Comparaison entre les différents filtrages - Partie 1.  
 $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$



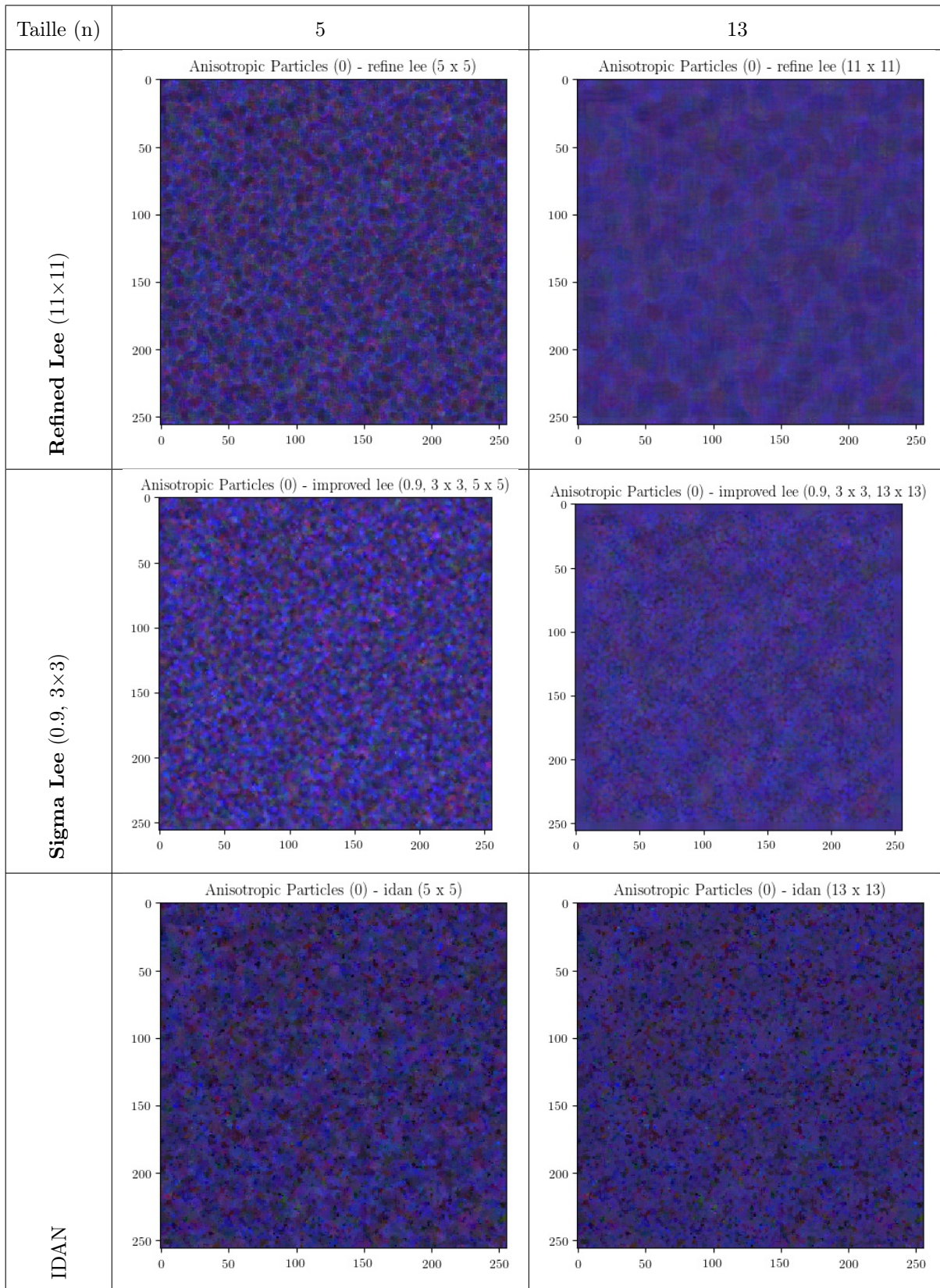
**Fig. 4.18. Diffuseurs double bond.** Comparaison entre les différents filtrages - Partie 2.  
 $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$



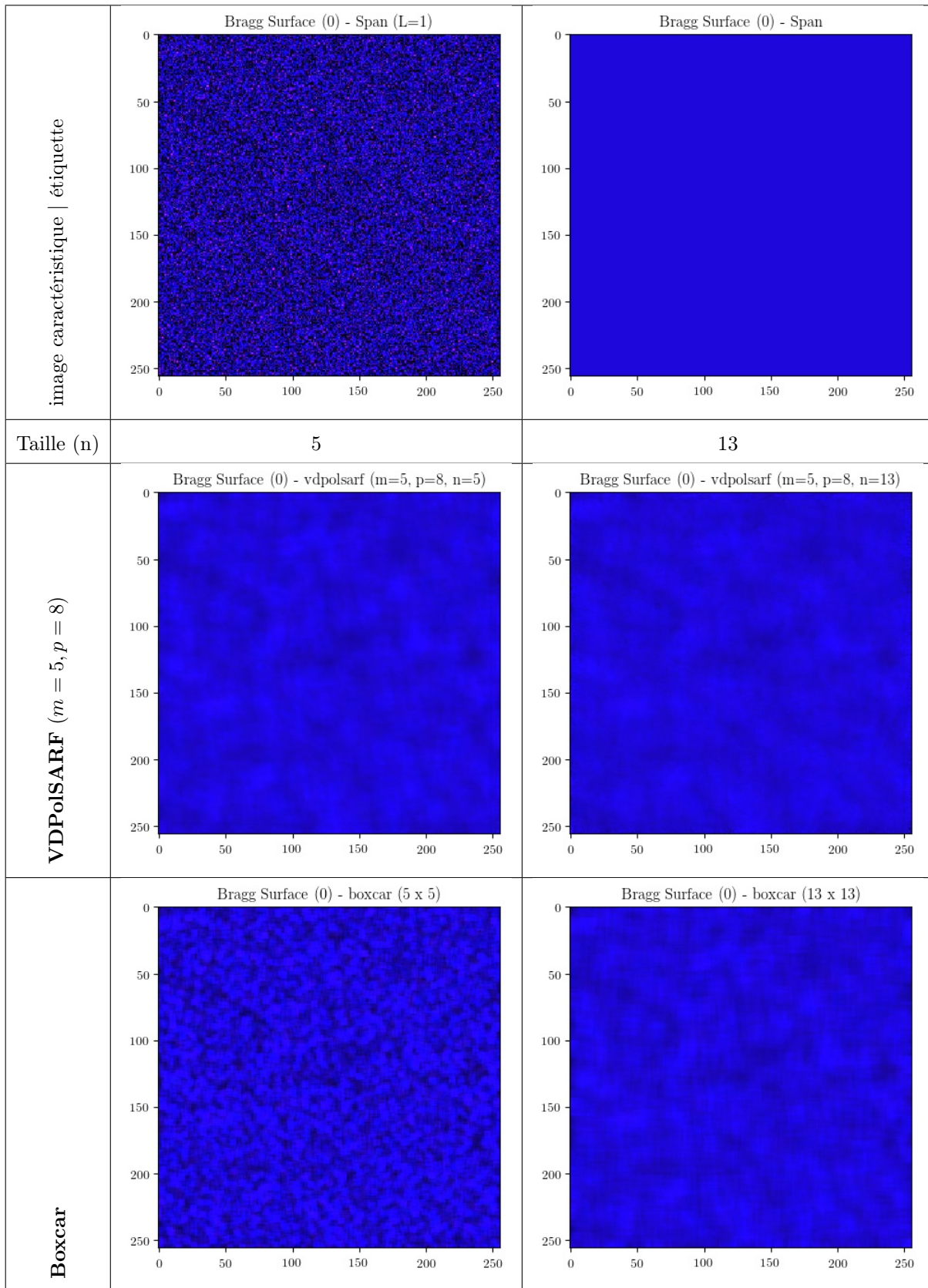
**Fig. 4.19. Diffuseurs volumétriques.** Comparaison entre les différents filtrages - Partie 1.

$(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$

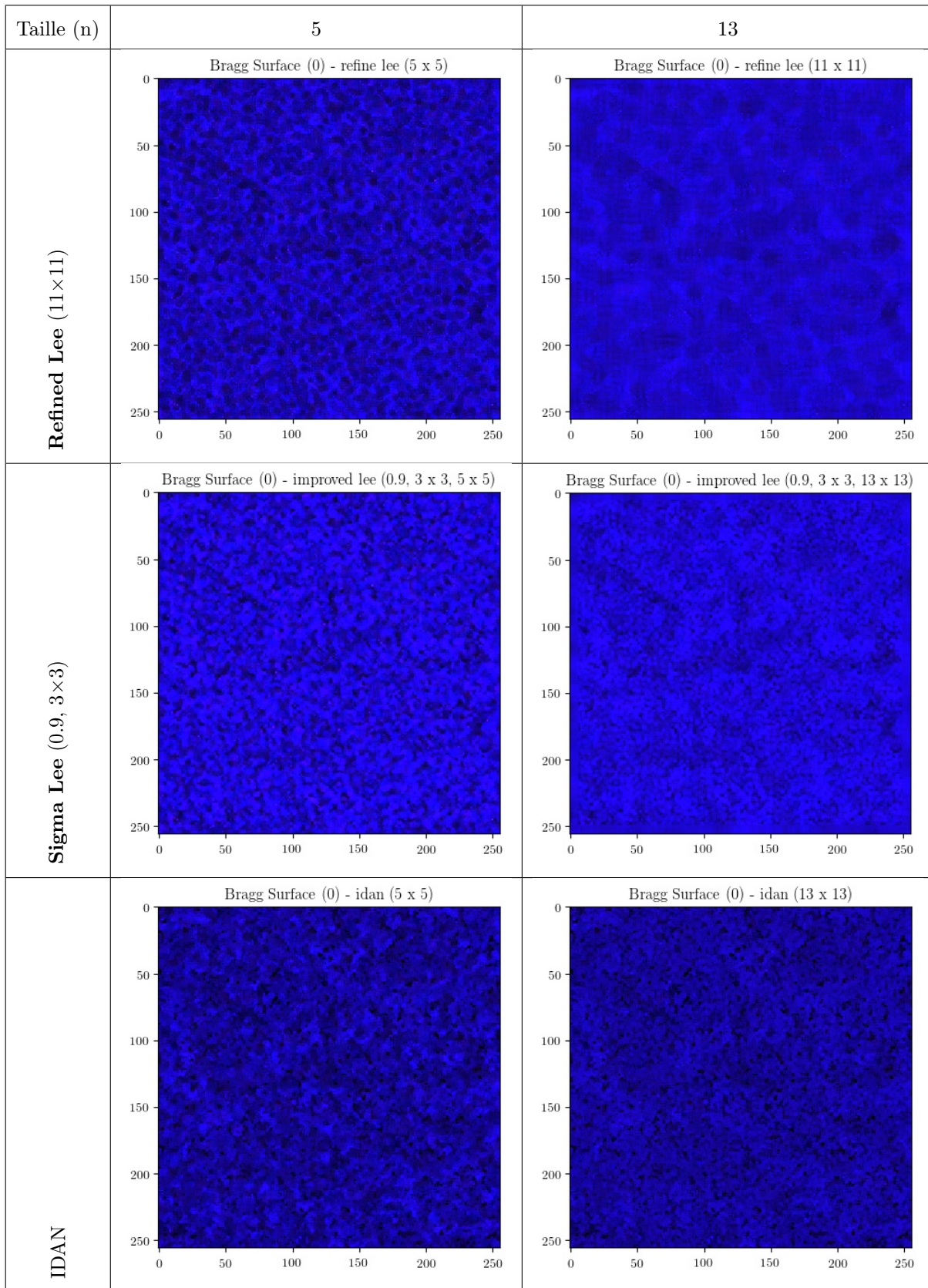




**Fig. 4.20. Diffuseurs volumétriques.** Comparaison entre les différents filtrages - Partie 2.  
 $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$



**Fig. 4.21. Diffuseurs surfaciques.** Comparaison entre les différents filtrages - Partie 1.  
 $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$



**Fig. 4.22. Diffuseurs surfaciques.** Comparaison entre les différents filtrages - Partie 2.  
 $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$

## 4.5. L'estimation des paramètres de la décomposition H-A-Alpha sur les données homogènes

Cette section présente les résultats relatifs à la décomposition  $H/A/\bar{\alpha}$  des échantillons filtrés pour nos trois diffuseurs types. Les Figures 4.23, 4.25 et 4.27 montrent la moyenne des erreurs relatives en fonction de la taille  $n$  de la fenêtre d'entrée et ceci pour chacun des filtres "standards" et appris. De façon similaire, les Figures 4.24, 4.26 et 4.28 montrent l'écart-type des erreurs relatives en fonction de la taille  $n$  de la fenêtre d'entrée et ceci pour chacun de filtres "standards" et appris. Les graphes présentent les résultats pour chaque terme de la décomposition, soit l'entropie, l'anisotropie et l'angle  $\bar{\alpha}$  (voir section 2.1.3 pour les aspects théoriques).

Dans le cas de l'estimation de l'entropie des diffuseurs de type *double bond* et volumétrique, l'observation générale est que les modèles appris se comportent mieux que les filtres "standards" ou sont similaires au filtre **Boxcar**. Par contre pour le diffuseur surfacique, on remarque des erreurs plus élevées et des résultats plus incertains après que la taille  $n$  de la première couche convolutive dépasse 7 pour les modèles courts ( $m = 0, 1$ ). Cependant pour les modèles profonds ( $m = 5$ ) les résultats sont semblables au filtre **Boxcar** à large fenêtre  $15 \times 15$ .

	<b>Double bond</b>	<b>Volume</b>	<b>Surface</b>
<b>VDPLSARF</b> ( $m = 5, p = 8, n = 15$ )	$1.7 \pm 1.3\%$	$1.1 \pm 0.1\%$	$5.3 \pm 4.1\%$
<b>Boxcar</b> $15 \times 15$	$2.1 \pm 1.6\%$	$1.4 \pm 0.1\%$	$4.6 \pm 3.4\%$

**Tab. 4.6.** Biais relatifs sur l'estimé de l'entropie. Comparaison entre le modèle **VDPLSARF** ( $m = 5, p = 8, n = 15$ ) et le filtre **Boxcar**  $15 \times 15$

Dans le cas du diffuseur *double bond*, l'estimation de l'anisotropie est légèrement moins bonne pour les modèles courts ( $m = 0, 1$ ) ou égale ou supérieure pour les modèles plus profonds ( $m = 3, 5$ ) si nous les comparons aux résultats du **Boxcar**. Dans le cas du diffuseur volumétrique les résultats sont égaux ou supérieurs au **Boxcar**. Par contre pour le diffuseur surfacique les résultats sont nettement moins précis pour les modèles courts ( $m = 0, 1$ ).

Dans le cas du paramètres  $\bar{\alpha}$ , l'estimation est égale et supérieure au filtre **Boxcar**. Le biais relatif est semblable pour les trois type de diffuseur (voir Tab. 4.8).

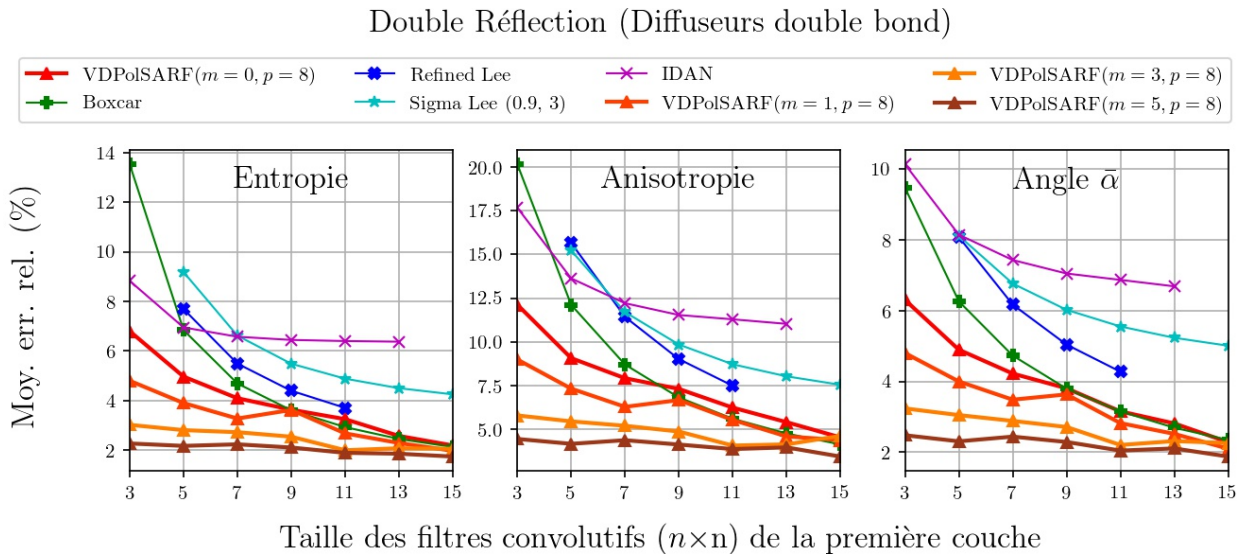
	Double bond	Volume	Surface
VDPLSARF ( $m = 5, p = 8, n = 15$ )	$3.4 \pm 2.7\%$	$15.7 \pm 11.2\%$	$19.3 \pm 14.3\%$
Boxcar $15 \times 15$	$4.1 \pm 3.1\%$	$19.5 \pm 14.4\%$	$14.8 \pm 11.1\%$

**Tab. 4.7.** Biais relatifs sur l'estimé de l'anisotropie. Comparaison entre le modèle **VDPLSARF** ( $m = 5, p = 8, n = 15$ ) et le filtre **Boxcar**  $15 \times 15$ .

	Double bond	Volume	Surface
VDPLSARF ( $m = 5, p = 8, n = 15$ )	$1.9 \pm 1.4\%$	$2.2 \pm 1.7\%$	$1.7 \pm 1.3\%$
Boxcar $15 \times 15$	$3.3 \pm 1.8\%$	$2.8 \pm 2.1\%$	$2.1 \pm 1.5\%$

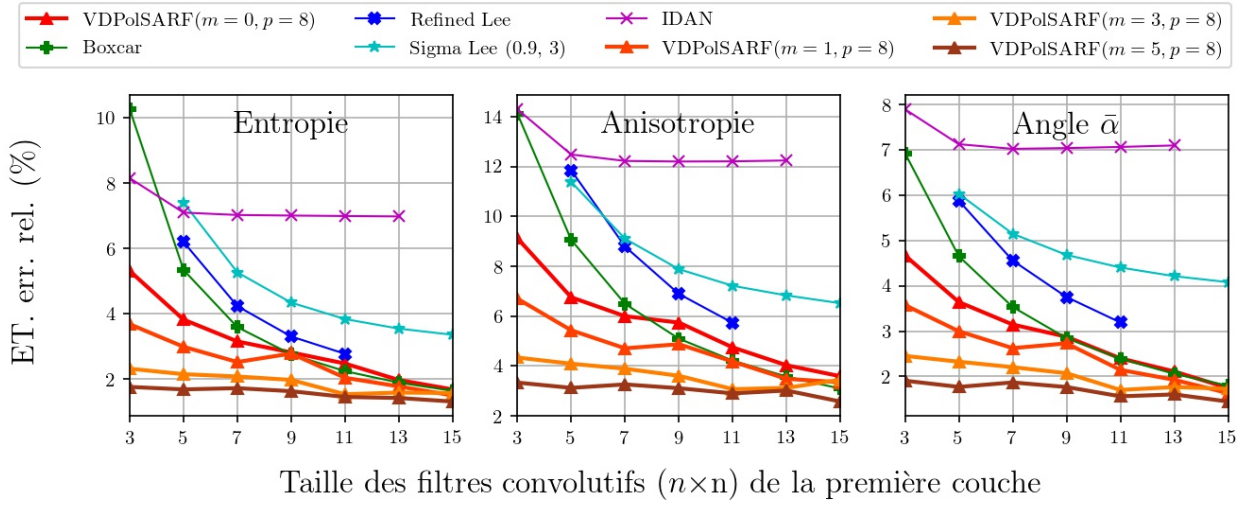
**Tab. 4.8.** Biais relatifs sur l'estimé de l'angle  $\bar{\alpha}$ . Comparaison entre le modèle **VDPLSARF** ( $m = 5, p = 8, n = 15$ ) et le filtre **Boxcar**  $15 \times 15$ .

En conclusion à cette section, les observations montrent que la précision des paramètres n'est pas garantie pour tous les modèles. En particulier pour les modèles courts ( $m = 0,1$ ) et les grandes fenêtres  $n$ . Cependant pour le modèle le plus long ( $m = 5, p = 8$ ) les résultats semblent stables pour l'anisotropie.



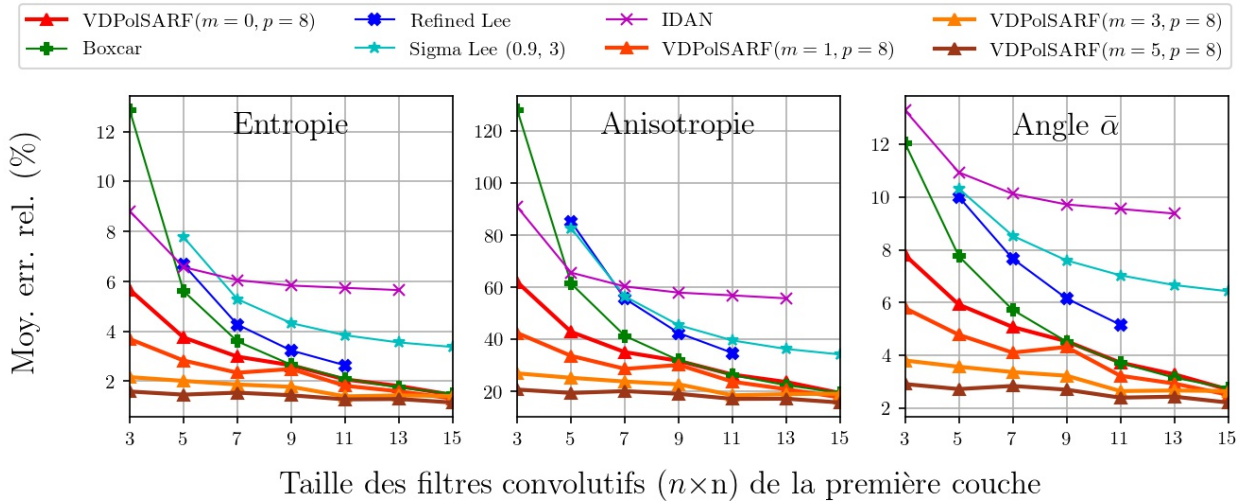
**Fig. 4.23. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



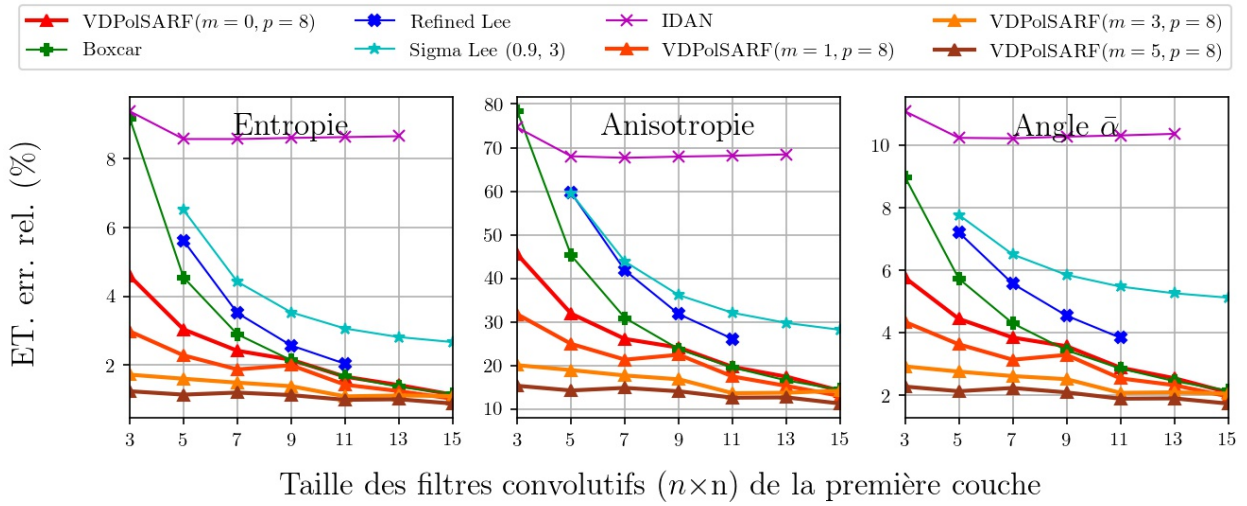
**Fig. 4.24. Diffuseurs double bond.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



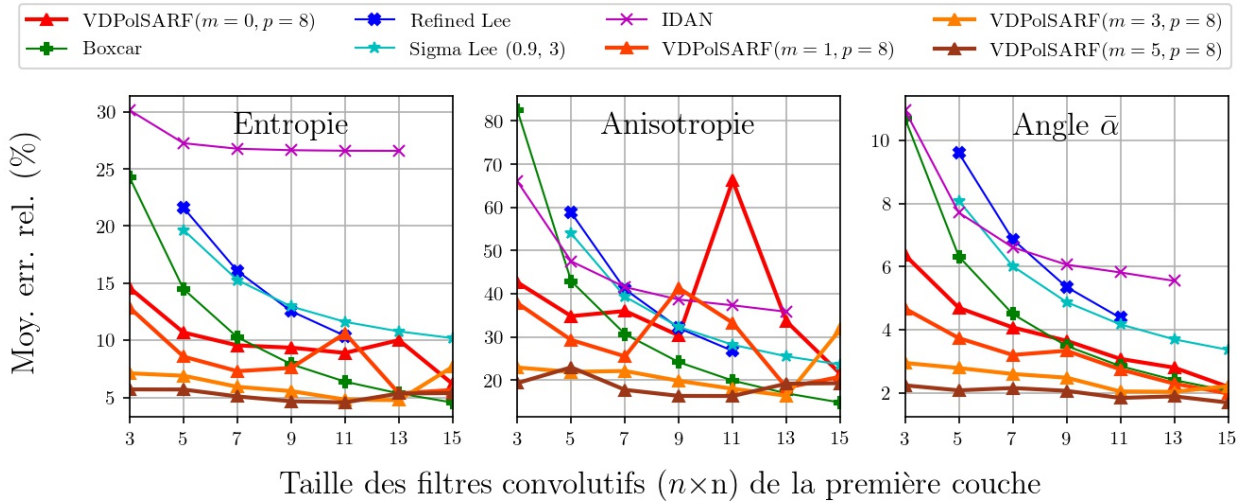
**Fig. 4.25. Diffuseurs volumétriques.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



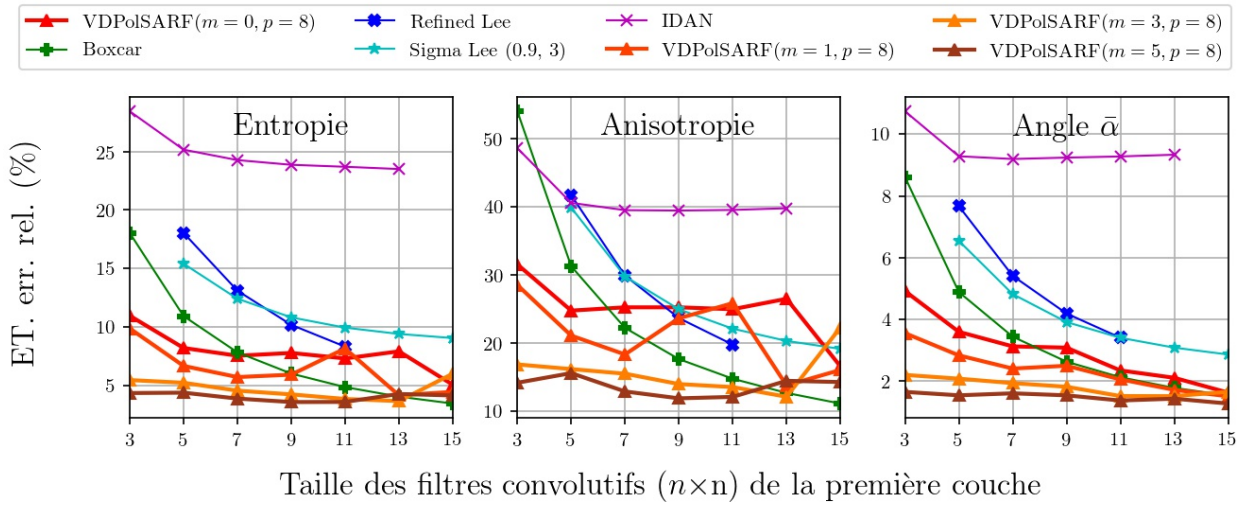
**Fig. 4.26. Diffuseurs volumétriques.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. 4.27. Diffuseurs surfaciques.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. 4.28. Diffuseurs surfaciques.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .



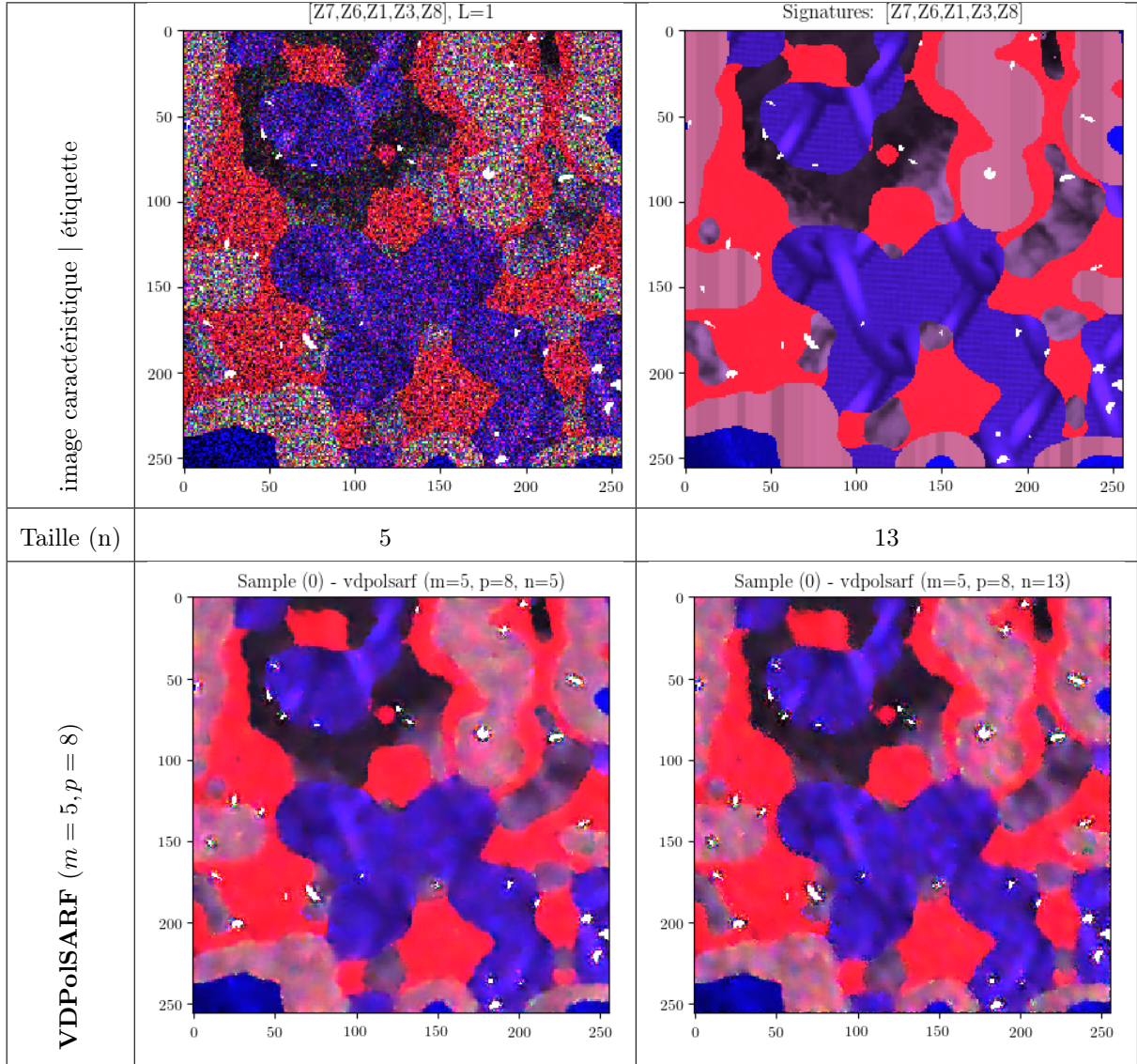
## 4.6. La qualité de la reconstruction des détails

Cette section présente les résultats des modèles appliqués sur les images synthétiques hétérogènes (multi-signatures) avec inclusion de cibles ponctuelles. Les cibles sont ajoutées aléatoirement aux images et leurs intensités peuvent varier entre 3db à 10 db selon le cas. Nous avons produits originalement 10 images selon la procédure exposée à la Section 3.1.4. Par contre notre analyse ne portera que sur deux images afin d'éviter la redondance des résultats.

Les modèles entraînés sur les données hétérogènes avec inclusion de cibles ponctuelles sont utilisés pour filtrer nos images. Les Figures 4.29 et 4.30 montrent les résultats obtenus pour la première image synthétique. La première des figures présente côte-à-côte sur la première ligne du tableau, l'image caractéristique avec le chatoiement une vue et l'image étiquette. La seconde ligne montre les résultats du filtrage réalisé par les filtres **VDPOLSAR** ( $m = 5, p = 8$ ) de taille  $n$  égale à 5 et 13. La seconde figure montre les résultats comparatifs des filtres **Boxcar** et **Sigma Lee**. La même disposition est employée pour la présentation du filtrage de notre second exemple (Fig. 4.31 et 4.32).

L'analyse qualitative par l'inspection visuelle des images montre que les images traitées par les filtres appris se comparent favorablement aux images filtrées par le **Boxcar**. Dans le cas de ce dernier on observe nettement l'effet de la perte de résolution spatiale et de l'étalement de la radiométrie des cibles sur un large voisinage. Pour le filtre **VDPoISARF**, les contours sont mieux définis et les cibles ponctuelles sont conservées presque entièrement. Dans le cas du filtre  $5 \times 5$ , les zones homogènes paraissent mieux filtrées par rapport au filtre **Sigma Lee**. Cependant nous observons que les filtres appris fabriquent des artéfacts autour des cibles. A la différence des filtres "standards" qui détectent la présence des cibles pour éviter de les filtrer, les filtres appris tentent plutôt de les restituer à partir du signal d'entrée: il n'y a pas de détection mais une reconstruction. Cette opération s'avère être une opération plus délicate.

Les Figures 4.33 et 4.34 présentent l'analyse quantitative sur chacune de nos deux images simulées. Premièrement, nous effectuons une comparaison entre l'image caractéristique filtrée et l'image étiquette afin d'évaluer la reconstruction général. Deuxièmement, nous comparons les résultats du filtre d'un Sobel ( $3 \times 3$ ) appliqué à l'image caractéristique filtrée et à l'image étiquette afin d'évaluer la reconstruction des contours.



**Fig. 4.29. Image simulée 0.** Comparaison entre les différents filtrages - Partie 1. ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

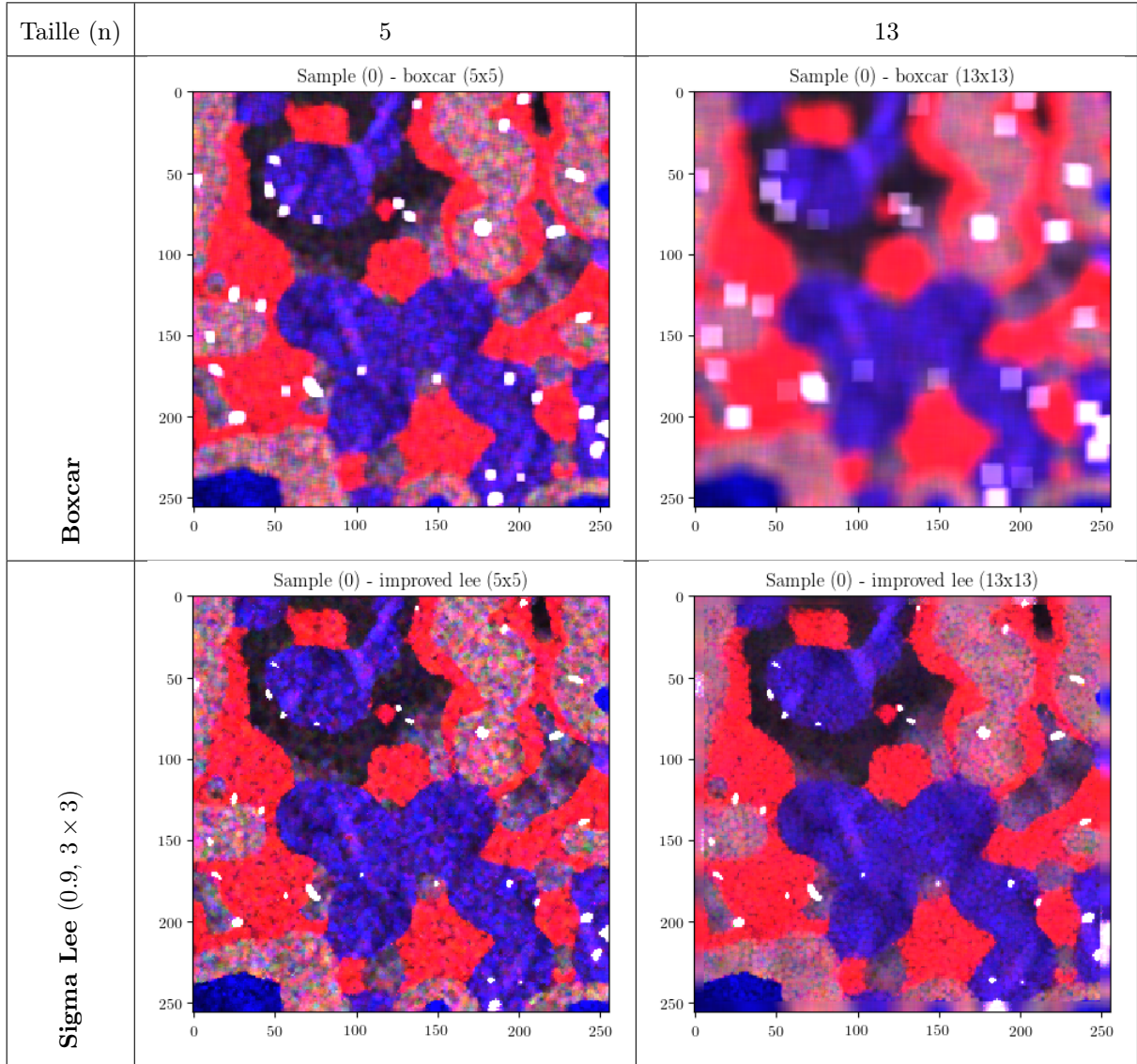
Nous utilisons les deux mesures suivantes pour l'évaluation de la reconstruction des détails:

- (1) **PSNR** (*Peak Signal to Noise Ratio*):
- (2) **SIMM** (*Structural Similarity Index Measure*):

Le **PSNR** est formulé comme suit:

$$\text{PSNR} = 10 \times \log_{10} \left( \frac{d^2}{\text{EQM}} \right) \quad (4.6.1)$$

où **EQM** est l'erreur quadratique moyenne:



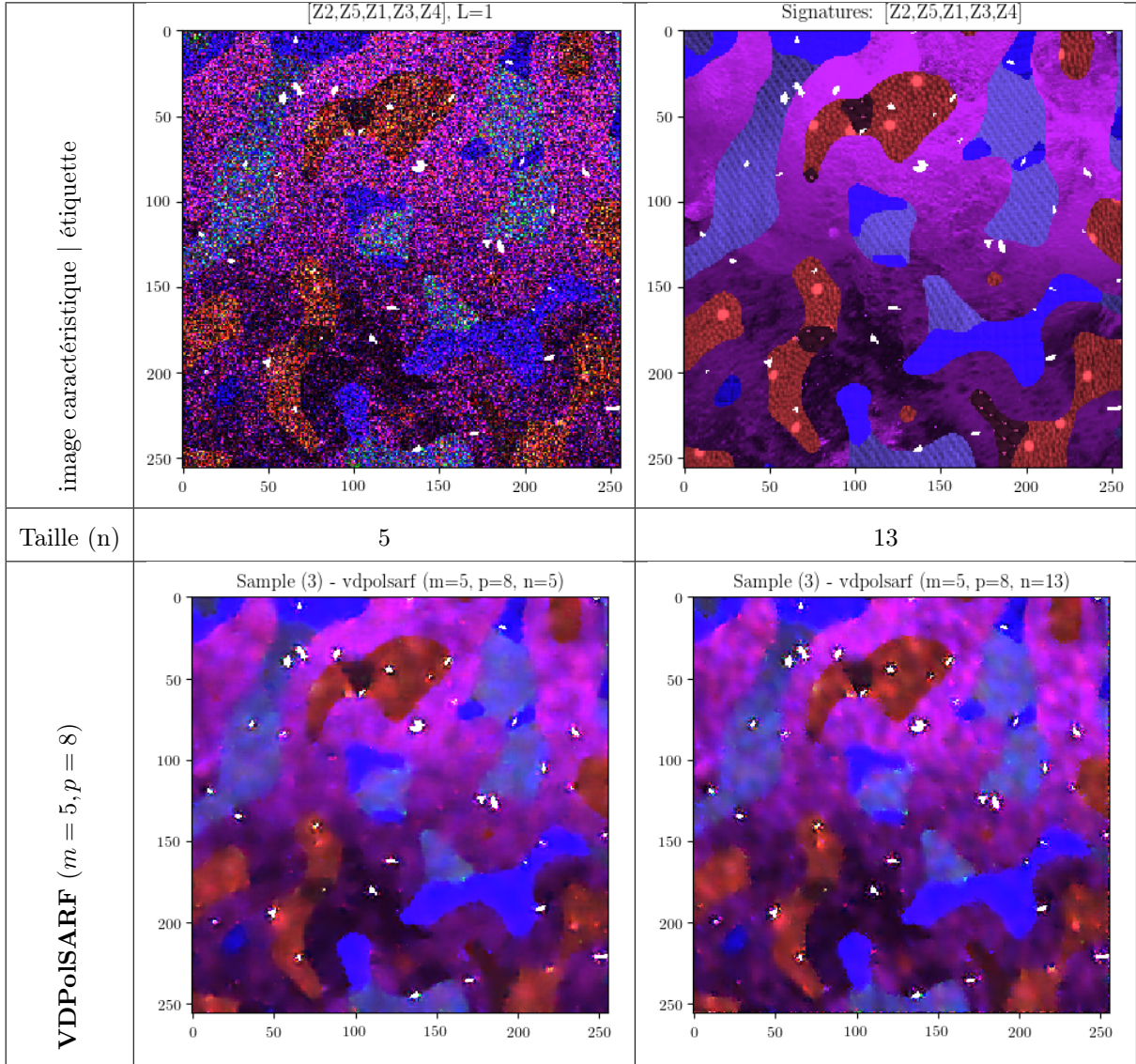
**Fig. 4.30. Image simulée 0.** Comparaison entre les différents filtres - Partie 2. (Improved Lee = Sigma Lee) ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

$$\mathbf{EQM} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left( I_0(i,j) - I_r(i,j) \right)^2 \quad (4.6.2)$$

et  $I_0$  correspond au images caractéristiques avec chatoiement et  $I_r$  aux images étiquettes. Nous avons fixé la valeur  $d = 3dB$  et les images ont été seuillées en conséquence.

Le **SSIM** est formulé comme suit:

$$SSIM = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (4.6.3)$$

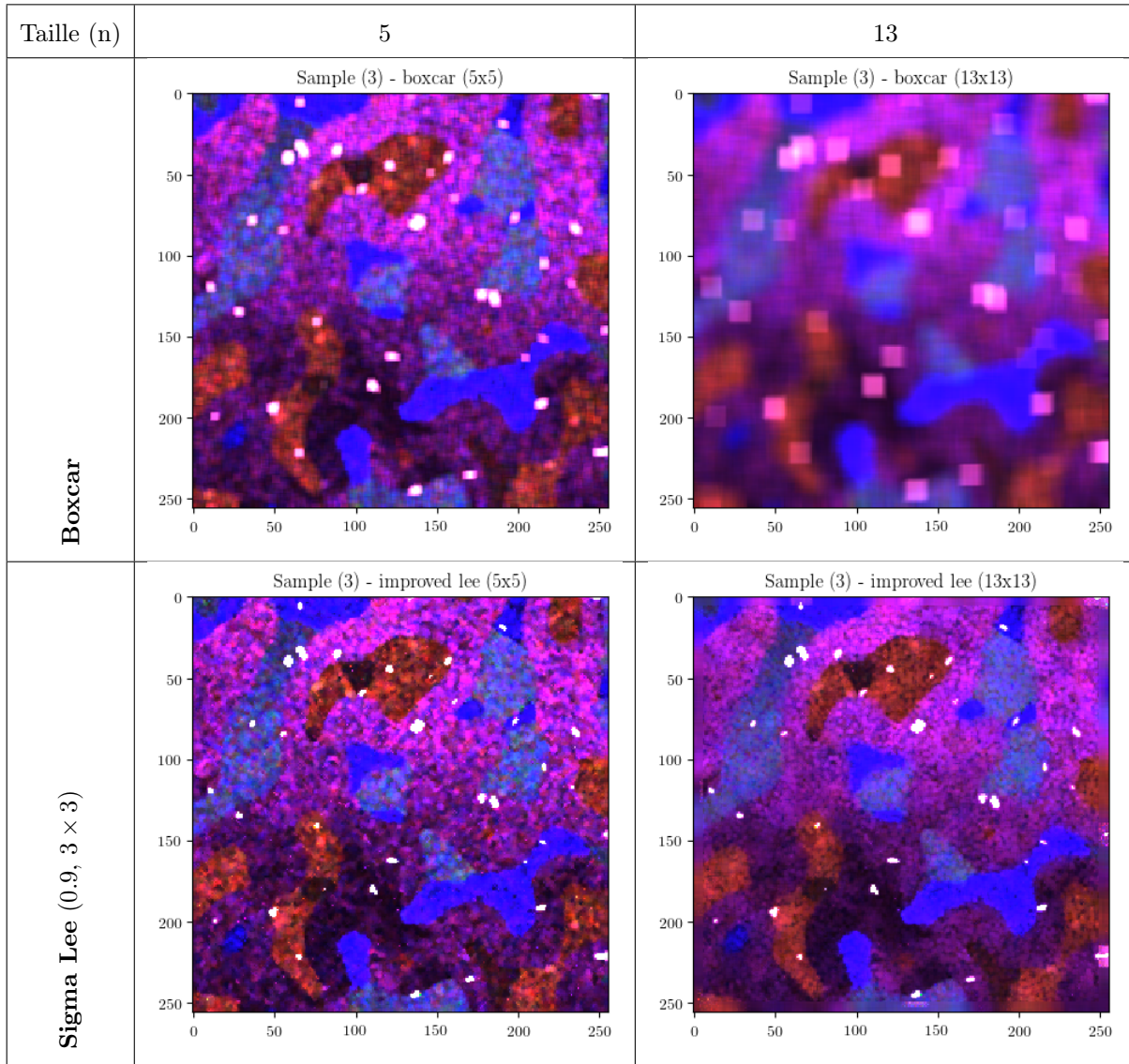


**Fig. 4.31. Image simulée 3.** Comparaison entre les différents filtrages - Partie 1. ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

où  $\mu_x$  est la moyenne de  $I_0$  et  $\mu_y$  est la moyenne de  $I_r$ ,  $\sigma_x^2$  est la variance de  $I_0$  et  $\sigma_y^2$  est la variance de  $I_r$ ,  $C_1$  et  $C_2$  sont deux constantes.

Les Figures 4.33 et 4.34 présentent les résultats suivants:

- (1) La comparaison des **PSNR** pour chacun des filtres en fonction de la taille  $n$  de l'entrée du filtre par rapport à l'image étiquette;
- (2) La comparaison des **SSIM** pour chacun des filtres en fonction de la taille  $n$  de l'entrée du filtre par rapport à l'image étiquette;

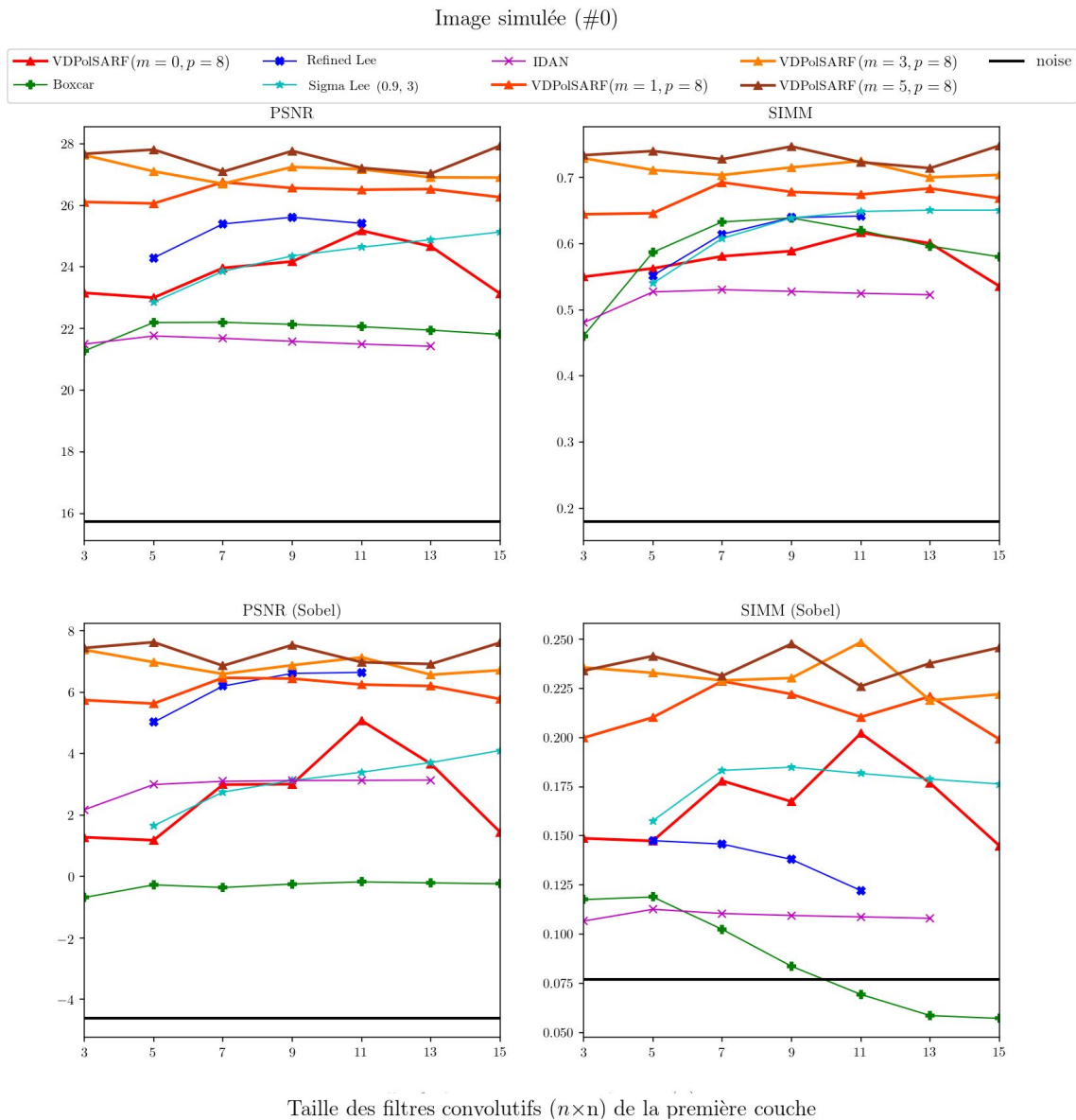


**Fig. 4.32. Image simulée 3.** Comparaison entre les différents filtres - Partie 2. (Improved Lee = Sigma Lee) ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

- (3) La comparaison des **PSNR** sur les images de Sobel pour chacun des filtres en fonction de la taille  $n$  de l'entrée du filtre par rapport à l'image de Sobel étiquette;
- (4) La comparaison des **SSIM** sur les images de Sobel pour chacun des filtres en fonction de la taille  $n$  de l'entrée du filtre par rapport à l'image de Sobel étiquette;

Nous remarquons que dans le cas du **PSNR**, le **Boxcar** et le filtre **IDAN** donnent les pires résultats. Ce qui est prévisible pour le **Boxcar** mais étonnant pour le filtre **IDAN** dont la performance est encore moindre. Le filtre **Refined LEE** performe légèrement mieux que les filtres **Sigma Lee** et **VDPolSARF** le plus court. Ce sont les filtres appris les plus

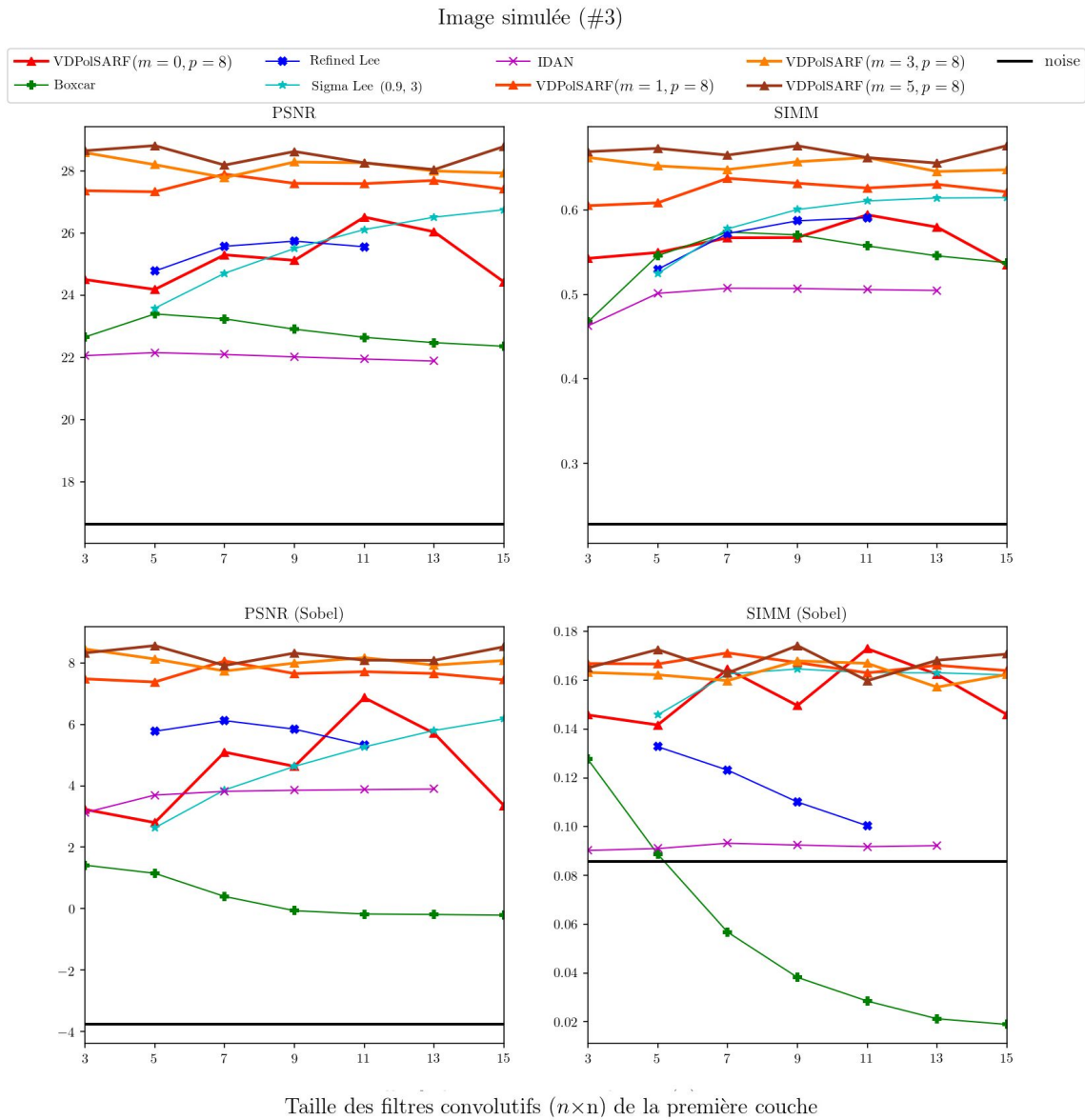
profonds qui dominent au niveau de la performance. Le filtre **VDPoISARF** ( $m = 5, p = 8$ ) donne les meilleurs résultats et ceci indépendamment de la taille de la fenêtre.



**Fig. 4.33.** Image simulée 0). Moyenne du **PSNR** et du **SSIM** sur l'image du **span** et sur l'image de Sobel du **span**. Les résultats sont présentés en fonction de la taille  $n$  de l'entrée des filtres.

Dans le cas du **PSNR** sur les images de Sobel, nous observons que les performances sont faibles avec des scores inférieurs à 8. Les performances se séparent en trois groupes: le filtre **Boxcar**, les filtres "standards" avec le filtre **VDPoISARF** le plus court et les filtres appris

plus profonds **VDPolSARF** ( $m = 1,3,5$ ). Par contre pour l'image #0 où le filtre **Refined LEE** rejoint les performances du filtre **VDPolSARF** ( $m = 3$ ).



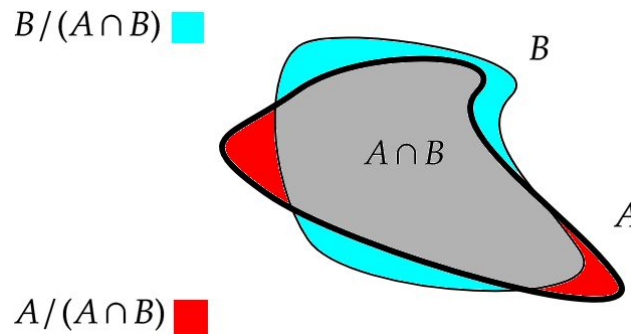
**Fig. 4.34.** Image simulée 3. Moyenne du **PSNR** et du **SSIM** sur l'image du **span** et sur l'image de Sobel du **span**. Les résultats sont présentés en fonction de la taille  $n$  des filtres.

Dans le cas des résultats des **SSIM** sur les images filtrées et des **SSIM** sur les images filtrées de Sobel, nos observations sont semblables à celles émises pour les **PSNR**. De manière systématique les filtres appris les plus profonds **VDPolSARF** obtiennent les meilleures performances si nous les comparons aux filtres "standards" et au filtre **Boxcar**.

## 4.7. La qualité de la restitution des cibles ponctuelles

Cette section présente quelques résultats montrant la capacité des algorithmes à préserver les cibles ponctuelles. Nous avons comparé la conservation de la superficie des cibles entre les différentes approches. La Figure 4.35 schématise les trois mesures que nous avons réalisées. Si nous considérons la surface  $A$  comme la cible étiquette et forme  $B$  comme la cible restituée, nous pouvons calculer les mesures suivantes:

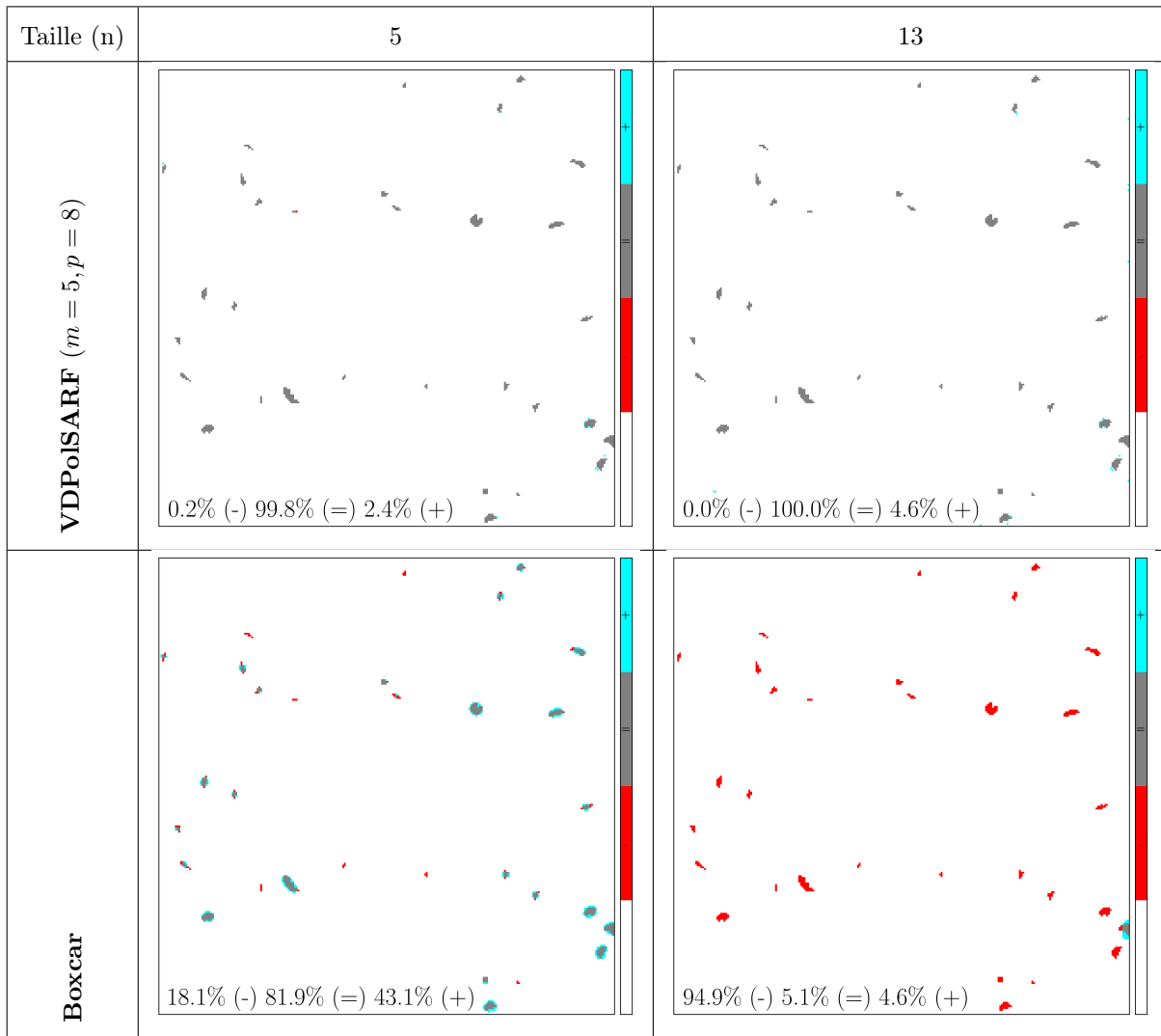
- (1) La superficie préservée:  $\text{superficie}(A \cap B) / \text{superficie}(A)$
- (2) La superficie additionnée:  $\text{superficie}(B/(A \cap B)) / \text{superficie}(A)$
- (3) La superficie supprimée:  $\text{superficie}(A/(A \cap B)) / \text{superficie}(A)$



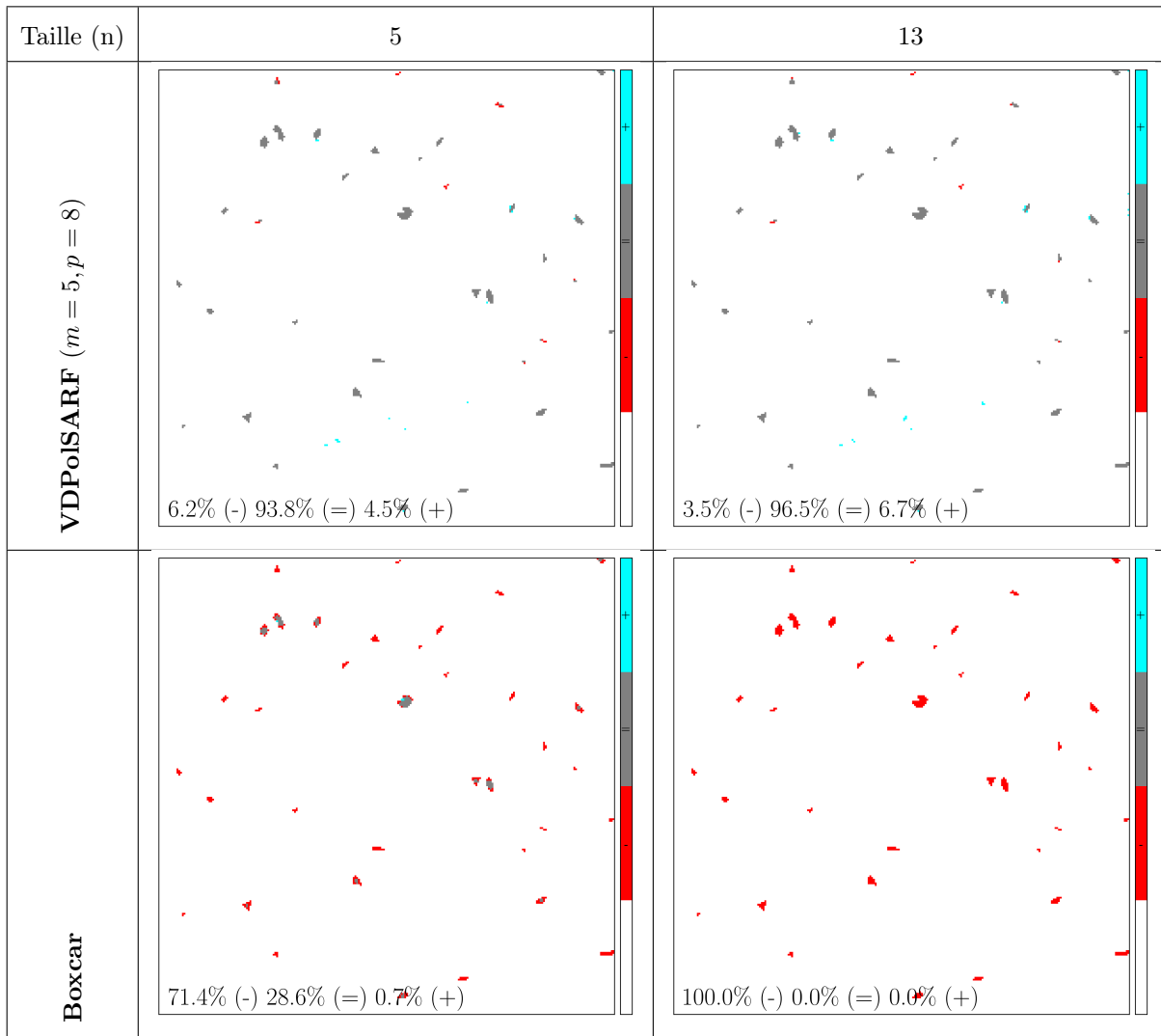
**Fig. 4.35.** Schéma représentant les changements possibles de la géométrie de la cible.

Les Figures 4.36 et 4.37 montrent l'analyse sur nos deux images synthétiques. Nous comparons les résultats obtenus avec le modèle **VDPolSARF** ( $m = 5, p = 8$ ) et le filtre **Boxcar** pour deux tailles de fenêtre  $n = 5, 13$ . Les images ont été seuillées à 3 dB afin de ne conserver que les cibles ponctuelles. Nous observons les résultats prévisibles du **Boxcar** qui perd graduellement les cibles en fonction de la taille du filtre. Dans le cas du filtre  $13 \times 13$  les cibles sont complètement diluées. Dans le cas des filtres appris, les résultats varient entre 99.8 % et 93.8 % de préservation des cibles pour la taille de fenêtre de  $5 \times 5$  et de 100 % à 96.5 % pour une taille de fenêtre de  $13 \times 13$ .





**Fig. 4.36. Image simulée 0. Résultats sur la préservation des cibles ponctuelles**



**Fig. 4.37. Image simulée 3. Résultats sur la préservation des cibles ponctuelles**

## 4.8. L'effet de la complexité de l'apprentissage sur l'estimation des paramètres polarimétriques

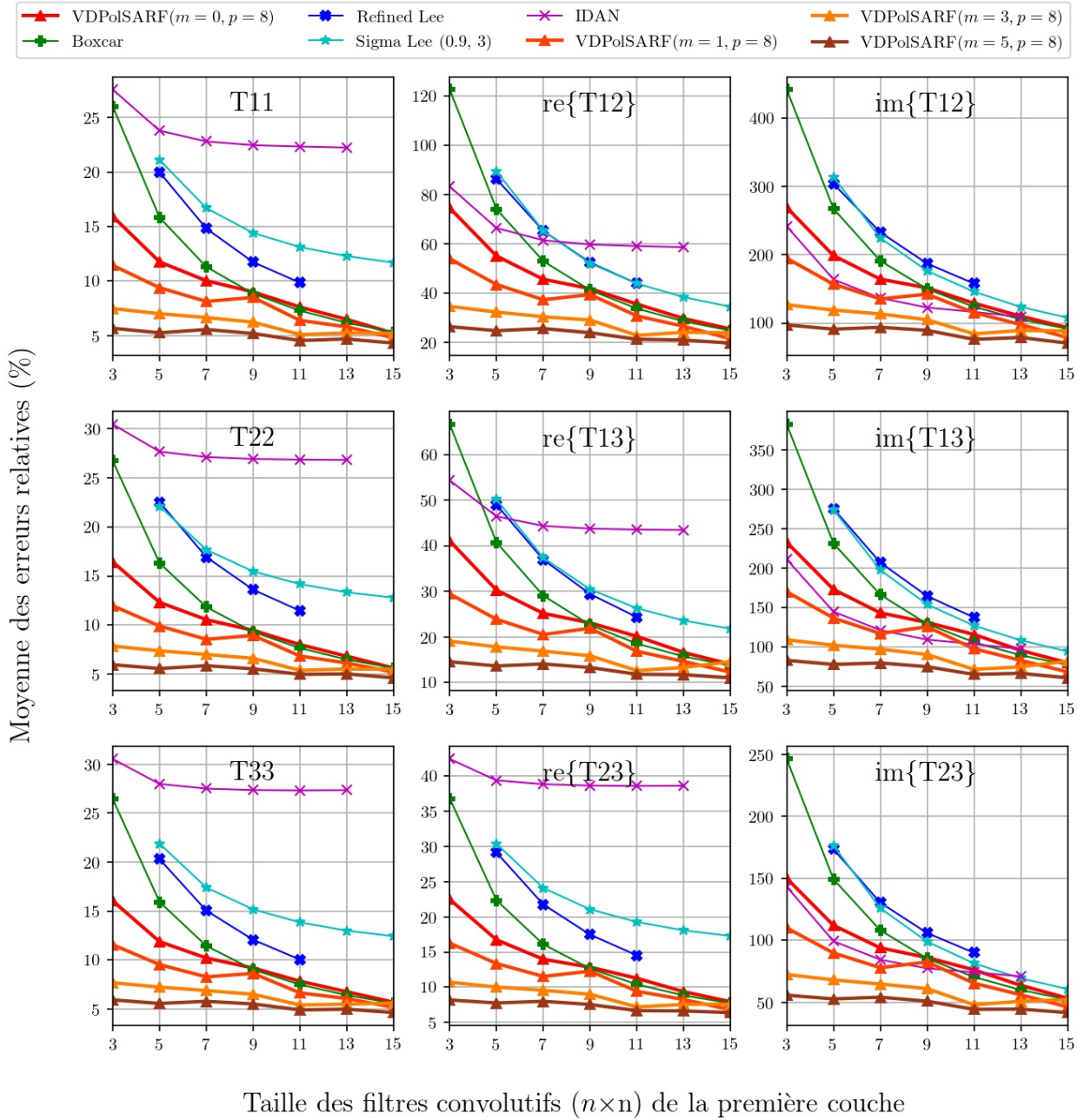
Cette dernière section sur les données synthétiques met en relief l'effet du type d'apprentissage sur les résultats des estimées des paramètres polarimétriques sur des surfaces homogènes. Nous rappelons que les apprentissages ont été faits sous les mêmes conditions (Sec. 3.3). La seule variable qui a changé est l'ensemble de données utilisé pour les apprentissages. Afin d'observer l'influence sur chacun des paramètres estimés, nous avons refait les analyses sur les régions homogènes mais en utilisant les modèles entraînés sur les données hétérogènes et les modèles entraînés sur les données hétérogènes avec inclusion de cibles ponctuelles. Nous ne présentons que les résultats sur le diffuseur *double bond* car les résultats sont similaires pour les autres types de diffuseurs. Les Figures 4.38, 4.39, 4.40 présentent le biais estimé sur les termes de la matrice de cohérence. Les Figures 4.41, 4.42, 4.43 présentent les valeurs du **NEV** sur les termes diagonaux de la matrice de cohérence et les Figures 4.44, 4.45, 4.46 présentent les biais calculés sur les paramètres de la décomposition de  $H/A/\bar{\alpha}$ .

Nous remarquons la tendance générale de la perte de performance en fonction de la complexité des données d'apprentissage. Le passage entre l'apprentissage à partir des données homogènes et les données hétérogènes pour l'estimation des termes de la matrice de cohérence et des paramètres de la décomposition de  $H/\bar{\alpha}$  montrent une légère perte de performance de l'ordre de 2 %. Dans le cas du calcul du **NEV** on observe une perte de plus de 35 % pour le modèle ( $m = 5, p = 8, n = 3$ ) et de plus de 50 % pour le modèle ( $m = 5, p = 8, n = 15$ ). Cependant les résultats demeurent supérieurs aux autres filtres "standards" et au filtre **Boxcar** pour les modèles **VDPolSARF** ( $m = 3,5, p = 8$  et  $n < 9$ ).

L'introduction de cibles ponctuelles affecte de façon notable les résultats des modèles courts ( $m = 0, m = 1,3$ ) et nuisent plus aux modèles avec une large fenêtre d'entrée  $n > 9$ . On remarque que les modèles profonds ( $m = 5, p = 8, n < 9$ ) réussissent malgré tout à maintenir de bons estimés pour les termes de la matrice de cohérence et les paramètres  $H/A/\bar{\alpha}$ . Les valeurs du **NEV** demeurent supérieures pour des tailles de fenêtre inférieures ou égales à 9 si nous les comparons aux filtres "standards".

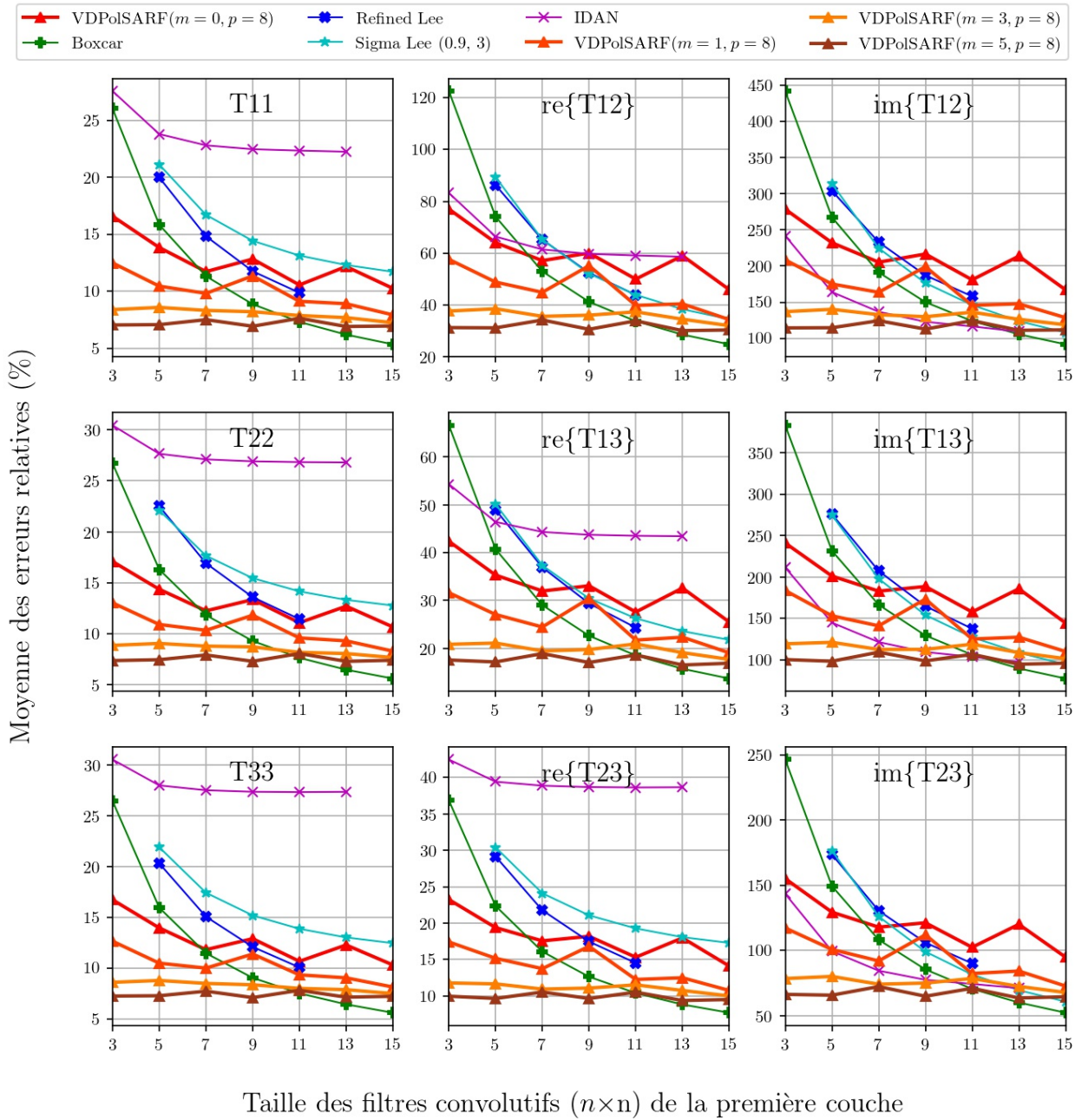
La Figure 4.47 montre un exemple visuel qui met en relief l'influence des données d'apprentissage sur la capacité des modèles à filtrer les régions homogènes.

### Double Réflexion (Diffuseurs double bond)



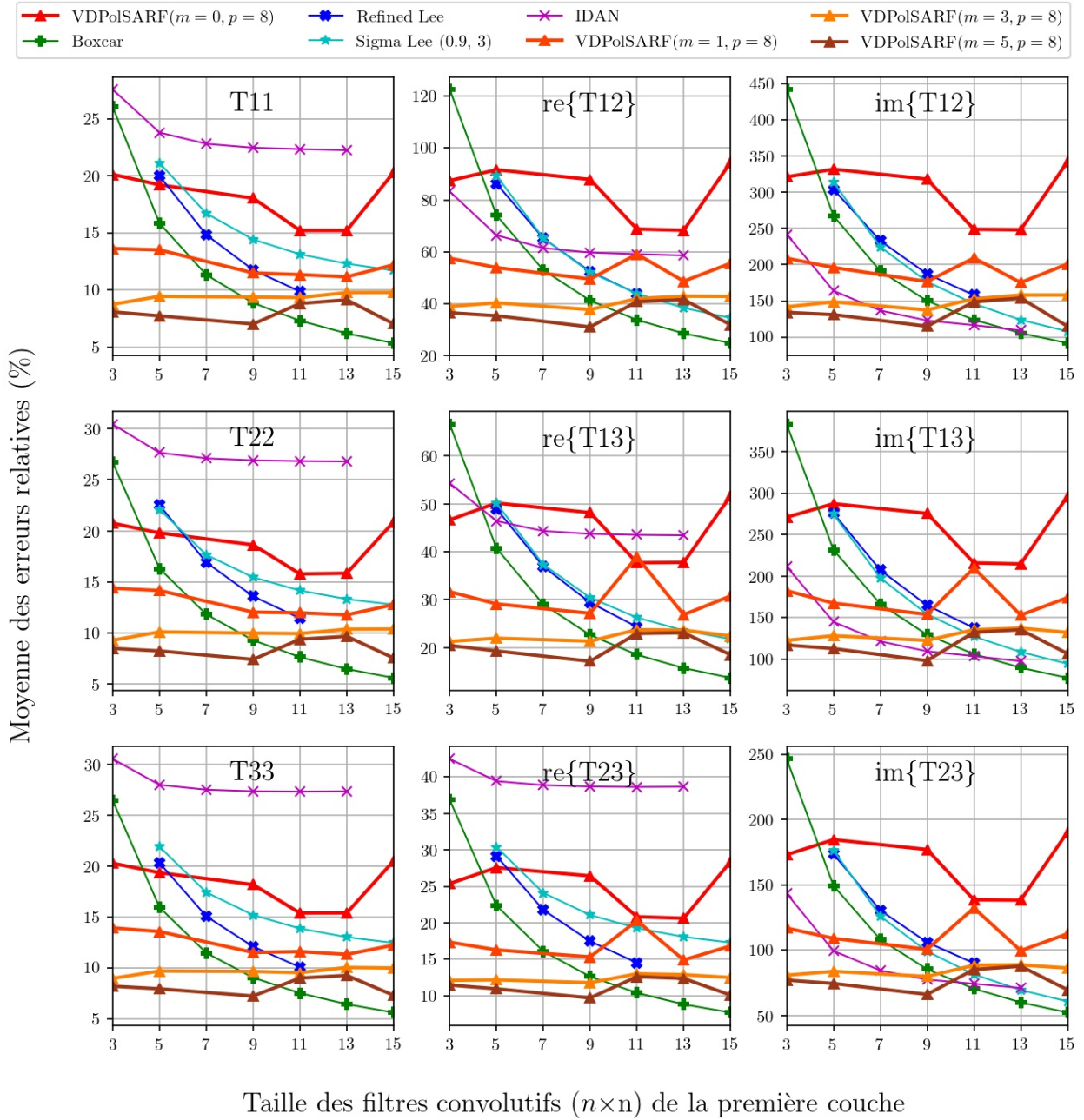
**Fig. 4.38. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données homogènes).

### Double Réflexion (Diffuseurs double bond)



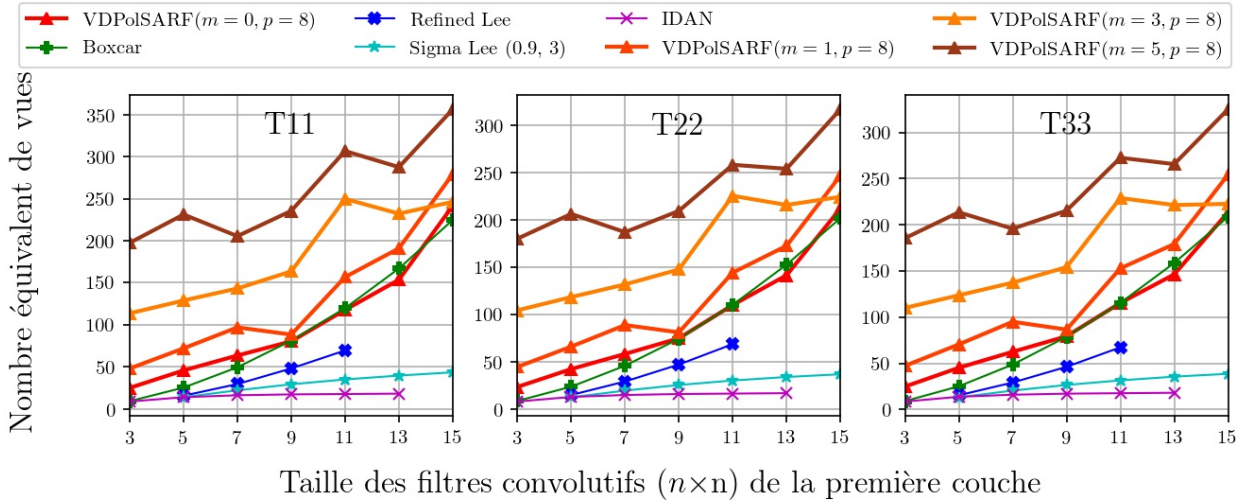
**Fig. 4.39. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes).

### Double Réflexion (Diffuseurs double bond)



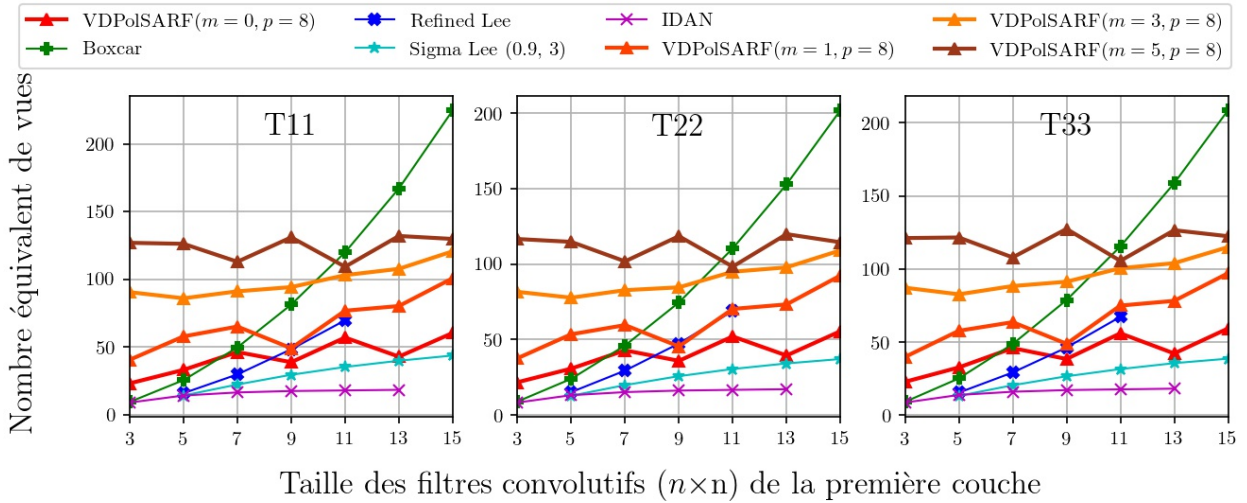
**Fig. 4.40. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles)

### Double Réflexion (Diffuseurs double bond)



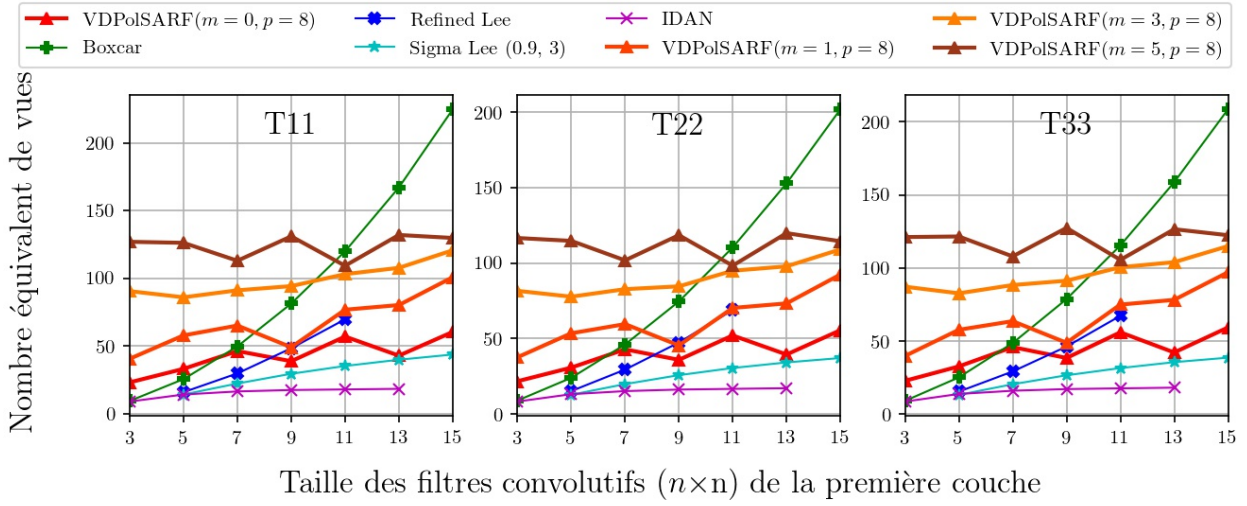
**Fig. 4.41. Diffuseurs double bond.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données homogènes).

### Double Réflexion (Diffuseurs double bond)



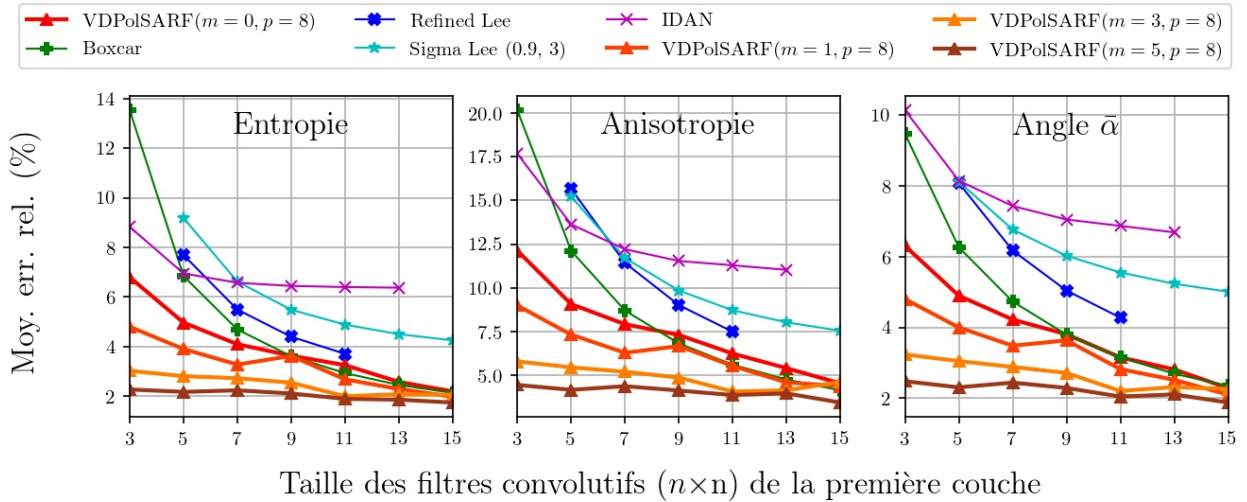
**Fig. 4.42. Diffuseurs double bond.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes).

### Double Réflexion (Diffuseurs double bond)



**Fig. 4.43. Diffuseurs double bond.** Nombre équivalent de vues calculé sur les termes diagonaux de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles).

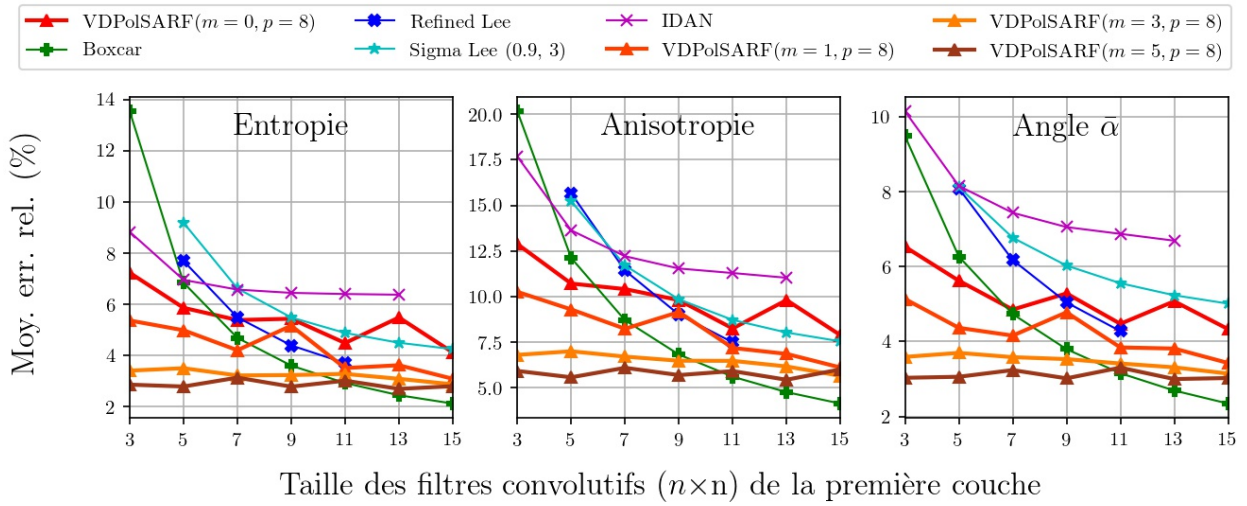
### Double Réflexion (Diffuseurs double bond)



**Fig. 4.44. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données homogènes).

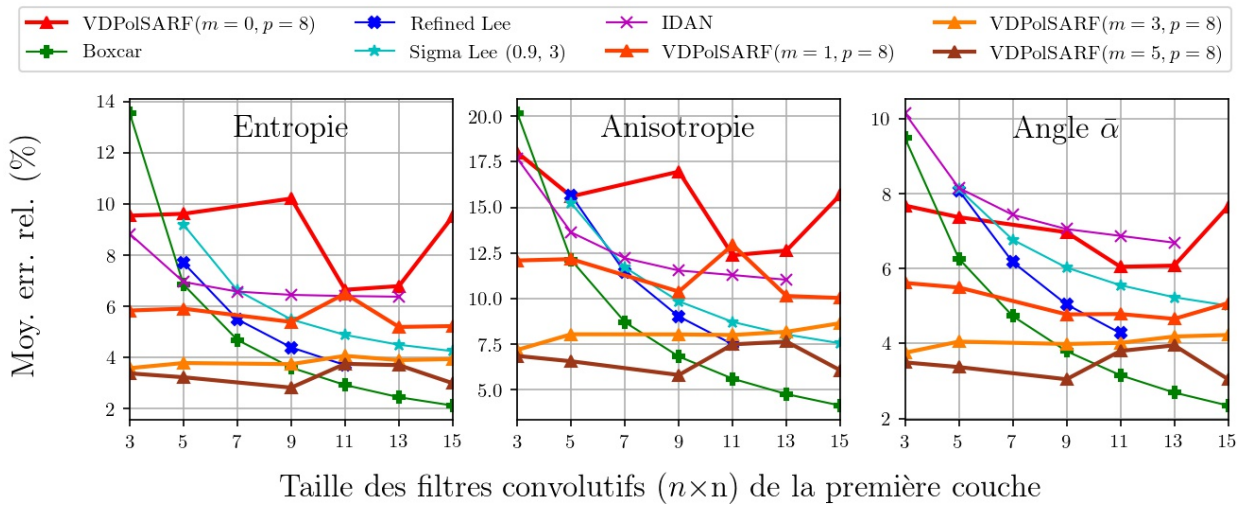


### Double Réflexion (Diffuseurs double bond)

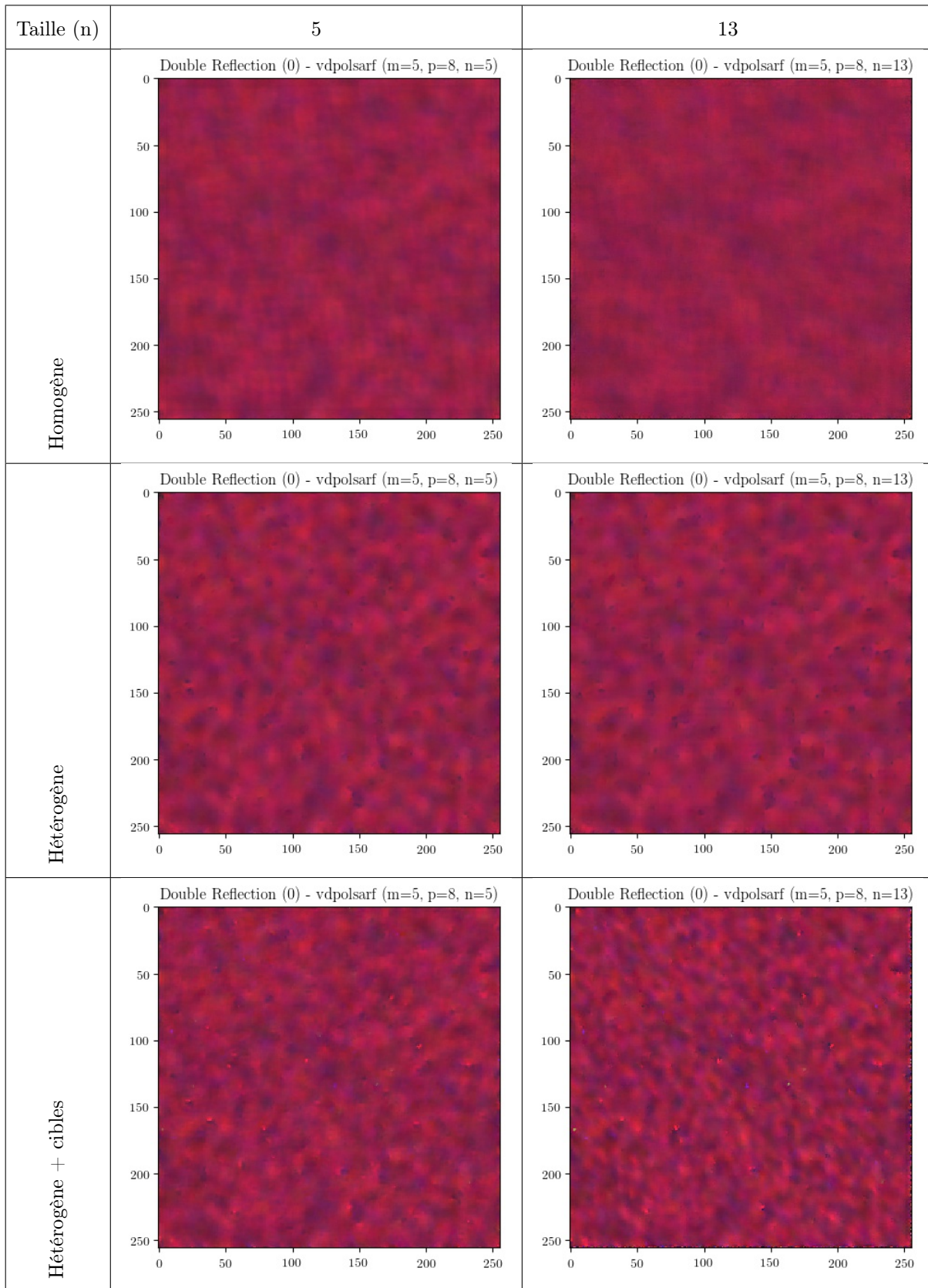


**Fig. 4.45. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes).

### Double Réflexion (Diffuseurs double bond)



**Fig. 4.46. Diffuseurs double bond.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$  (apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles).

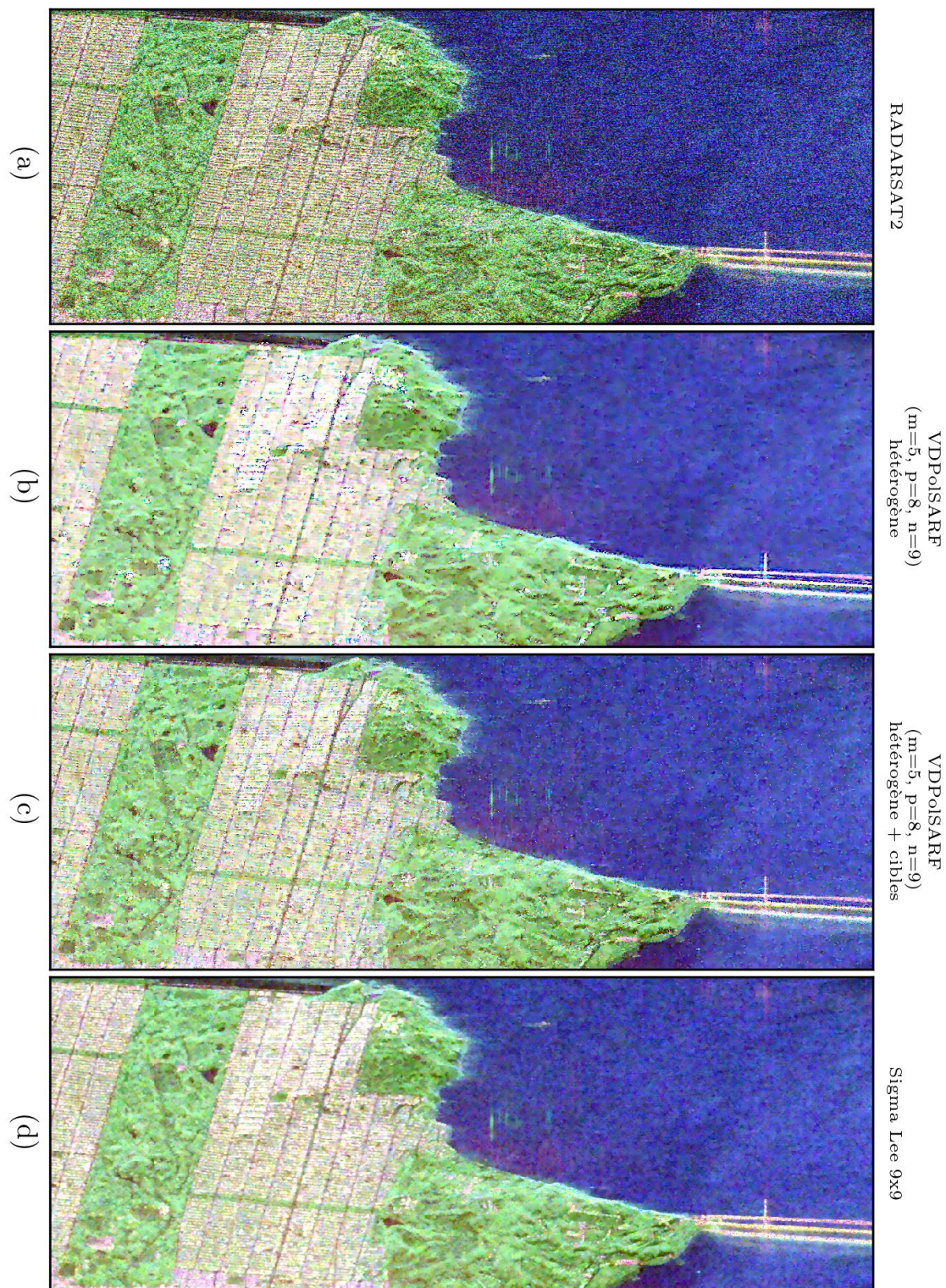


**Fig. 4.47. Diffuseurs double bond.** Comparaison entre les différents apprentissages pour le modèle **VDPolSARF** ( $m = 5, p = 8$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ )

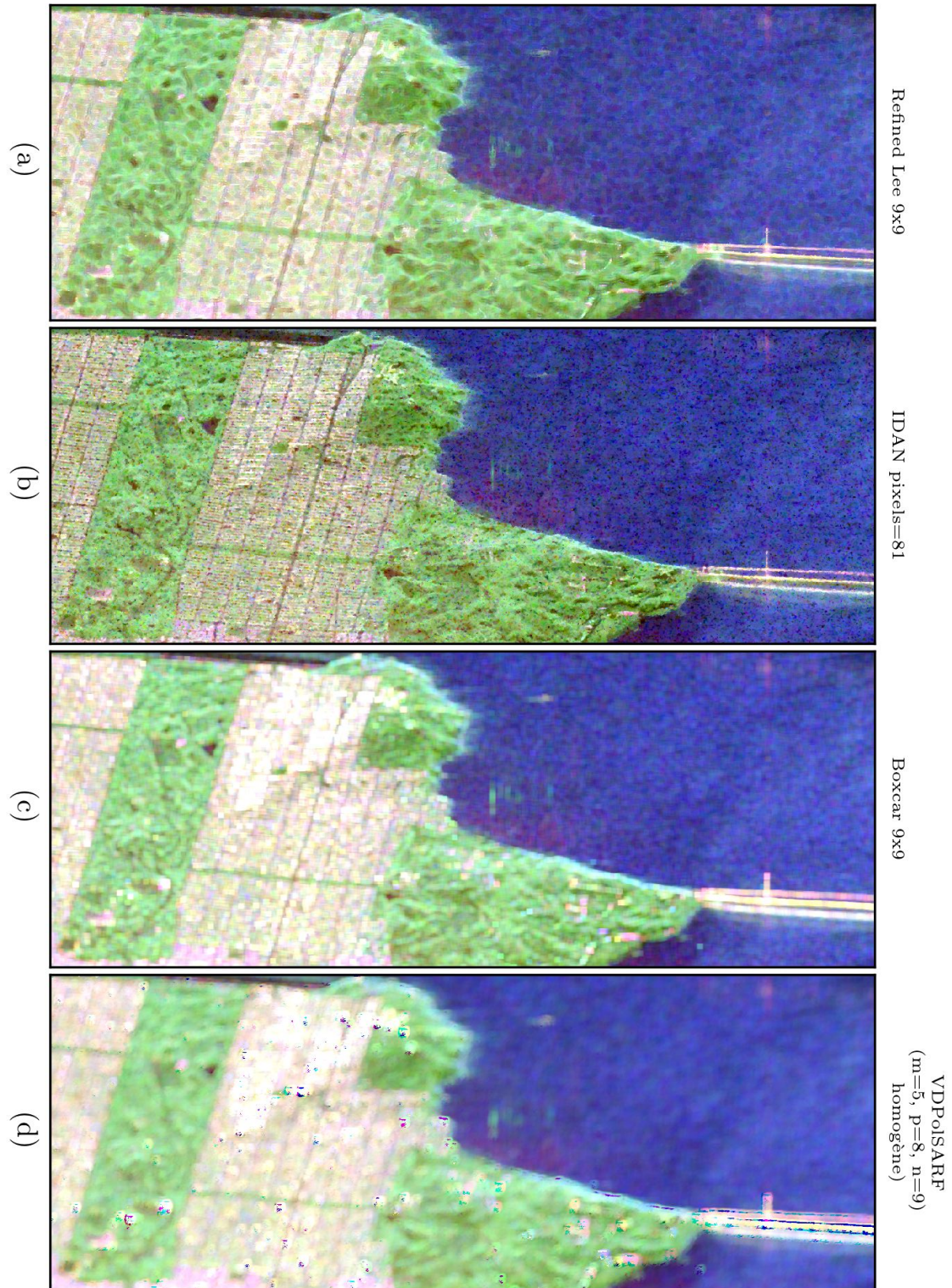
## 4.9. Évaluation qualitative sur les images polarimétriques réelles

Cette section présente l'application des modèles entraînés sur nos images polarimétriques test. Ces images ont été décrites à la section 3.2. Les Figures 4.48 et 4.49, 4.50 et 4.51, 4.52 et 4.53 illustrent les résultats sur l'image en QuadPol RADARSAT2, ALOS2 et GAF3. Nous y comparons le modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données hétérogènes et le même modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données homogènes avec inclusion de cibles ponctuelles par rapport aux différents filtres polarimétriques standards. A cette échelle les résultats semblent assez bien filtrés par nos deux modèles **VDPolSARF** (Figs. 4.48, 4.50 et 4.52 (b) et (c) ), ils sont comparables aux résultats du filtre **Sigma Lee** (Fig. 4.48 (d)). Nous donnons aussi les résultats du modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données homogènes (Figs. 4.49, 4.51 et 4.53 (d)) à titre de comparaison avec le filtre Boxcar 9x9 (Figs. 4.49, 4.51 et 4.53 (c)). Nous pouvons dès lors remarquer que ce modèle filtre beaucoup plus les zones homogènes que tous les autres modèles entraînés ainsi que les autres filtres standards et le filtre **Boxcar**. Par contre il crée de nombreux artefacts autour des cibles ponctuelles. Ce qui est normal, puisque l'apprentissage a été effectué sur des images complètement homogènes.

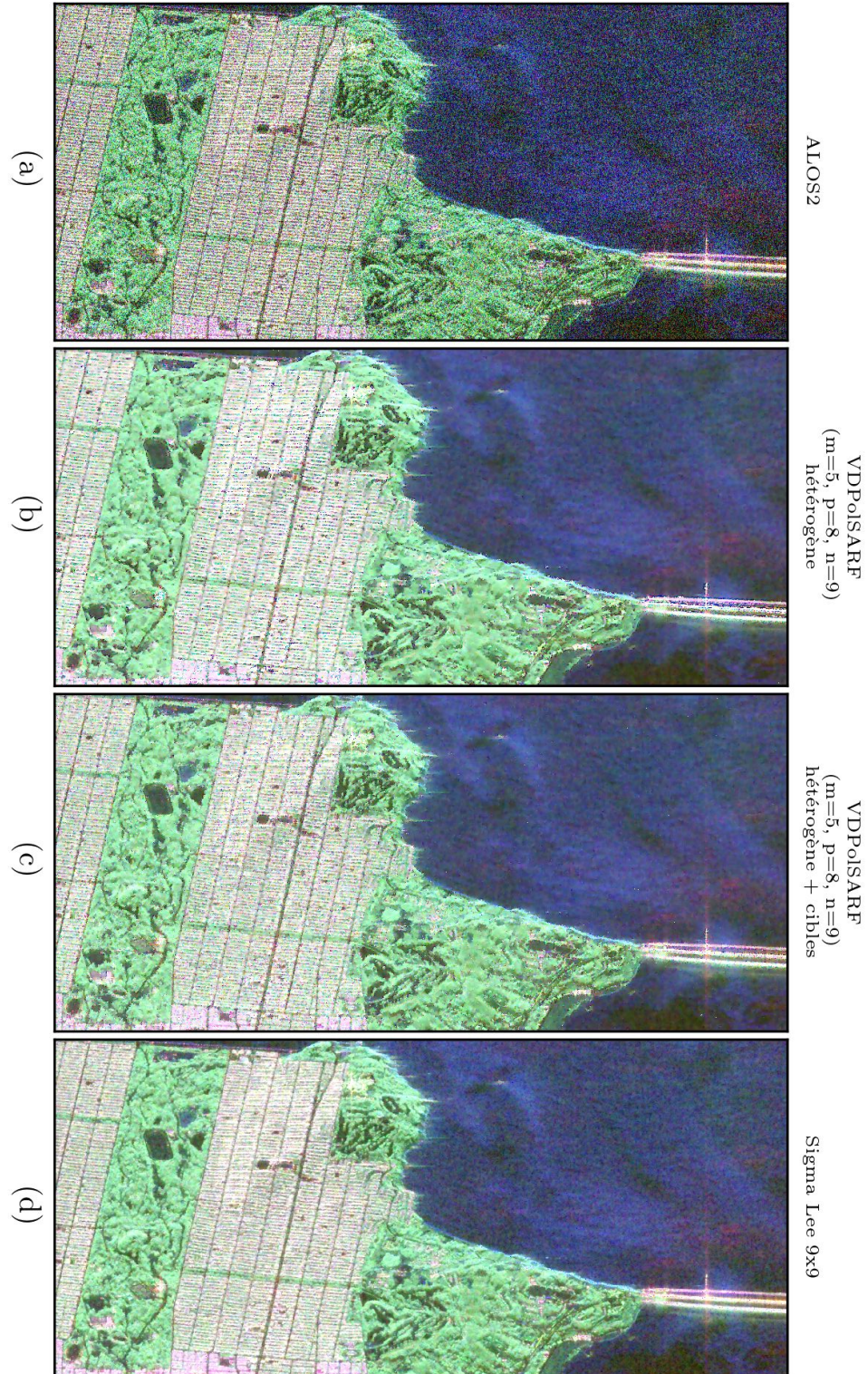
Maintenant, lorsque nous analysons plus attentivement les résultats à plus grande échelle (Figs. 4.54, 4.55 et 4.56), nous pouvons remarquer que chacun de nos deux modèles produit des résultats quelques peu différents. Premièrement le modèle entraîné sur les données hétérogènes semble meilleur pour filtrer les zones homogènes, par contre les cibles ponctuelles sont dégradées quelque peu par des artefacts sur le pourtour des cibles. On le remarque particulièrement sur l'image ALOS2 (Fig. 4.55 (b)). Dans le cas de notre second modèle, il préserve assez bien les cibles mais il semble ne pas suffisamment filtrer les régions homogènes. On remarque une forme de pixelisation qui est particulièrement visible sur l'image de RADARSAT2 (4.54 (c)) et GAF3 (4.56 (c)). Sur le plan de la préservation des détails, nos modèles entraînés semblent donner des résultats aussi bons que le filtre **Sigma Lee**. Il nous apparaît que le modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données hétérogènes donne de meilleurs résultats sur les images ALOS2 pour la préservation des arêtes que les autres filtres.



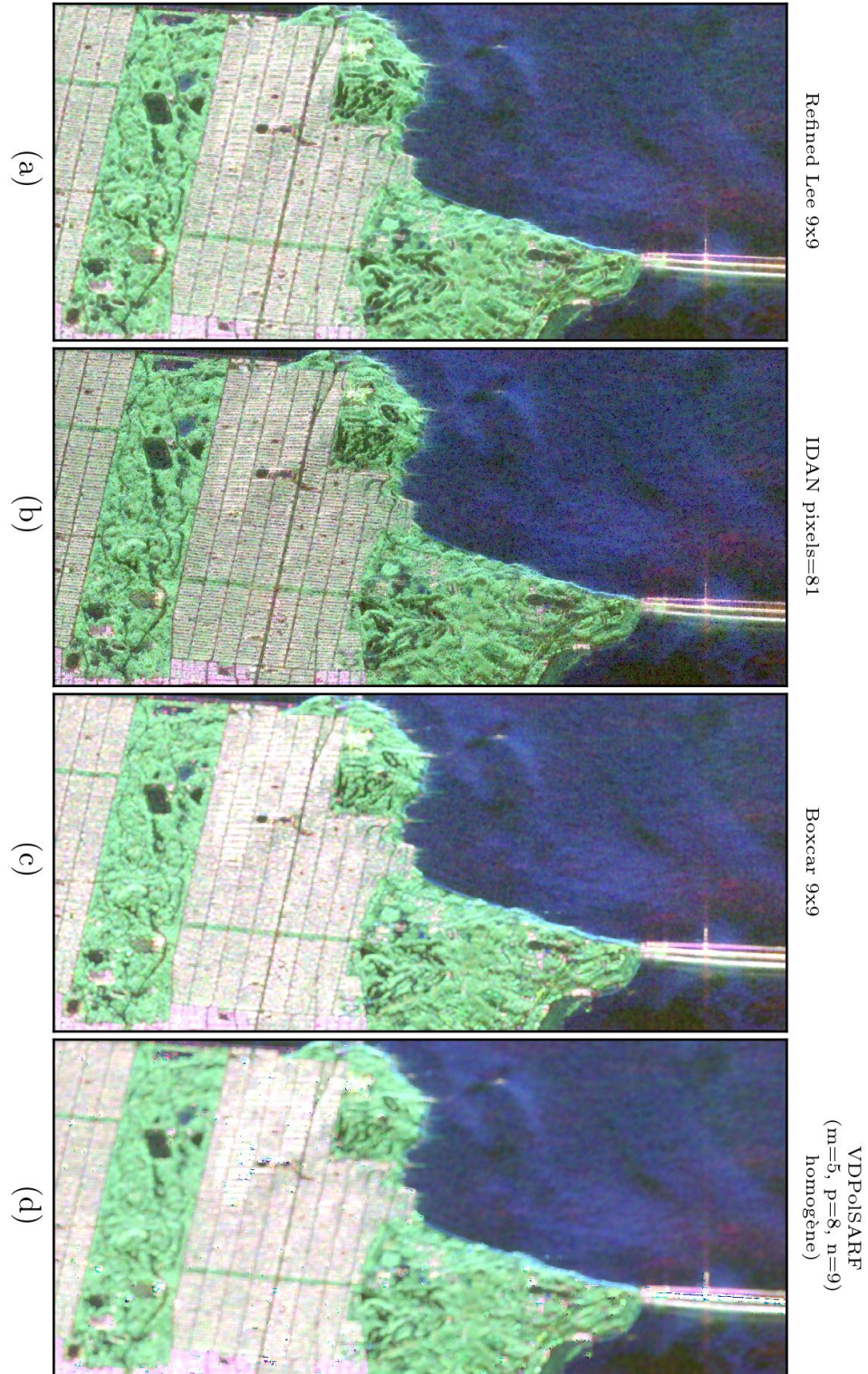
**Fig. 4.48.** Comparaison entre les filtrages de l'image RADARSAT2 (1): (a) image originale, (b) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre **Sigma Lee** ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ).



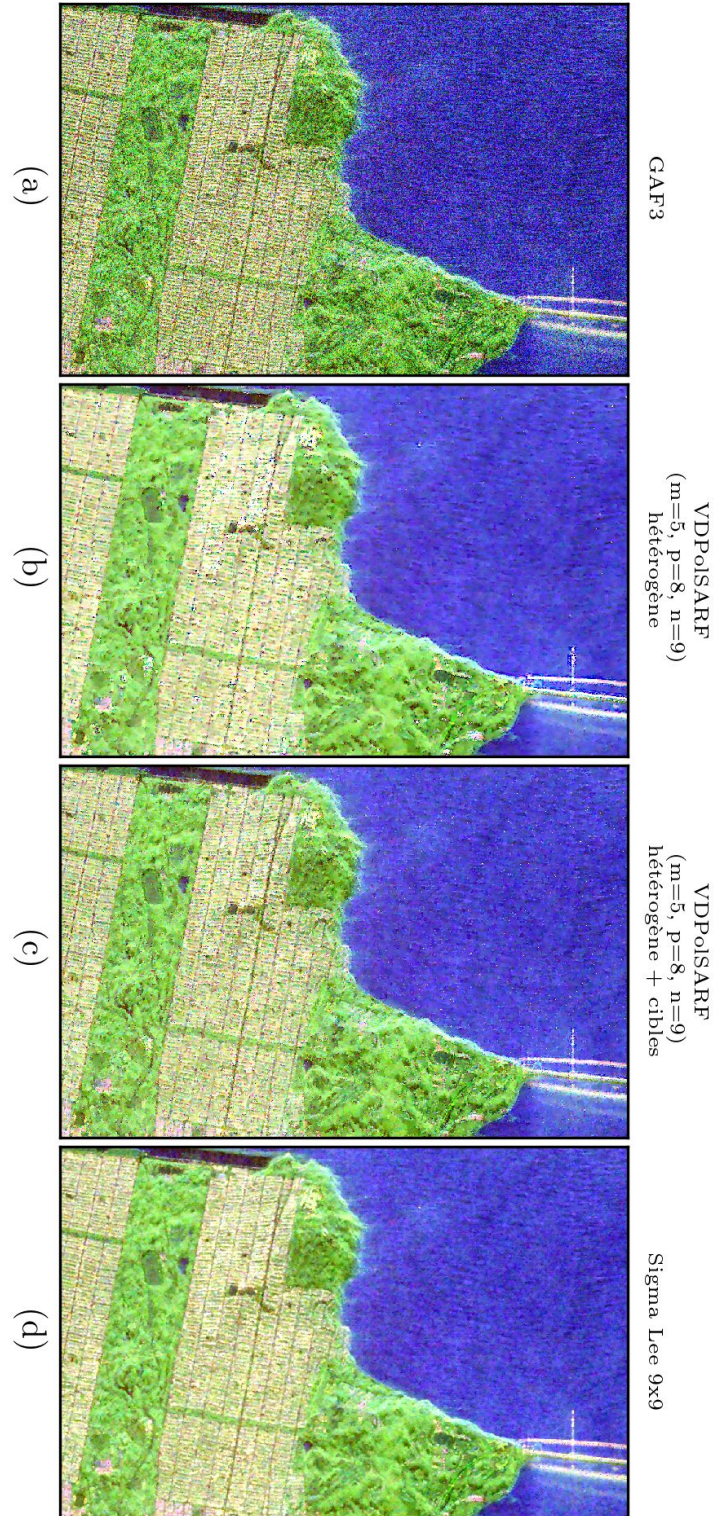
**Fig. 4.49.** Comparaison entre les filtrages de l'image RADARSAT2 (2): (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre **Sigma Lee** ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).



**Fig. 4.50.** Comparaison entre les filtrages de l'image ALOS2 (1): (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre Sigma Lee ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).

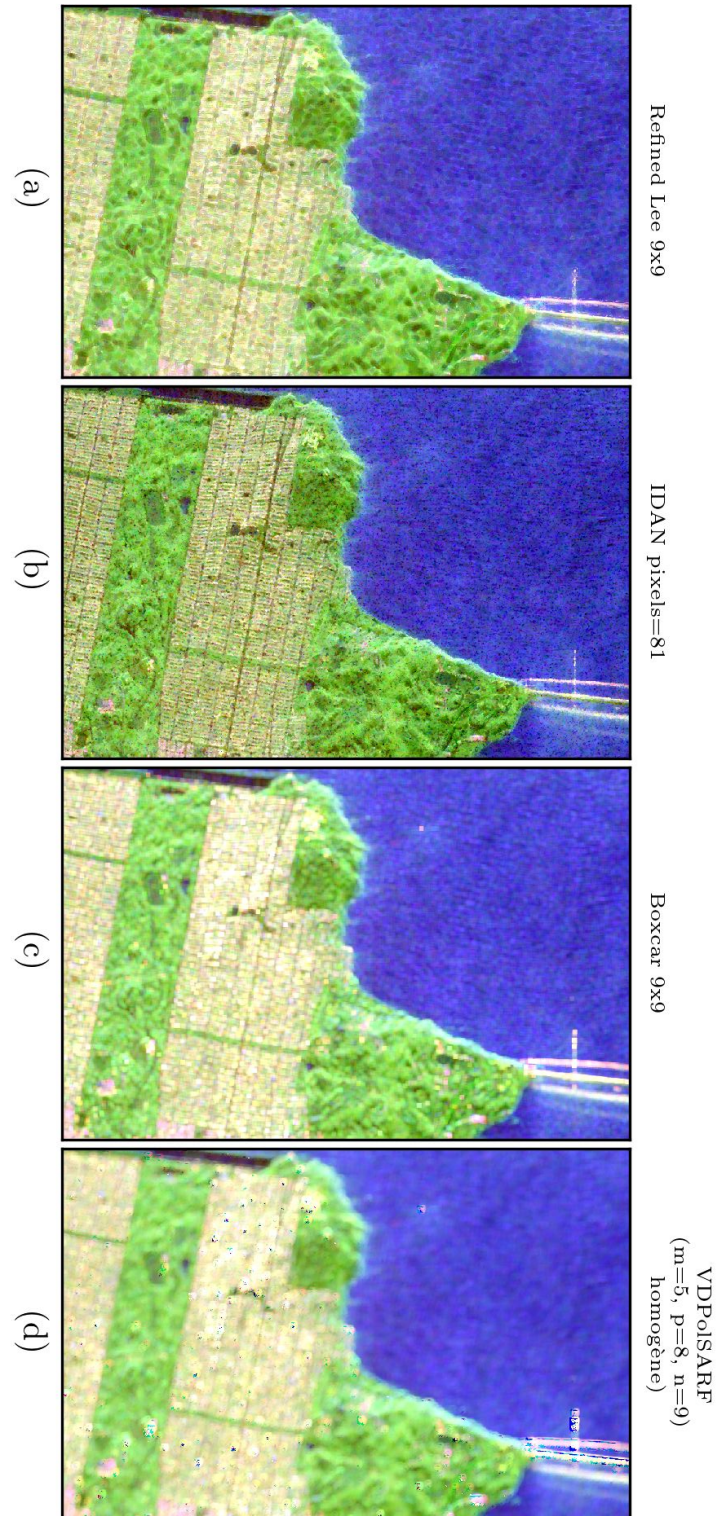


**Fig. 4.51.** Comparaison entre les filtrages de l'image ALOS2 (2): (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre Sigma Lee ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).

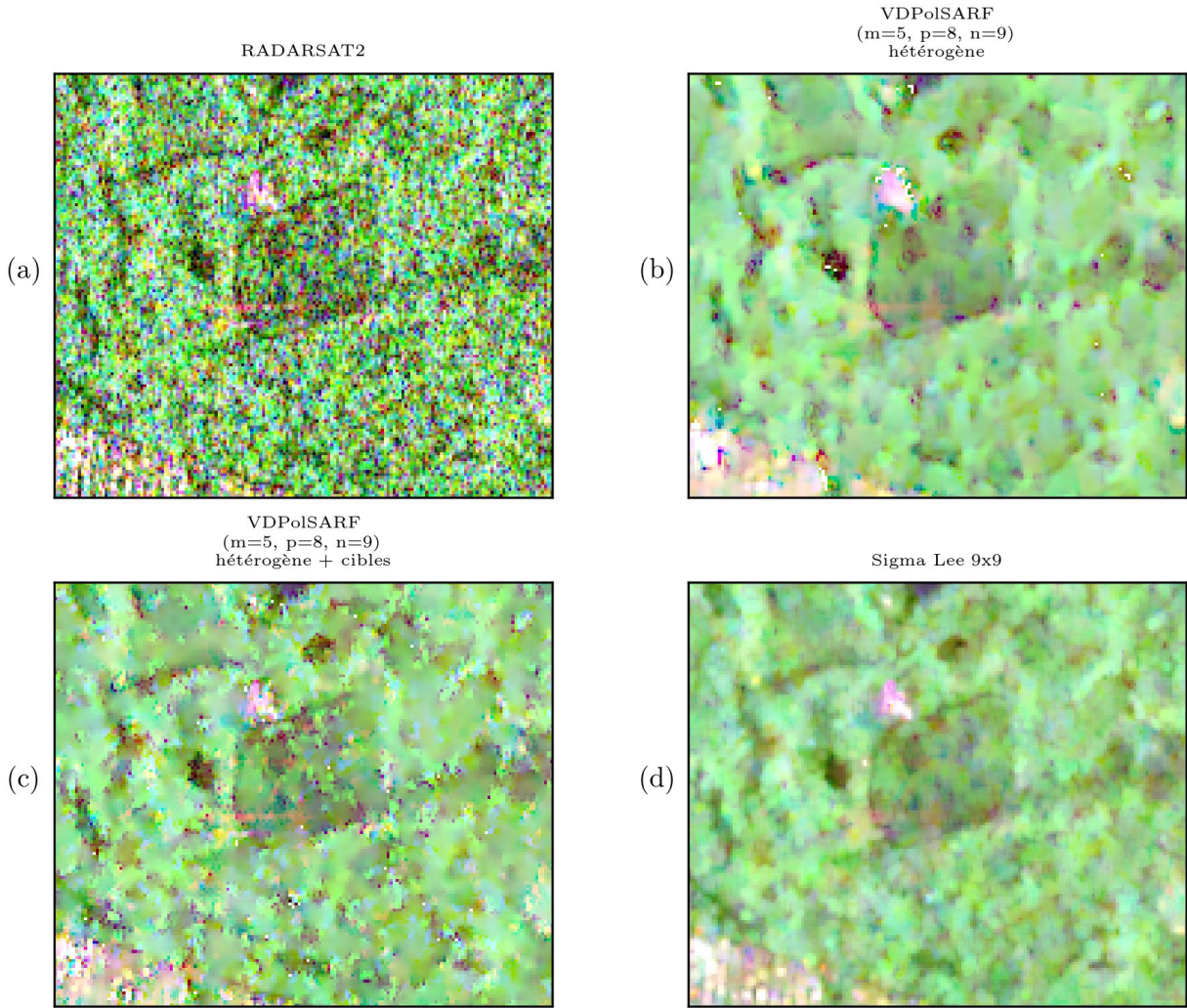


**Fig. 4.52.** Comparaison entre les filtrages de l'image GAF3 (1): (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre Sigma Lee ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).

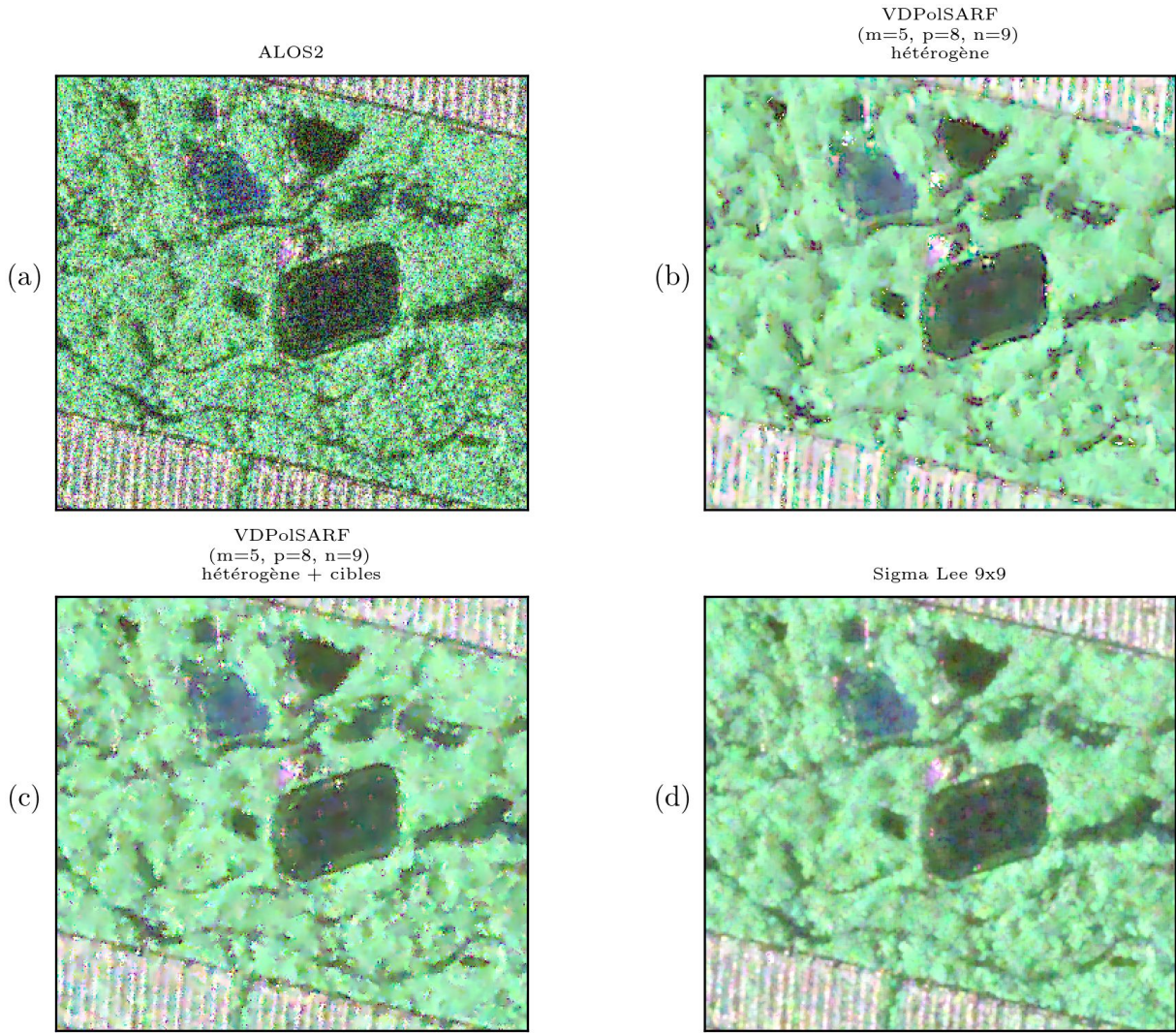




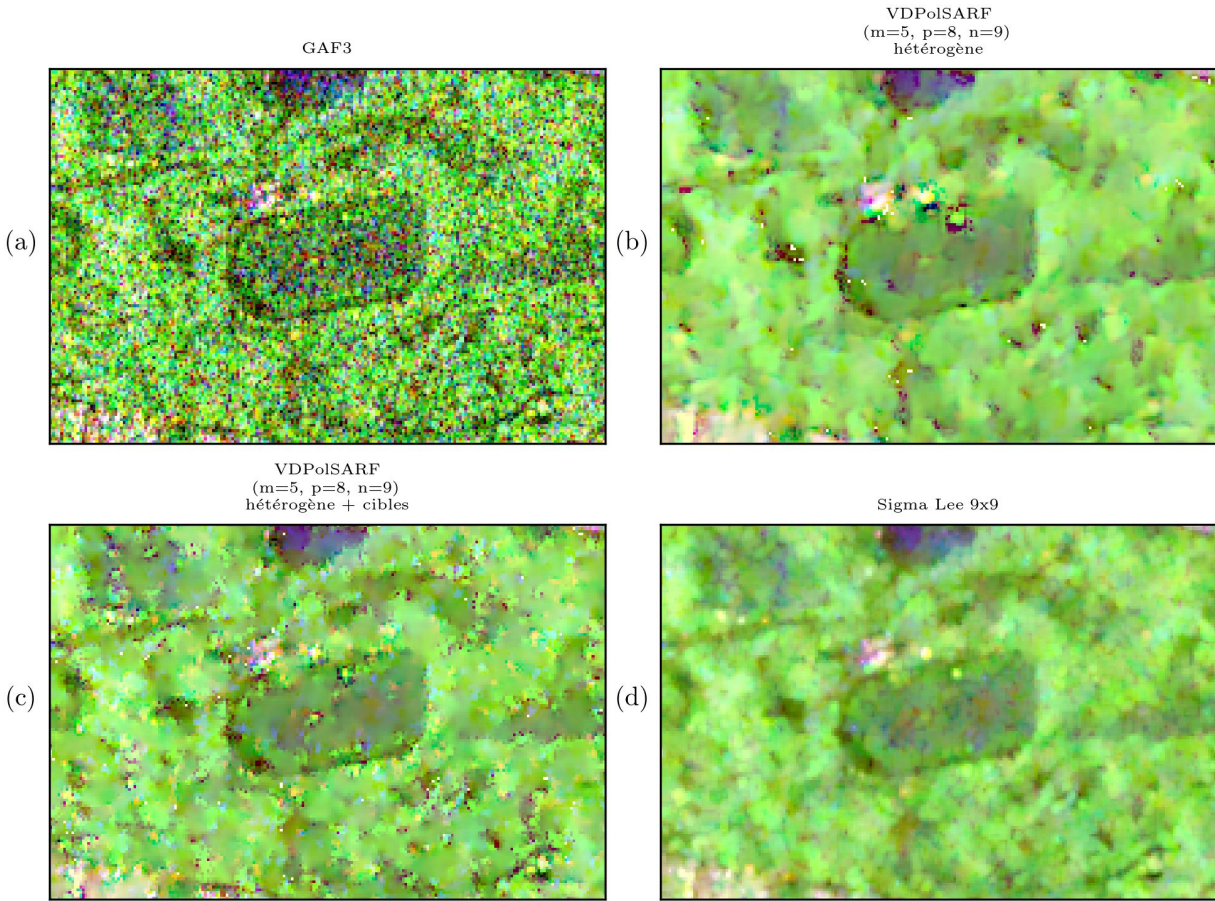
**Fig. 4.53.** Comparaison entre les filtrages de l'image GAF3 (2): (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre Sigma Lee ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).



**Fig. 4.54.** Comparaison entre les filtrages de l'image RADARSAT2 (1): (a) image originale, (b) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre **Sigma Lee** ( $9 \times 9$ ).  $(S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV})$ .



**Fig. 4.55.** Comparaison entre les filtrages de l'image ALOS2 (1): (a) image originale, (b) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5, p=8, n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre **Sigma Lee** ( $9 \times 9$ ). ( $S_{HH} - S_{VV}, \sqrt{2}S_{HV}, S_{HH} + S_{VV}$ ).



**Fig. 4.56.** Comparaison entre les filtrages de l'image **GAF3 (1)**: (a) image originale, (b) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes, (c) modèle **VDPolSARF** ( $m=5$ ,  $p=8$ ,  $n=9$ ) apprentissage sur les données hétérogènes avec inclusion de cibles ponctuelles, (d) filtre **Sigma Lee** ( $9 \times 9$ ). ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ ).

## 4.10. Évaluation du nombre équivalent de vues sur les images polarimétriques réelles

Les tableaux suivants (Tab. 4.9, 4.10 et 4.11) présentent le nombre équivalent de vues calculé pour chacune de nos expérimentations. Sur chaque image polarimétrique nous avons sélectionné la même région homogène et exécuté le calcul sur les éléments en puissance des matrices de cohérence. Pour chacun des capteurs, nous obtenons des résultats supérieurs avec le modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données hétérogènes par rapport aux filtres standards (**Sigma Lee**, **Refined Lee** et **IDAN**). Il n'est dépassé que par le **Boxcar**, sauf dans le cas des données ALOS2. Les résultats pour le modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur les données hétérogènes avec inclusion de cible corrobore notre problème de sous-filtrage identifié à la section précédente (sec. 4.9). Les résultats sont nettement inférieurs à nos attentes. Dans le cas du modèle **VDPolSARF** ( $m = 5, p = 8, n = 9$ ) entraîné sur le données homogènes, il démontrent nettement son efficacité à filtrer les zones homogènes. Les **NEV** obtenus sont nettement supérieurs au **Boxcar**. Ce qui confirme nos résultats précédents sur la qualité du filtrage.

	POLARSAR			VDPolSARF (m=5, p=8, n=9) hétérogène			VDPolSARF (m=5, p=8, n=9) hétérogène + cibles			Sigma Lee 9 × 9		
	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV
$T_{11}$	2.24e-01	2.50e-01	<b>0.80</b>	2.19e-01	5.63e-02	<b>15.19</b>	2.22e-01	1.48e-01	<b>2.25</b>	2.08e-01	7.70e-02	<b>7.31</b>
$T_{22}$	8.38e-03	8.73e-03	<b>0.92</b>	8.27e-03	1.76e-03	<b>22.06</b>	8.34e-03	4.12e-03	<b>4.09</b>	7.90e-03	2.40e-03	<b>10.80</b>
$T_{33}$	2.11e-03	2.24e-03	<b>0.88</b>	2.06e-03	4.85e-04	<b>18.08</b>	2.09e-03	1.12e-03	<b>3.50</b>	1.96e-03	5.86e-04	<b>11.13</b>
	Refined Lee 9 × 9			IDAN pixels=81			Boxcar 9 × 9			VDPolSARF (m=5, p=8, n=9) homogène		
$T_{11}$	1.84e-01	5.64e-02	<b>10.69</b>	1.54e-01	5.39e-02	8.19	2.23e-01	4.55e-02	<b>24.13</b>	2.22e-01	2.99e-02	<b>55.07</b>
$T_{22}$	7.95e-03	1.99e-03	<b>15.98</b>	6.14e-03	2.02e-03	9.27	8.40e-03	1.46e-03	<b>33.17</b>	8.40e-03	8.90e-04	<b>89.08</b>
$T_{33}$	2.05e-03	5.35e-04	<b>14.73</b>	1.57e-03	5.31e-04	8.71	2.09e-03	3.97e-04	<b>27.75</b>	2.09e-03	2.50e-04	<b>69.45</b>

**Tab. 4.9.** Mesure du nombre équivalent de vues sur une zone de type surfacique sur les termes en puissance de l'image RADARSAT2. La moyenne, l'écart-type et le **NEV** est calculé pour chaque filtre.

	POLARSAR			VDPolSARF (m=5, p=8, n=9) hétérogène			VDPolSARF (m=5, p=8, n=9) hétérogène + cibles			Sigma Lee 9 × 9		
	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV
$T_{11}$	7.33e-02	7.74e-02	<b>0.90</b>	7.26e-02	1.79e-02	<b>16.42</b>	7.30e-02	2.47e-02	<b>8.76</b>	6.98e-02	2.17e-02	<b>10.38</b>
$T_{22}$	1.60e-02	1.61e-02	<b>0.99</b>	1.59e-02	2.43e-03	<b>43.09</b>	1.60e-02	3.22e-03	<b>24.71</b>	1.54e-02	3.65e-03	<b>17.93</b>
$T_{33}$	3.82e-03	3.84e-03	<b>0.99</b>	3.80e-03	5.92e-04	<b>41.24</b>	3.81e-03	6.44e-04	<b>35.03</b>	3.71e-03	8.93e-04	<b>17.20</b>
	Refined Lee 9 × 9			IDAN pixels=81			Boxcar 9 × 9			VDPolSARF (m=5, p=8, n=9) homogène		
$T_{11}$	6.20e-02	1.69e-02	<b>13.39</b>	5.59e-02	1.92e-02	8.45	7.33e-02	1.77e-02	<b>17.14</b>	7.33e-02	1.46e-02	<b>25.19</b>
$T_{22}$	1.53e-02	3.14e-03	<b>23.83</b>	1.25e-02	3.57e-03	12.25	1.60e-02	2.63e-03	<b>37.27</b>	1.60e-02	1.77e-03	<b>81.50</b>
$T_{33}$	3.76e-03	7.76e-04	<b>23.53</b>	3.05e-03	8.85e-04	11.84	3.82e-03	6.50e-04	<b>34.44</b>	3.82e-03	4.77e-04	<b>64.08</b>

**Tab. 4.10.** Mesure du nombre équivalent de vues équivalent sur une zone de type surfacique sur les termes en puissance de l'image ALOS2. La moyenne, l'écart-type et le **NEV** est calculé pour chaque filtre.

	POLARSAR			VDPolSARF (m=5, p=8, n=9) hétérogène			VDPolSARF (m=5, p=8, n=9) hétérogène + cibles			Sigma Lee 9 × 9		
	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV	$\mu$	$\sigma$	NEV
$T_{11}$	6.18e-01	7.16e-01	<b>0.74</b>	6.05e-01	1.97e-01	<b>9.47</b>	6.14e-01	4.02e-01	<b>2.33</b>	5.74e-01	2.48e-01	<b>5.36</b>
$T_{22}$	1.07e-02	1.13e-02	<b>0.90</b>	1.06e-02	2.53e-03	<b>17.55</b>	1.07e-02	5.30e-03	<b>4.05</b>	1.02e-02	3.42e-03	<b>8.82</b>
$T_{33}$	1.25e-03	1.30e-03	<b>0.92</b>	1.23e-03	3.32e-04	<b>13.72</b>	1.24e-03	7.00e-04	<b>3.15</b>	1.18e-03	3.94e-04	<b>8.98</b>
	Refined Lee 9 × 9			IDAN pixels=89			Boxcar 9 × 9			VDPolSARF (m=5, p=8, n=9) homogène		
$T_{11}$	5.03e-01	1.79e-01	<b>7.94</b>	4.09e-01	1.70e-01	5.80	6.19e-01	1.58e-01	<b>15.35</b>	6.18e-01	1.11e-01	<b>30.85</b>
$T_{22}$	9.86e-03	2.70e-03	<b>13.33</b>	7.69e-03	2.59e-03	8.78	1.08e-02	2.29e-03	<b>22.05</b>	1.08e-02	1.57e-03	<b>46.71</b>
$T_{33}$	1.23e-03	3.45e-04	<b>12.64</b>	9.54e-04	3.34e-04	8.15	1.25e-03	2.90e-04	<b>18.70</b>	1.25e-03	2.13e-04	<b>34.68</b>

**Tab. 4.11.** Mesure du nombre de vues équivalent sur une zone de type surfacique sur les termes en puissance de l'image GAF3. La moyenne, l'écart-type et le **NEV** est calculé pour chaque filtre.

## 4.11. Conclusion

Nous avons présenté dans ce chapitre un survol sur l'ensemble des résultats produits. La première partie portait sur les résultats des apprentissages du point de vue du *machine learning*. La seconde partie et la plus importante portait sur la capacité des modèles à bien estimer les matrices de cohérence sur les données polarimétriques synthétiques. Et la

dernière partie de ce chapitre portait sur une évaluation des modèles sur les données réelles. Les résultats obtenus s'avèrent bons et très prometteurs pour l'application des **RNC** comme outils de filtrage des images polarimétriques et l'estimation des paramètres de la matrice de cohérence.





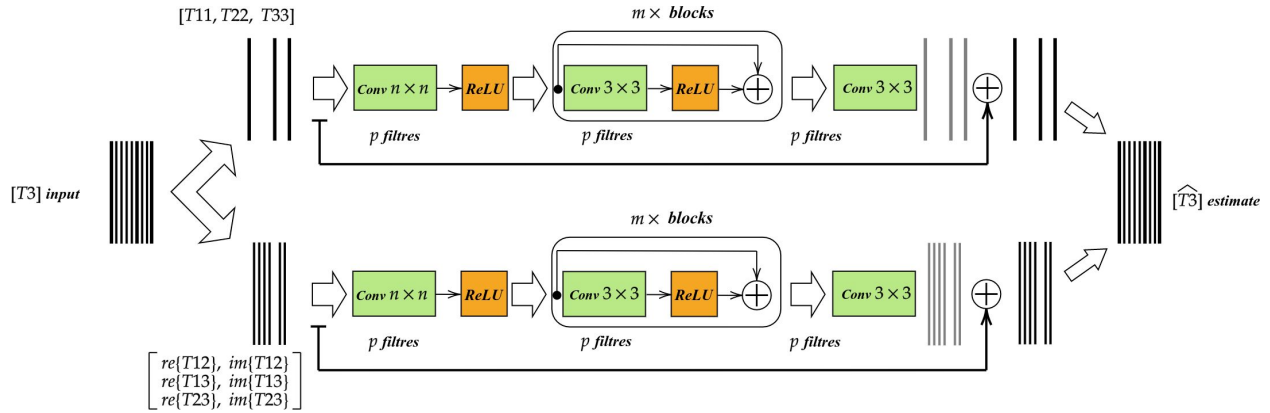
# Chapitre 5

---

## CONCLUSION

Cette étude a permis d’explorer le potentiel des réseaux neuronaux à convolution appliqués au filtrage des images QuadPol en milieu urbain. Nous avons implémenté une architecture simple et modulaire nommée **VDPolSARF** inspirée des réseaux **VDSR** et **RESNET**. Cette architecture nous a permis d’explorer l’effet de différents paramètres dont la taille de la convolution d’entrée, le nombre de convolutions par couche et le nombre de couches du réseau. Comparées aux approches classiques, les architectures testées ont démontré que les approches en apprentissage machine profond permettent d’obtenir une bonne estimation des paramètres polarimétriques tout en préservant la résolution spatiale aussi bien sinon mieux que ces dernières. De plus, les travaux ont montré qu’il n’est pas nécessaire d’avoir des réseaux extrêmement profonds (5 couches suffisent) pour réaliser un bon filtrage. Cependant, certains aspects de ces filtres restent à explorer et à améliorer.

Du point de vue de l’apprentissage, nous pourrions visiter l’espace des paramètres qui sont demeurés fixes durant nos expérimentations afin d’optimiser celui-ci. En particulier, il serait intéressant de vérifier l’impact du paramètre du taux d’apprentissage plus adéquatement sur la vitesse de convergence et la précision. Nous pourrions de plus utiliser des fonctions de coût basées sur des métriques mieux adaptées à la polarimétrie, comme par exemple la distance de Frobenius. Un autre aspect à améliorer serait le balancement des classes polarimétriques, car selon la distribution de nos échantillons sur le plan  $H/\bar{\alpha}$  (Fig. 3.12), certaines de nos classes sont mieux représentées. L’idéal serait de faire un apprentissage balancé uniformément et vérifier si ceci apporterait de meilleurs résultats. Toujours en lien avec la distribution des échantillons, nous remarquons qu’ils ne sont pas uniformément répartis sur le plan  $H/A$  (Fig. 3.13). Ce qui nous amène à réfléchir que la production des signatures devrait être distribuée

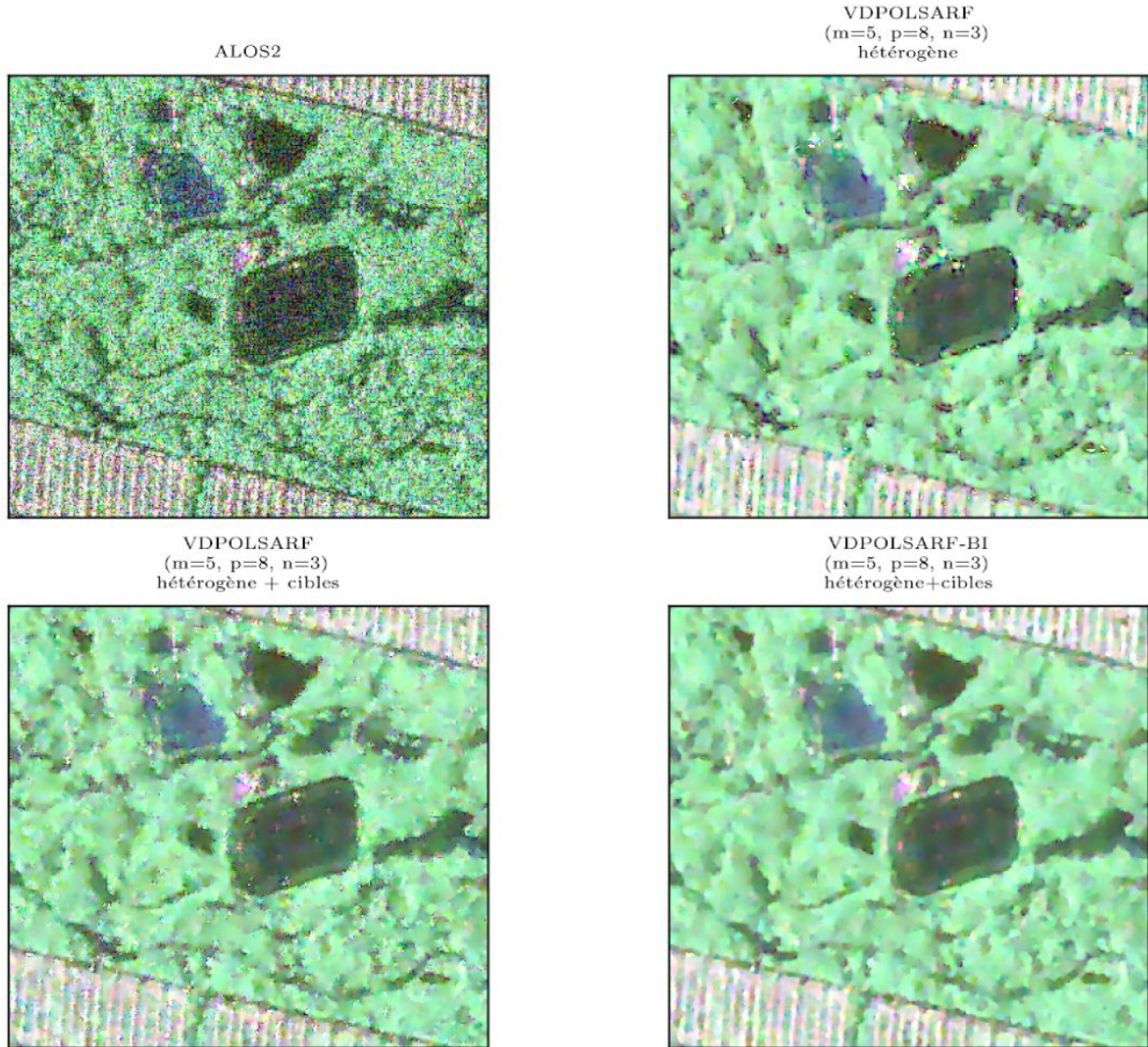


**Fig. 5.1.** Exemple d’une architecture permettant de filtrer indépendamment les termes copolaires et les termes croisés de la matrice de cohérence. La branche du haut modélise les 3 termes copolaires et la branche du bas modélise chacune des parties des nombres complexes des termes croisés.

uniformément selon les trois axes  $H/A/\bar{\alpha}$  afin qu’il n’y ait pas de biais introduit par rapport à la distribution des valeurs propres des matrices de cohérence.

Un problème en lien avec la grande différence des statistiques entre les termes co-polaires et les termes croisés des matrices de cohérence bruitées est que les réseaux se voient obliger de modéliser deux ensembles de données complètement différents. Nous pourrions réfléchir à une approche où chaque terme de type co-polaire et croisé serait filtré par leur propre réseau. Ceci nous rapproche de la proposition de Lopèz-Martinez et Fàbregas [25] qui modéliserait les termes croisés indépendamment des termes copolaires. La Figure 5.1 présente une architecture qui pourrait aider à résoudre le problème. Elle est une simple extension de notre meilleur réseau actuel. Ce nouveau réseau serait découpé en deux branches, chaque branche modélise indépendamment un estimateur des paramètres copolaires et un estimateur des paramètres croisés. En sortie du réseau les estimations sont réordonnées correctement pour former l’estimation complète de la matrice de cohérence.

Pour conclure ce mémoire nous présentons un résultat qualitatif préliminaire de ce réseau bi-céphale. La Figure 5.2 compare nos deux modèles **VDPolSARF** présentés auparavant avec le nouveau modèle que nous appellerons **VDPolSARF-Bi**. Ce dernier a été entraîné sur les données hétérogènes avec inclusion de cibles ponctuelles. La même méthodologie a été utilisée pour l’entraînement à la différence près que la puissance des cibles ponctuelles a été légèrement augmentée. La sous-figure en-bas à droite présente notre nouveau résultat qui se compare favorablement aux résultats précédents. On y remarque que les artéfacts autour



**Fig. 5.2.** Résultats comparatifs en les modèles antécédents et le résultat de la nouvelle architecture. En-haut à gauche, l'image ALOS2 une vue. En-haut à droite, le résultat du modèle **VDPOLSARF** entraîné sur les données hétérogènes. En-bas à gauche, le résultat du modèle **VDPOLSARF** entraîné sur les données hétérogènes avec inclusion de cibles ponctuelles. Et finalement, en-bas à droite, le résultat du modèle bi-céphale **VDPOLSARF-BI** entraîné sur les données hétérogènes avec inclusion de cibles ponctuelles. ( $S_{HH} - S_{VV}$ ,  $\sqrt{2}S_{HV}$ ,  $S_{HH} + S_{VV}$ )

des cibles ont disparu et que les zones homogènes semblent bien filtrées. Les contours et les arêtes semblent bien préservés aussi. Ce qui nous ouvrent d'autres voies à la recherche qui paraissent valoir la peine d'être explorées. Mais pour l'instant, nous concluons cette étape des travaux ici.



# Bibliographie

---

- [1] W. L. CAMERON, N. N. YOUSSEF et L. K. LEUNG : Simulated polarimetric signatures of primitive geometrical shapes. *IEEE Transactions on Geoscience and Remote Sensing*, 34(3):793–803, May 1996.
- [2] G. CHIERCHIA, D. COZZOLINO, G. POGGI et L. VERDOLIVA : Sar image despeckling through convolutional neural networks. In *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 5438–5441, July 2017.
- [3] M. CIMPOI, S. MAJI, I. KOKKINOS, S. MOHAMED, et A. VEDALDI : Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [4] S. R. CLOUDE et E. POTTIER : A review of target decomposition theorems in radar polarimetry. *IEEE Transactions on Geoscience and Remote Sensing*, 34(2):498–518, March 1996.
- [5] S. R. CLOUDE et E. POTTIER : An entropy based classification scheme for land applications of polarimetric sar. *IEEE Transactions on Geoscience and Remote Sensing*, 35(1):68–78, Jan 1997.
- [6] C. DONG, C. C. LOY, K. HE et X. TANG : Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, Feb 2016.
- [7] S. FOUCHER, M. BEAULIEU, M. DAHMANE et F. CAVAYAS : Deep speckle noise filtering. pages 5311–5314, July 2017.
- [8] S. FOUCHER et C. LÓPEZ-MARTÍNEZ : Analysis, evaluation, and comparison of polarimetric sar speckle filtering techniques. *IEEE Transactions on Image Processing*, 23(4):1751–1764, April 2014.
- [9] A. FREEMAN et S. L. DURDEN : A three-component scattering model for polarimetric sar data. *IEEE Transactions on Geoscience and Remote Sensing*, 36(3):963–973, May 1998.
- [10] K. FUKUSHIMA : Neocognitron: a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern*, 36(4):193–202, 1980.
- [11] Xavier GLOROT, Antoine BORDES et Yoshua BENGIO : Deep sparse rectifier neural networks. 15:315–323, 11–13 Apr 2011.
- [12] L. GOMEZ, M. E. BUEMI, J. C. JACOBO-BERLLES et M. E. MEJAIL : A new image quality index for objectively evaluating despeckling filtering in sar images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 9(3):1297–1307, March 2016.
- [13] J. W. GOODMAN : Some fundamental properties of speckle\*. *J. Opt. Soc. Am.*, 66(11):1145–1150, Nov 1976.

- [14] Kaiming HE, Xiangyu ZHANG, Shaoqing REN et Jian SUN : Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
- [15] D. HUBEL et T. WIESEL : Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.
- [16] J. KIM, J. K. LEE et K. M. LEE : Accurate image super-resolution using very deep convolutional networks. pages 1646–1654, June 2016.
- [17] Jiwon KIM, Jung Kwon LEE et Kyoung Mu LEE : Accurate image super-resolution using very deep convolutional networks. *In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral)*, June 2016.
- [18] Diederik P. KINGMA et Jimmy BA : Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.
- [19] Alex KRIZHEVSKY, Ilya SUTSKEVER et Geoffrey E HINTON : Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.
- [20] E. KROGAGER : New decomposition of the radar target scattering matrix. *Electronics Letters*, 26(18): 1525–1527, Aug 1990.
- [21] Y. LECUN, L. BOTTOU, Y. BENGIO et P. HAFFNER : Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.
- [22] Jong-Sen LEE, M. R. GRUNES et G. de GRANDI : Polarimetric sar speckle filtering and its implication for classification. *IEEE Transactions on Geoscience and Remote Sensing*, 37(5):2363–2373, Sept 1999.
- [23] Yangyang LI, Yanqiao CHEN, Guangyuan LIU et Licheng JIAO : A novel deep fully convolutional network for polsar image classification. *Remote Sensing*, 10(12), 2018.
- [24] Weibo LIU, Zidong WANG, Xiaohui LIU, Nianyin ZENG, Yurong LIU et Fuad E. ALSAADI : A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26, 2017.
- [25] C. LOPEZ-MARTINEZ et X. FABREGAS : Polarimetric sar speckle noise model. *IEEE Transactions on Geoscience and Remote Sensing*, 41(10):2232–2242, Oct 2003.
- [26] Warren S. MCCULLOCH et Walter PITTS : A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, Dec 1943.
- [27] A. MOREIRA, P. PRATS-IRAOLA, M. YOUNIS, G. KRIEGER, I. HAJNSEK et K. P. PAPATHANASSIOU : A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, 1(1):6–43, March 2013.
- [28] Pedro H. O. PINHEIRO et Ronan COLLOBERT : Recurrent convolutional neural networks for scene parsing. *CoRR*, abs/1306.2795, 2013.
- [29] Luca PIPIA et Xavier FABREGAS : Generation of pol-sar and pol-in-sar data for homogeneous distributed targets simulation. 586:35, 04 2005.

- [30] John A. RICHARDS : *Remote Sensing with Imaging Radar*. Springer Publishing Company, Incorporated, 1st édition, 2009.
- [31] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX : U-net: Convolutional networks for biomedical image segmentation. 2015.
- [32] Pierre SERMANET, David EIGEN, Xiang ZHANG, Michaël MATHIEU, Rob FERGUS et Yann LECUN : Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.
- [33] E. SHELHAMER, J. LONG et T. DARRELL : Fully convolutional networks for semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, April 2017.
- [34] Jamie SHERRAH : Fully convolutional networks for dense semantic labelling of high-resolution aerial imagery. *CoRR*, abs/1606.02585, 2016.
- [35] K. SIMONYAN et A. ZISSERMAN : Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [36] C. SZEGEDY, Wei LIU, Yangqing JIA, P. SERMANET, S. REED, D. ANGUELOV, D. ERHAN, V. VAN-  
HOUCHE et A. RABINOVICH : Going deeper with convolutions. pages 1–9, June 2015.
- [37] F. Van VEEN : The neural network zoo. 2016.
- [38] P. WANG, H. ZHANG et V. M. PATEL : Sar image despeckling using a convolutional neural network. *IEEE Signal Processing Letters*, 24(12):1763–1767, Dec 2017.
- [39] Wenming YANG, Xuechen ZHANG, Yapeng TIAN, Wei WANG et Jing-Hao XUE : Deep learning for single image super-resolution: A brief review. *CoRR*, abs/1808.03344, 2018.
- [40] Matthew D. ZEILER et Rob FERGUS : Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.
- [41] Matthew D. ZEILER, Dilip KRISHNAN, Graham W. TAYLOR et Rob FERGUS : Deconvolutional networks. 2010.
- [42] K. ZHANG, W. ZUO, Y. CHEN, D. MENG et L. ZHANG : Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017.
- [43] K. ZHANG, W. ZUO et L. ZHANG : Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, Sept 2018.
- [44] Qiang ZHANG, Qiangqiang YUAN, Jie LI, Zhen YANG et Xiaoshuang MA : Learning a dilated residual network for sar image despeckling. *Remote Sensing*, 10:196, 2018.





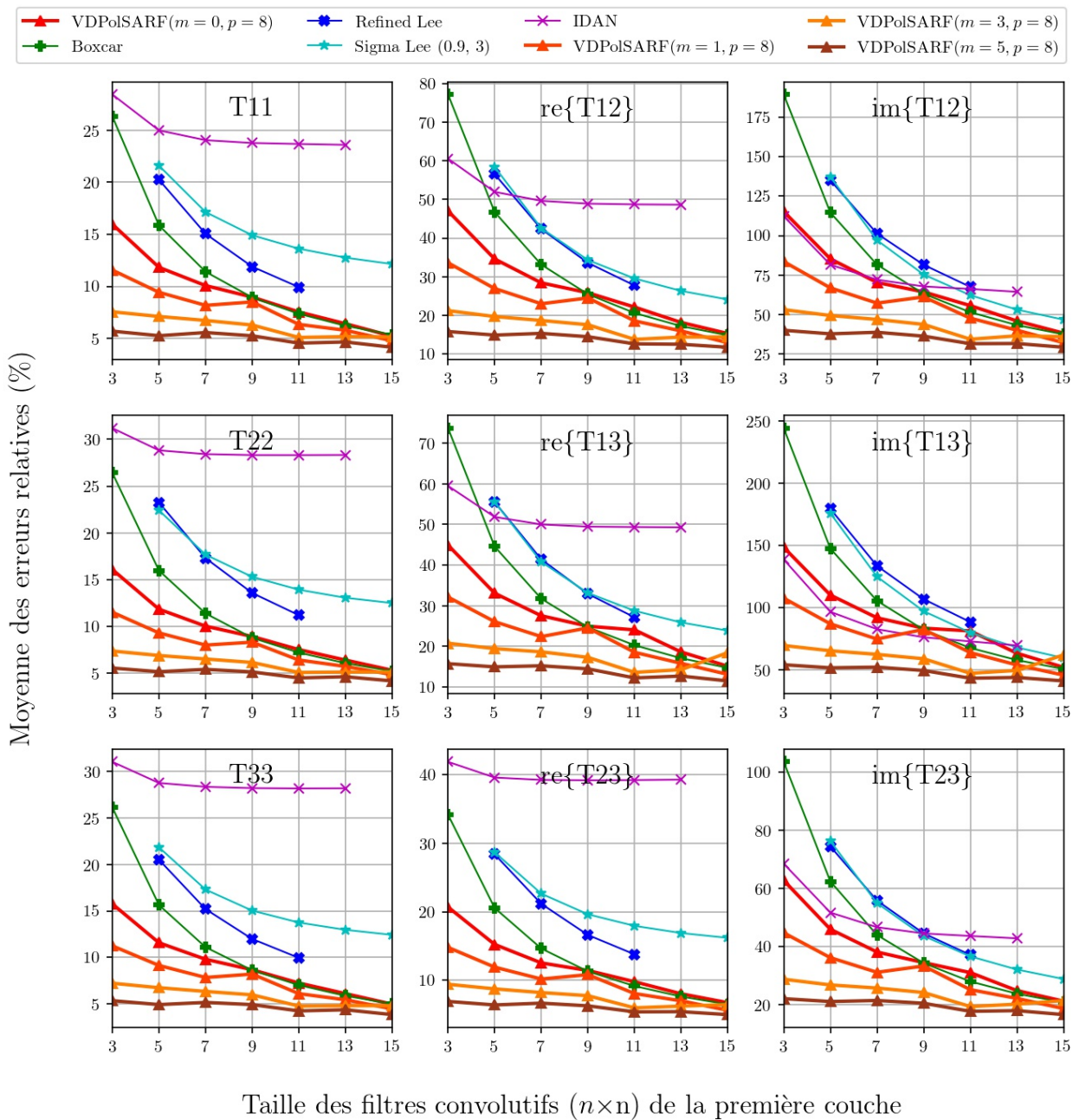
# Annexe A

---

## Données simulées homogènes

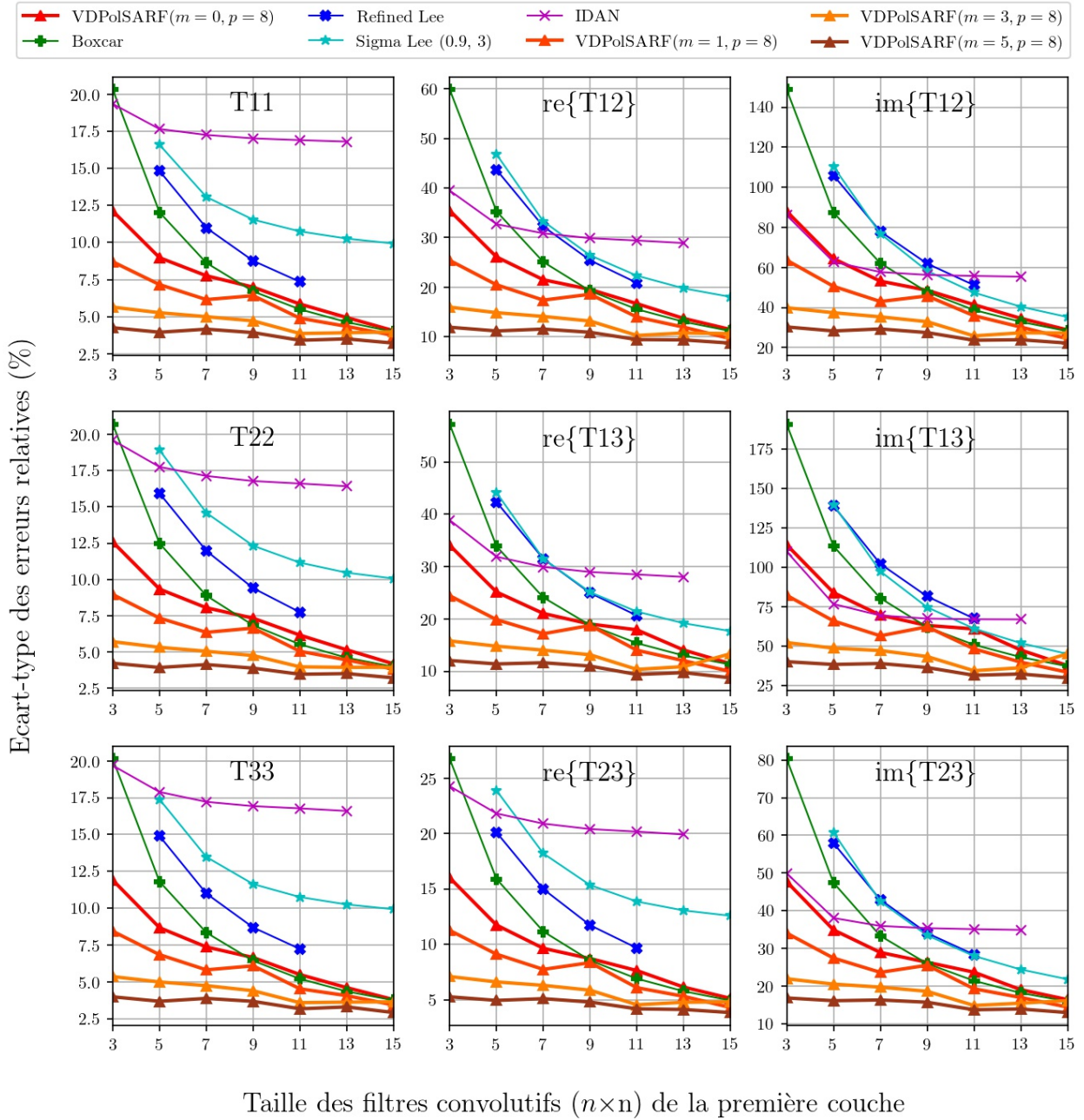
A.1. Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées homogènes des classes de diffuseurs polarimétriques.

## Réflexions diédrales



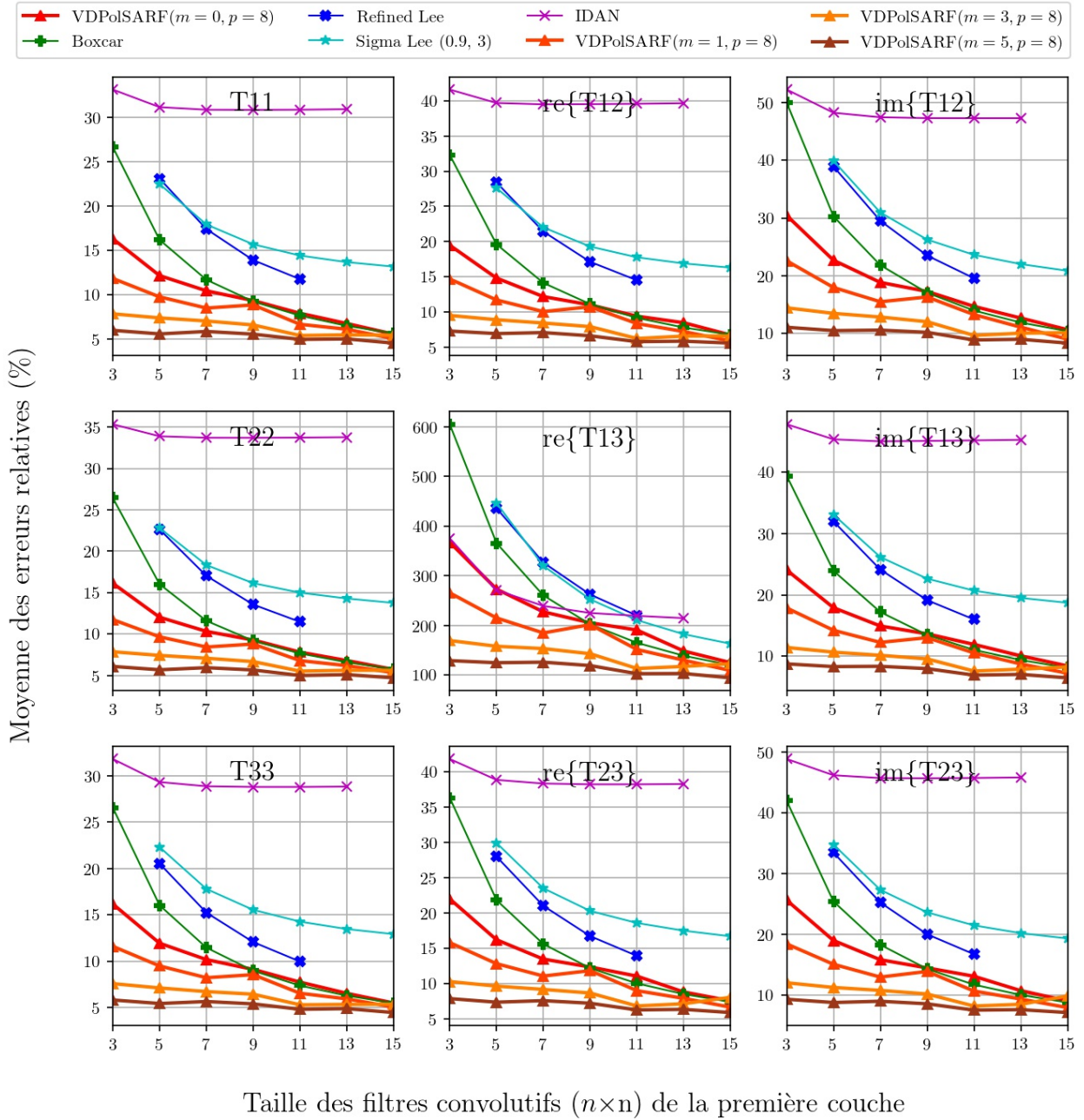
**Fig. A.1. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



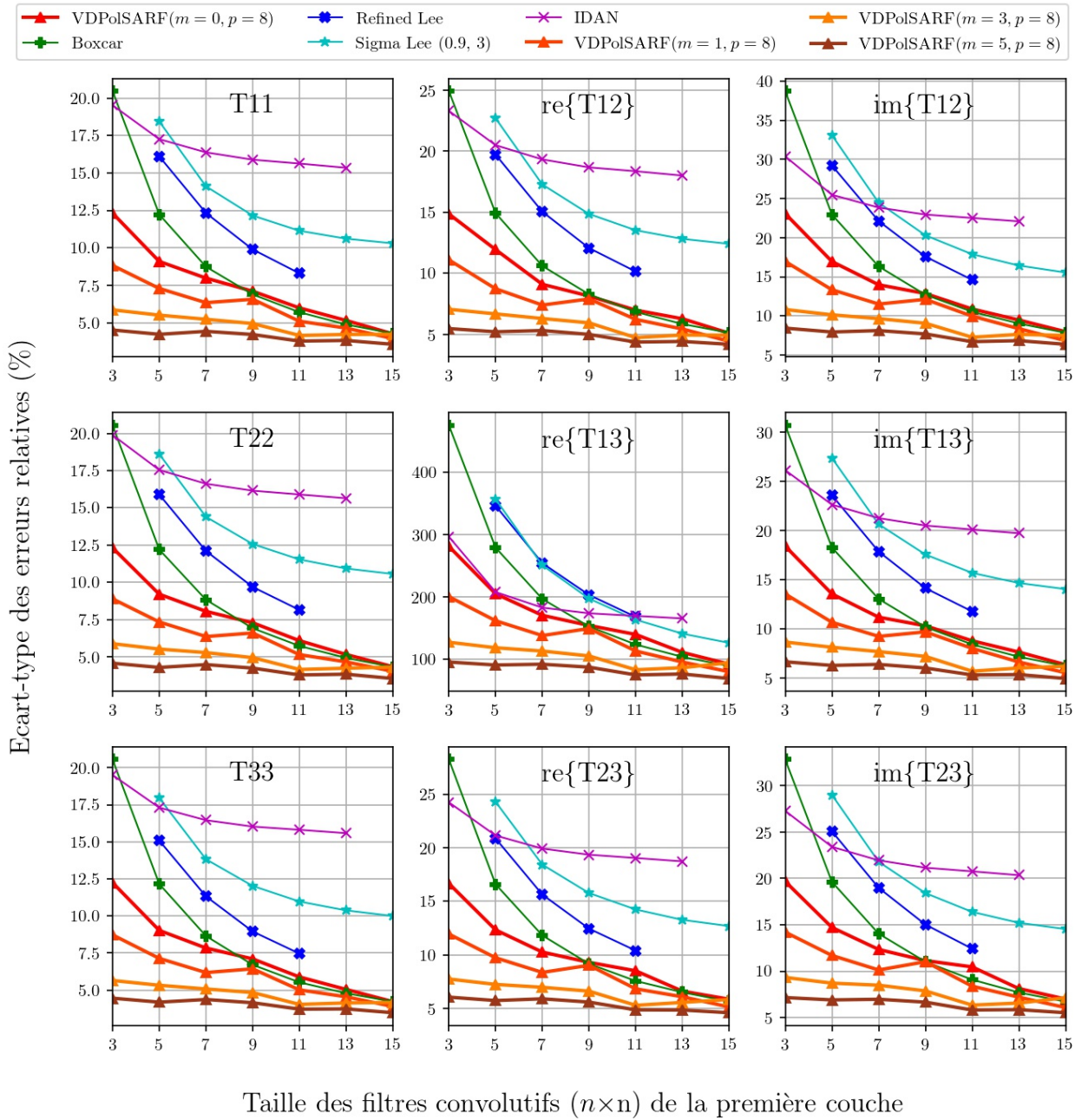
**Fig. A.2. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



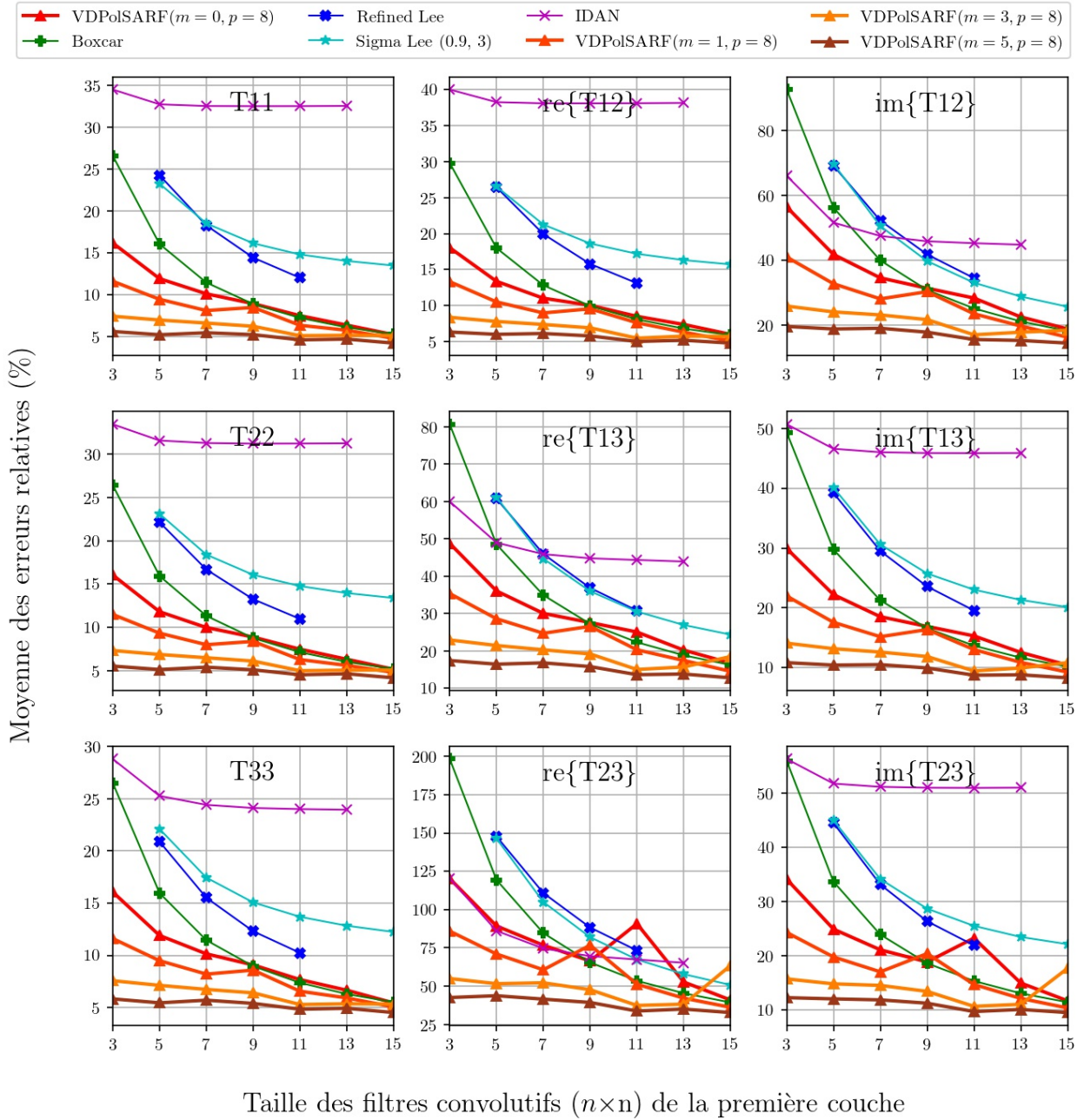
**Fig. A.3. Dipôles** . Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



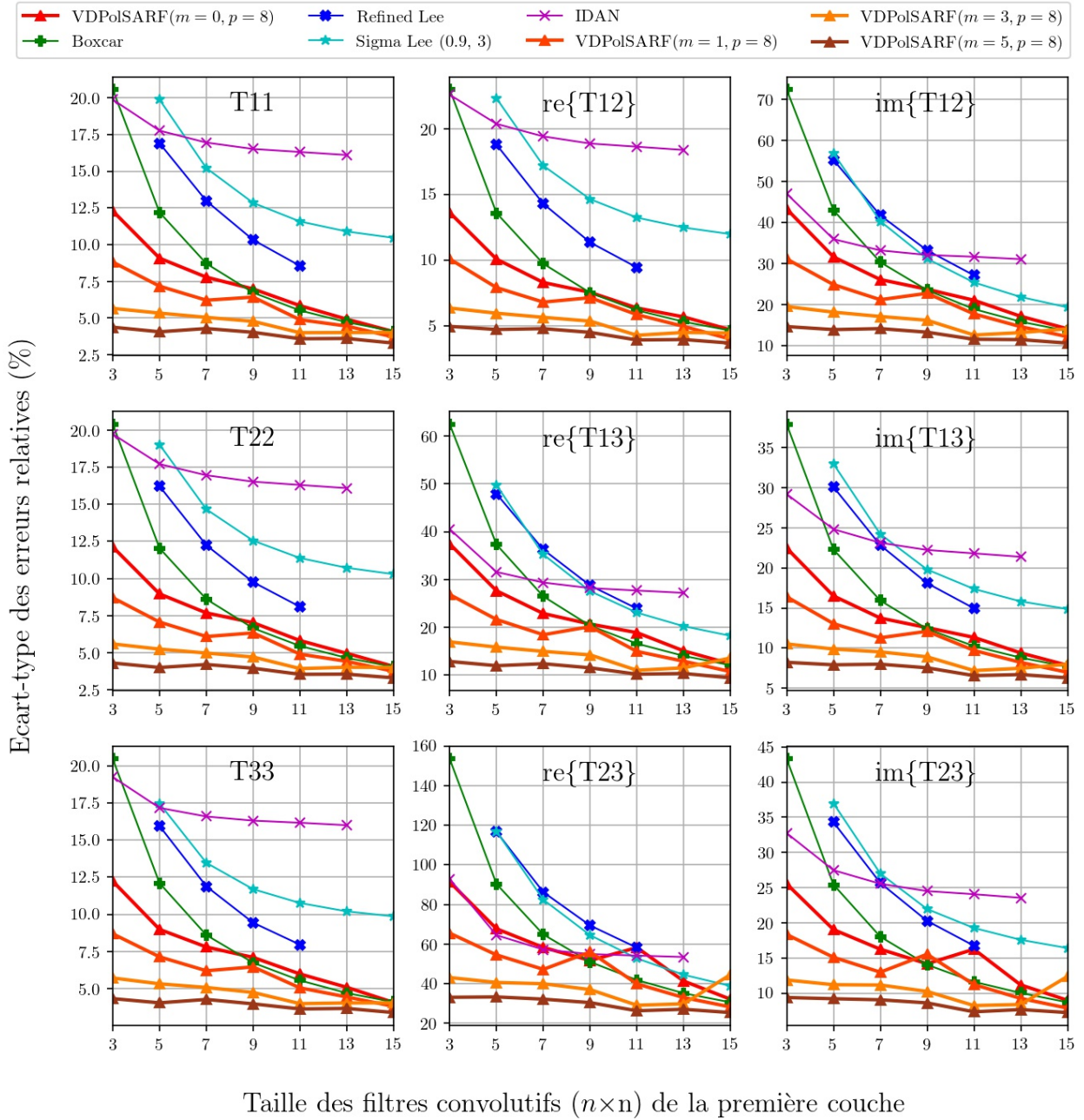
**Fig. A.4. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



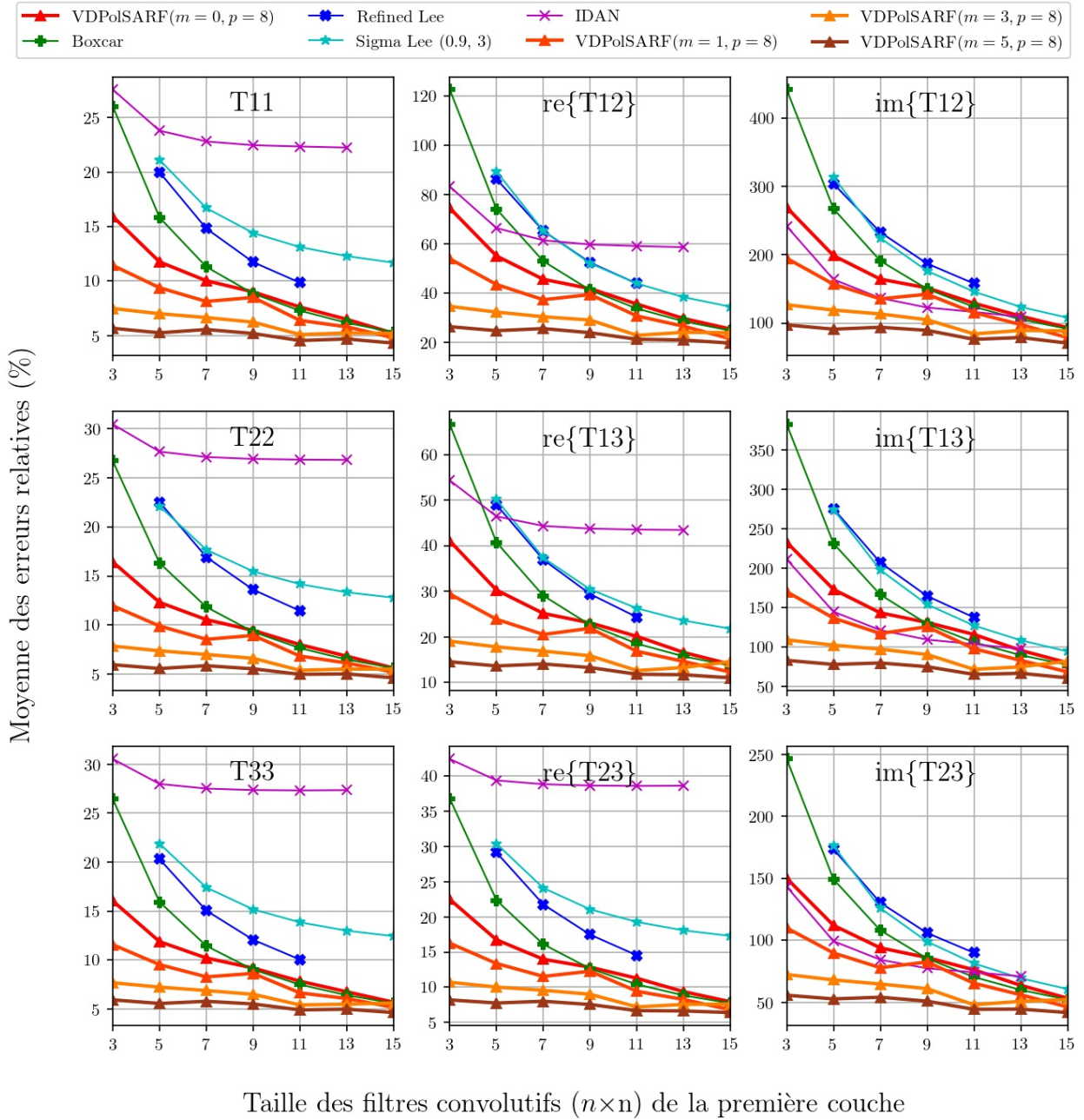
**Fig. A.5. Surfaces de Bragg.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.6. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

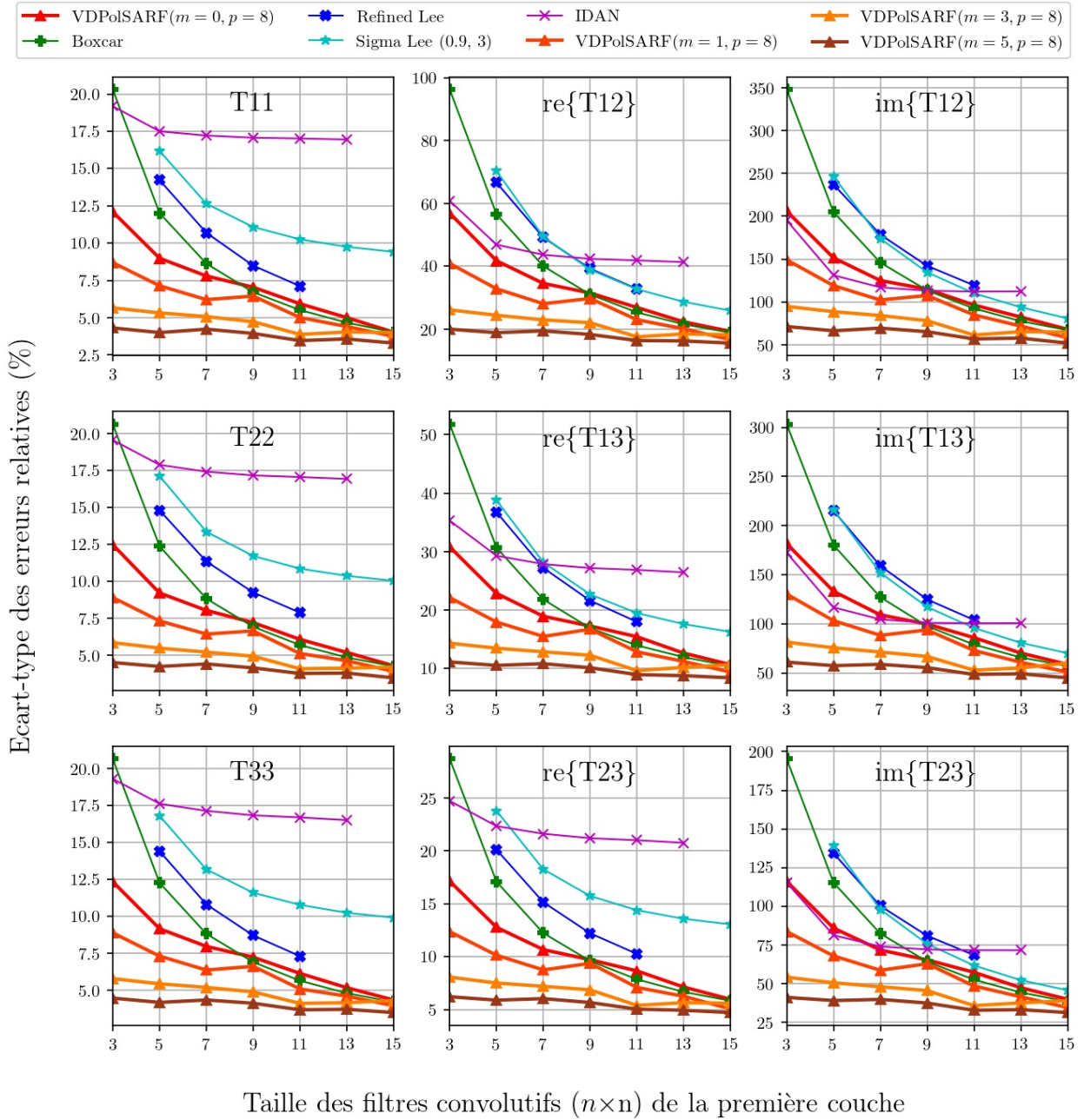
## Double Réflexion (Diffuseurs double bond)



**Fig. A.7. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

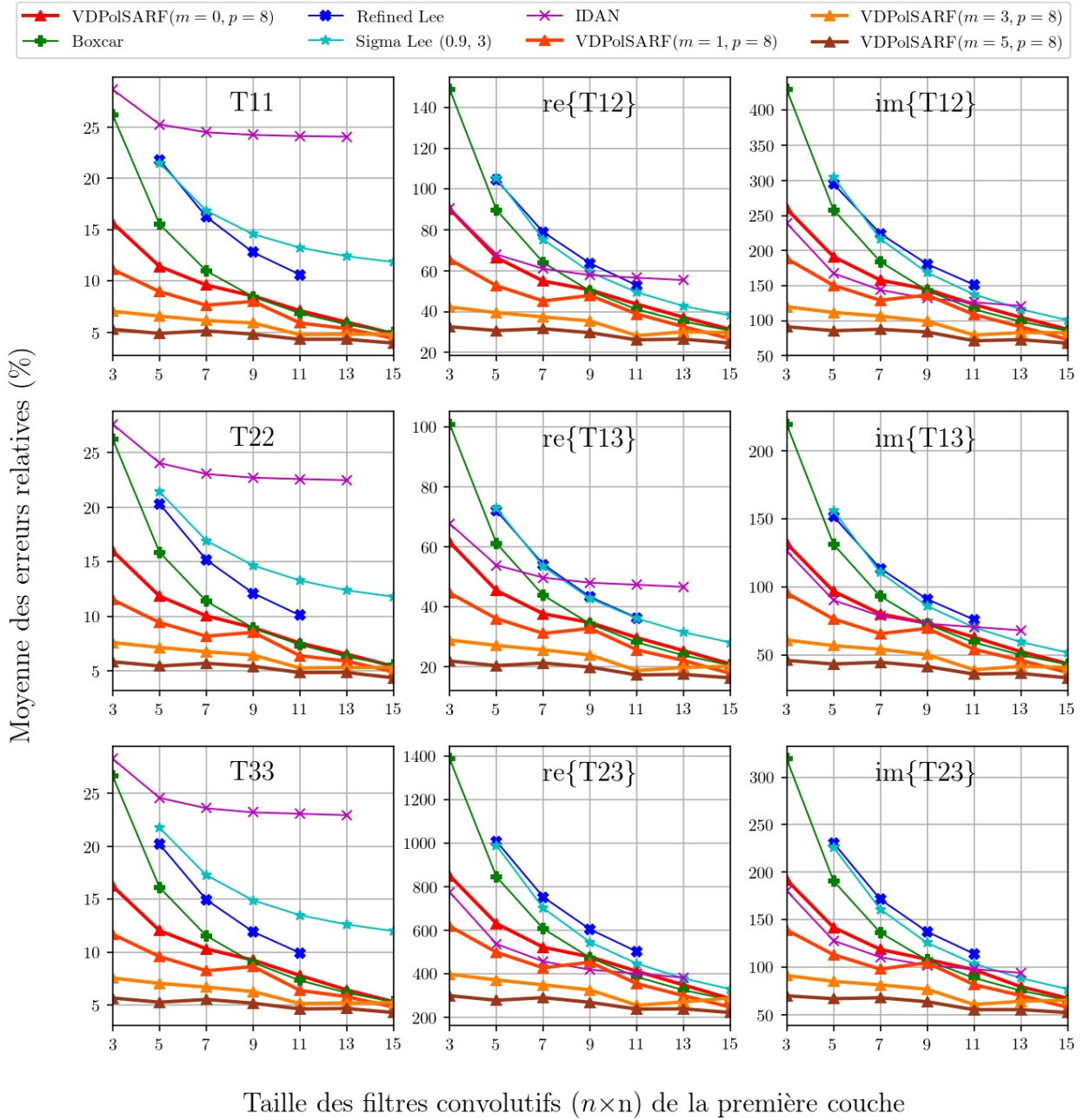


## Double Réflexion (Diffuseurs double bond)



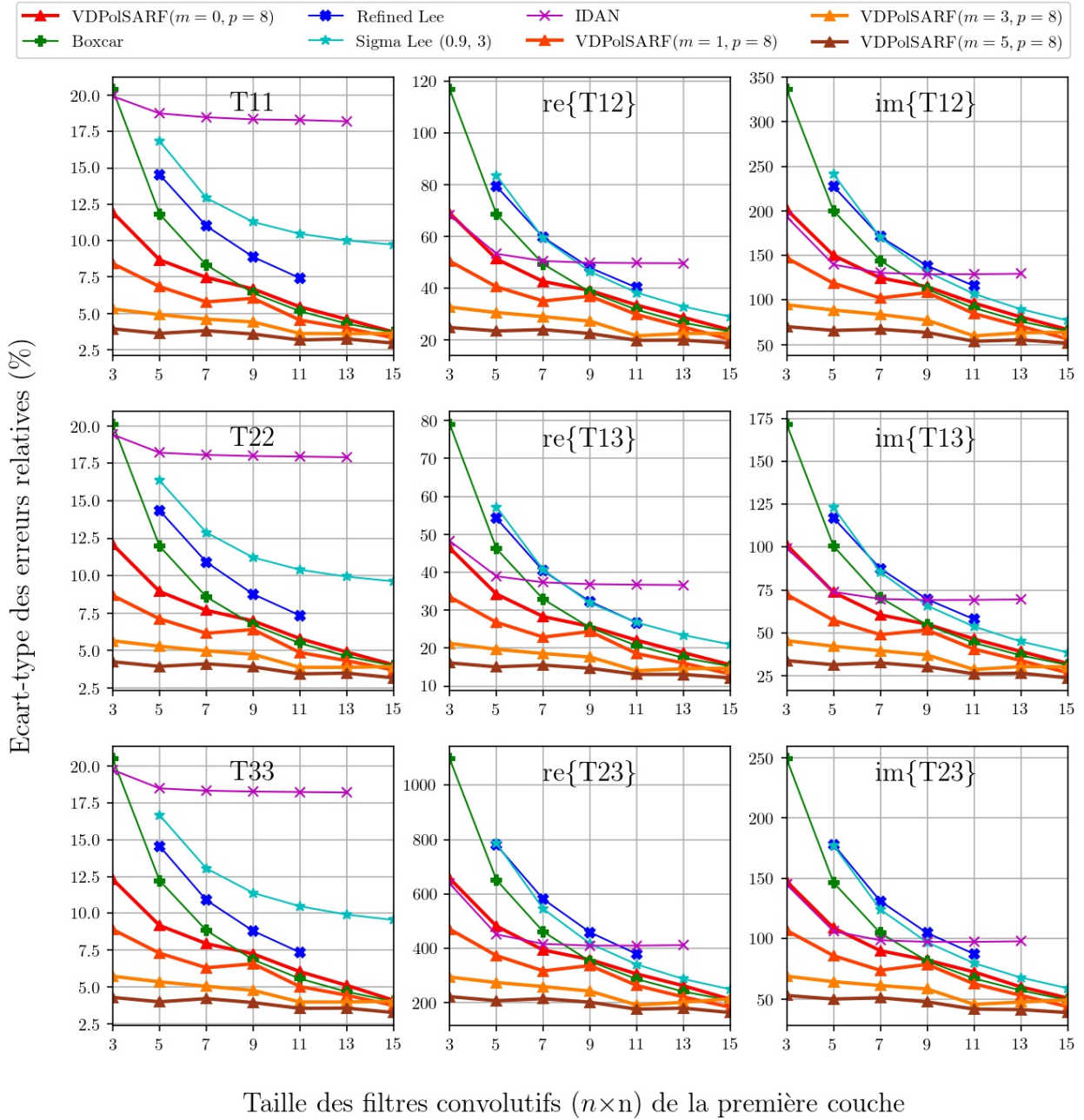
**Fig. A.8. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Particules Anisotropiques (Diffuseurs volumétriques)



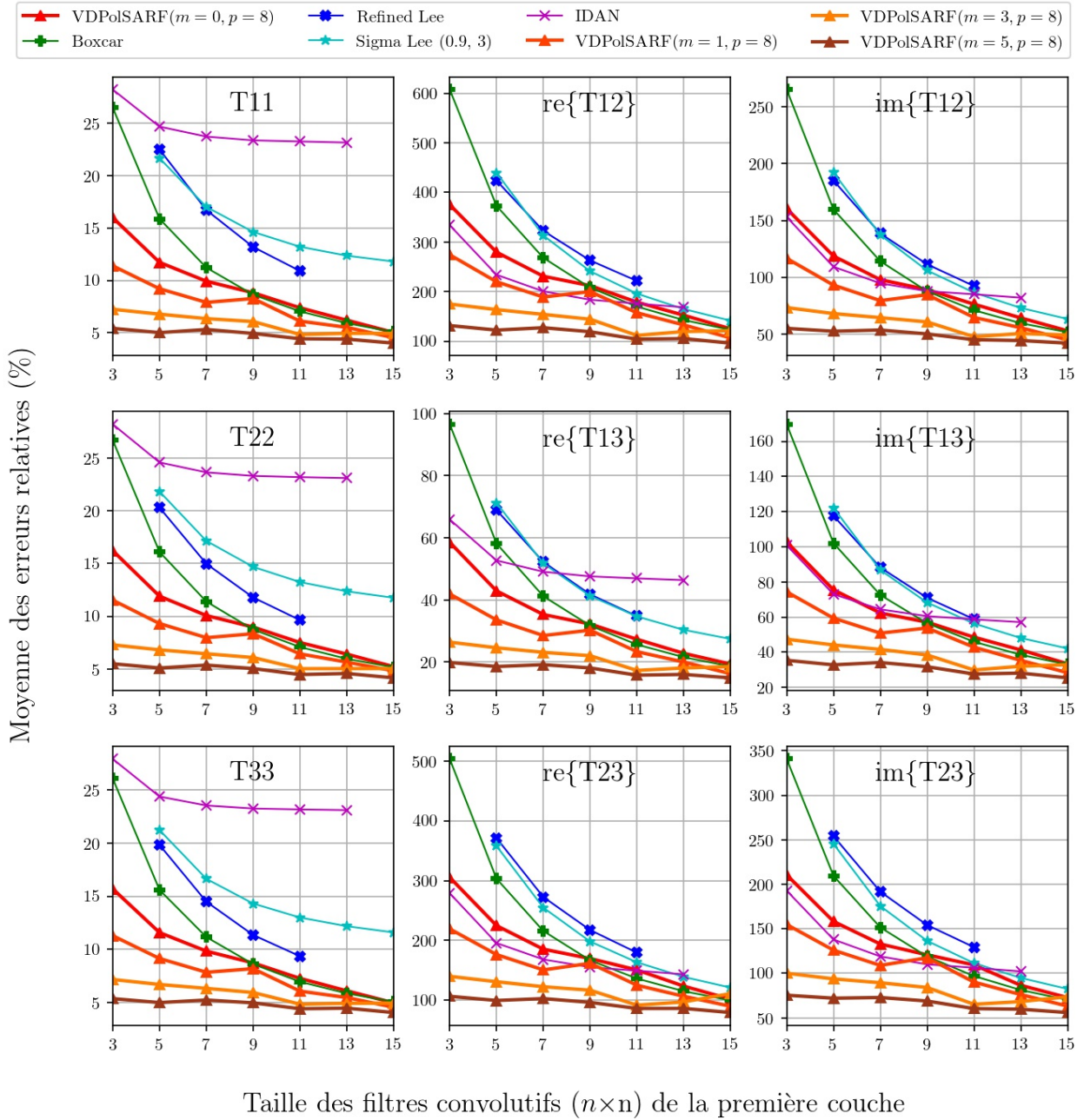
**Fig. A.9. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

Particules Anisotropiques (Diffuseurs volumétriques)



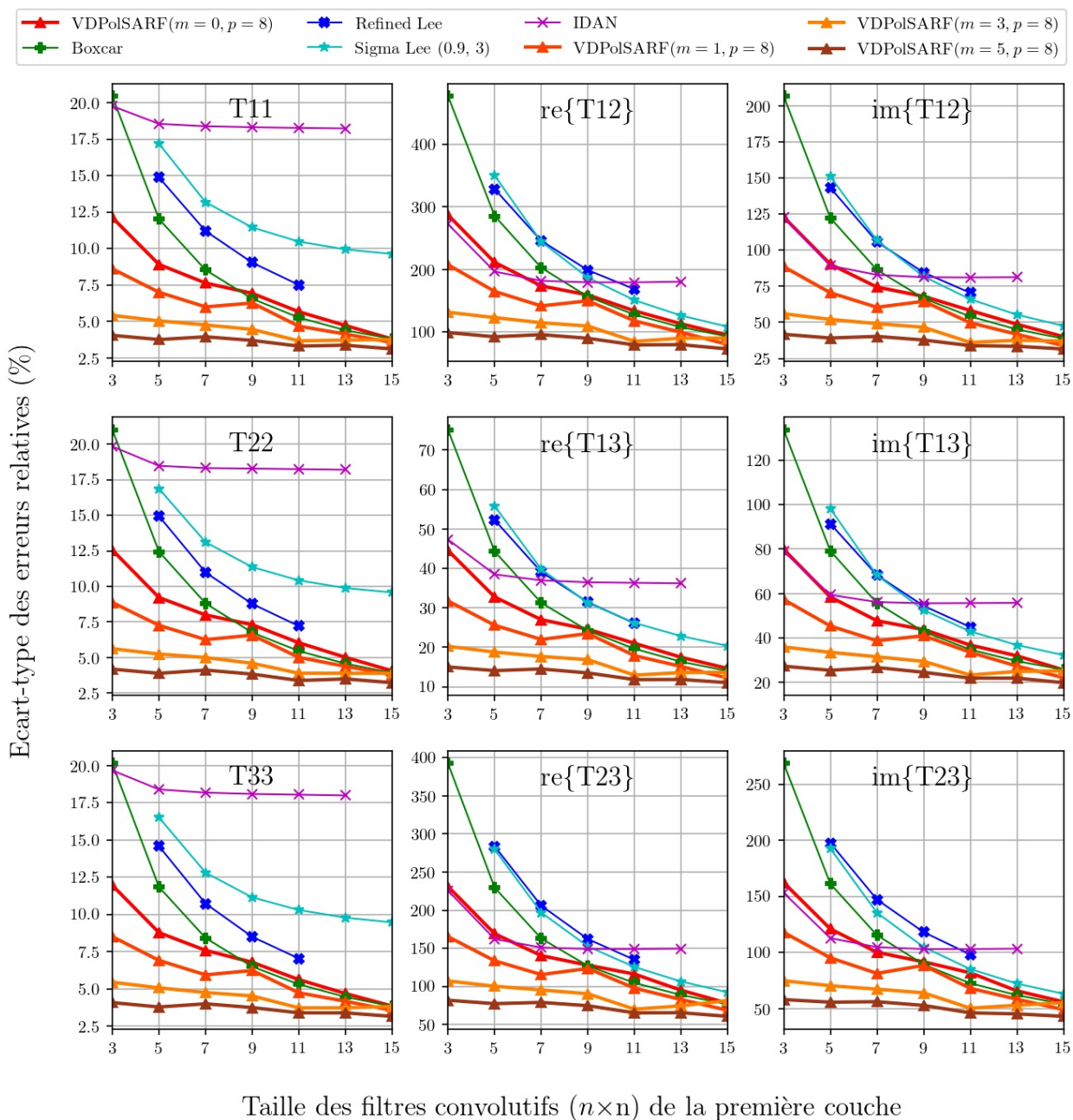
**Fig. A.10. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



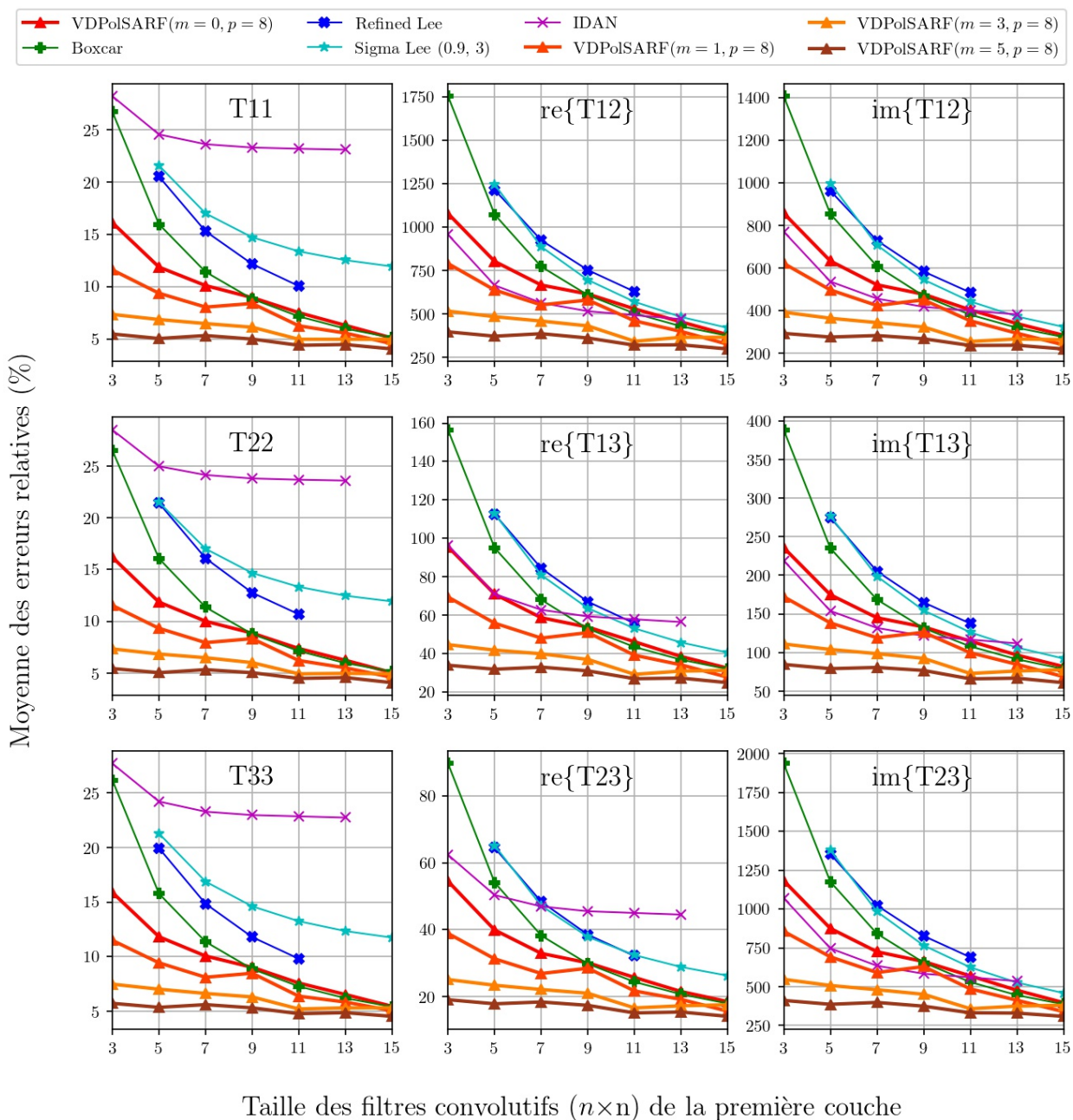
**Fig. A.11. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



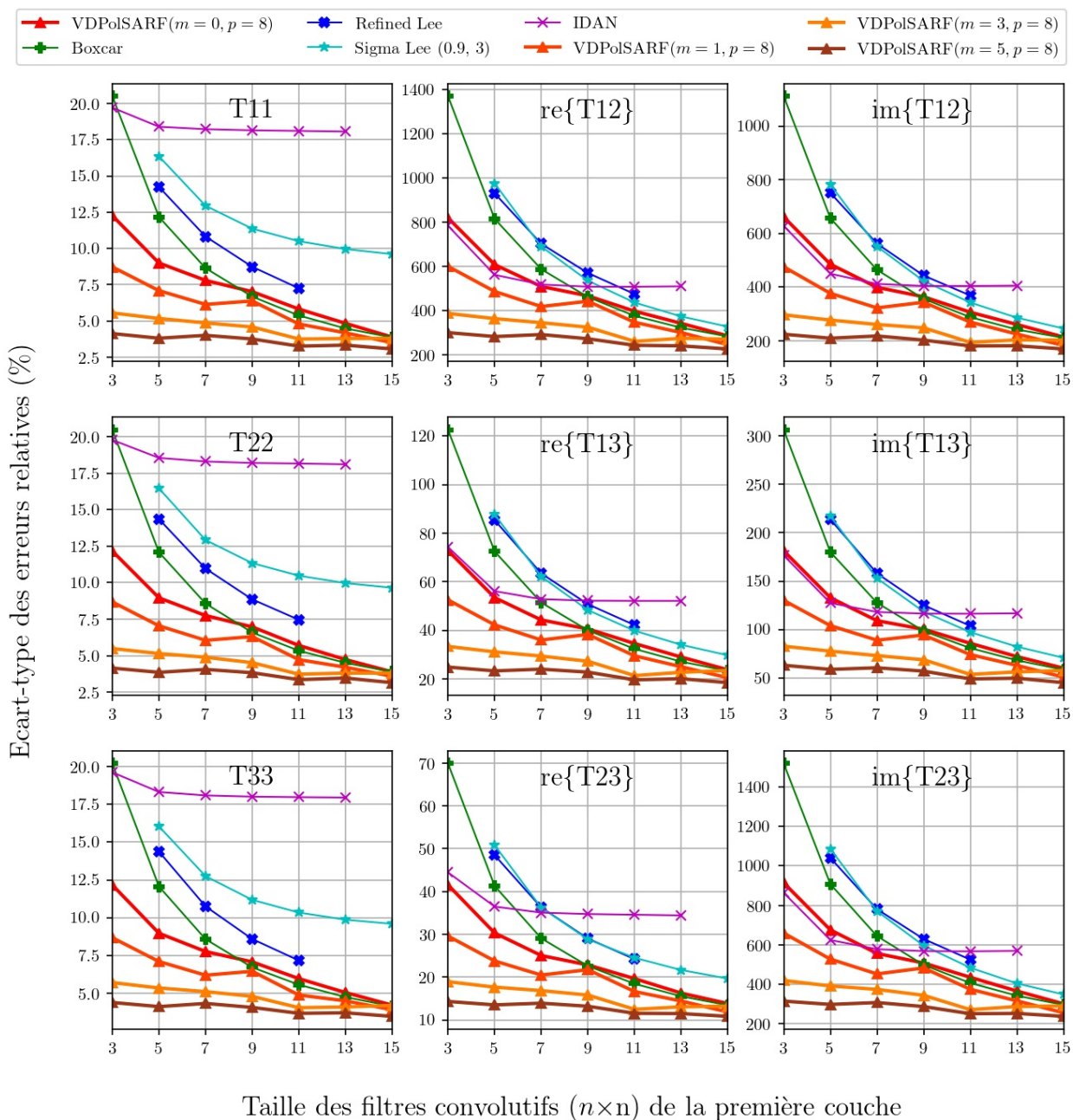
**Fig. A.12. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

## Structures Complexes



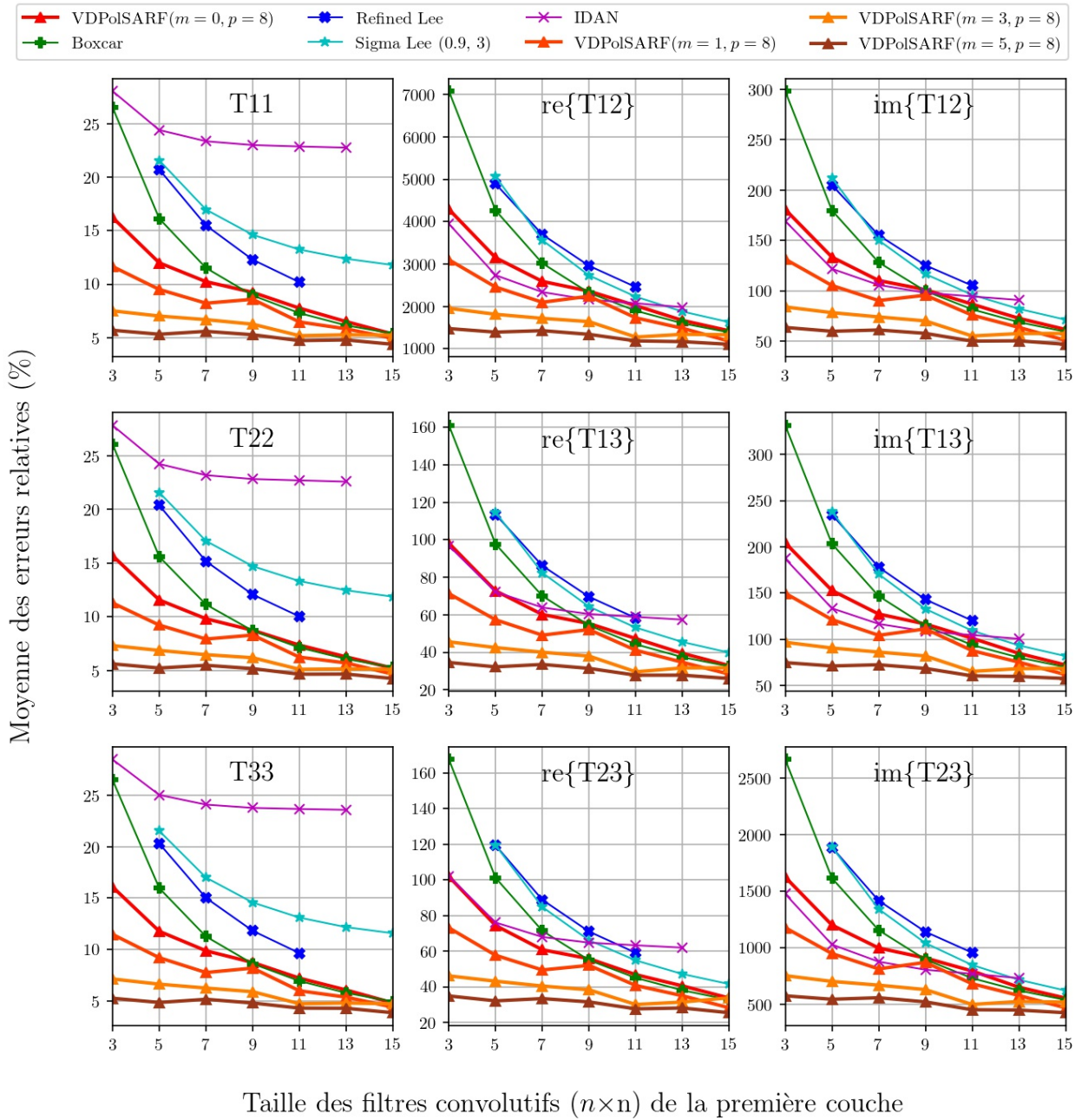
**Fig. A.13. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Structures Complexes



**Fig. A.14. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

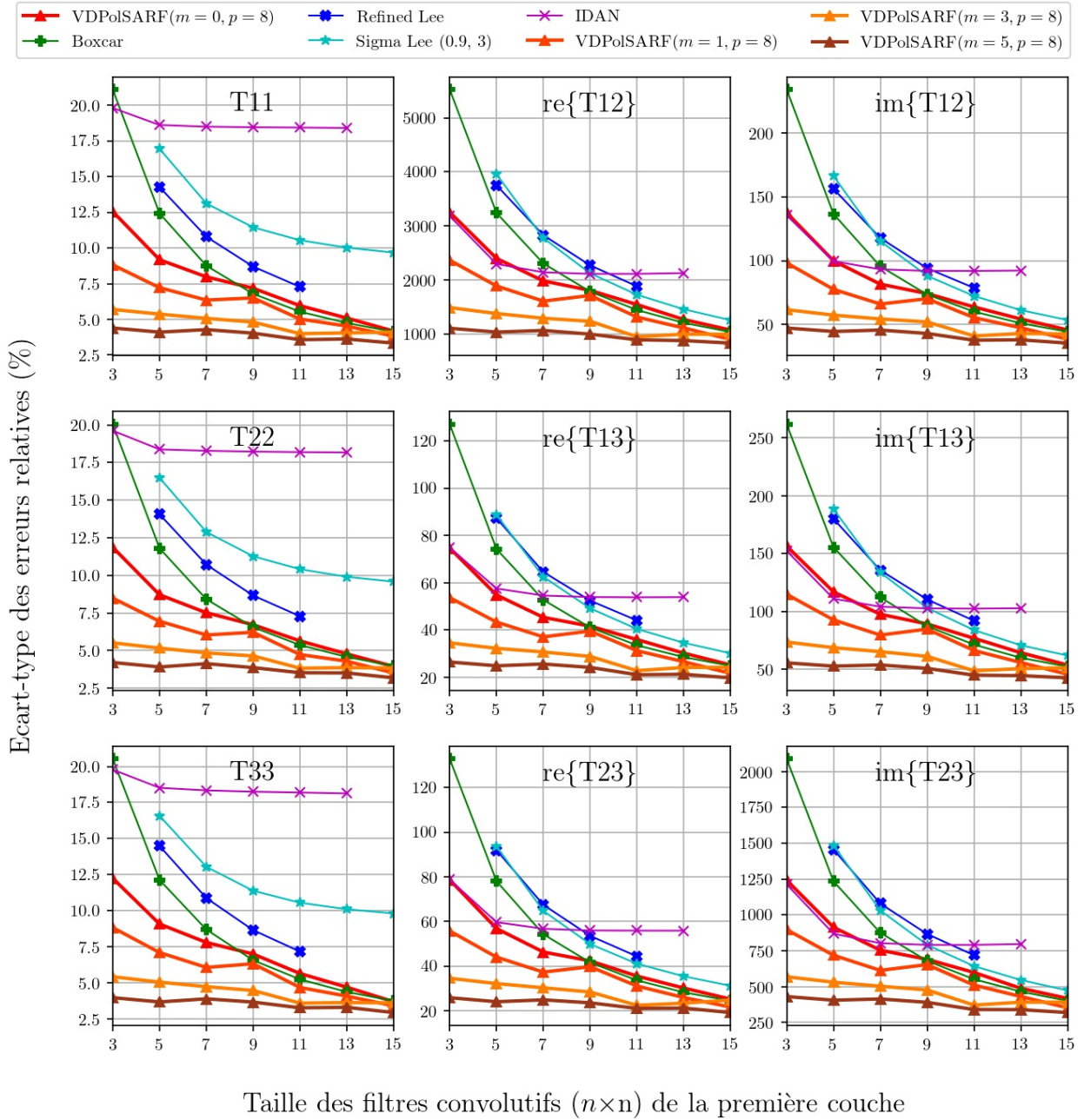
## Diffuseurs Anisotropiques aléatoires



**Fig. A.15. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .



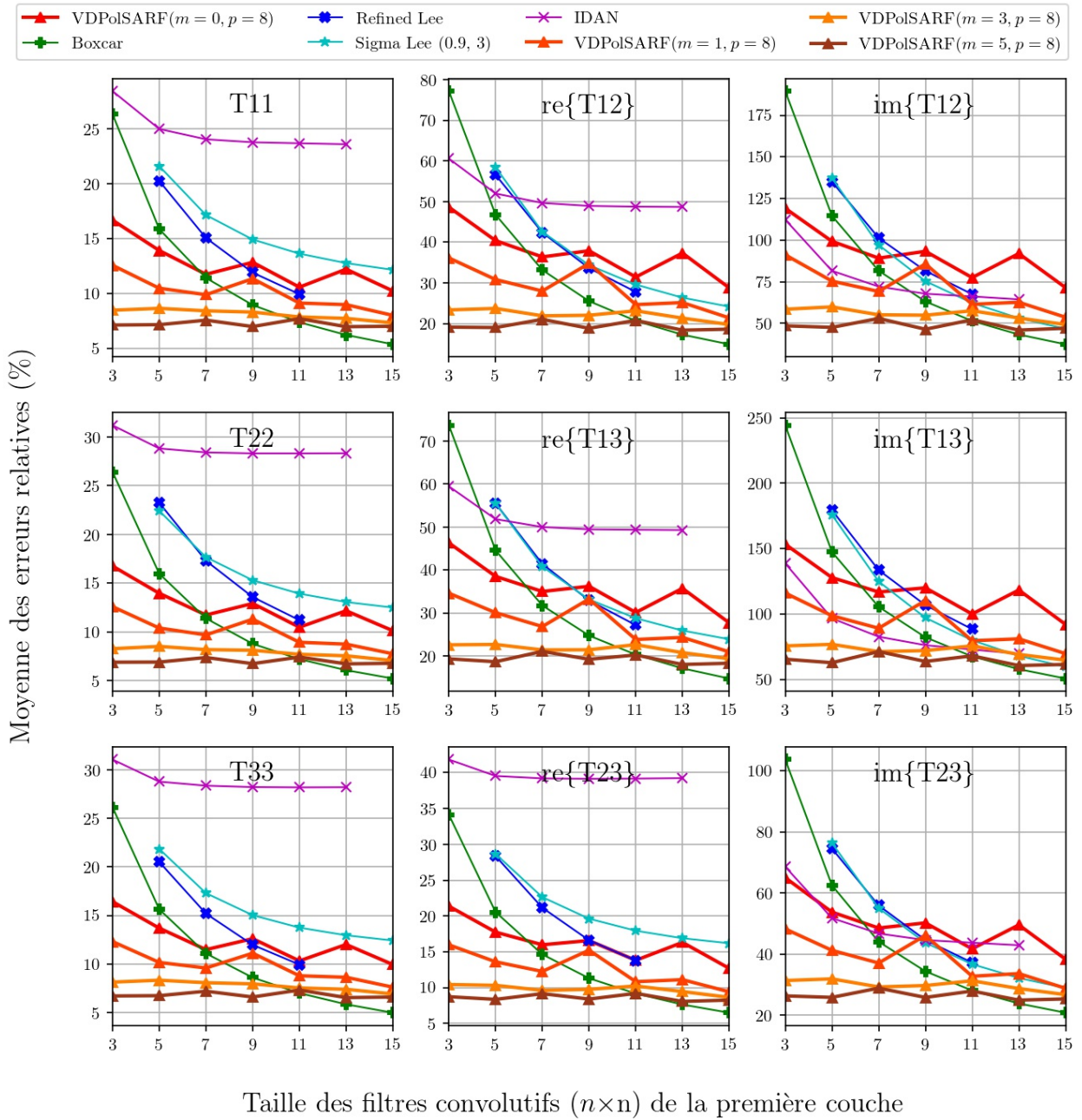
## Diffuseurs Anisotropiques aléatoires



**Fig. A.16. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

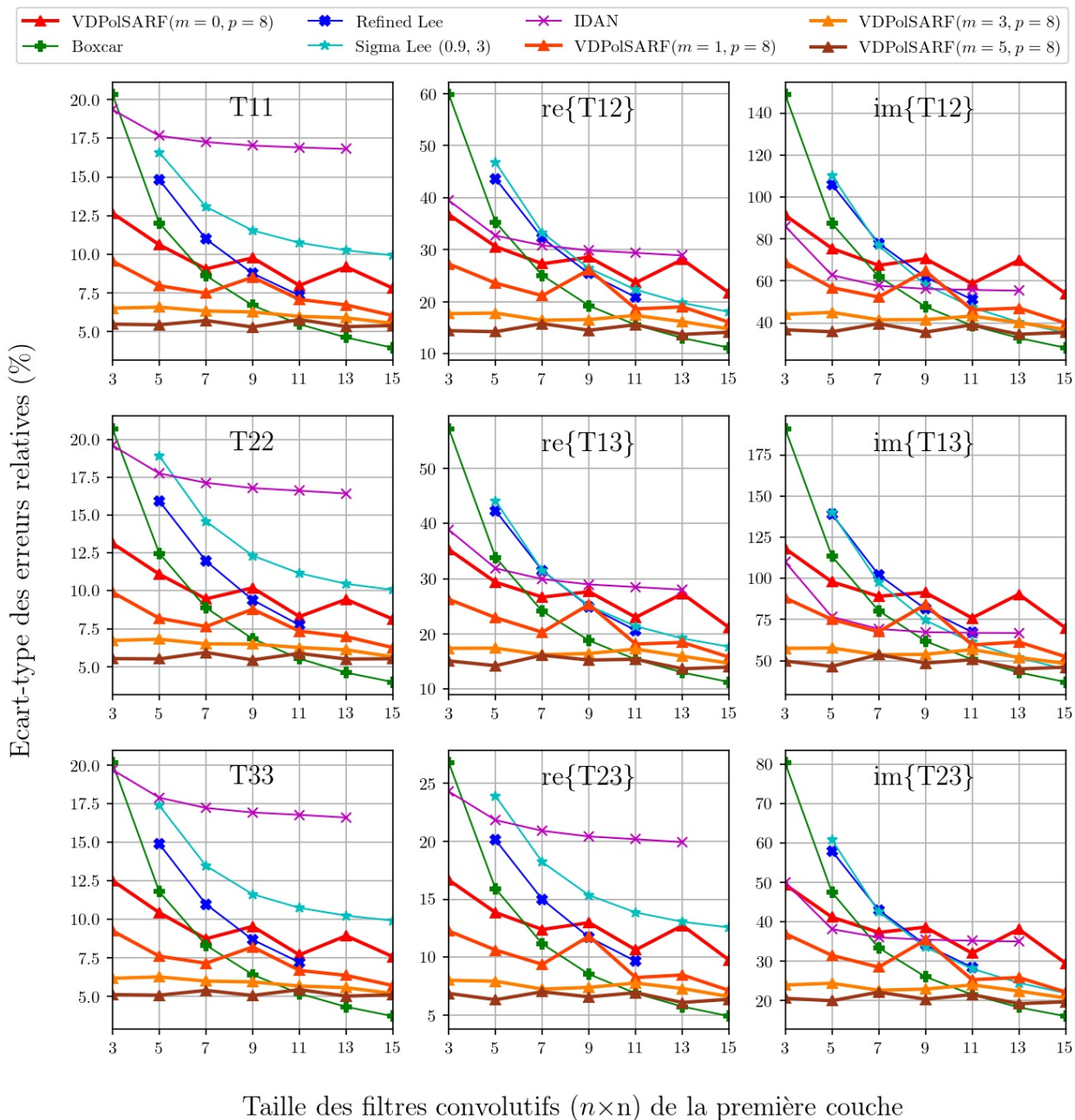
**A.2. Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes.**

## Réflexions diédrales



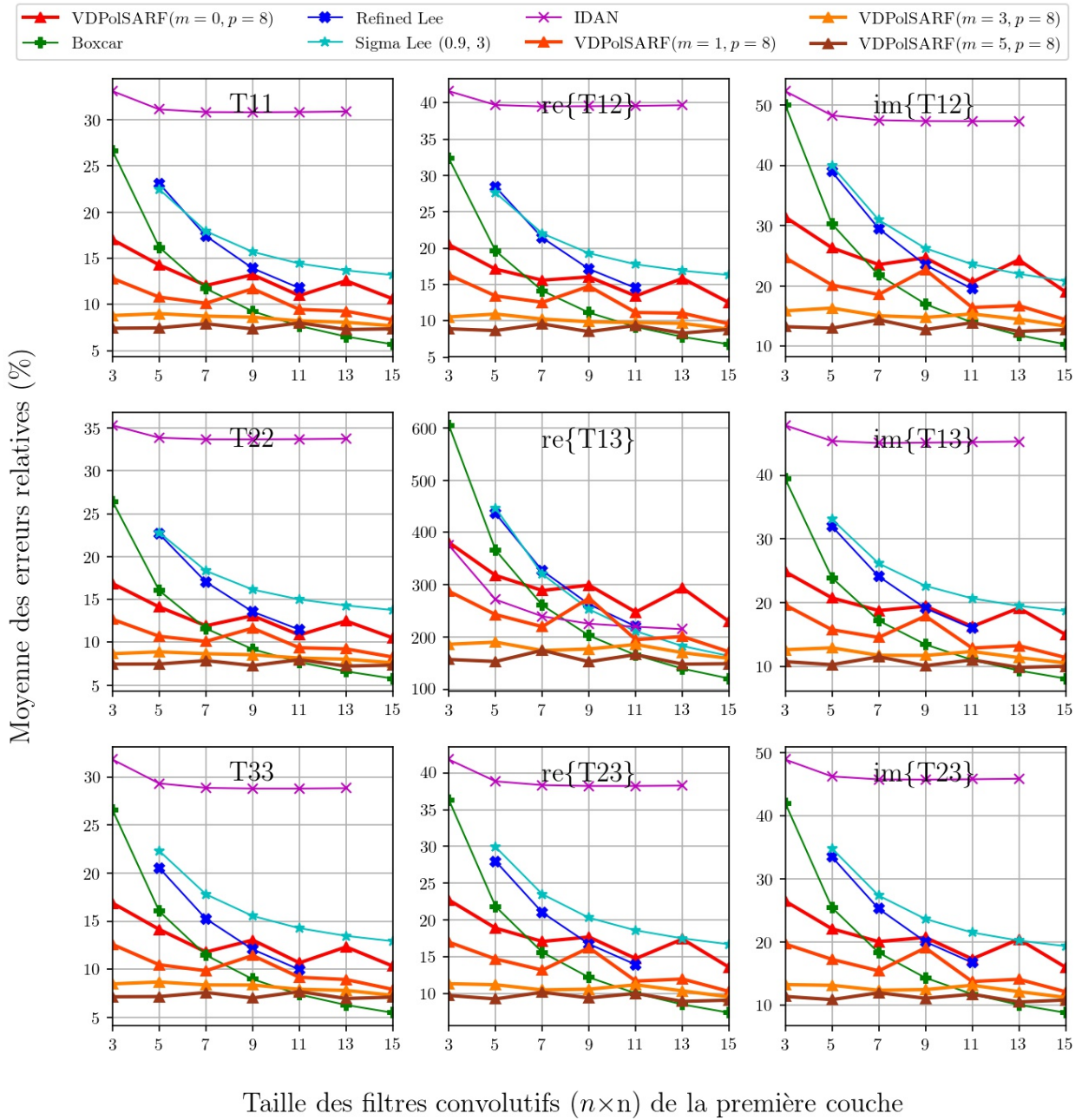
**Fig. A.17. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



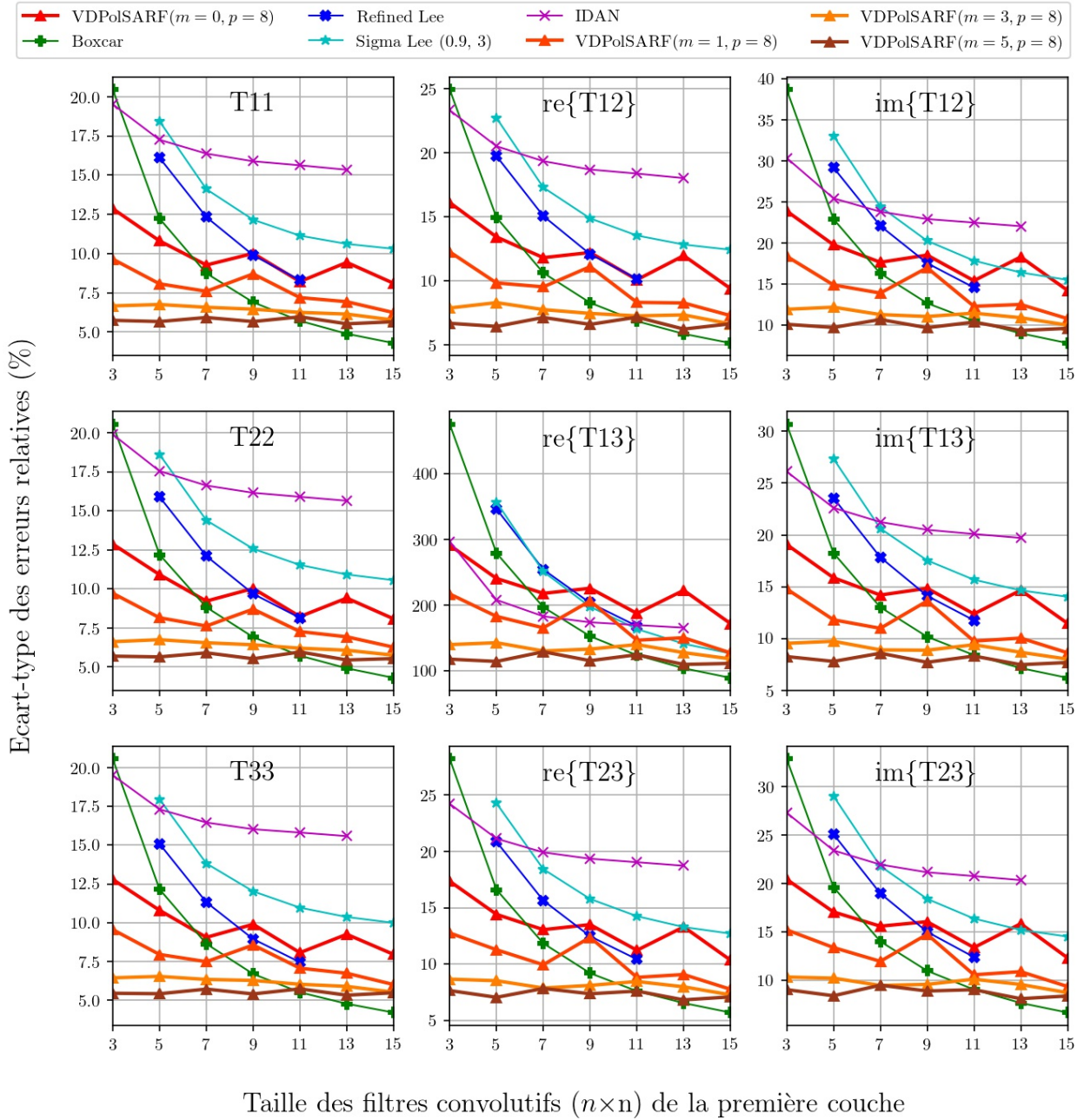
**Fig. A.18. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



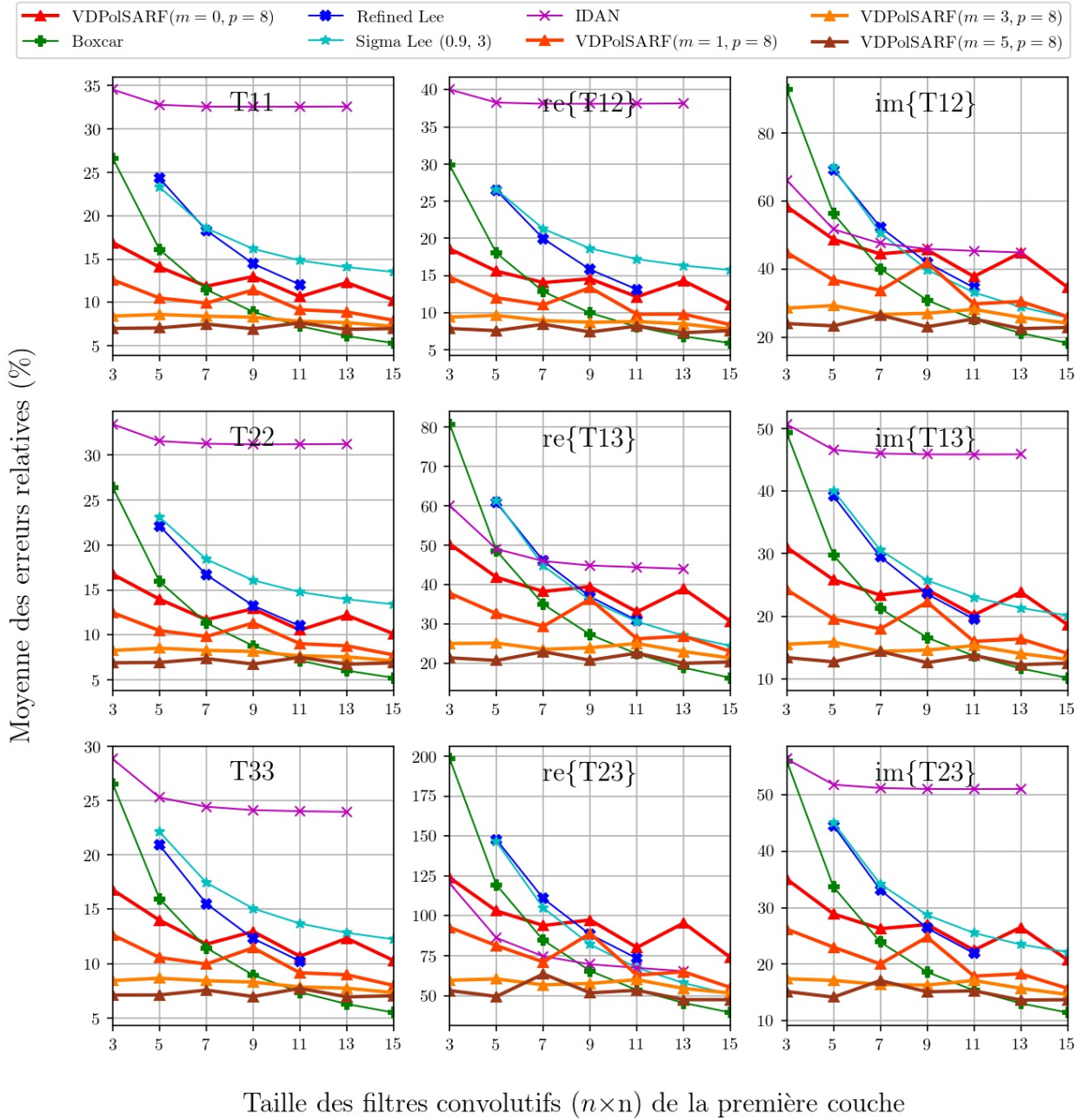
**Fig. A.19. Dipôles** . Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



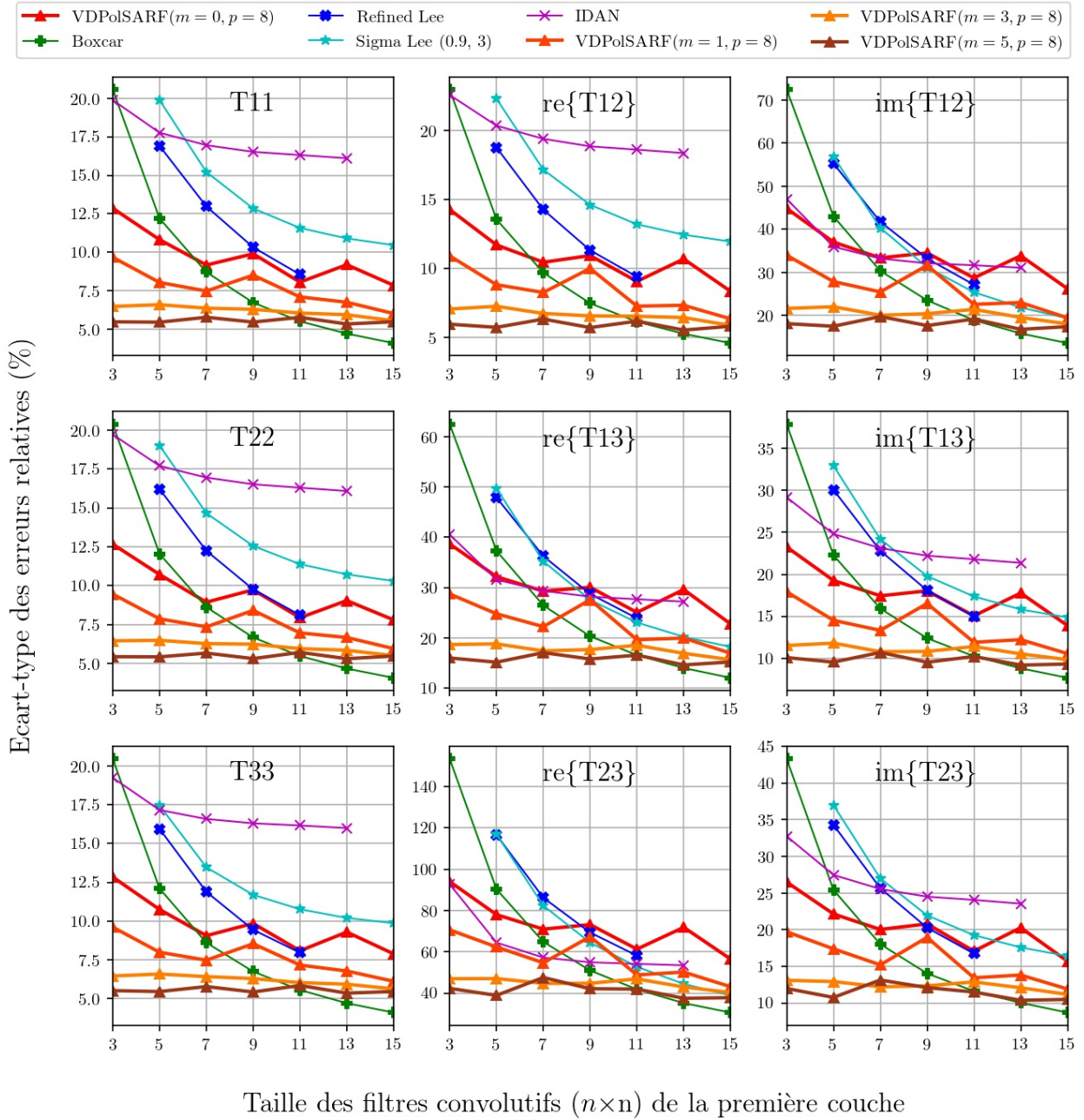
**Fig. A.20. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.21. Surfaces de Bragg.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

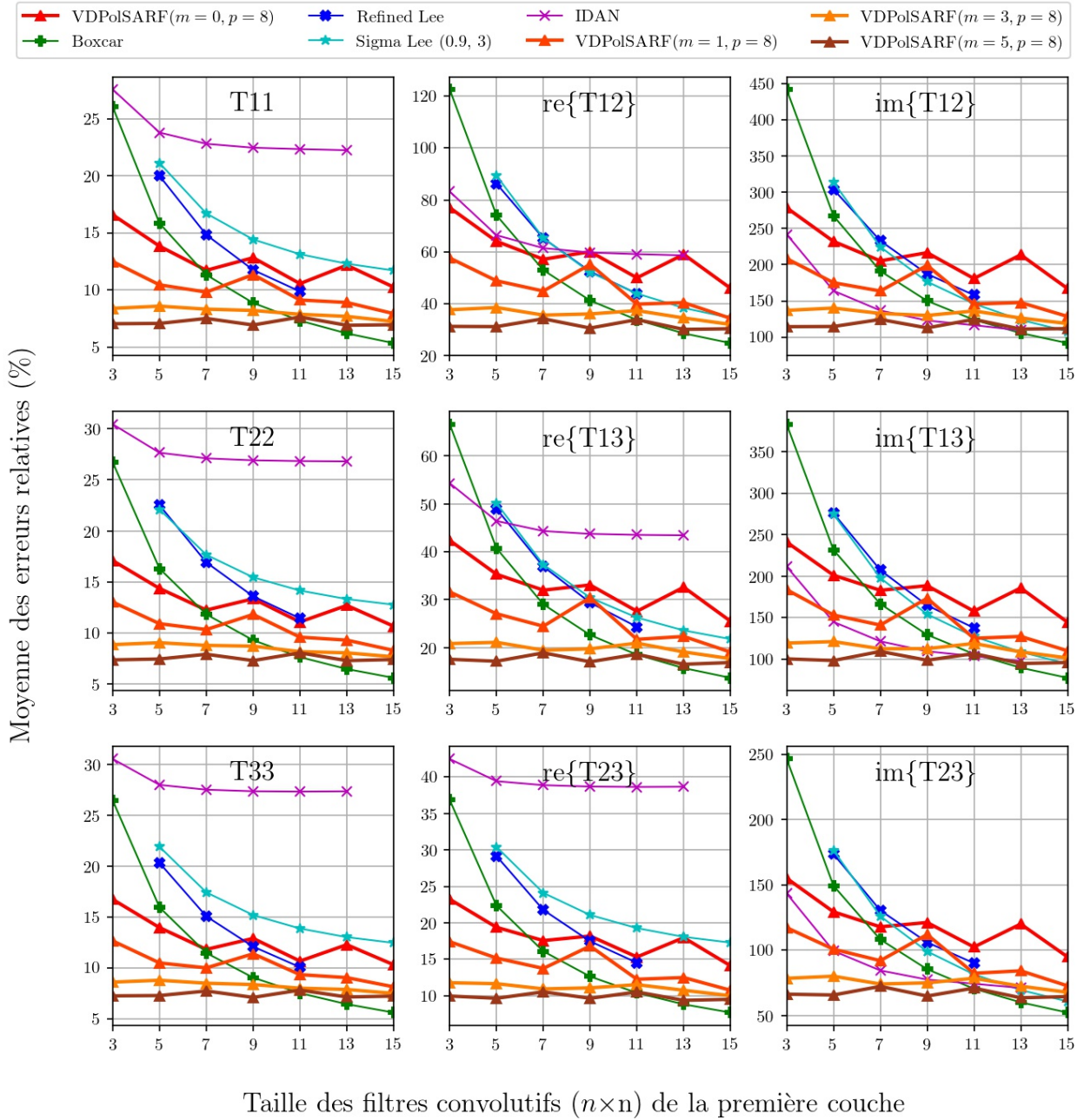
### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.22. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

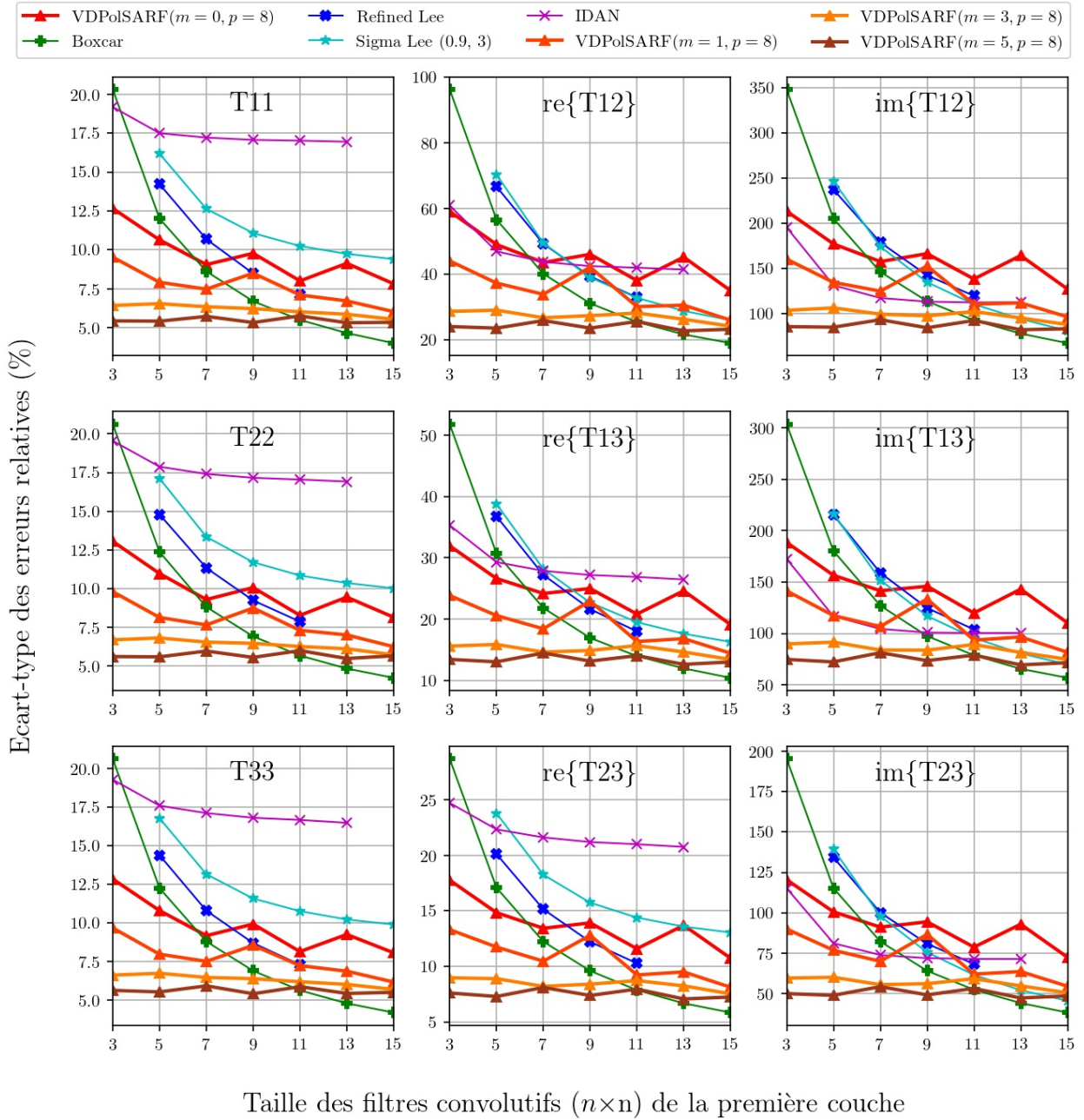


## Double Réflexion (Diffuseurs double bond)



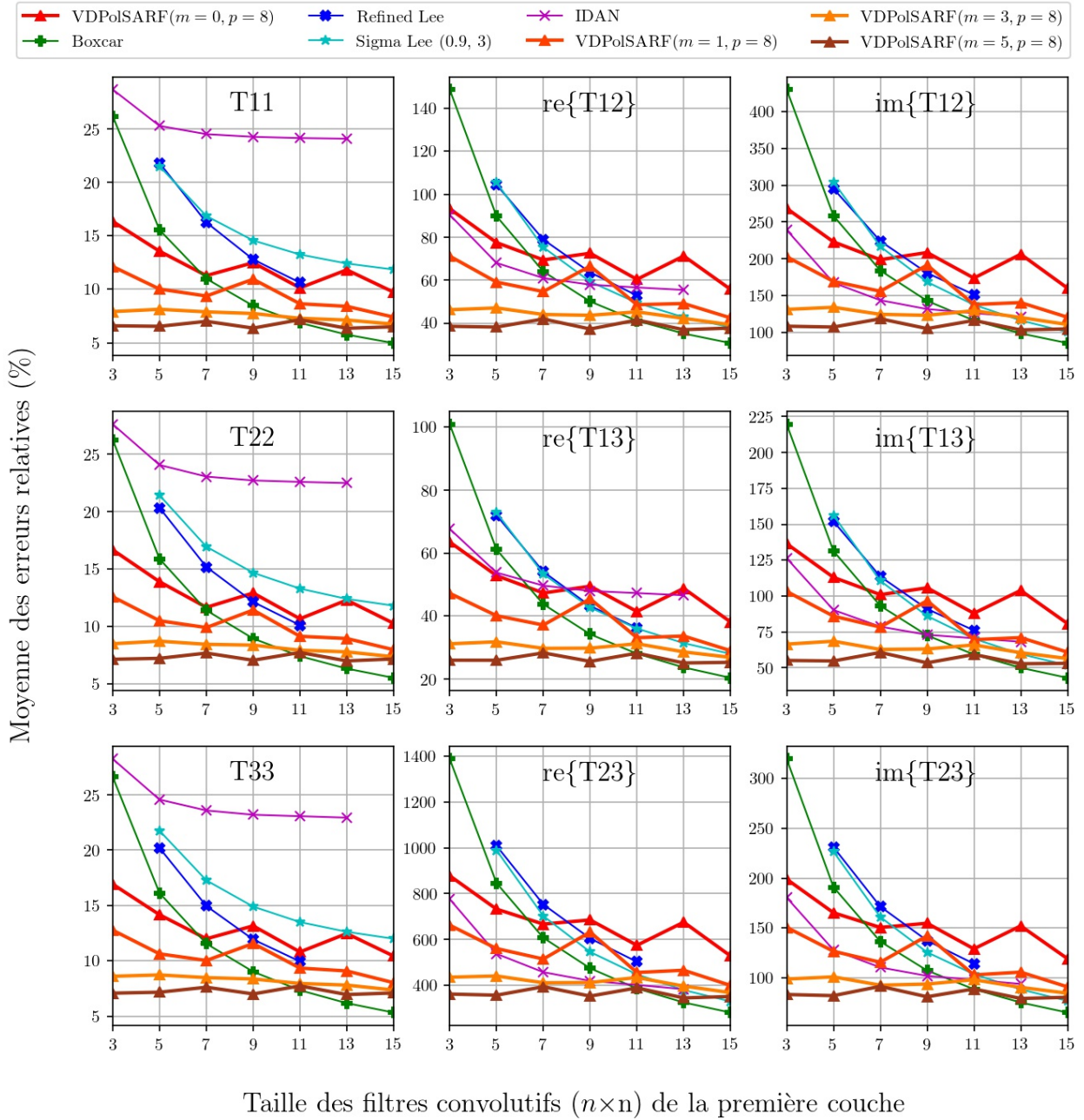
**Fig. A.23. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Double Réflexion (Diffuseurs double bond)



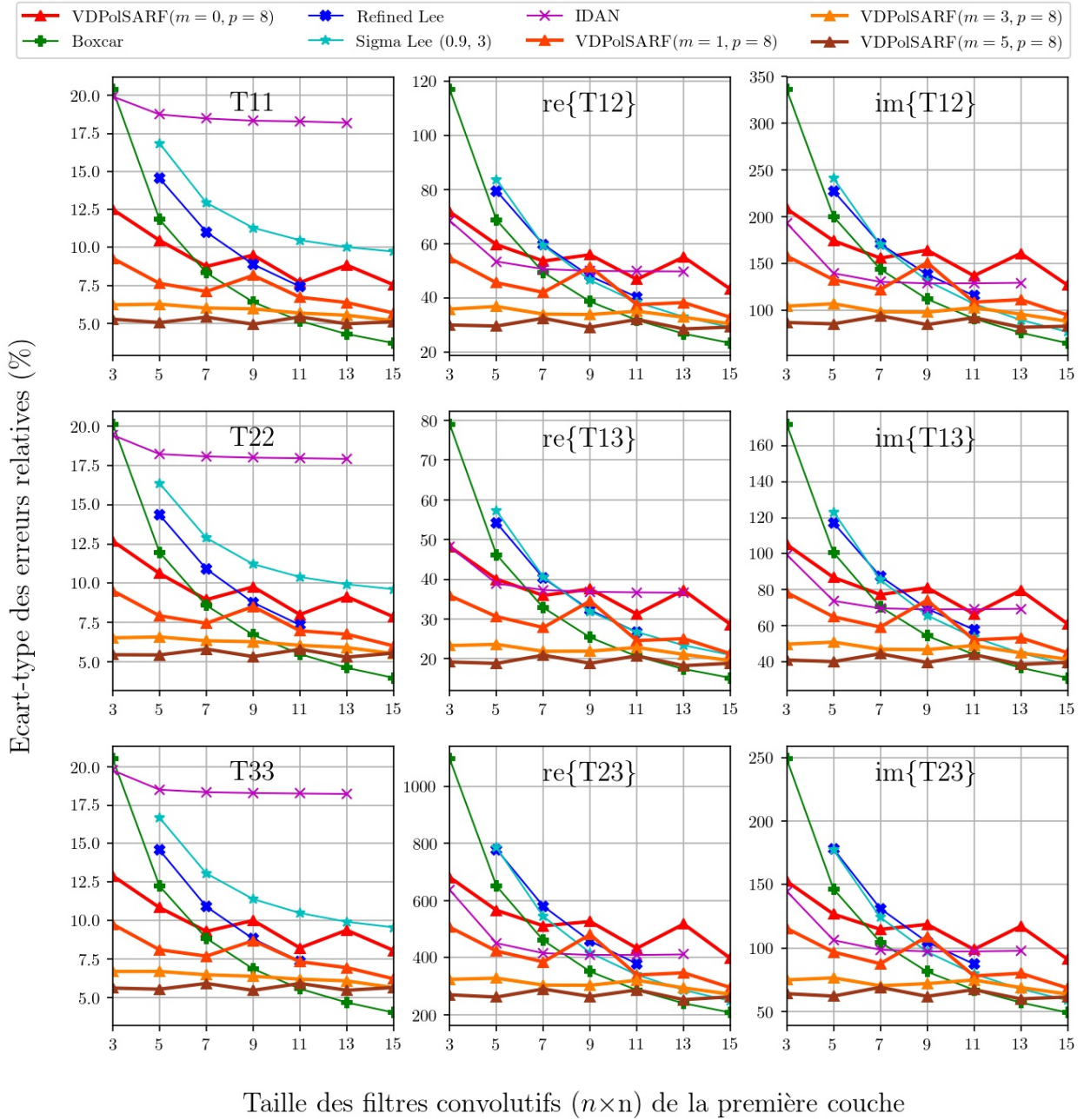
**Fig. A.24. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Particules Anisotropiques (Diffuseurs volumétriques)



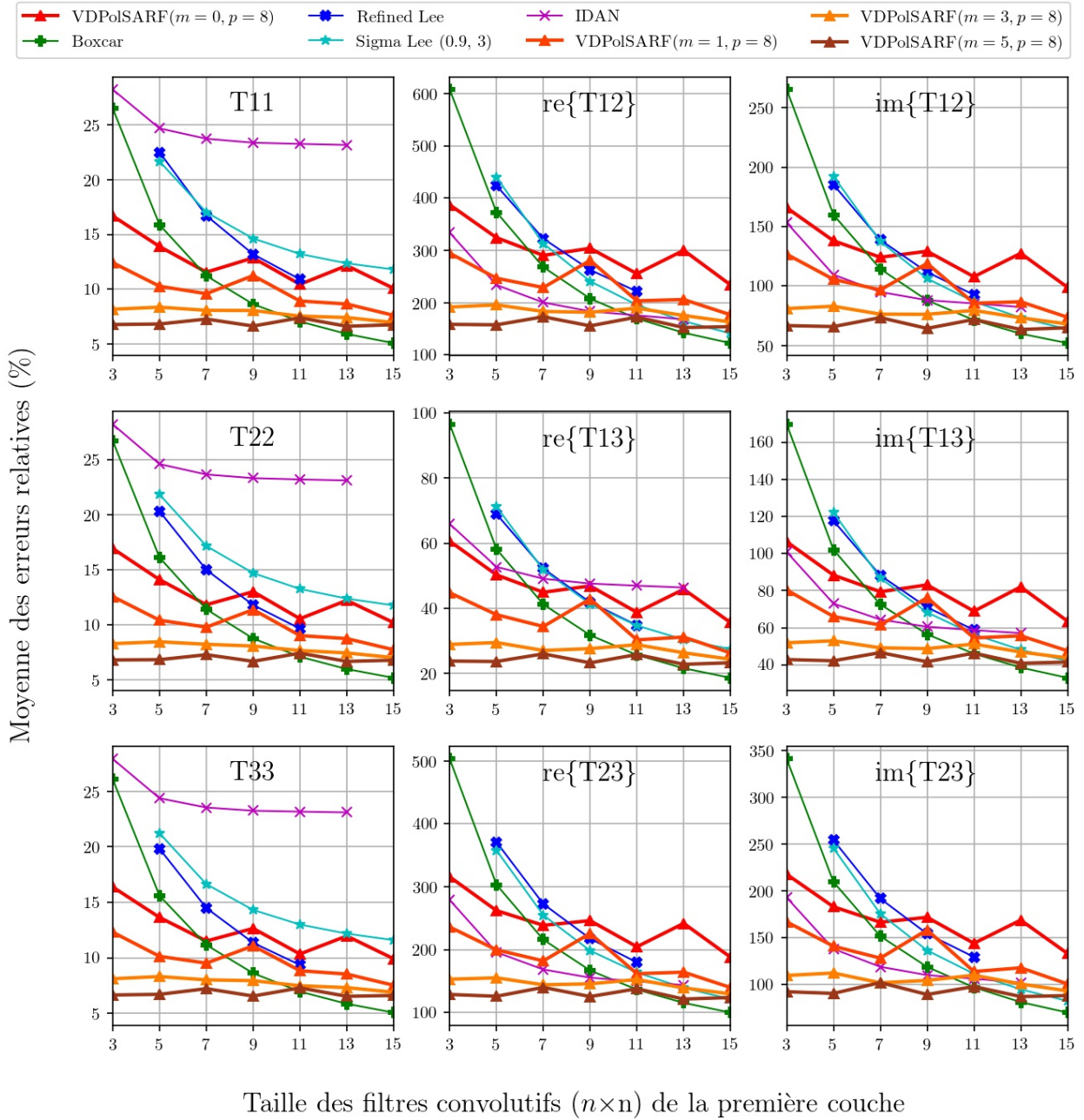
**Fig. A.25. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

Particules Anisotropiques (Diffuseurs volumétriques)



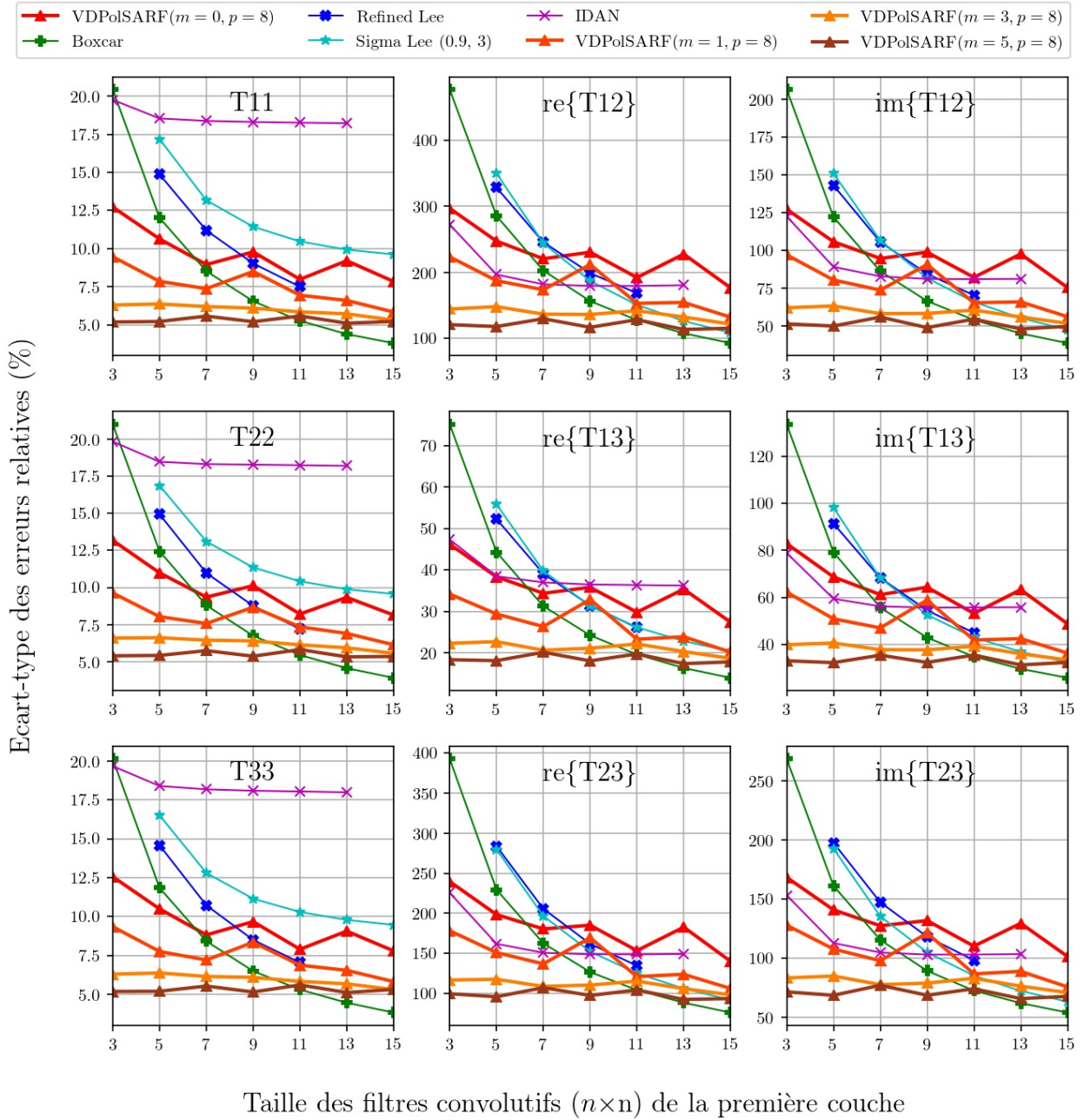
**Fig. A.26. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



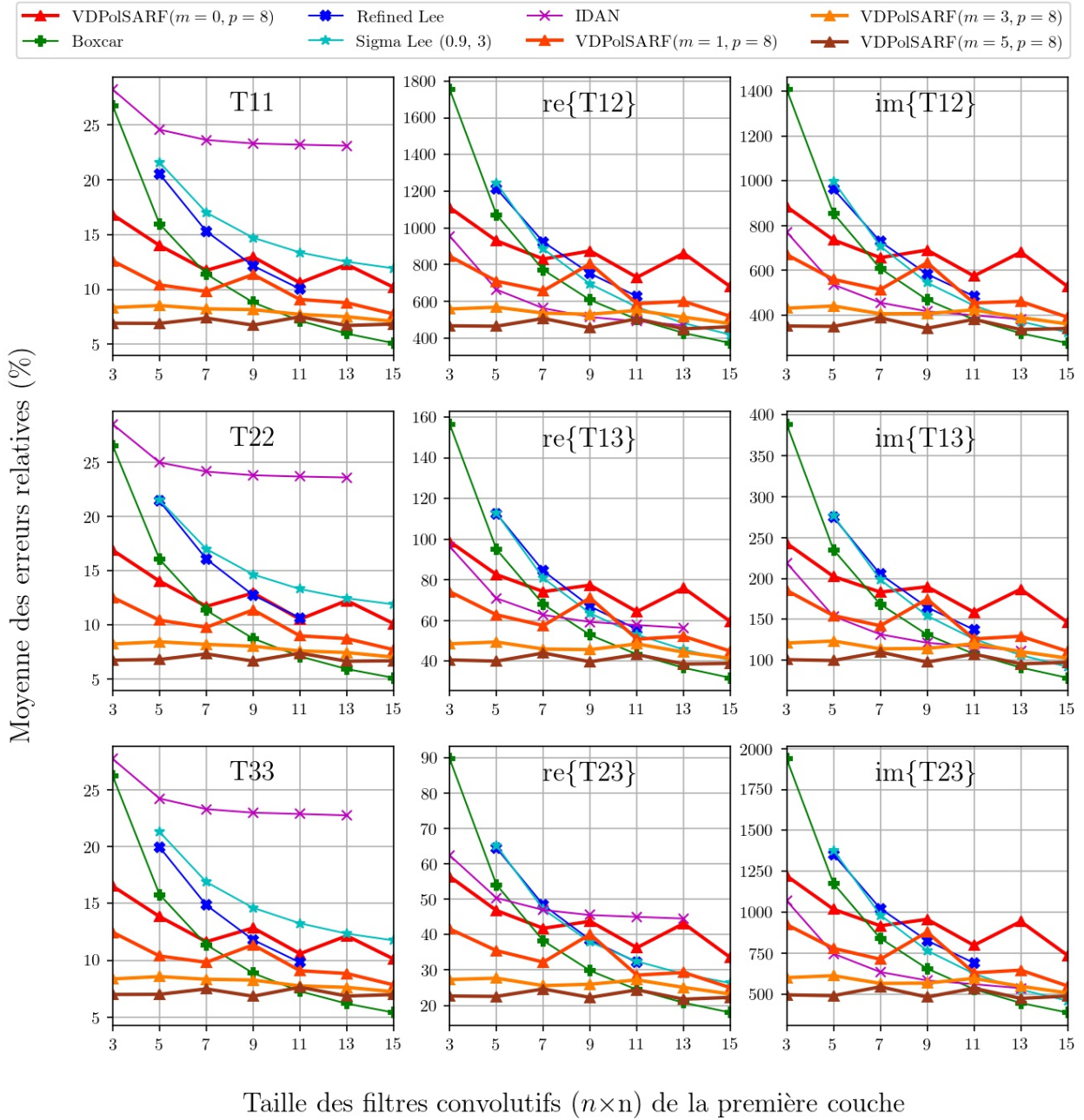
**Fig. A.27. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



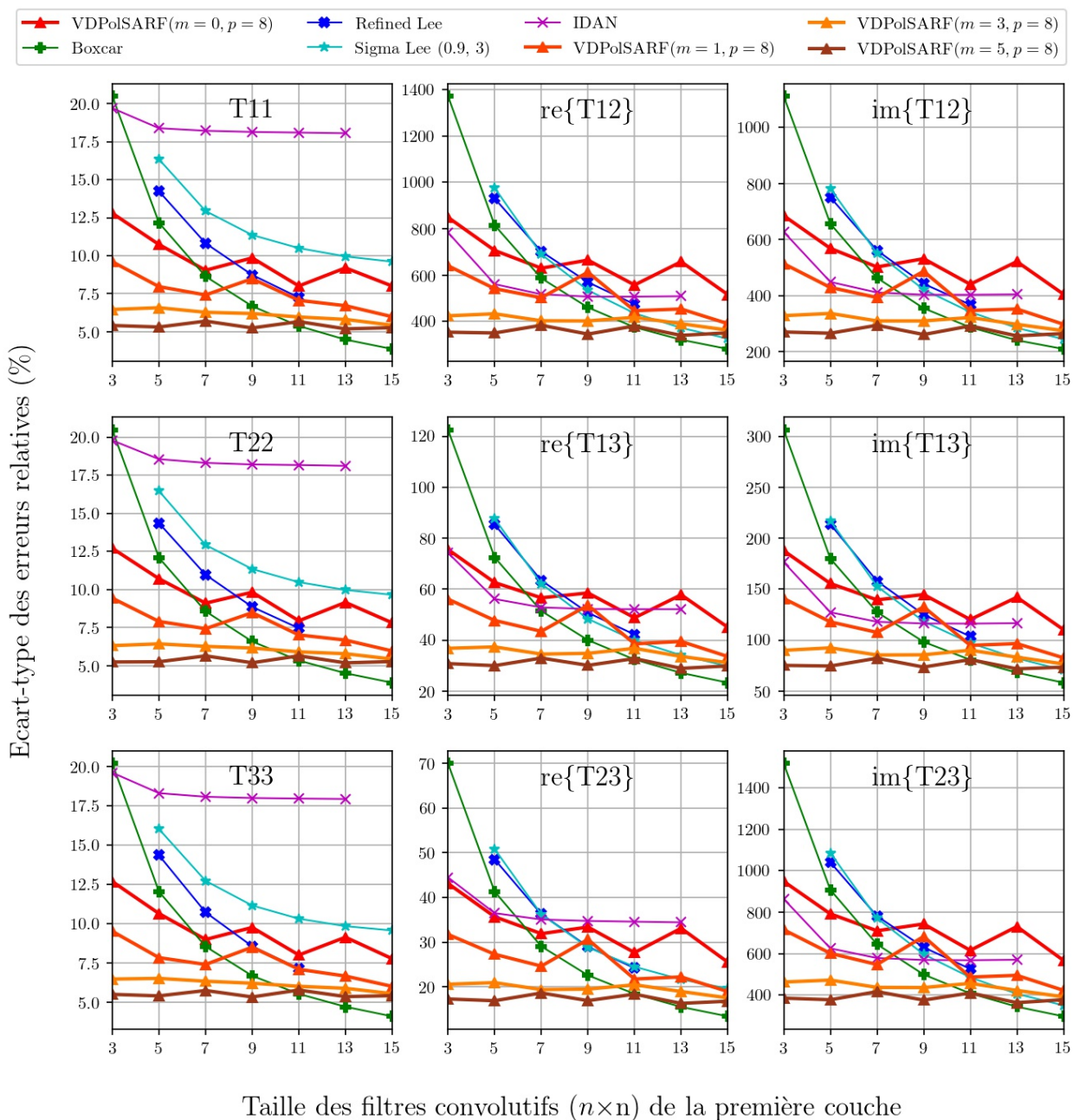
**Fig. A.28. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Structures Complexes



**Fig. A.29. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

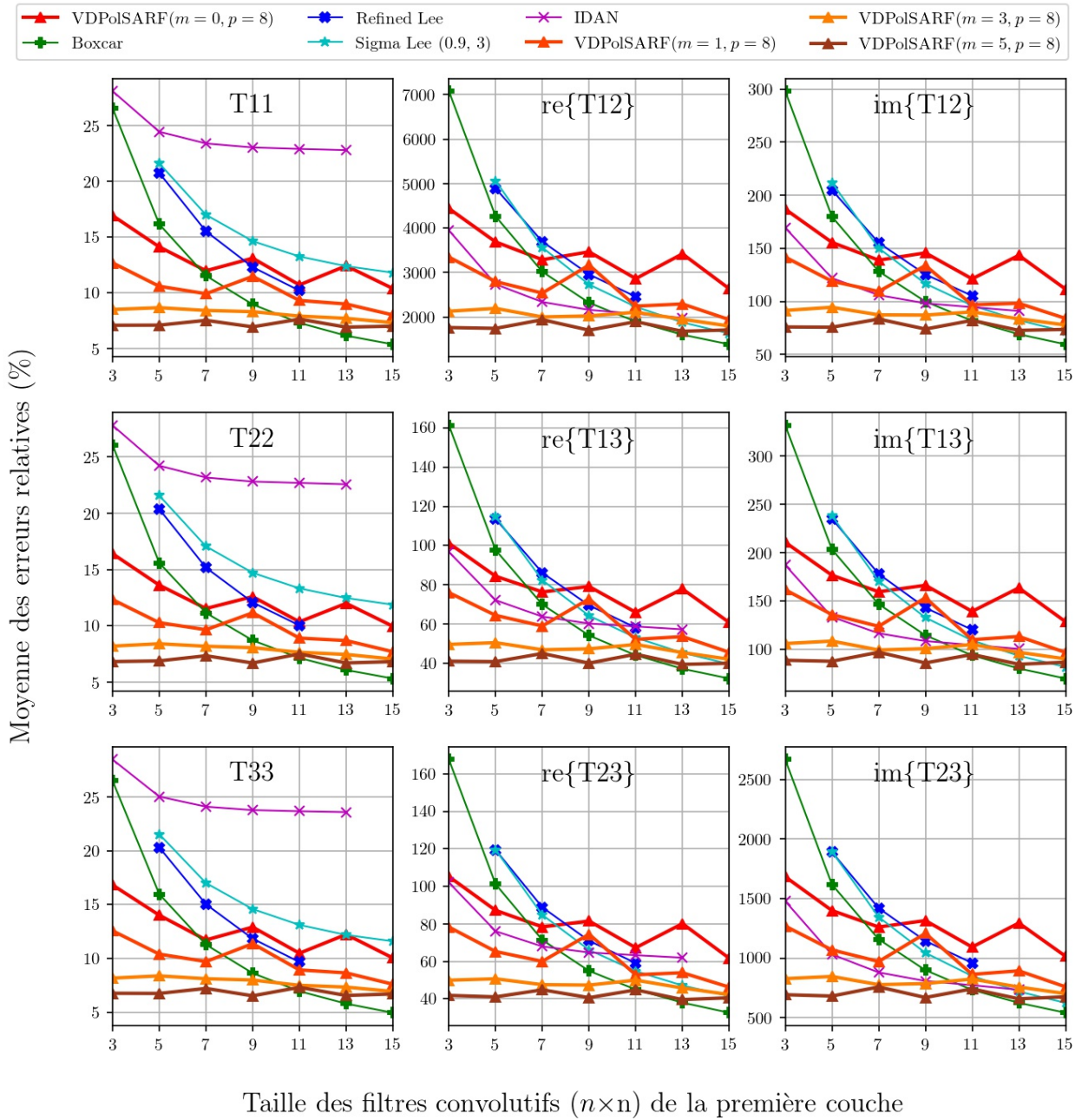
## Structures Complexes



**Fig. A.30. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

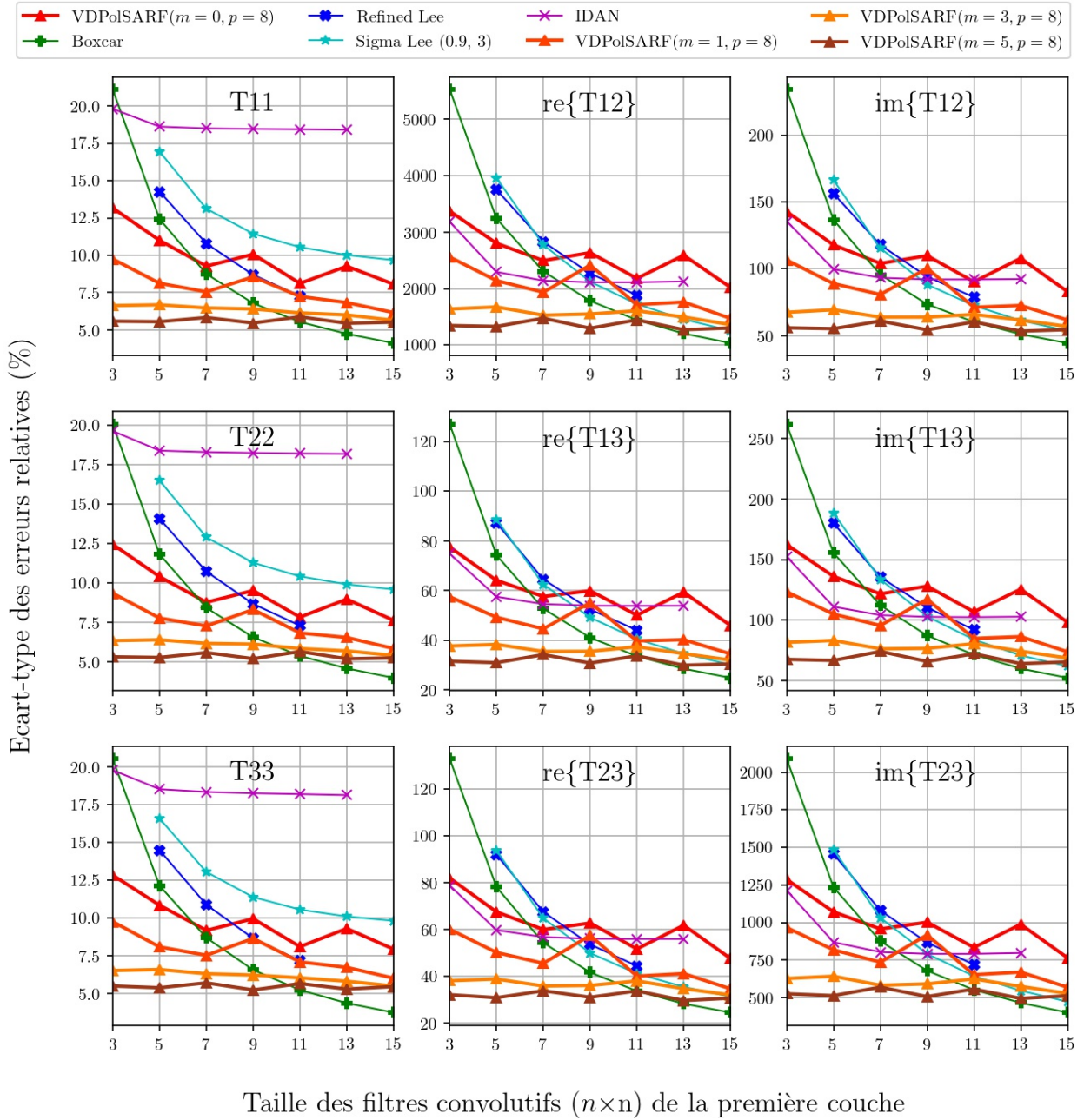


## Diffuseurs Anisotropiques aléatoires



**Fig. A.31. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

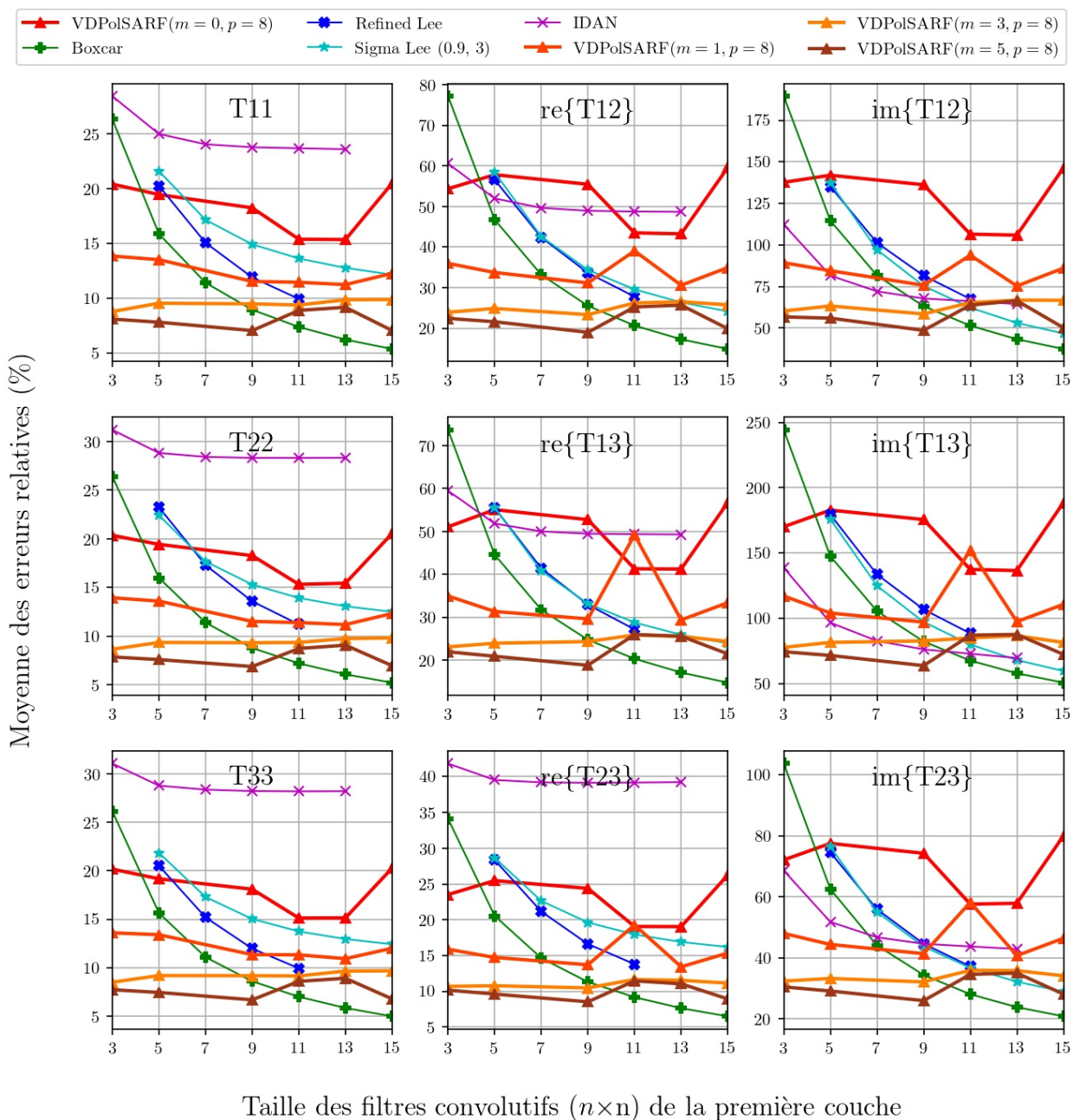
## Diffuseurs Anisotropiques aléatoires



**Fig. A.32. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

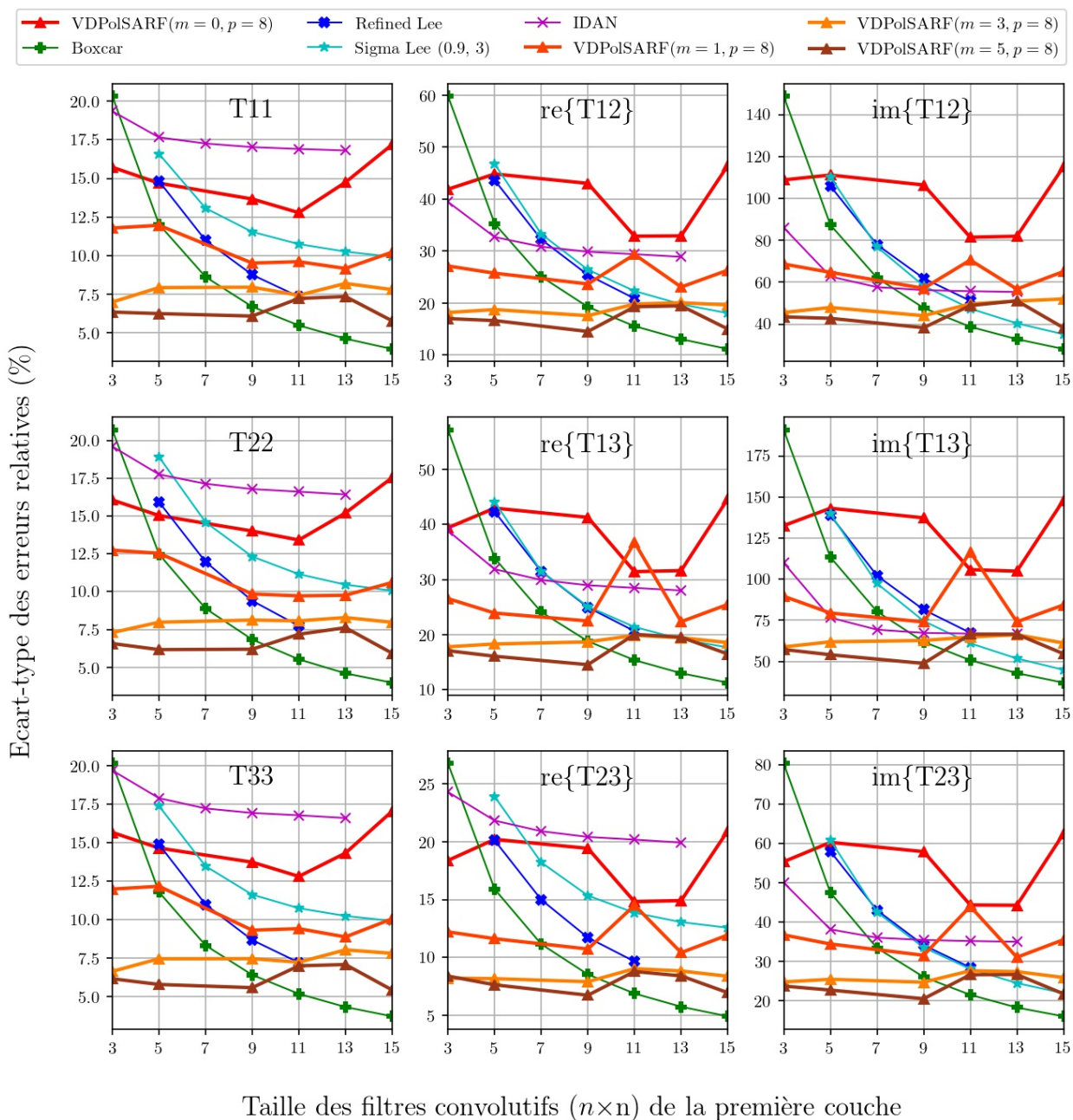
**A.3. Les résultats sur l'estimation de la matrices de cohérence sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes avec inclusion de cibles ponctuelles.**

## Réflexions diédrales



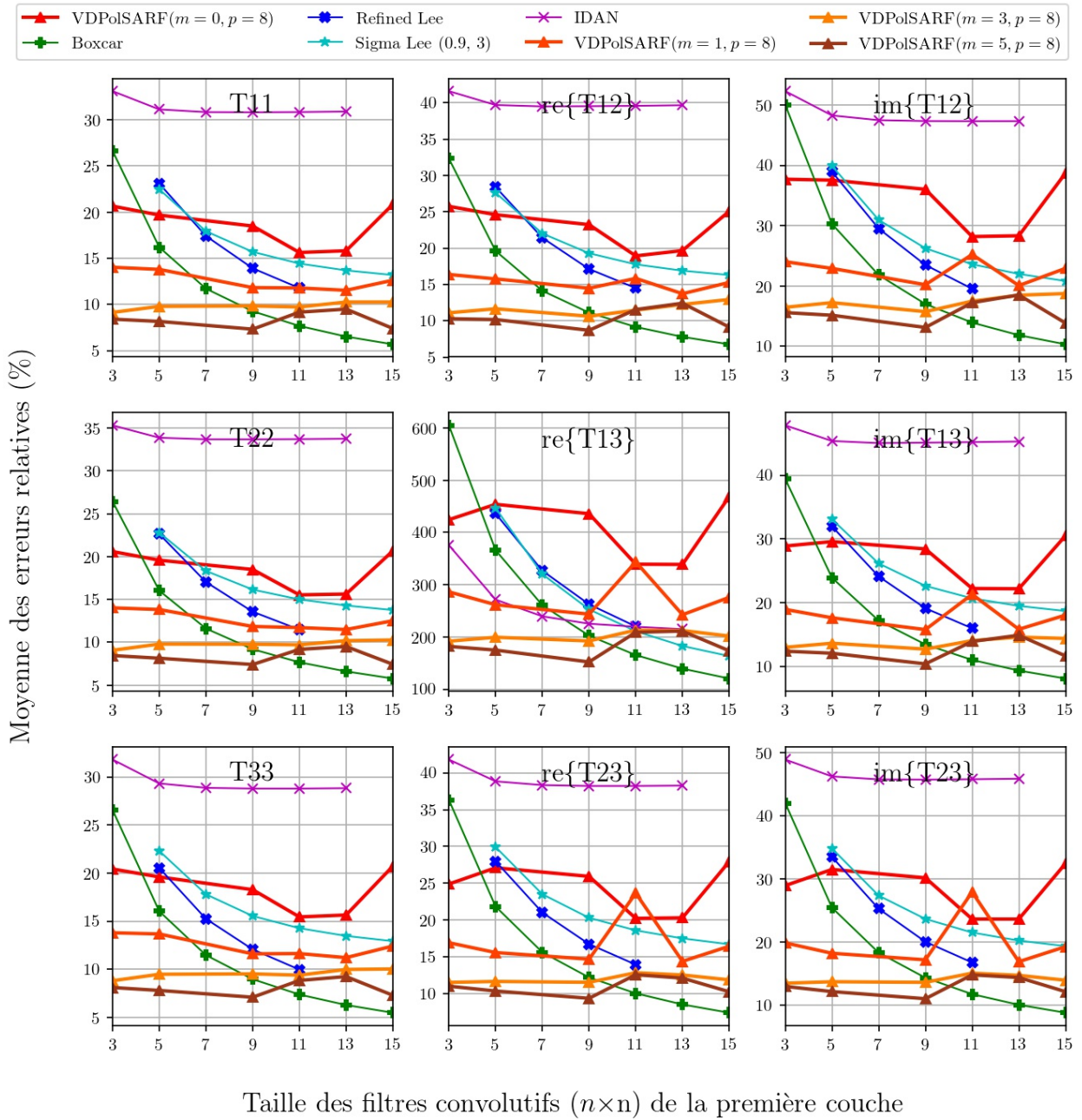
**Fig. A.33. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



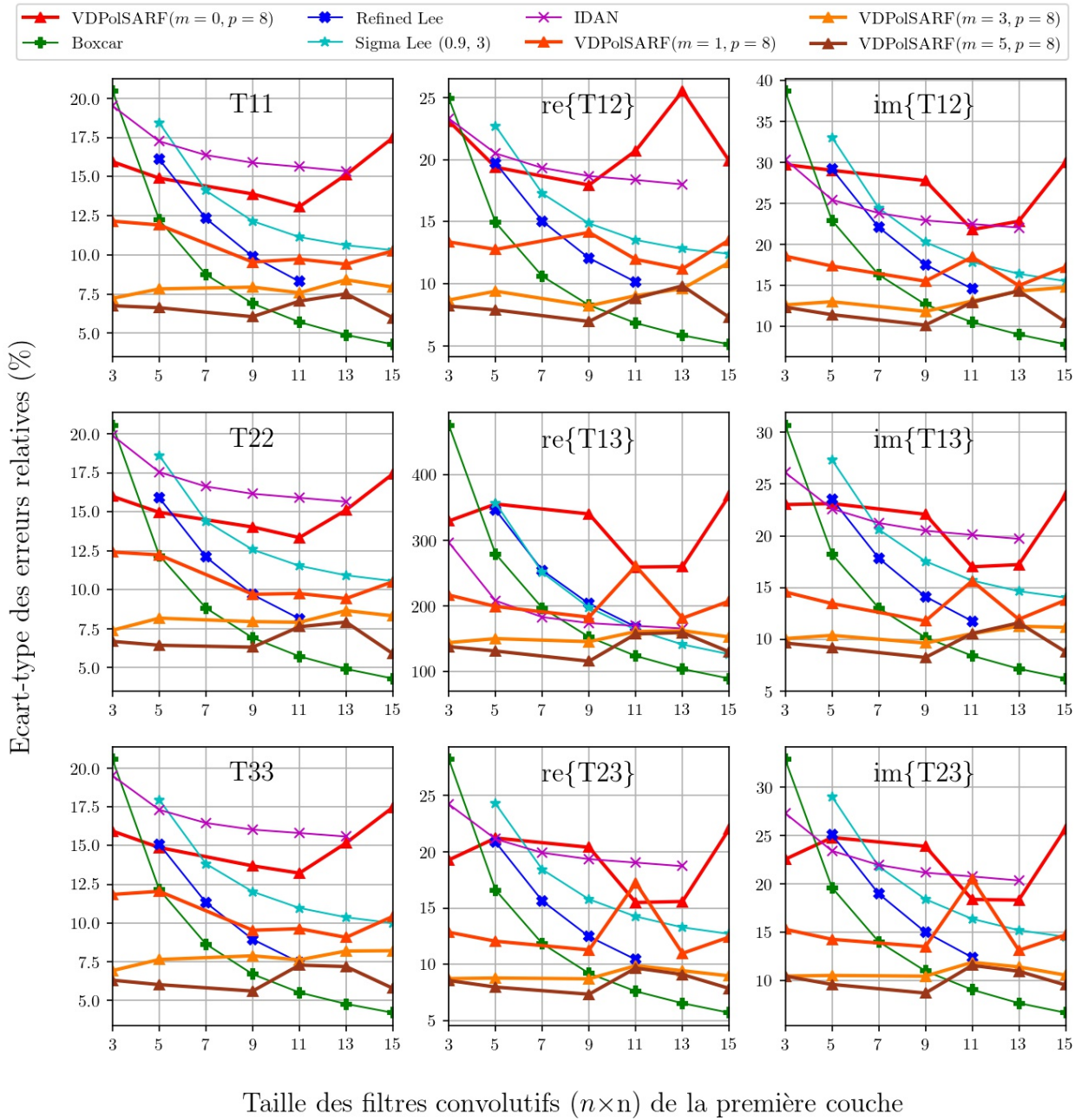
**Fig. A.34. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

## Dipôles



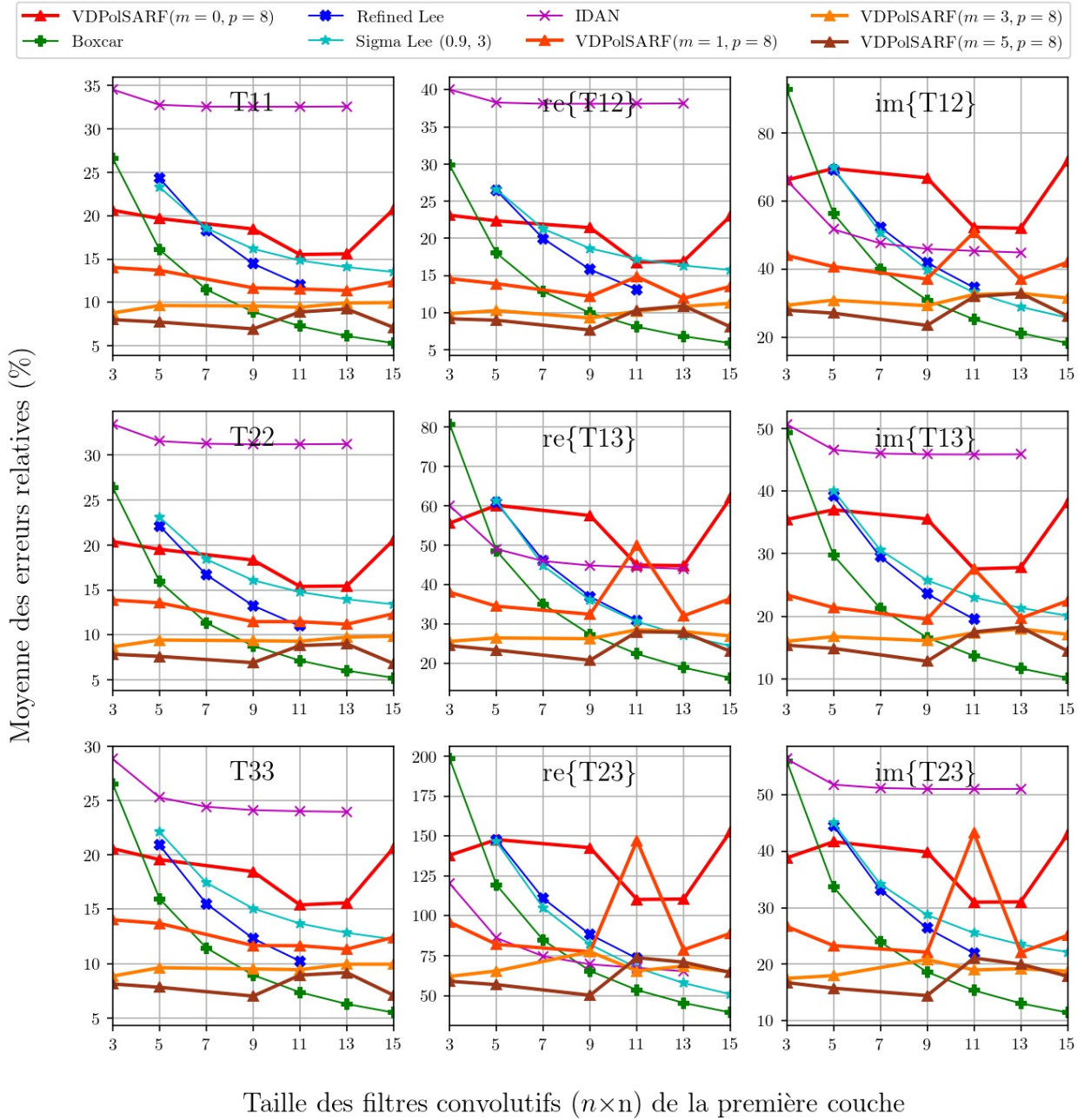
**Fig. A.35. Dipôles** . Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



**Fig. A.36. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

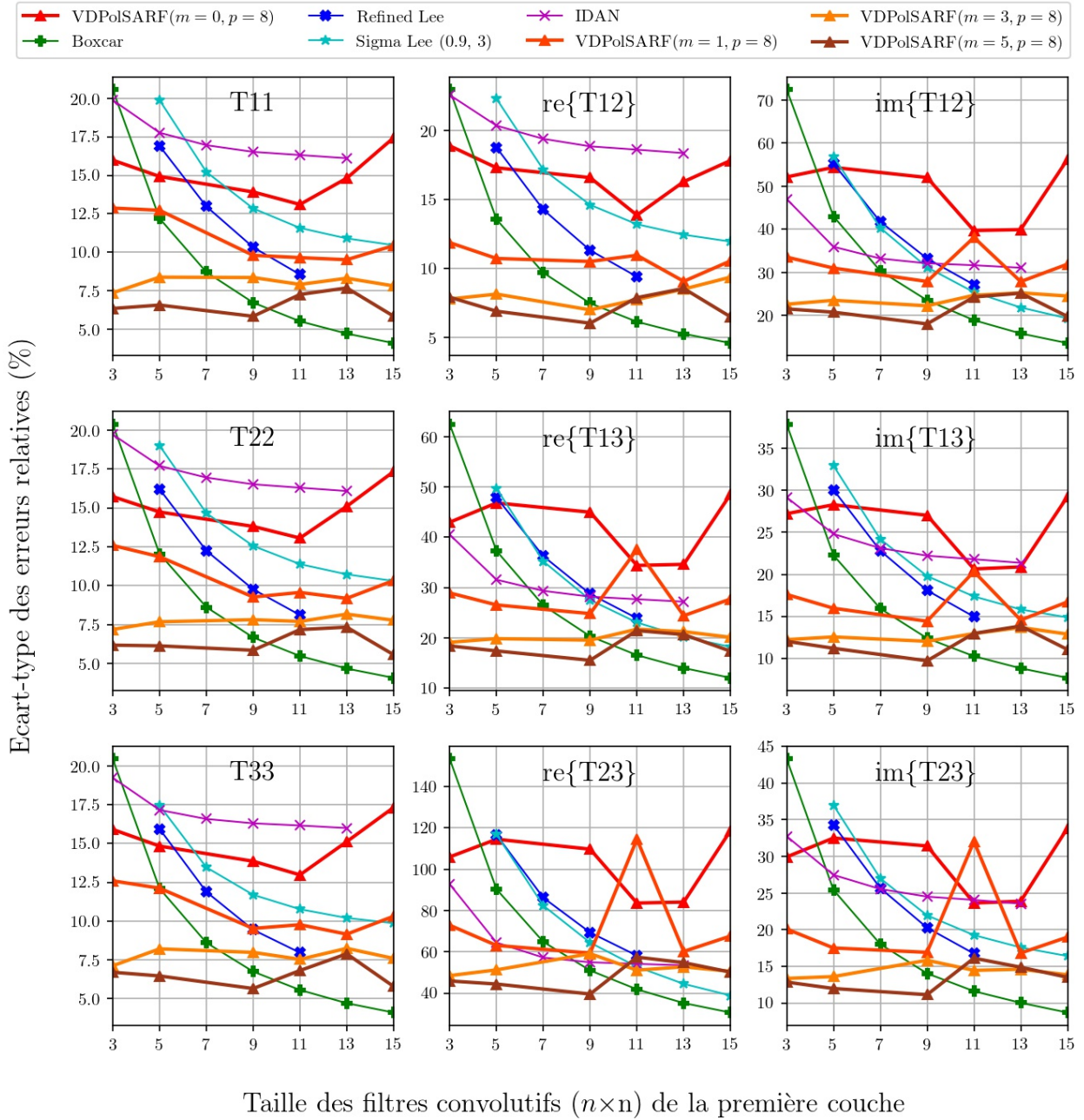
### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.37. Surfaces de Bragg.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

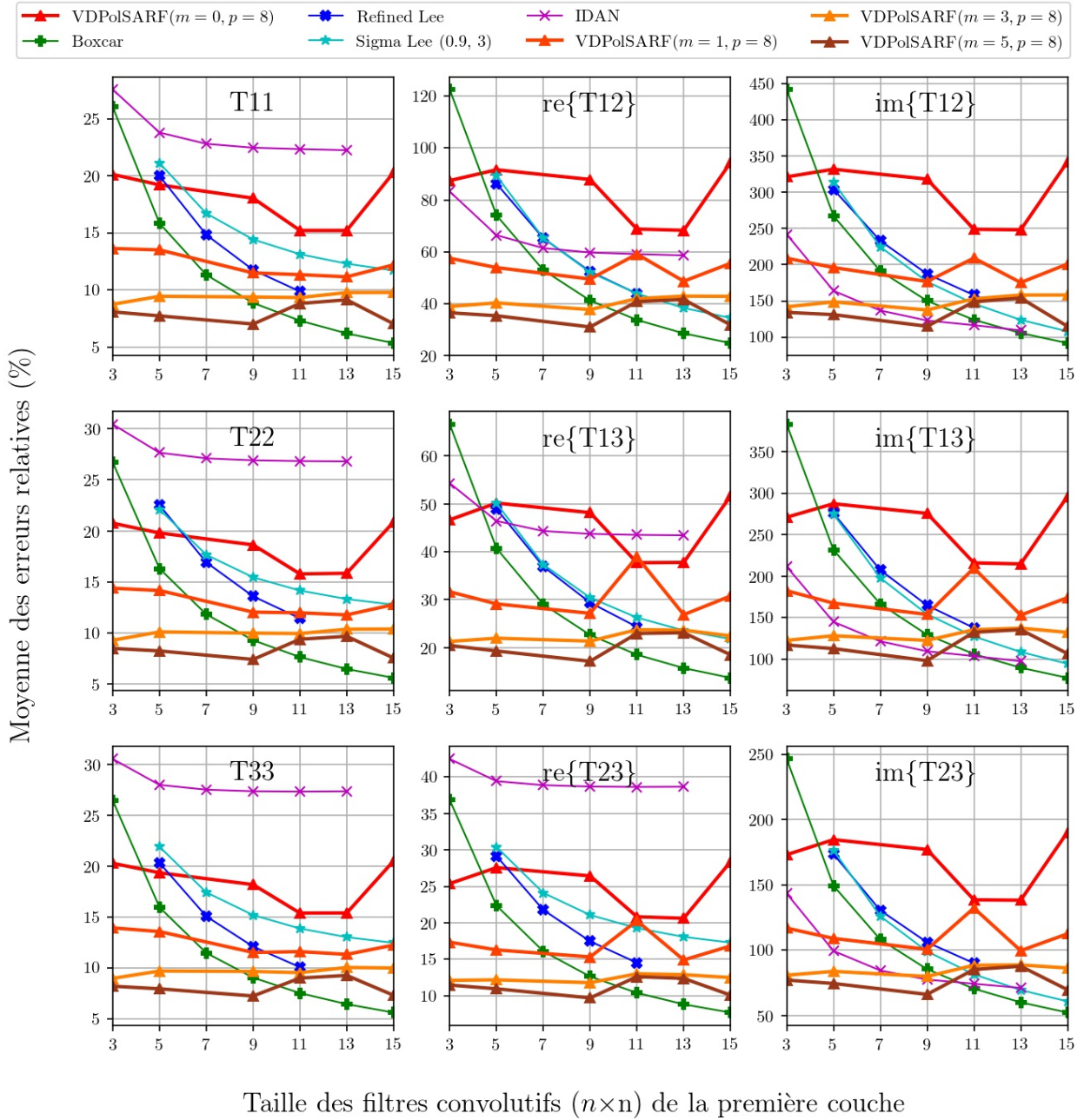


### Surfaces de Bragg (Diffuseurs surfaciques)



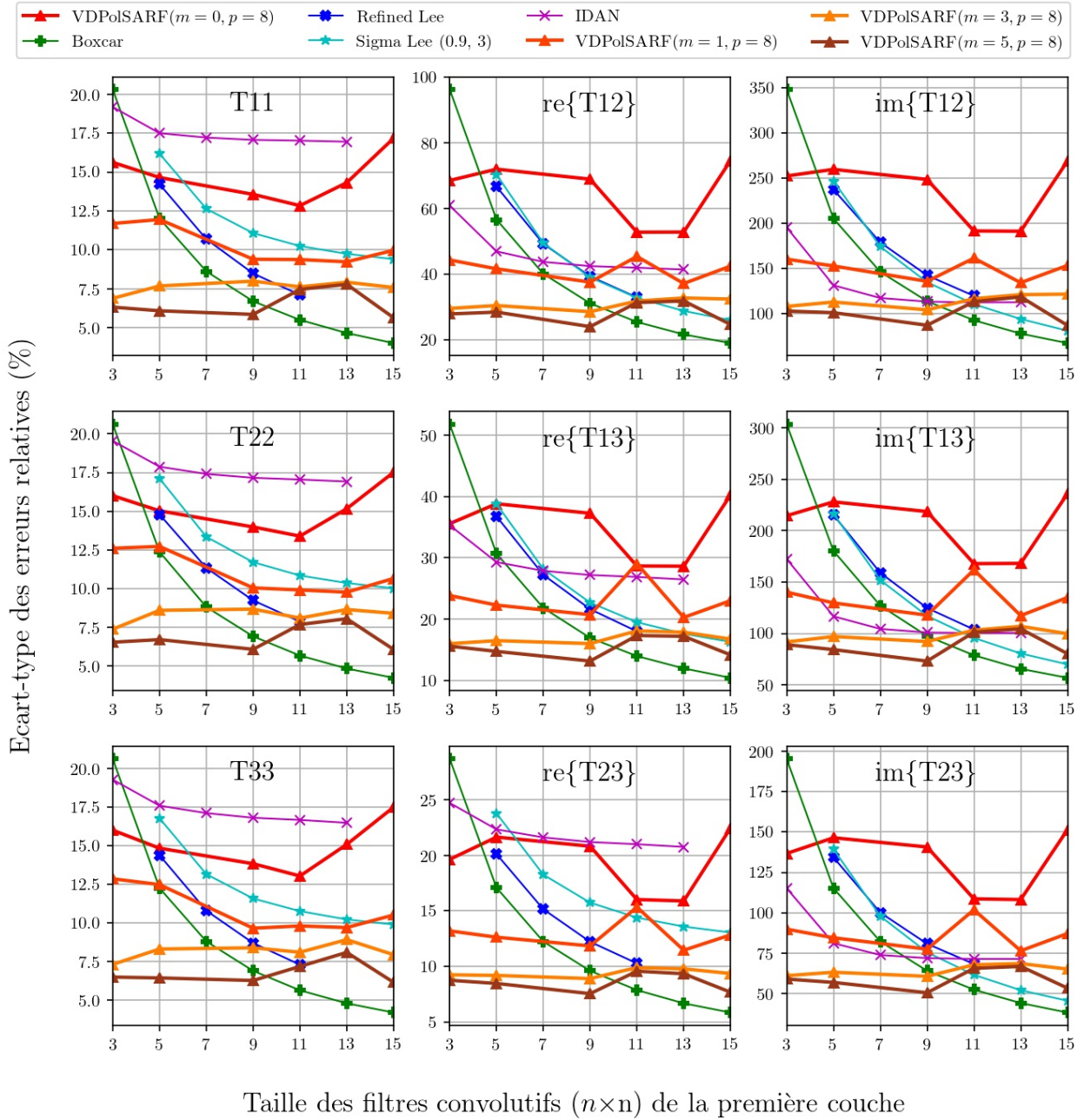
**Fig. A.38. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



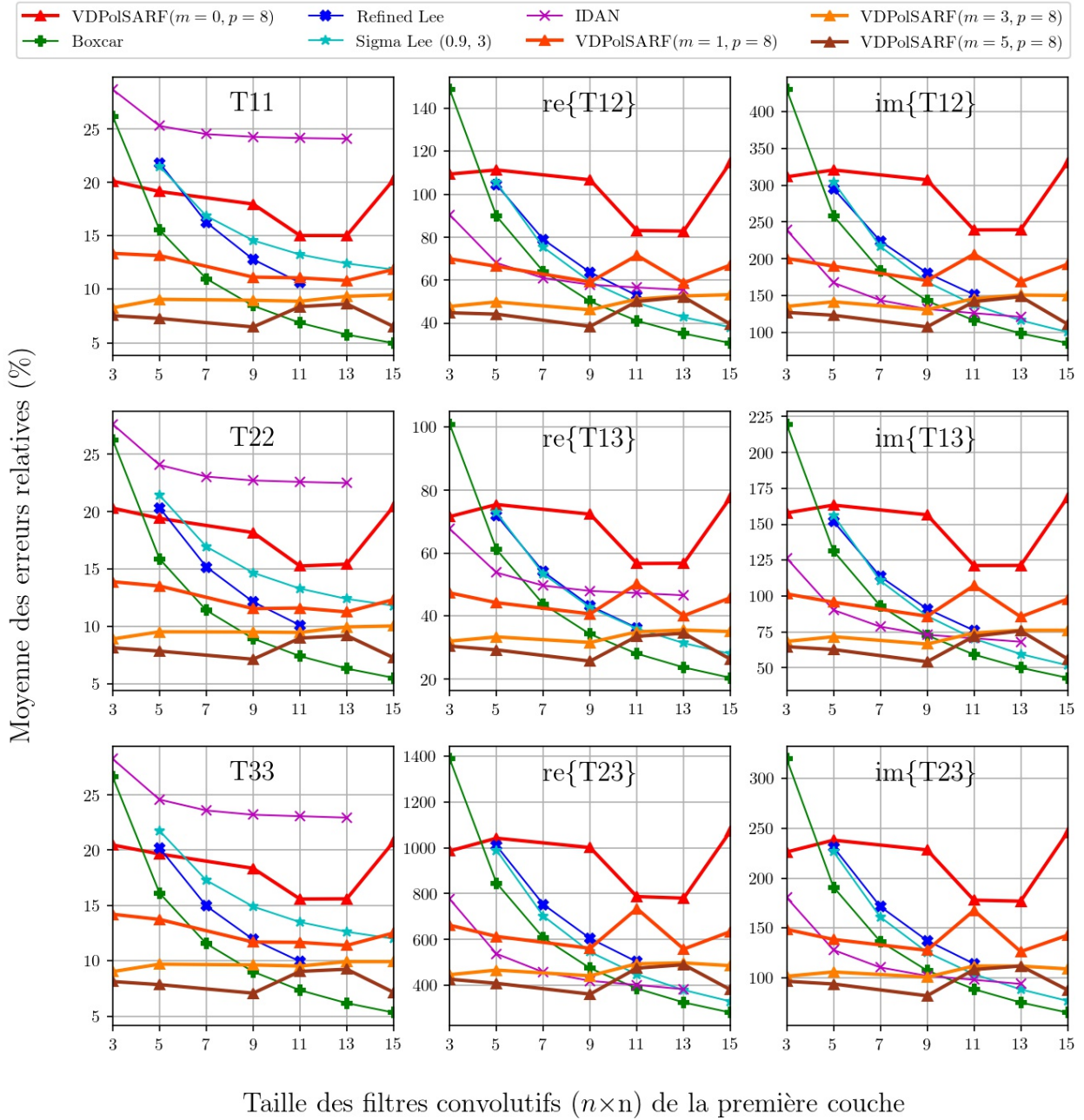
**Fig. A.39. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Double Réflexion (Diffuseurs double bond)



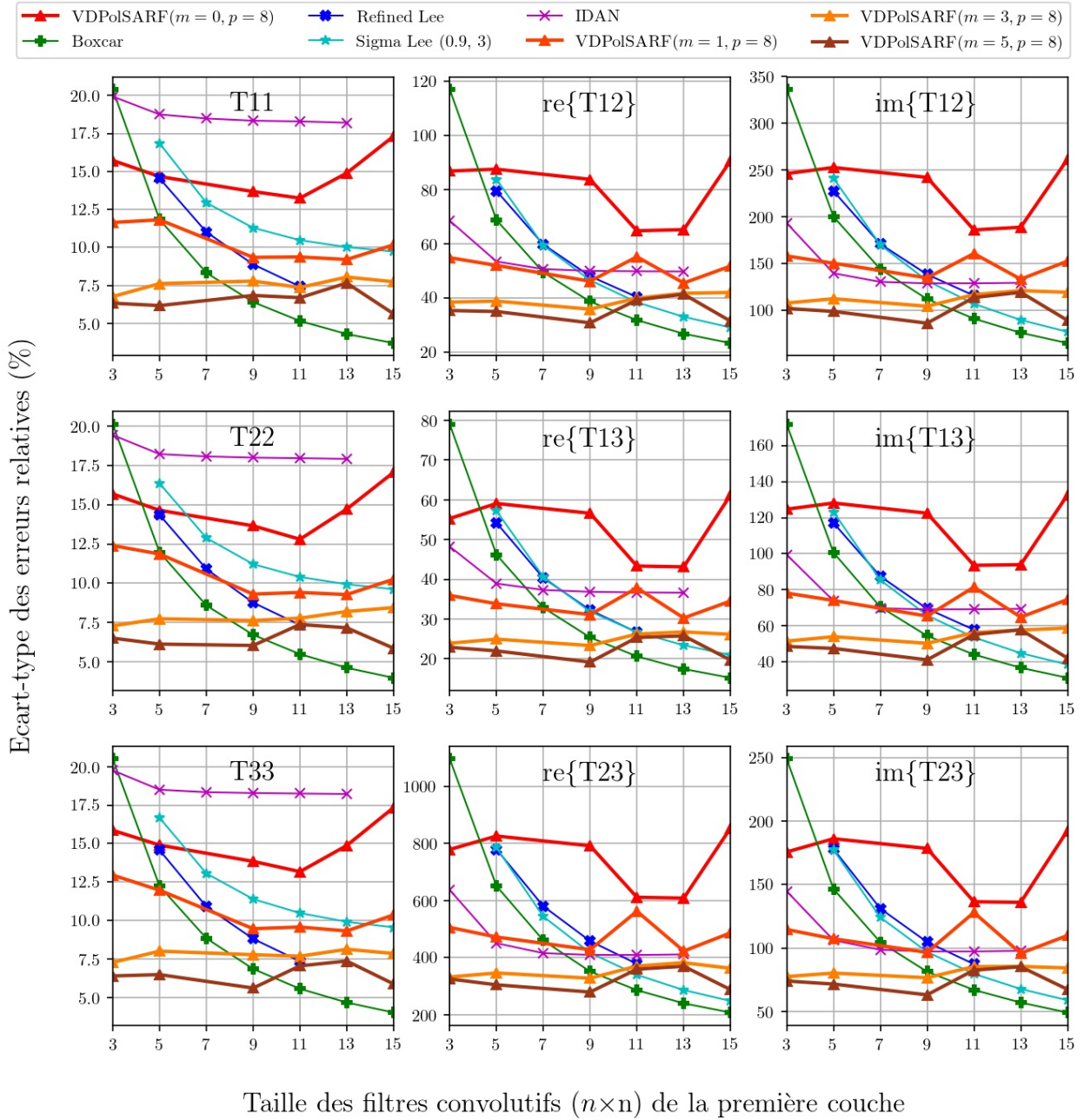
**Fig. A.40. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Particules Anisotropiques (Diffuseurs volumétriques)



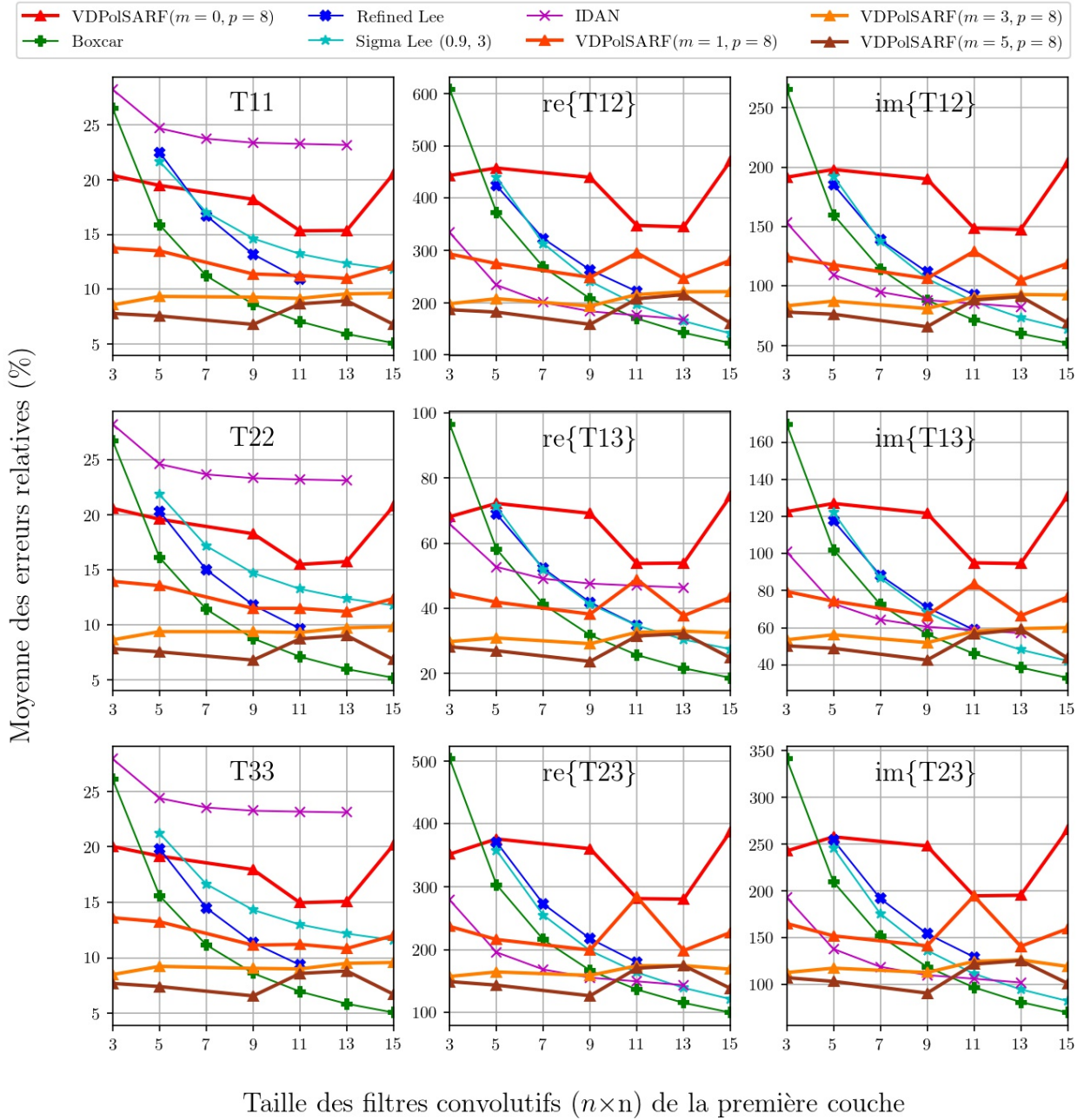
**Fig. A.41. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

Particules Anisotropiques (Diffuseurs volumétriques)



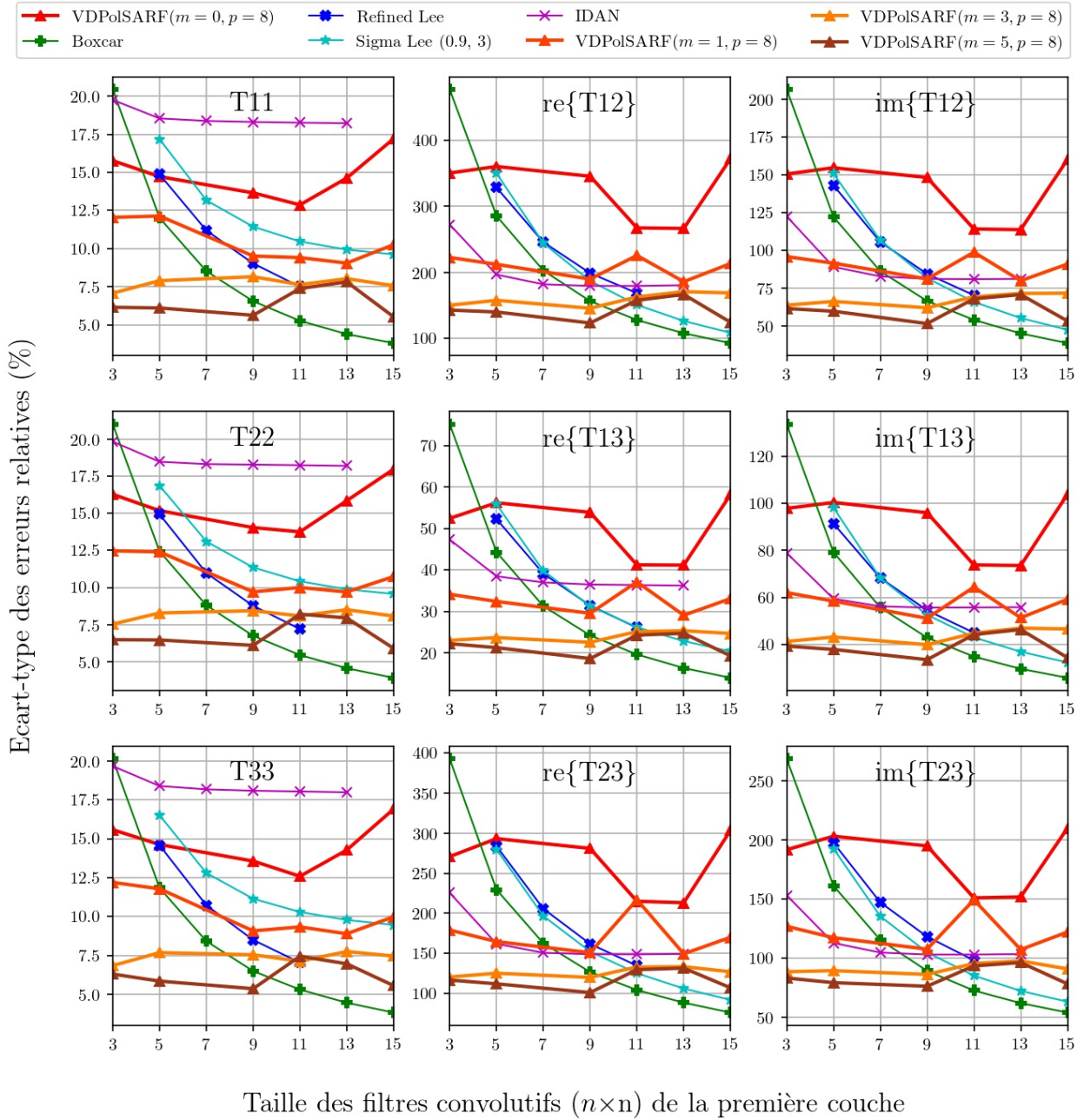
**Fig. A.42. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



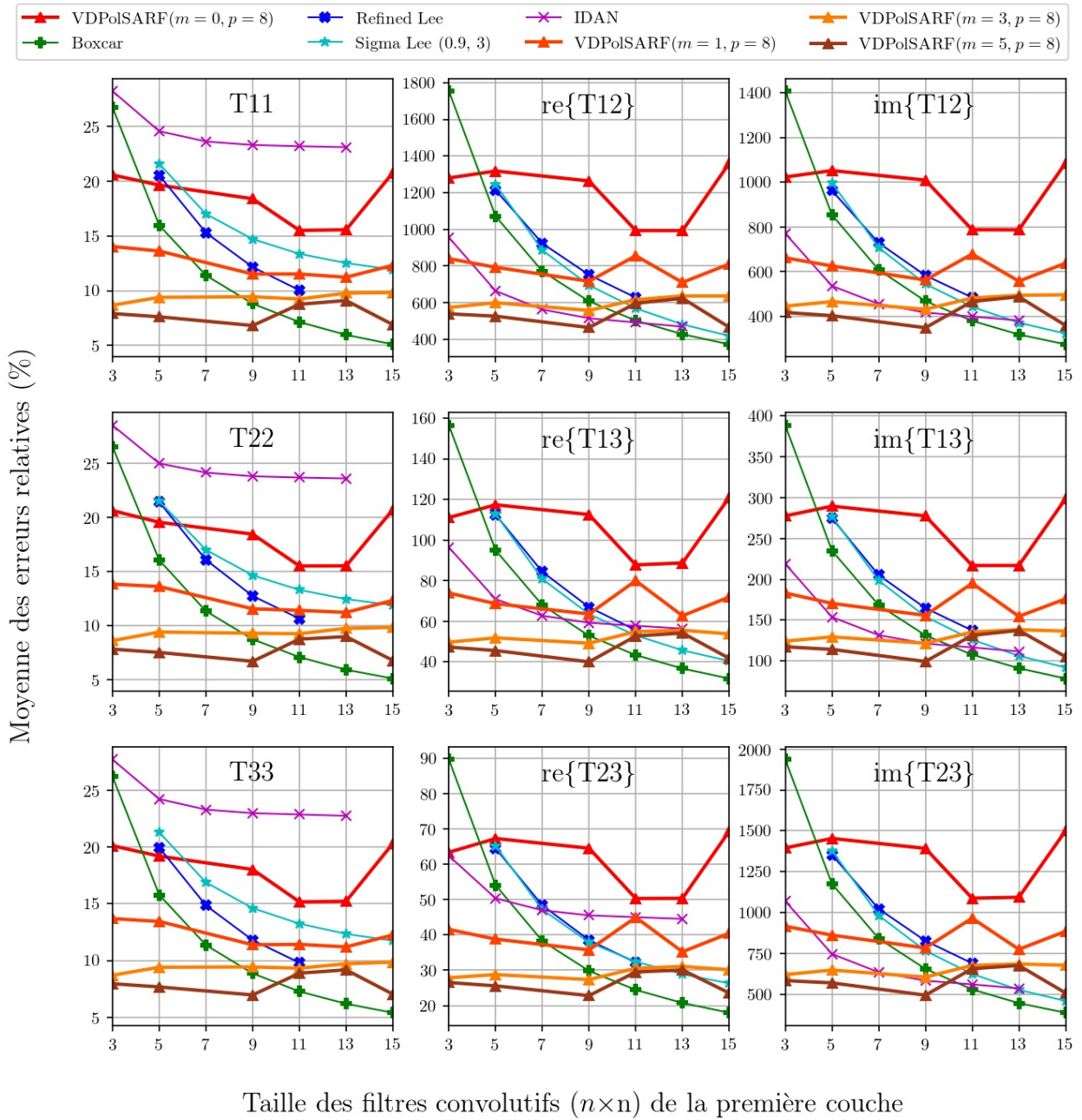
**Fig. A.43. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Surface aléatoires



**Fig. A.44. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

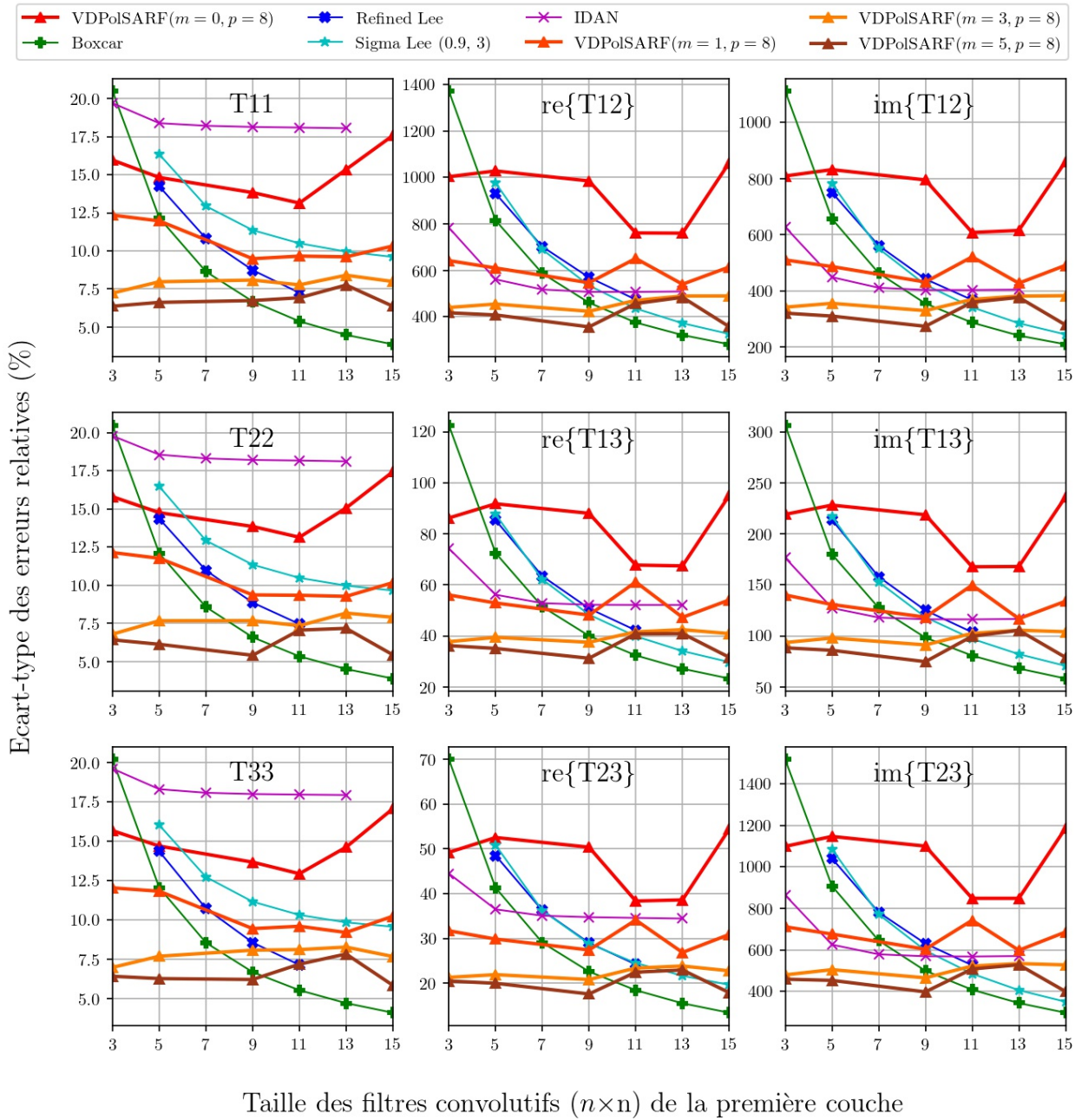
## Structures Complexes



**Fig. A.45. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

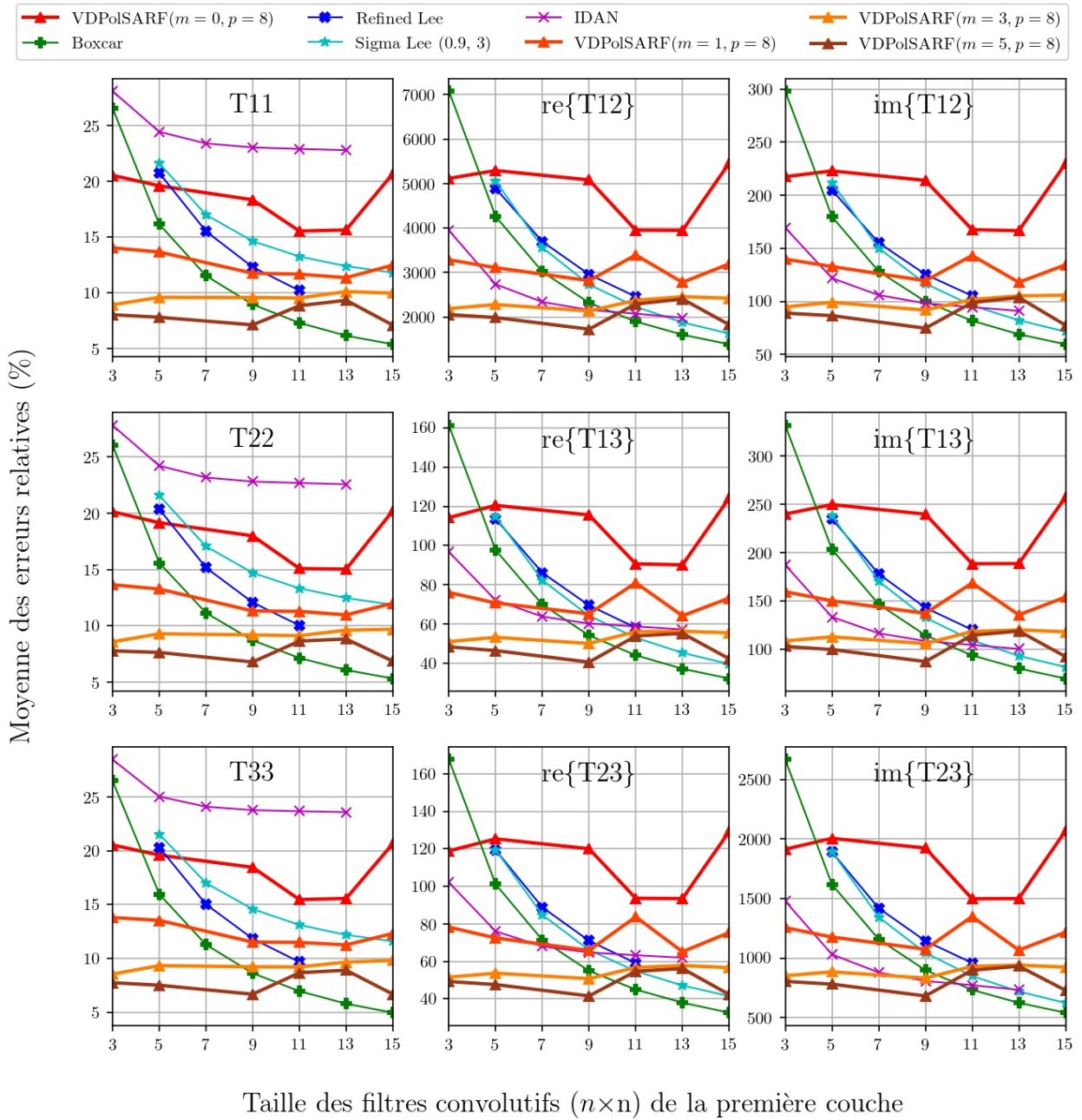


## Structures Complexes



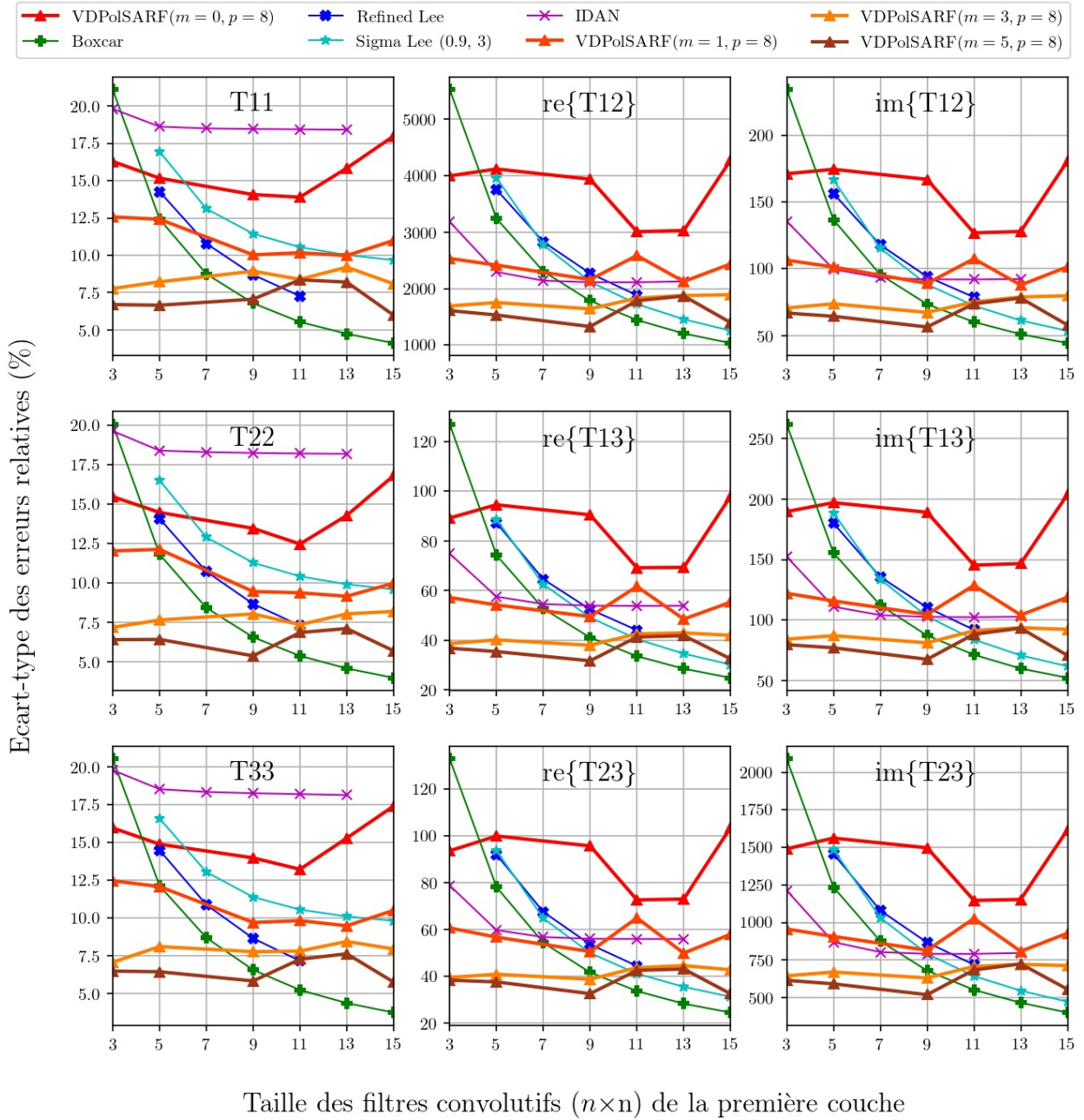
**Fig. A.46. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



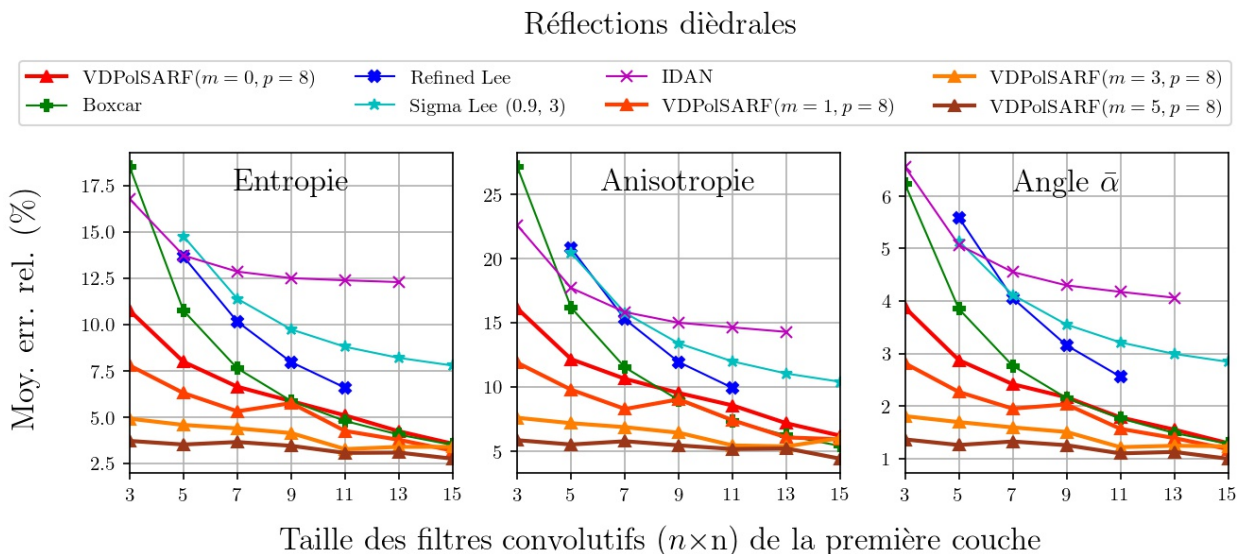
**Fig. A.47. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première convolution pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



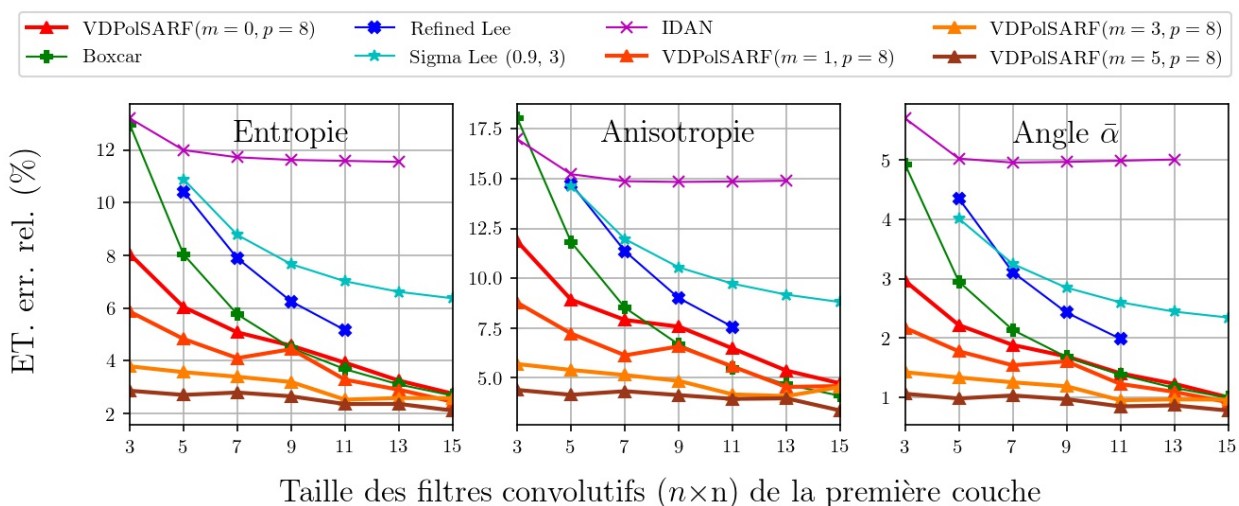
**Fig. A.48. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la matrice de cohérence en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

A.4. Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées homogènes des classes de diffuseurs polarimétriques.



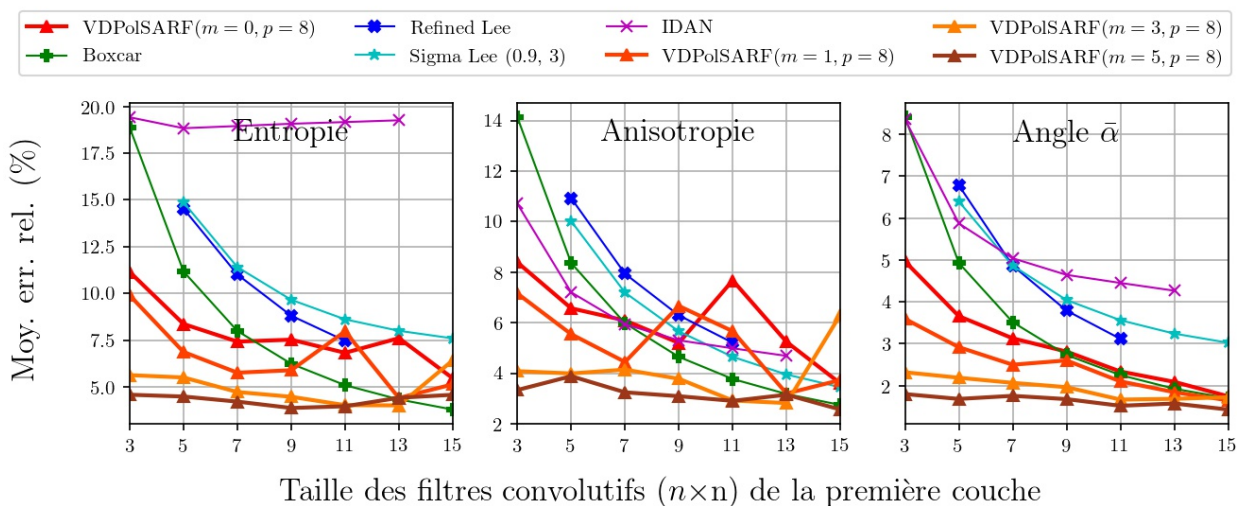
**Fig. A.49. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



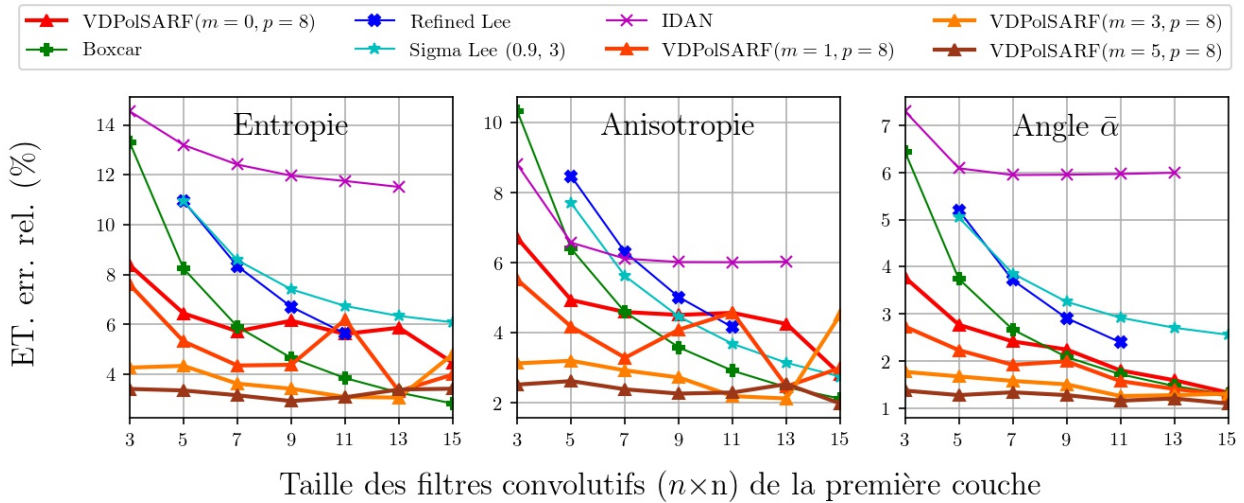
**Fig. A.50. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



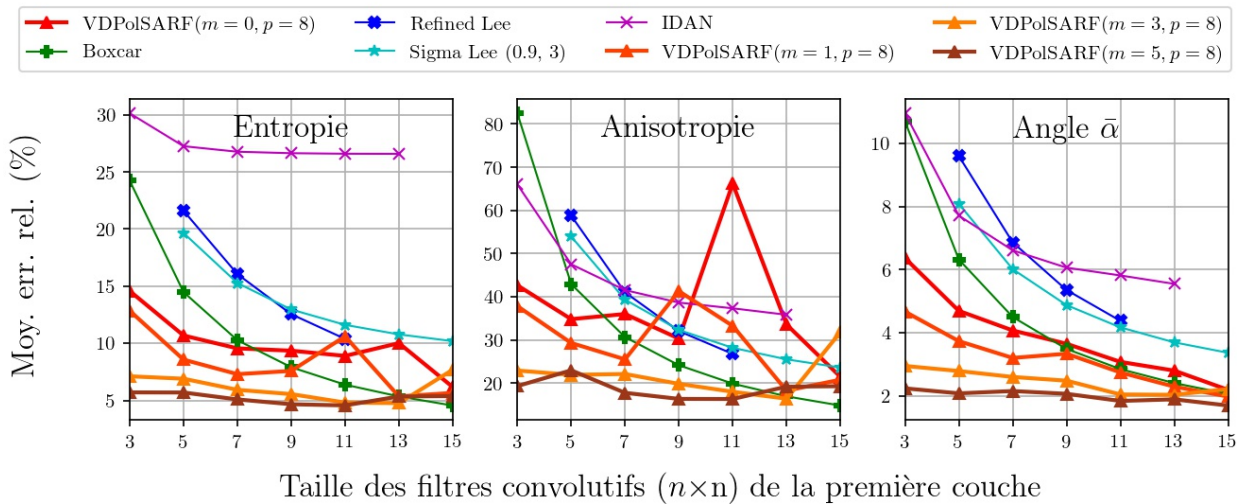
**Fig. A.51. Dipôles .** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Dipôles



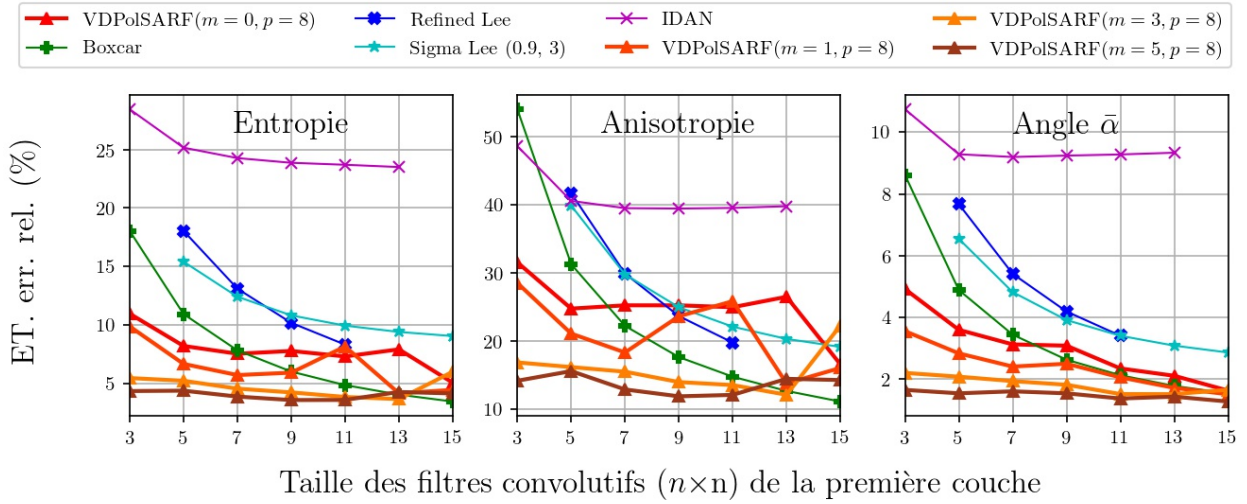
**Fig. A.52. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



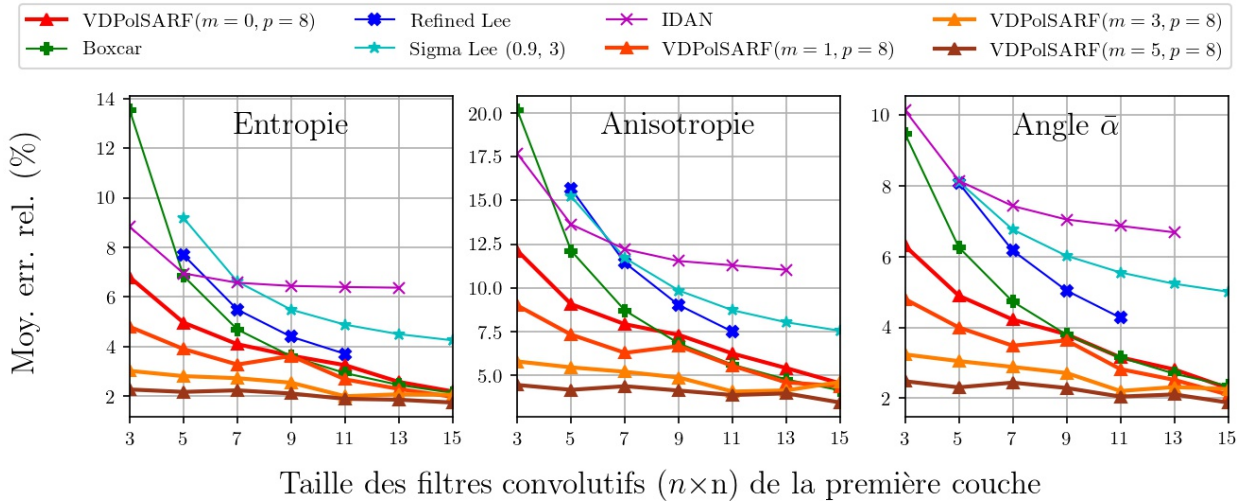
**Fig. A.53. Surfaces de Bragg** . Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



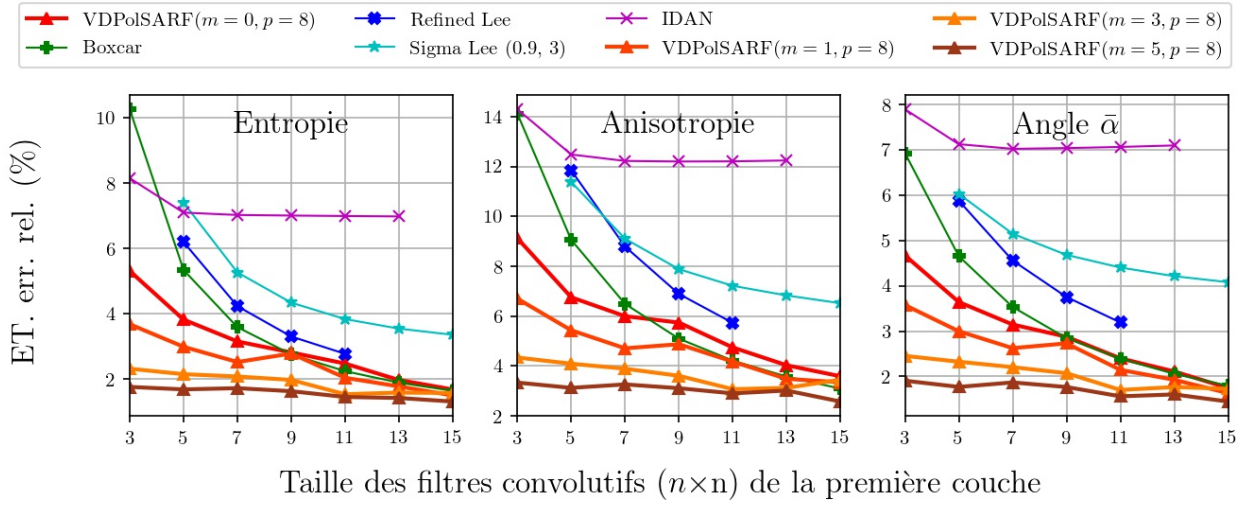
**Fig. A.54. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



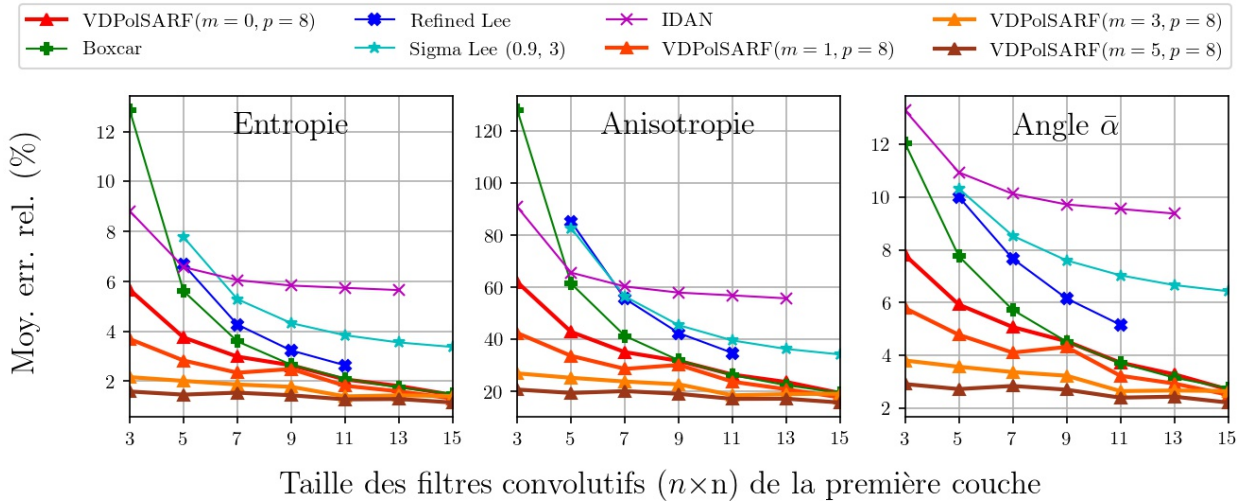
**Fig. A.55. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0, 1, 3, 5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



**Fig. A.56. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

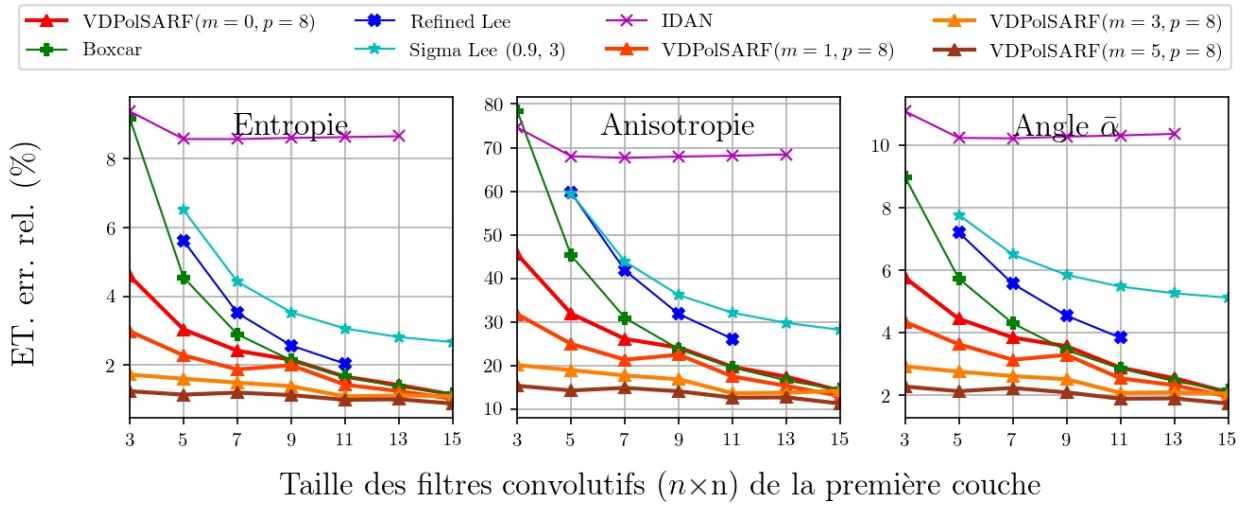
### Particules Anisotropiques (Diffuseurs volumétriques)



**Fig. A.57. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

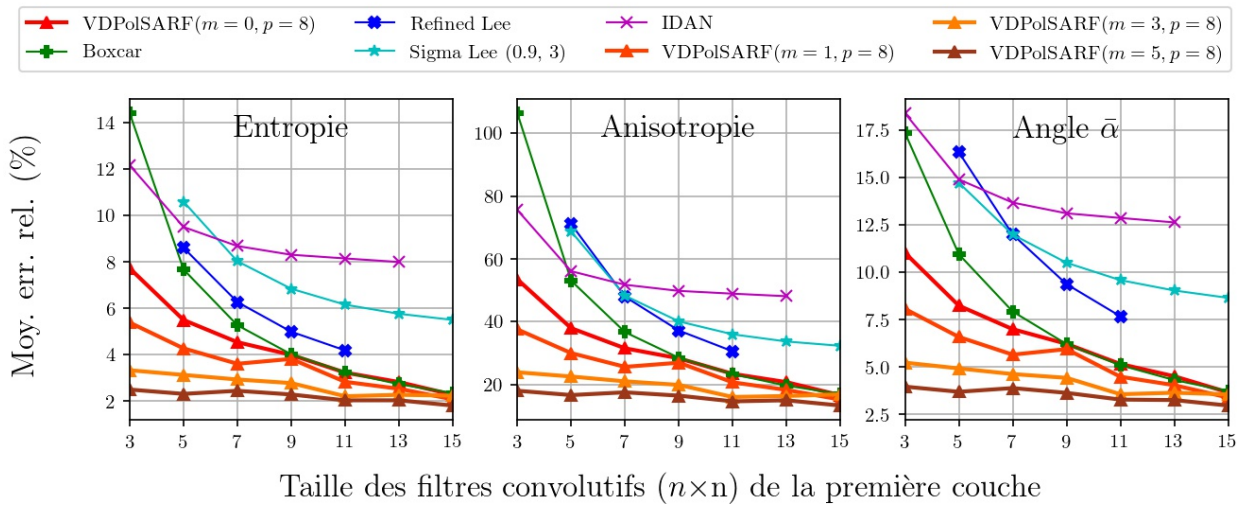


### Particules Anisotropiques (Diffuseurs volumétriques)



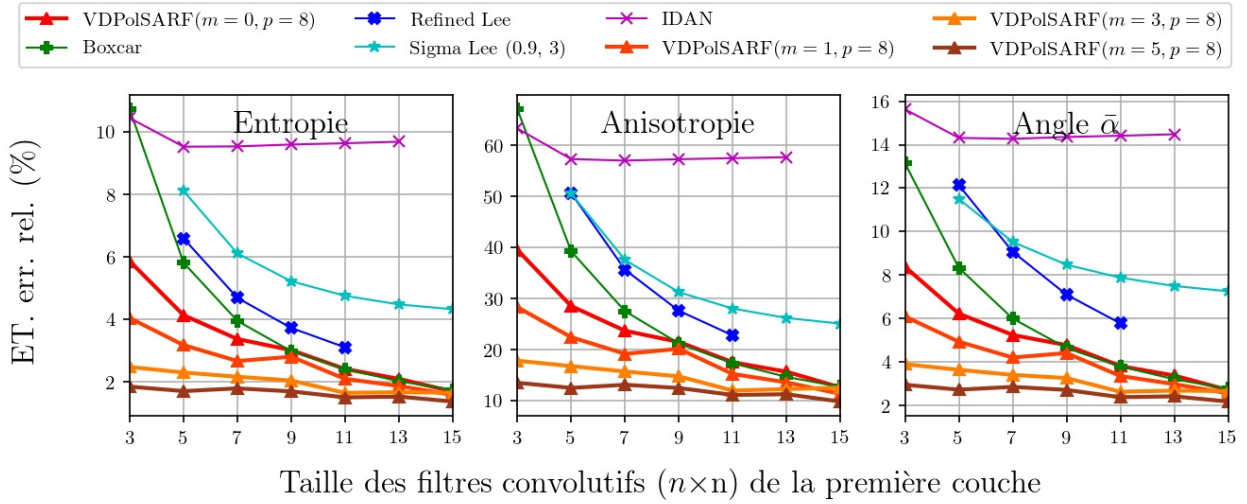
**Fig. A.58. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



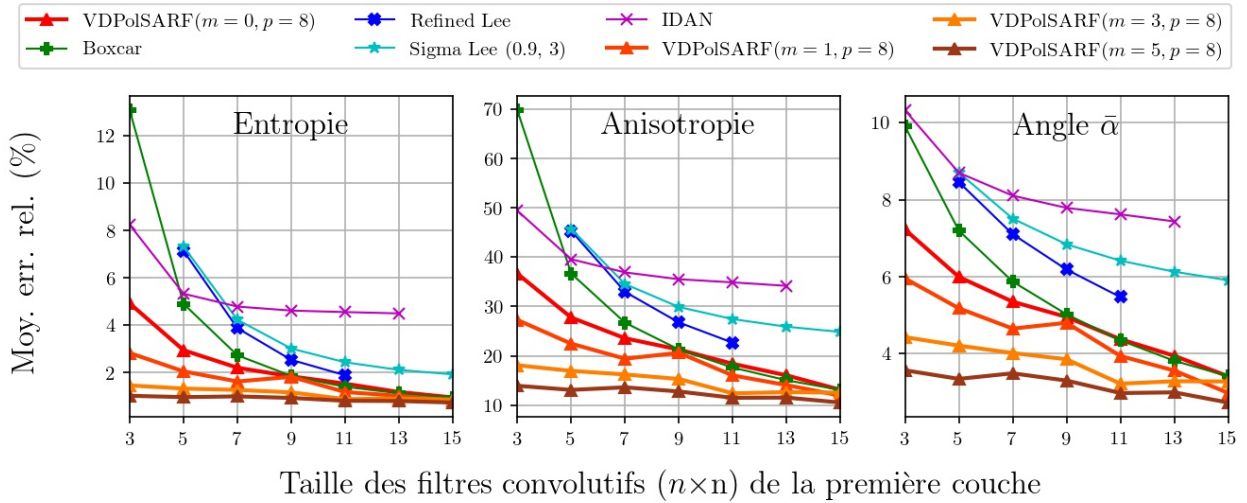
**Fig. A.59. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



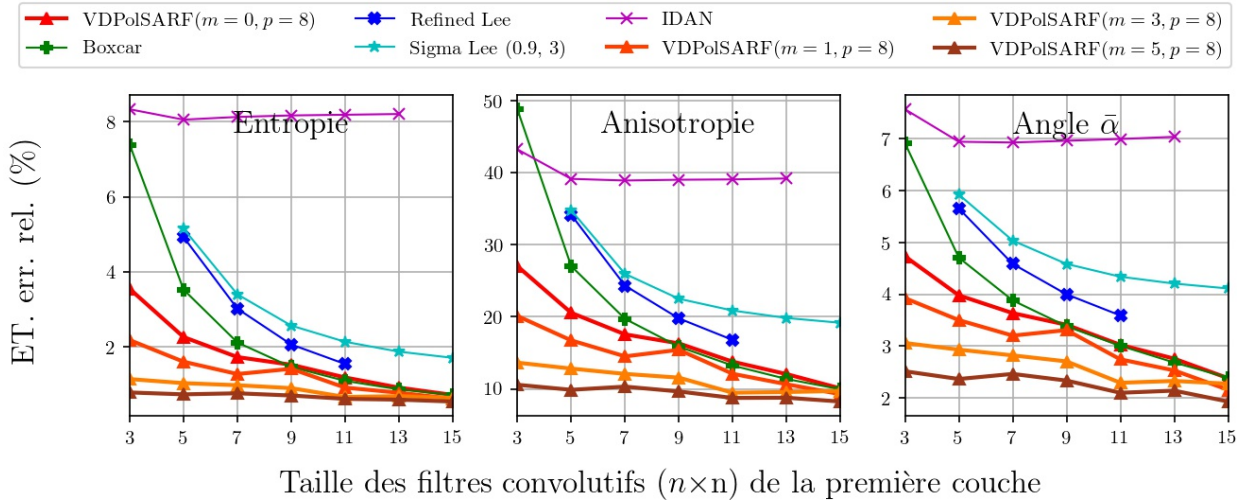
**Fig. A.60. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Structures Complexes



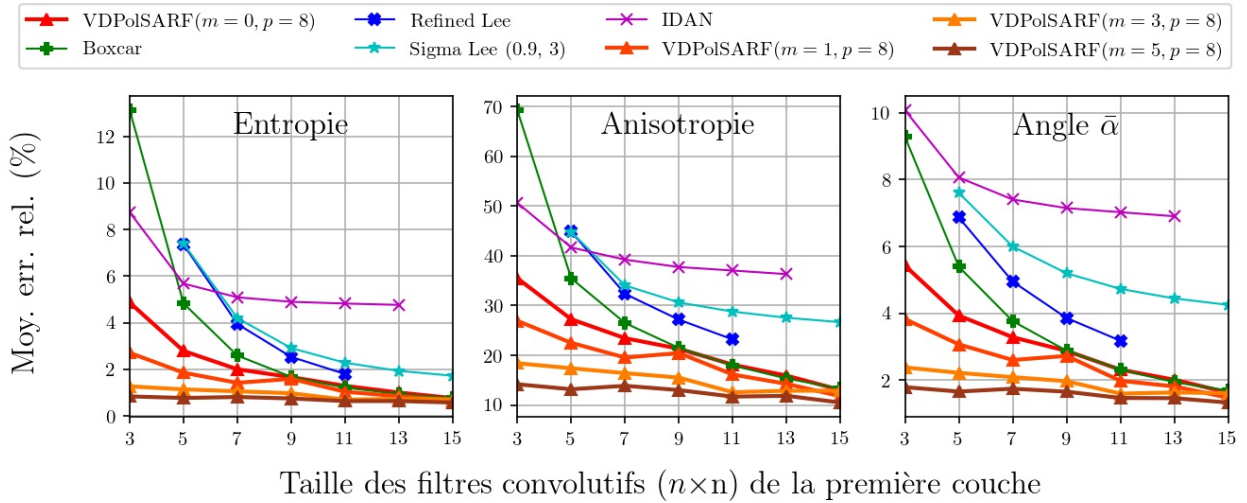
**Fig. A.61. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Structures Complexes



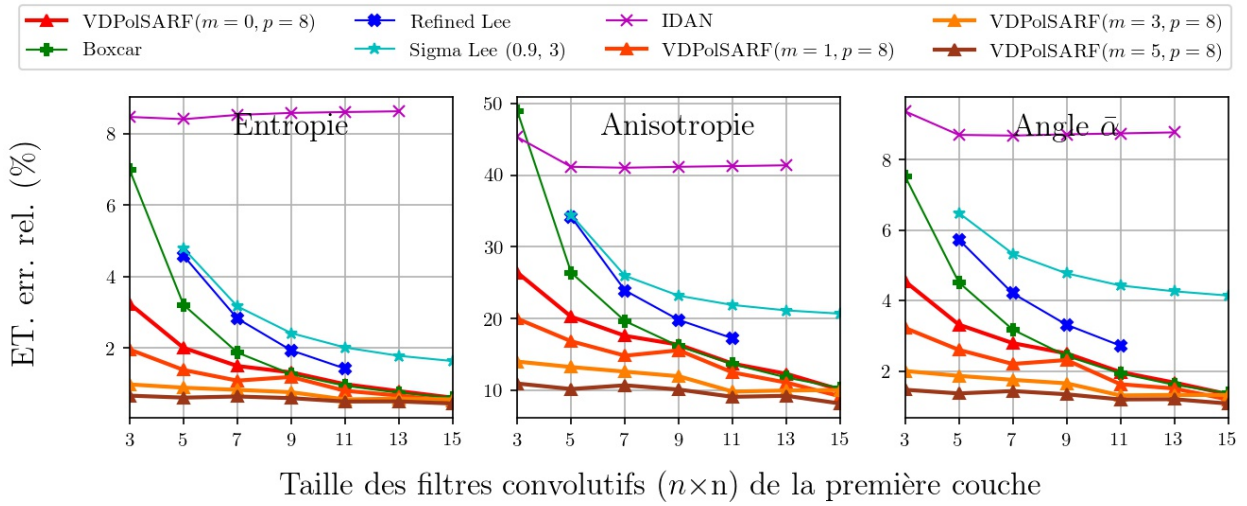
**Fig. A.62. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



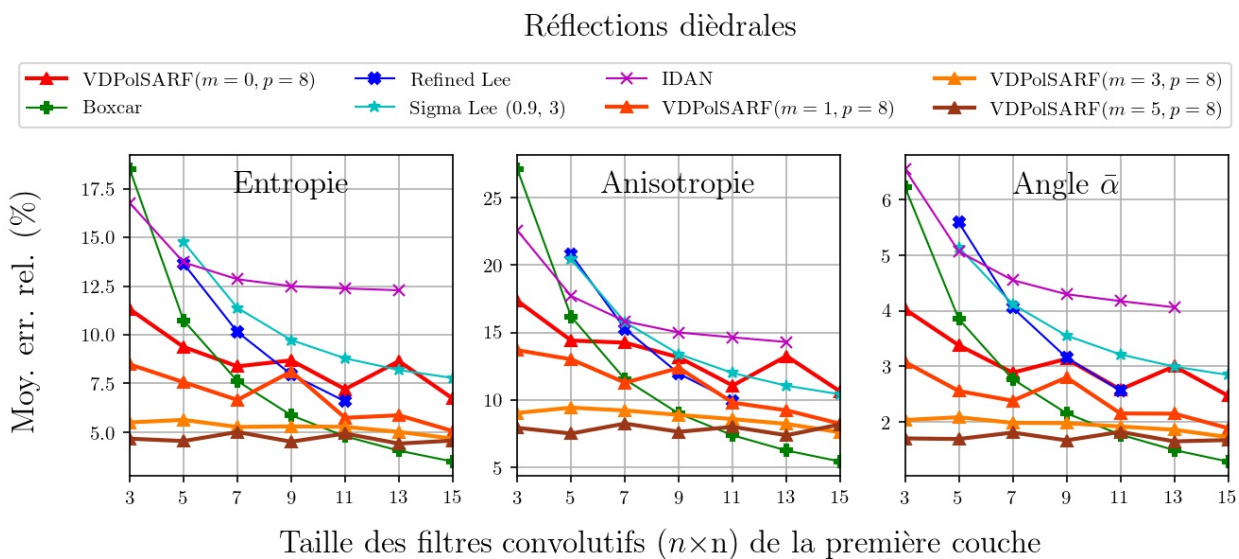
**Fig. A.63. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



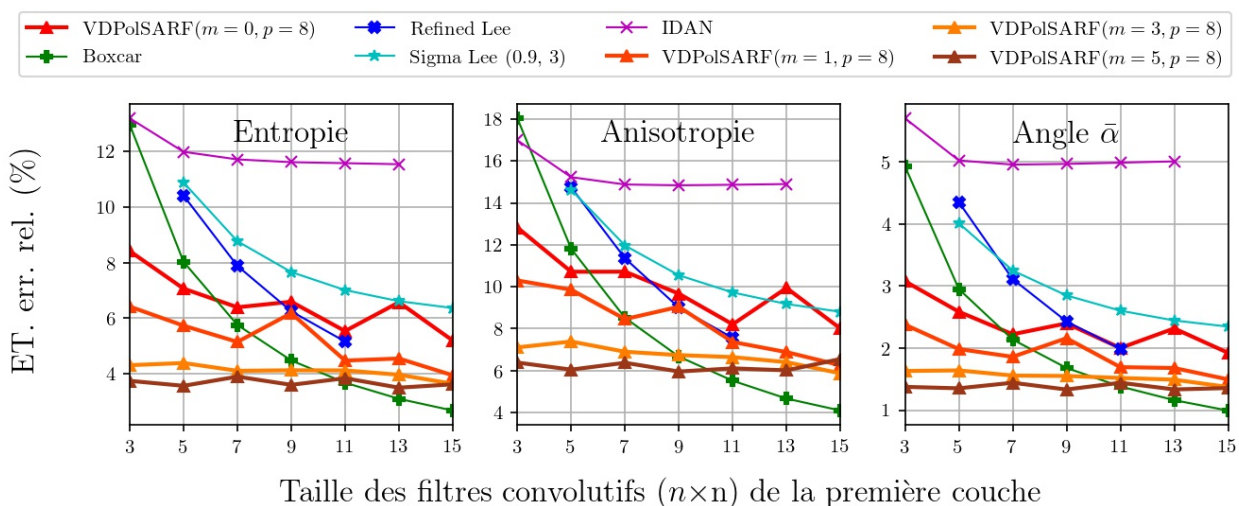
**Fig. A.64. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## A.5. Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes.



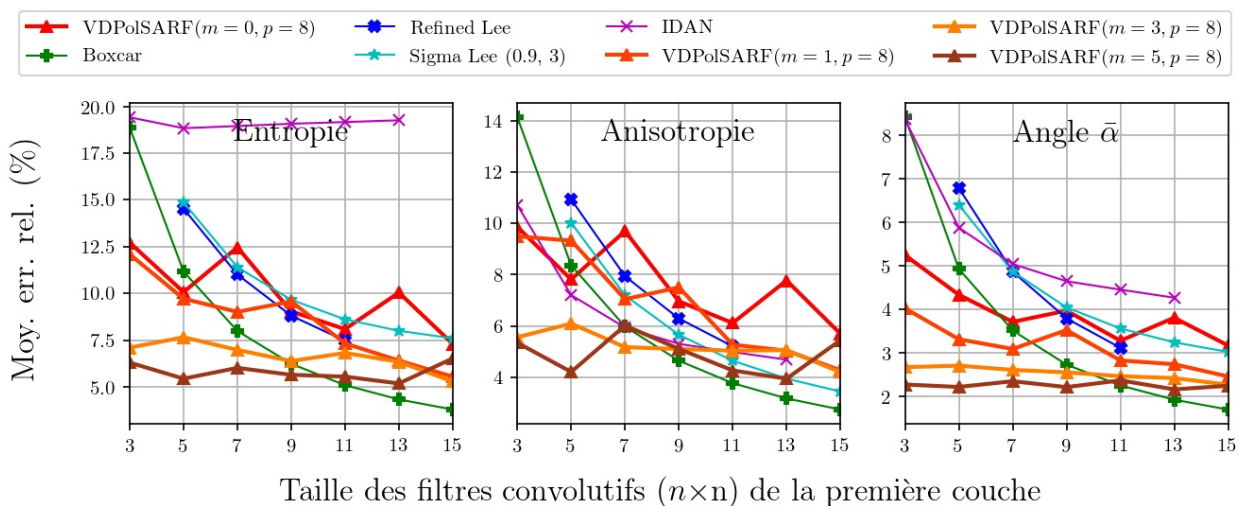
**Fig. A.65. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



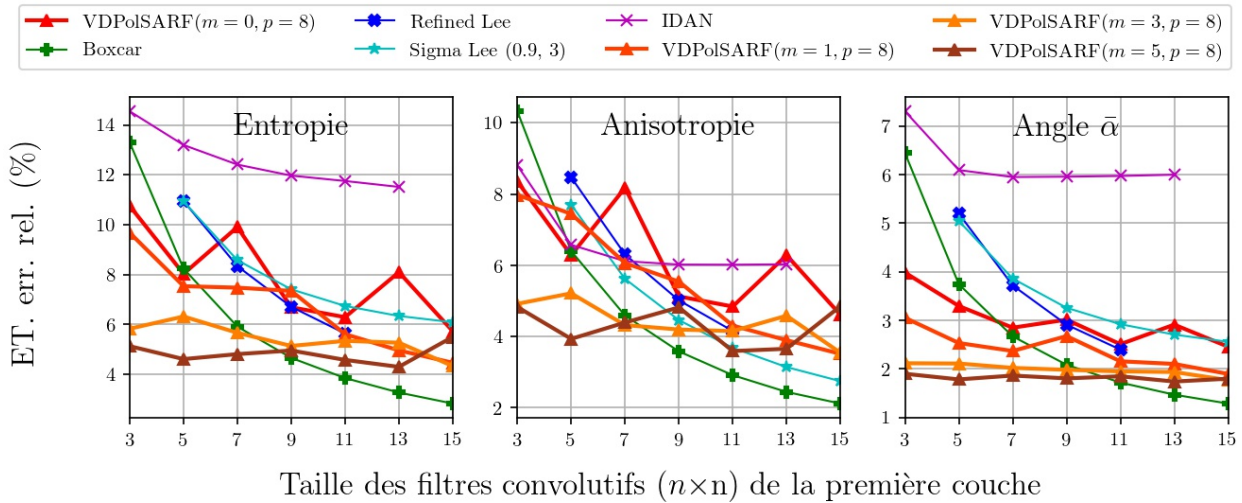
**Fig. A.66. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



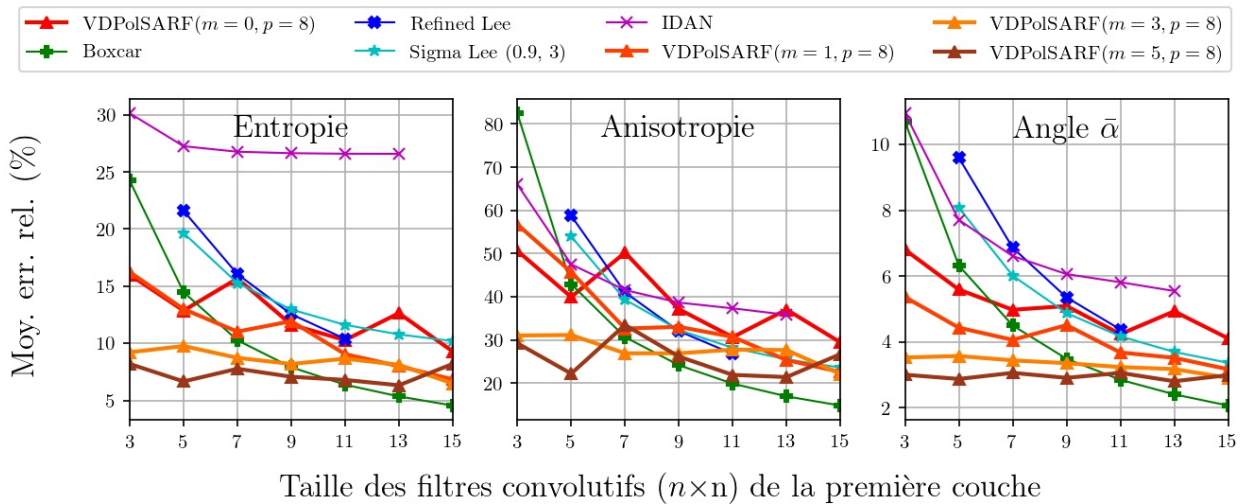
**Fig. A.67. Dipôles .** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Dipôles



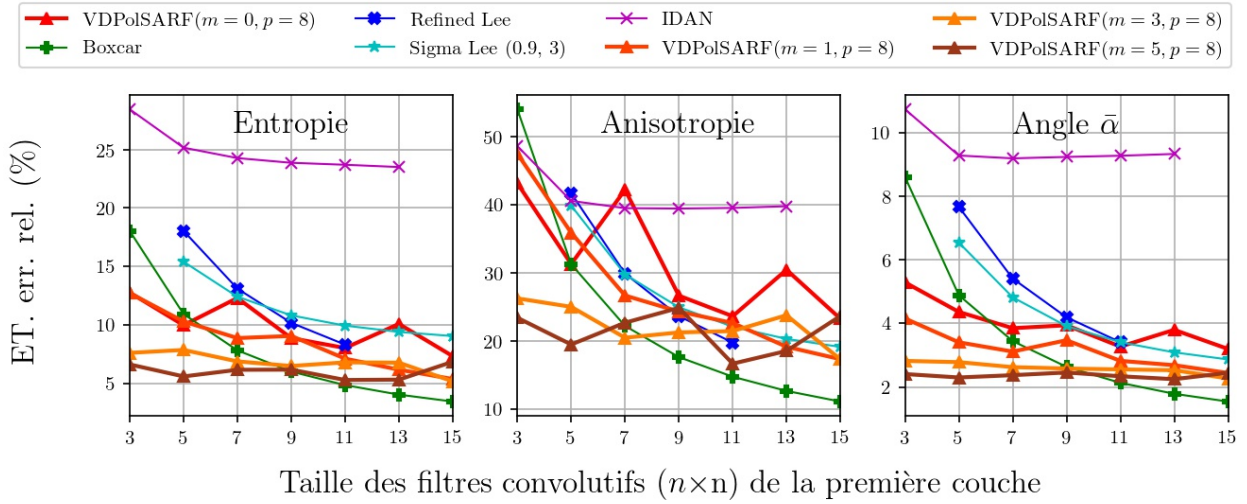
**Fig. A.68. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



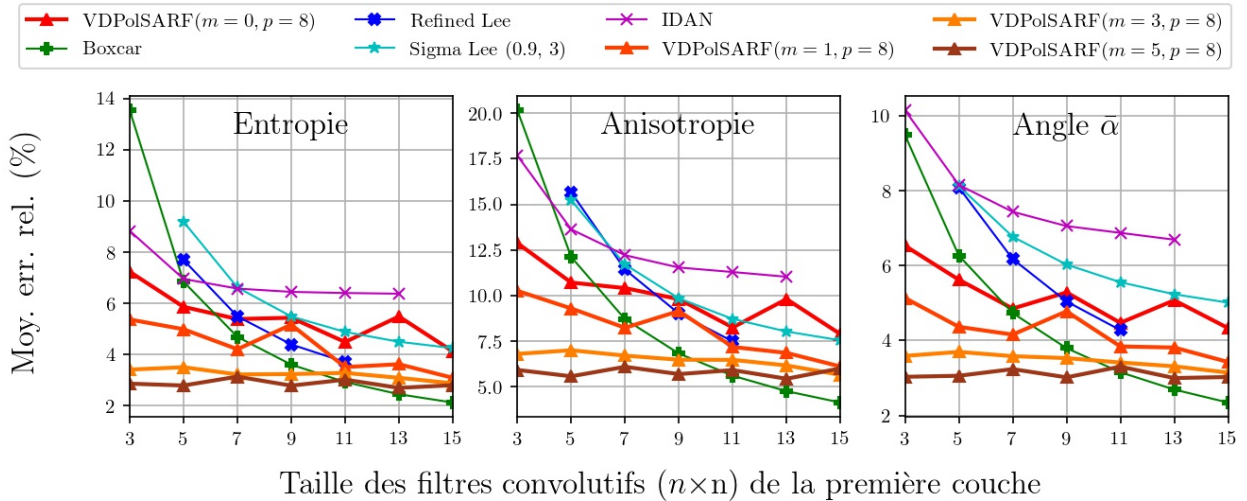
**Fig. A.69. Surfaces de Bragg** . Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.70. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

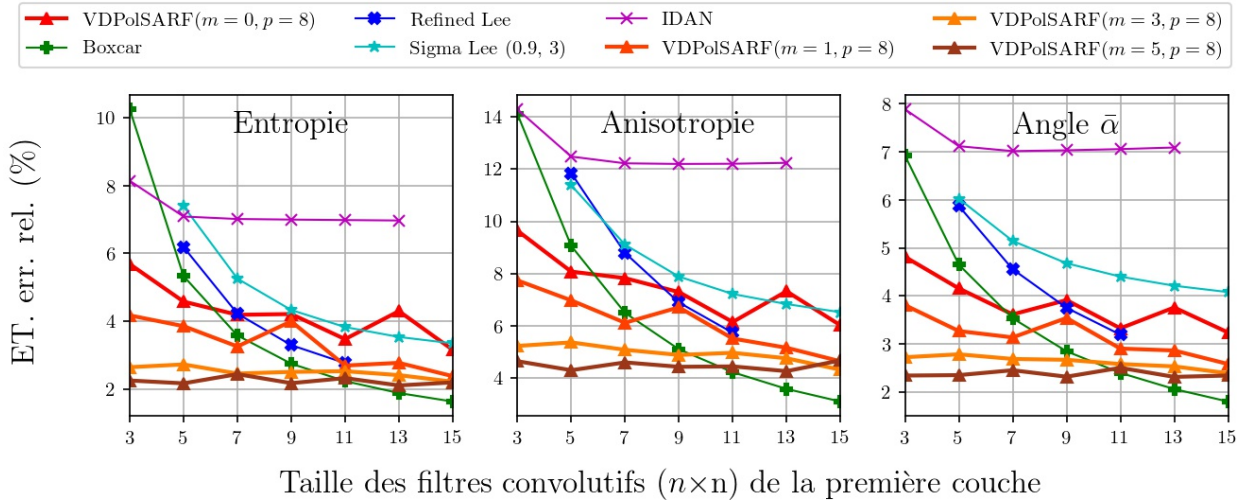
### Double Réflexion (Diffuseurs double bond)



**Fig. A.71. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

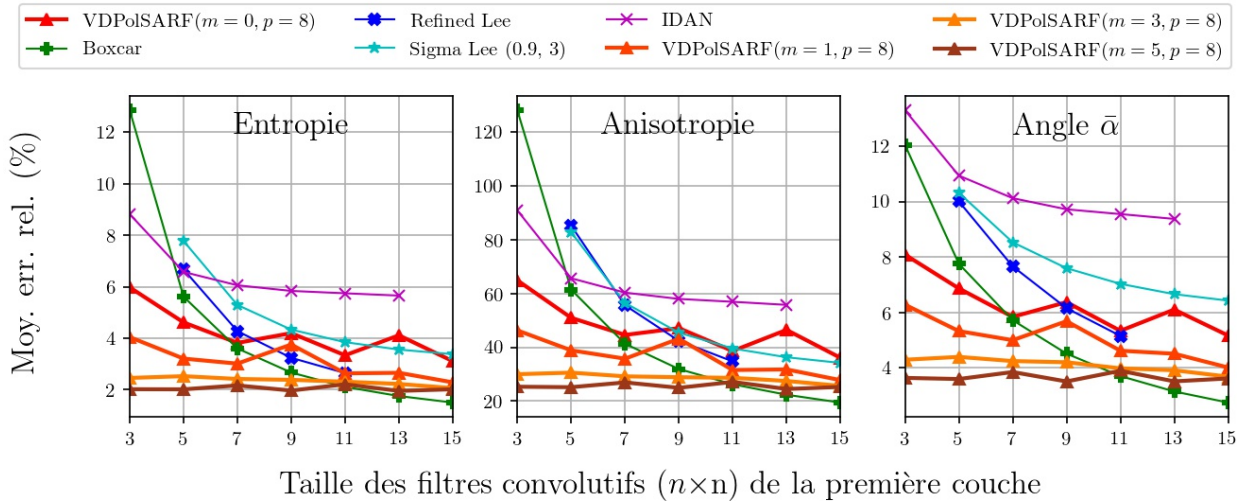


### Double Réflexion (Diffuseurs double bond)



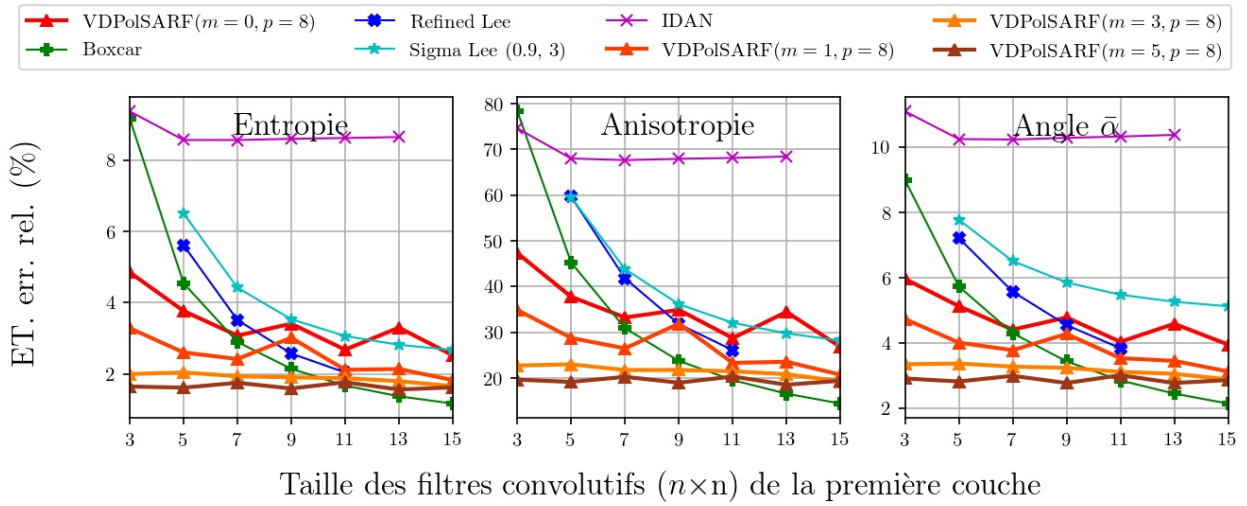
**Fig. A.72. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



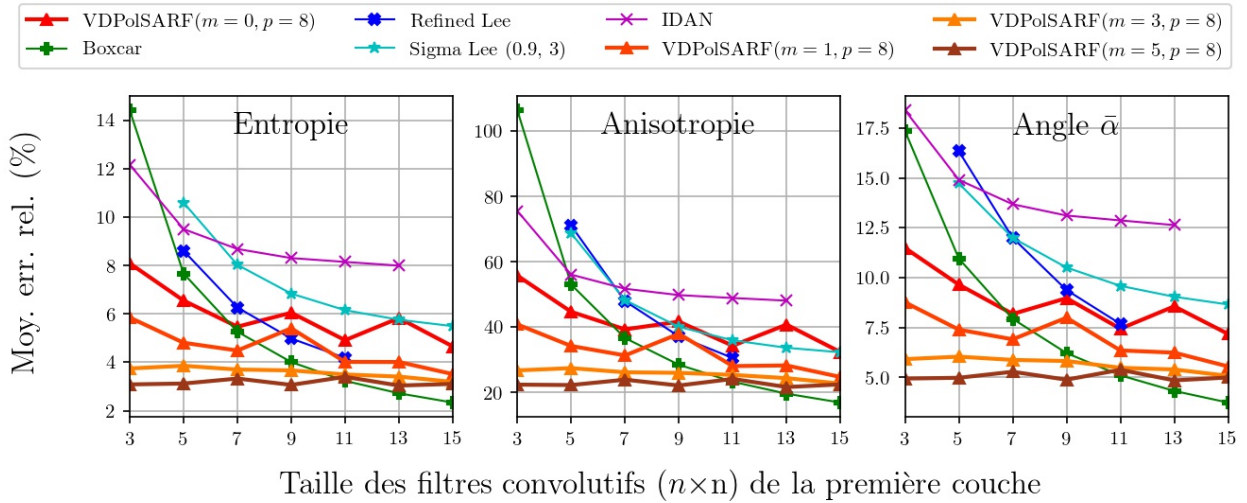
**Fig. A.73. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



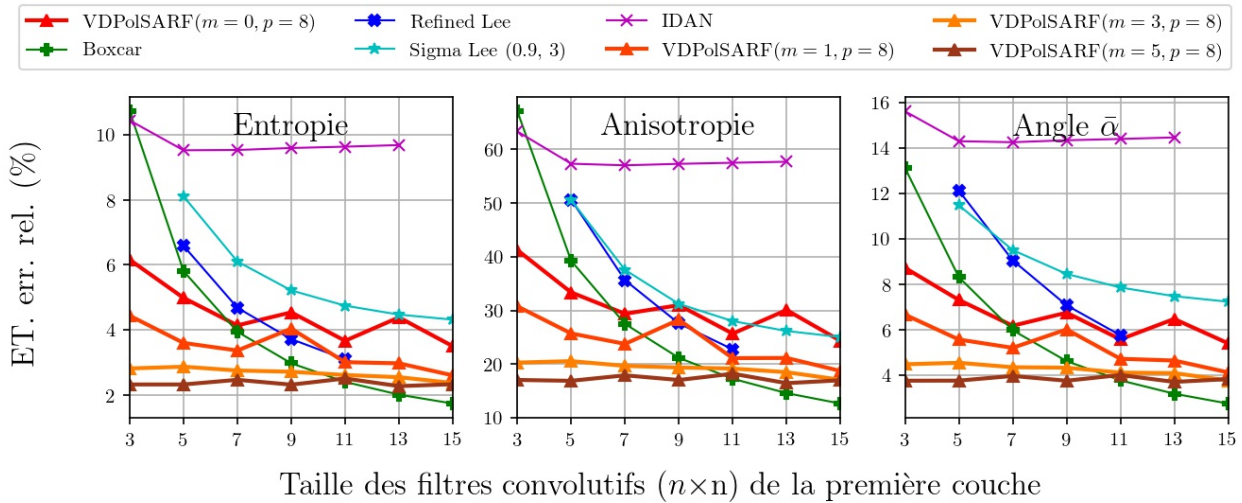
**Fig. A.74. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



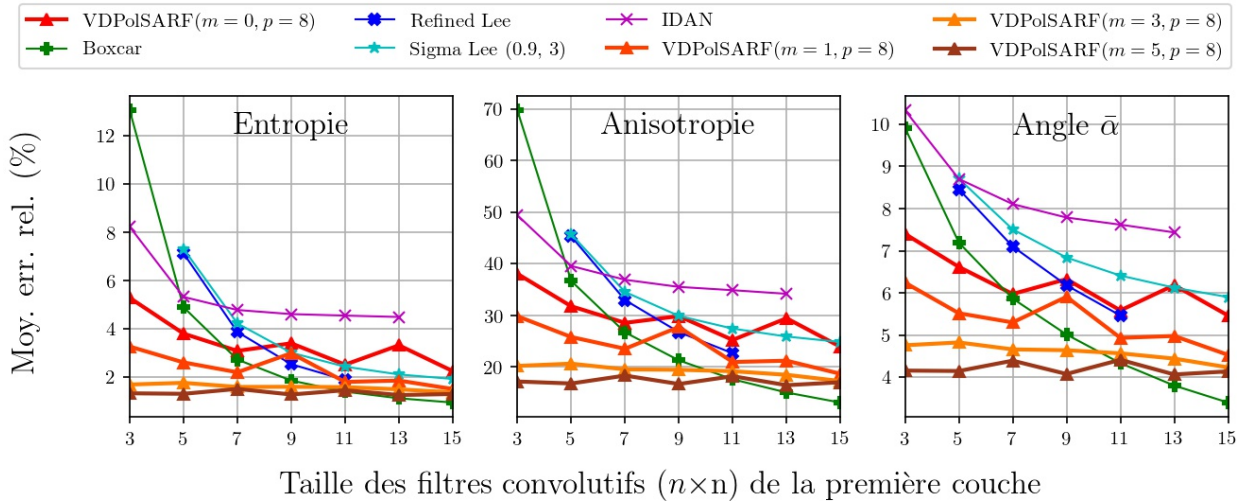
**Fig. A.75. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



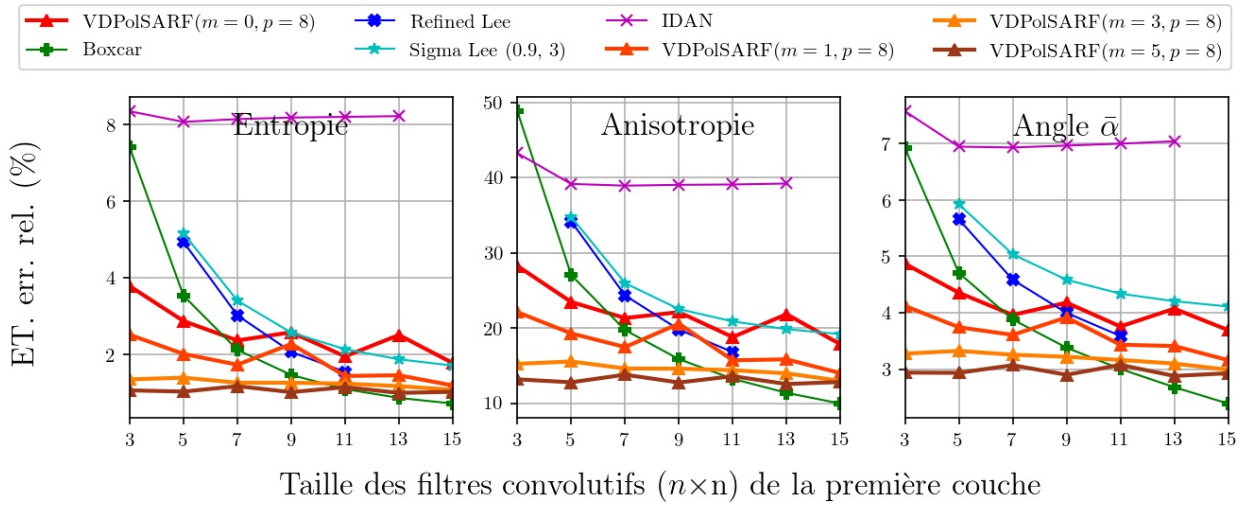
**Fig. A.76. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Structures Complexes



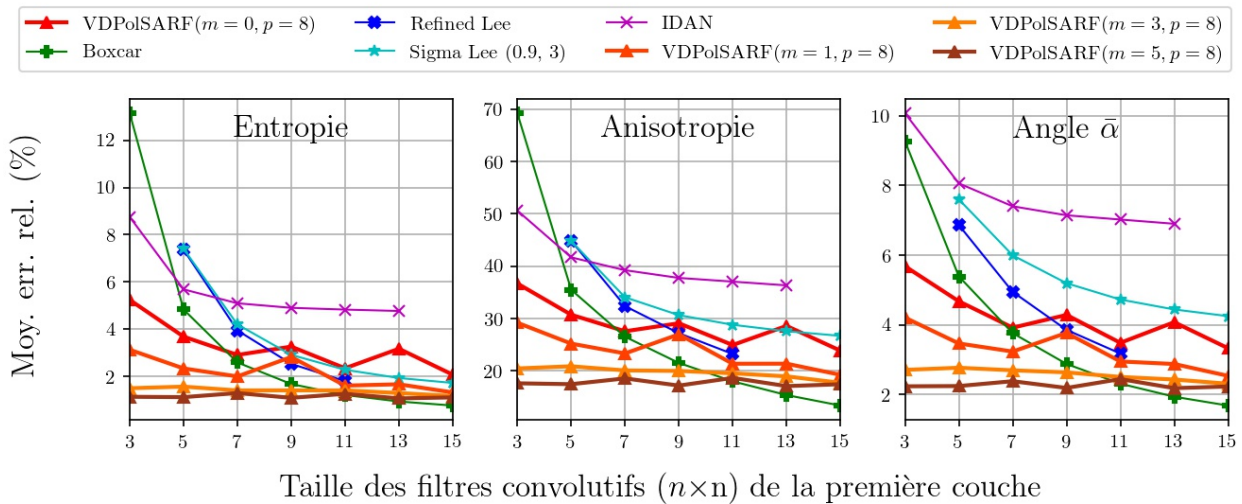
**Fig. A.77. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Structures Complexes



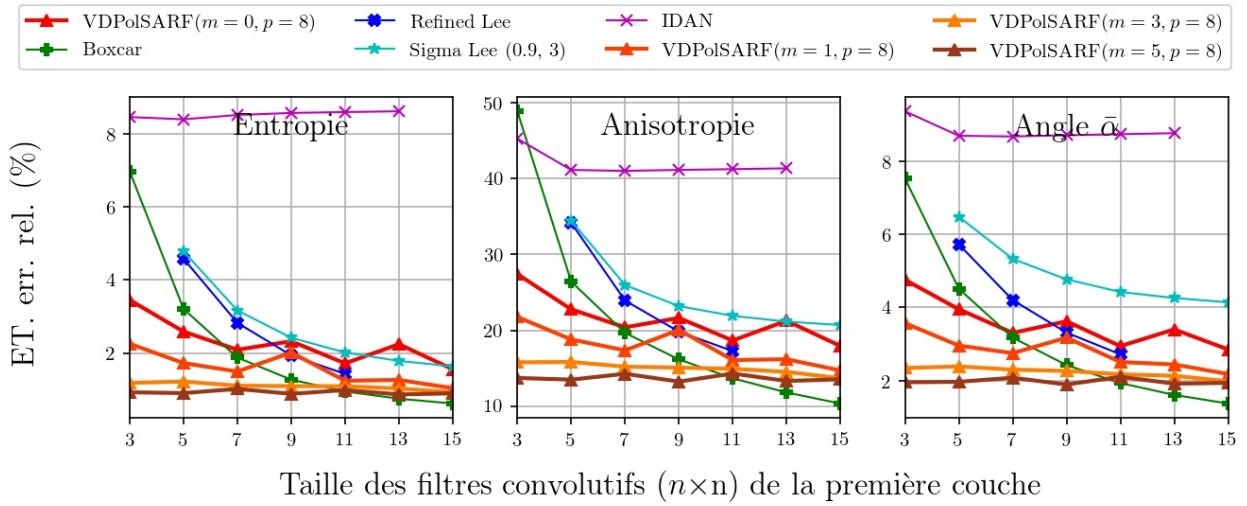
**Fig. A.78. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



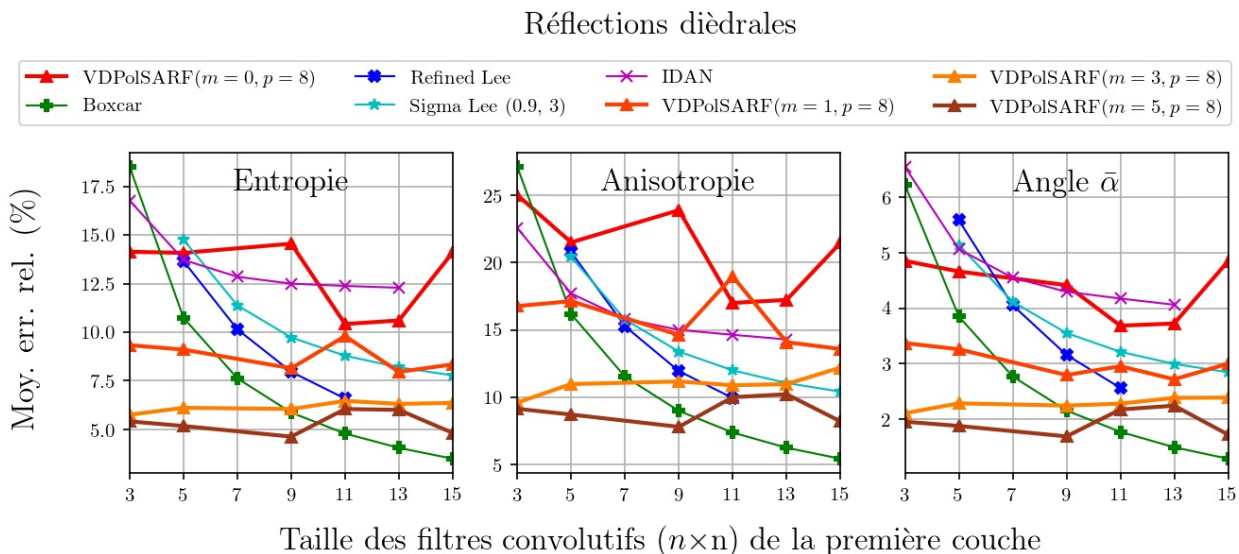
**Fig. A.79. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



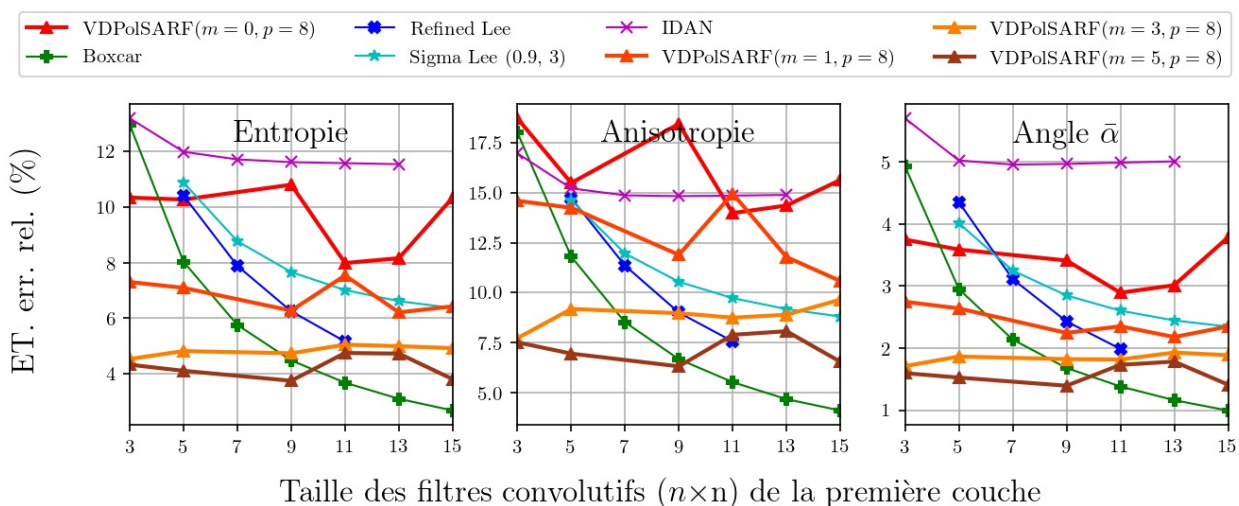
**Fig. A.80. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

A.6. Les résultats sur l'estimation des paramètres de la décomposition H-A-Alpha sur les données simulées homogènes par les modèles entraînés sur les données simulées hétérogènes avec inclusion de cibles ponctuelles.



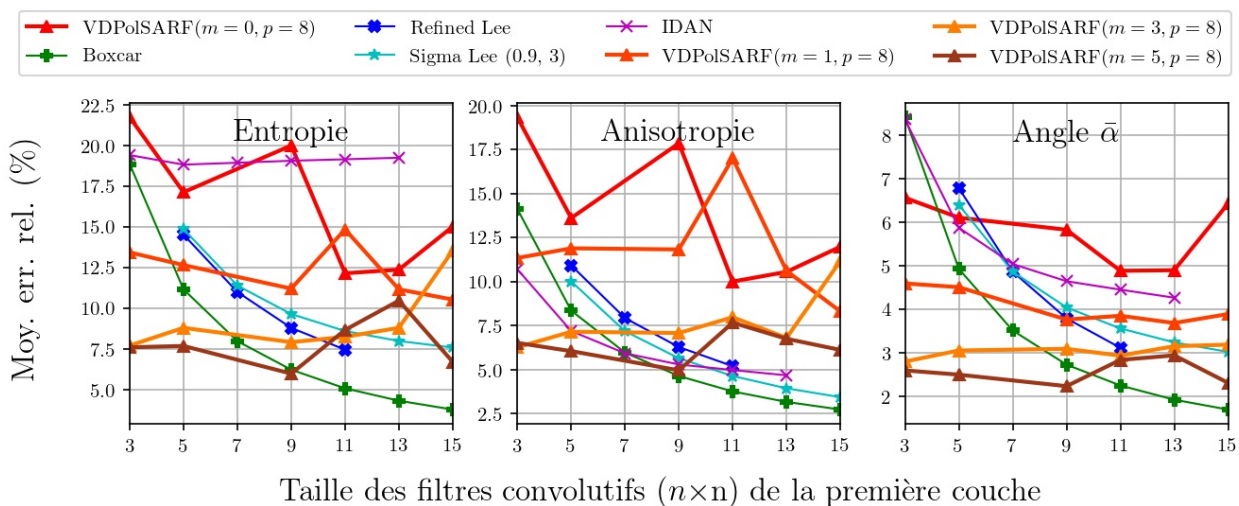
**Fig. A.81. Réflexions diédrales.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Réflexions diédrales



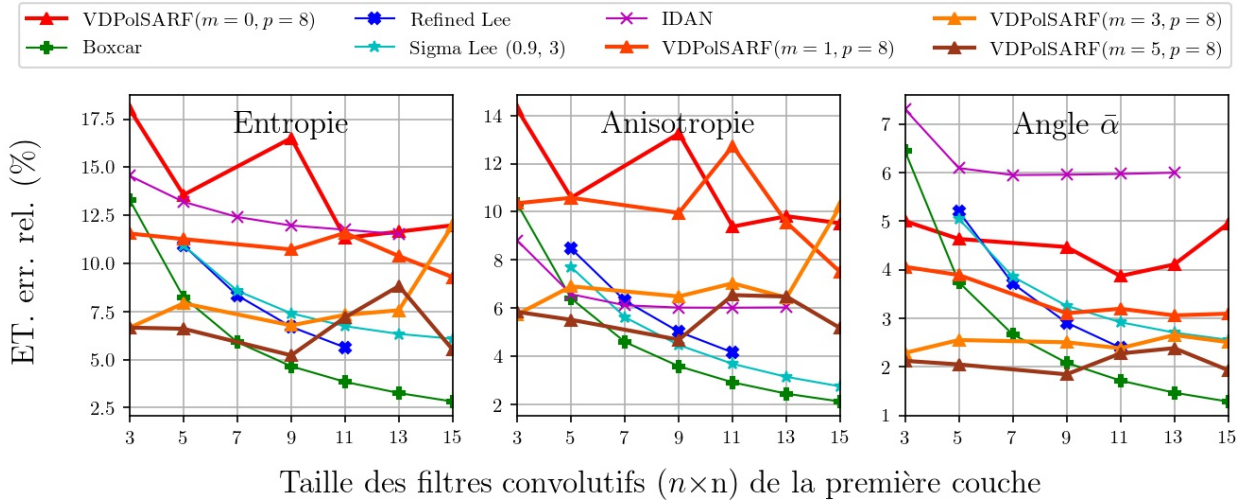
**Fig. A.82. Réflexions diédrales.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Dipôles



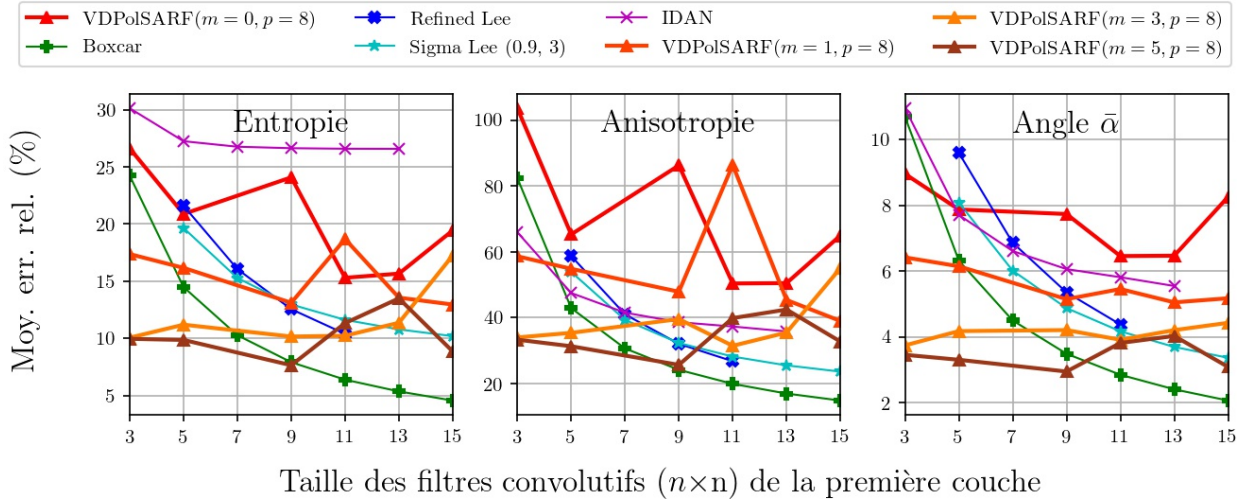
**Fig. A.83. Dipôles .** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Dipôles



**Fig. A.84. Dipôles** . Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

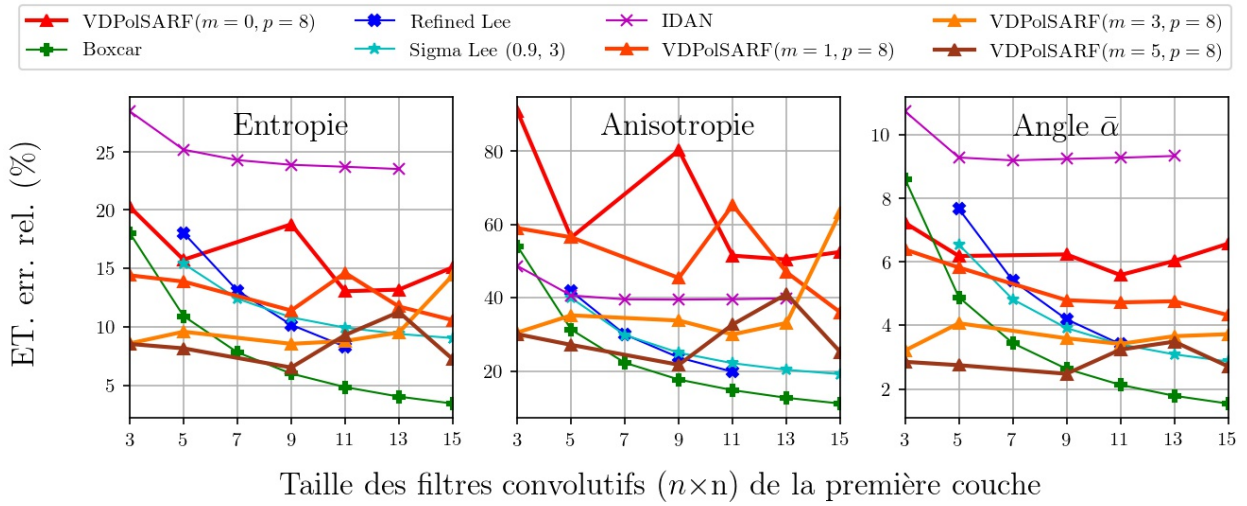
### Surfaces de Bragg (Diffuseurs surfaciques)



**Fig. A.85. Surfaces de Bragg** . Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

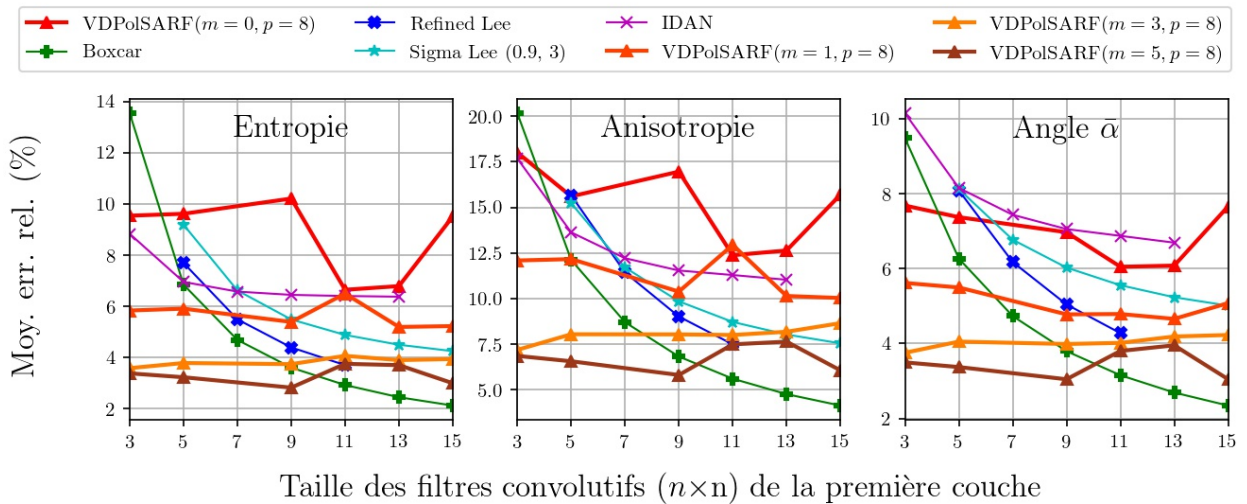


### Surfaces de Bragg (Diffuseurs surfaciques)



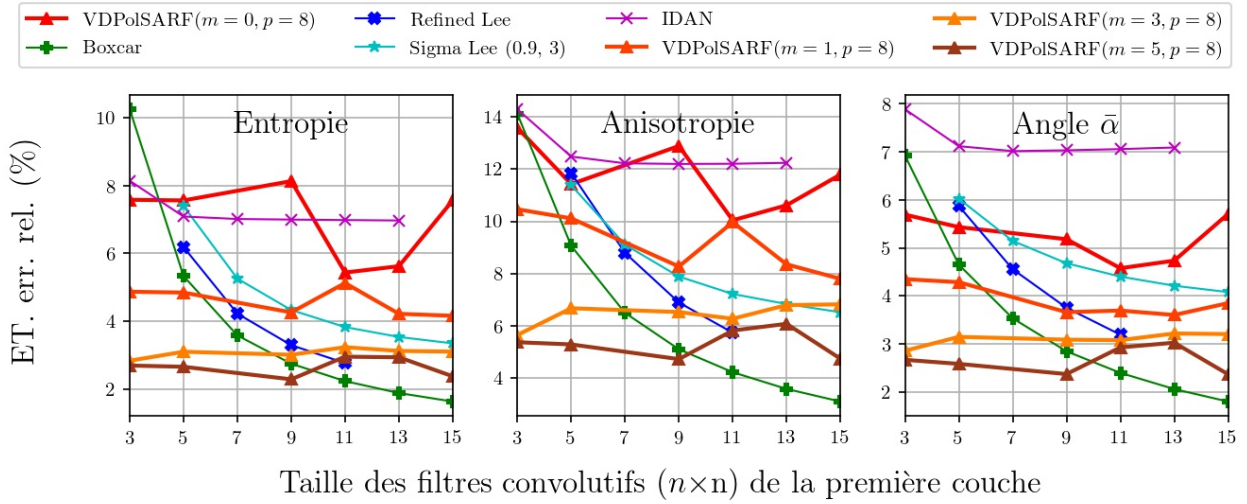
**Fig. A.86. Surfaces de Bragg .** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



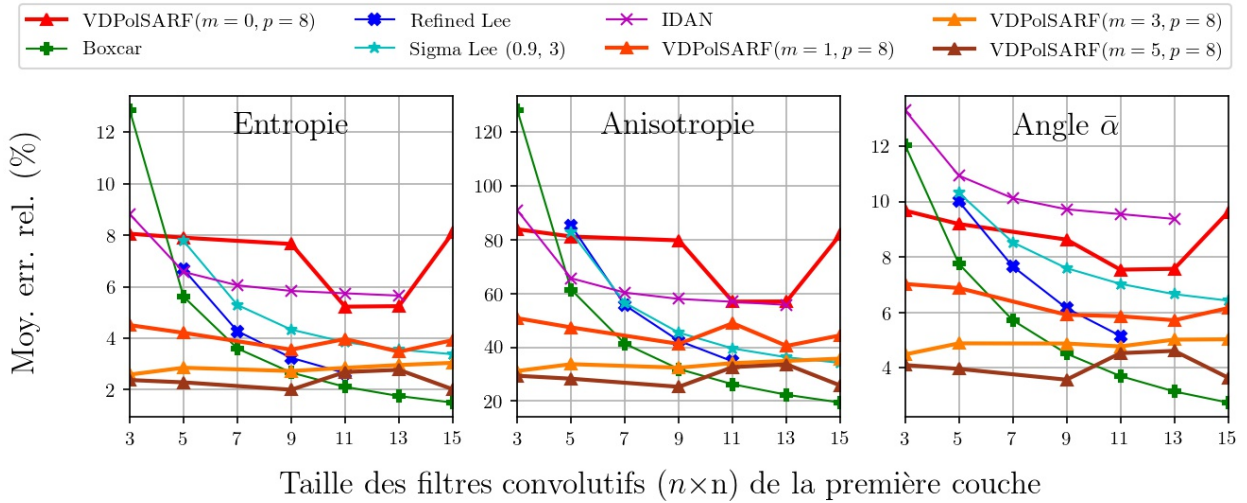
**Fig. A.87. Double Réflexion.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Double Réflexion (Diffuseurs double bond)



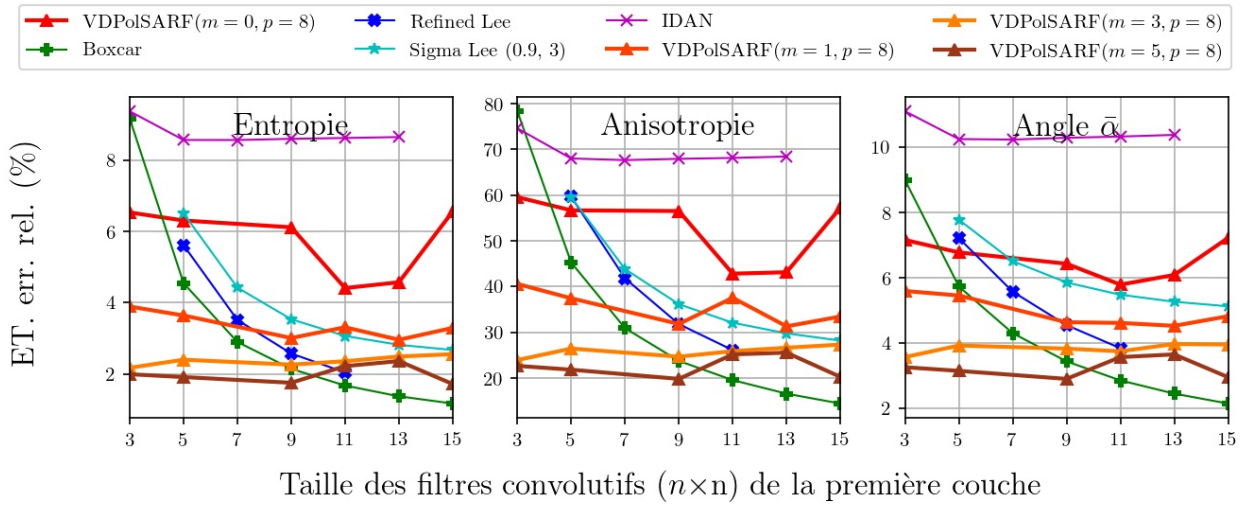
**Fig. A.88. Double Réflexion.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



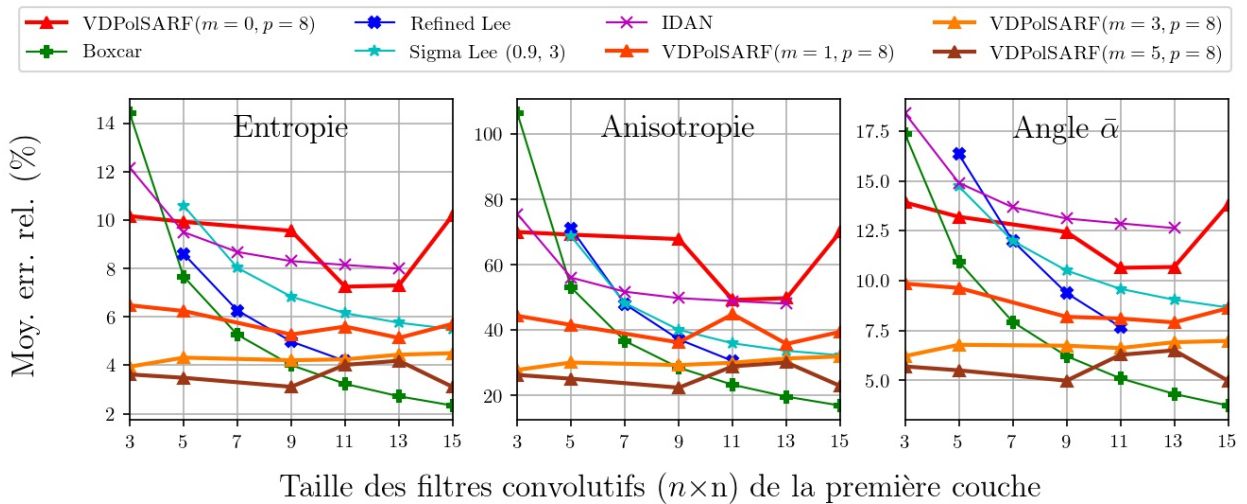
**Fig. A.89. Particules anisotropiques.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Particules Anisotropiques (Diffuseurs volumétriques)



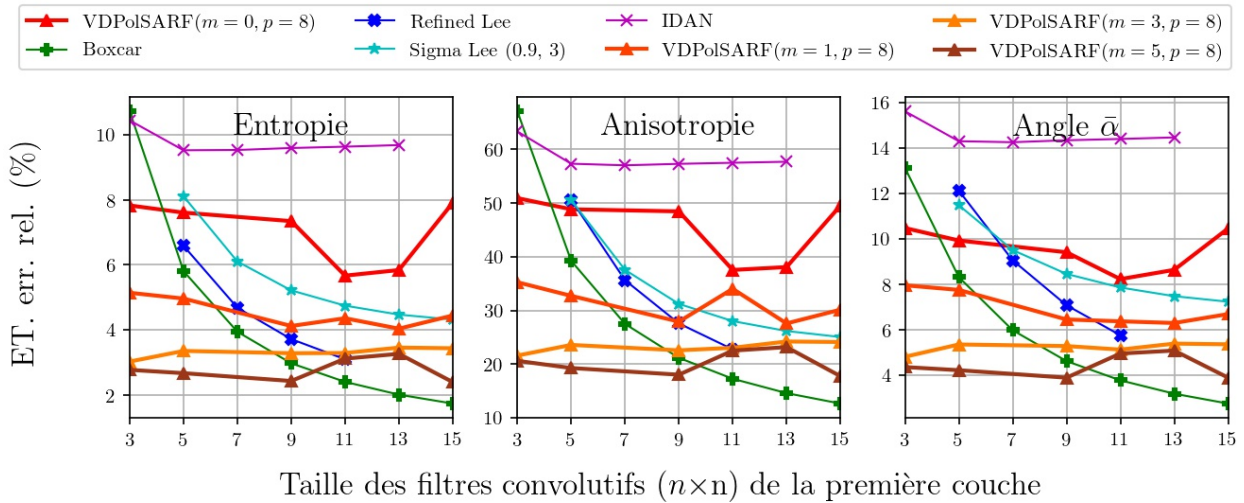
**Fig. A.90. Particules anisotropiques.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



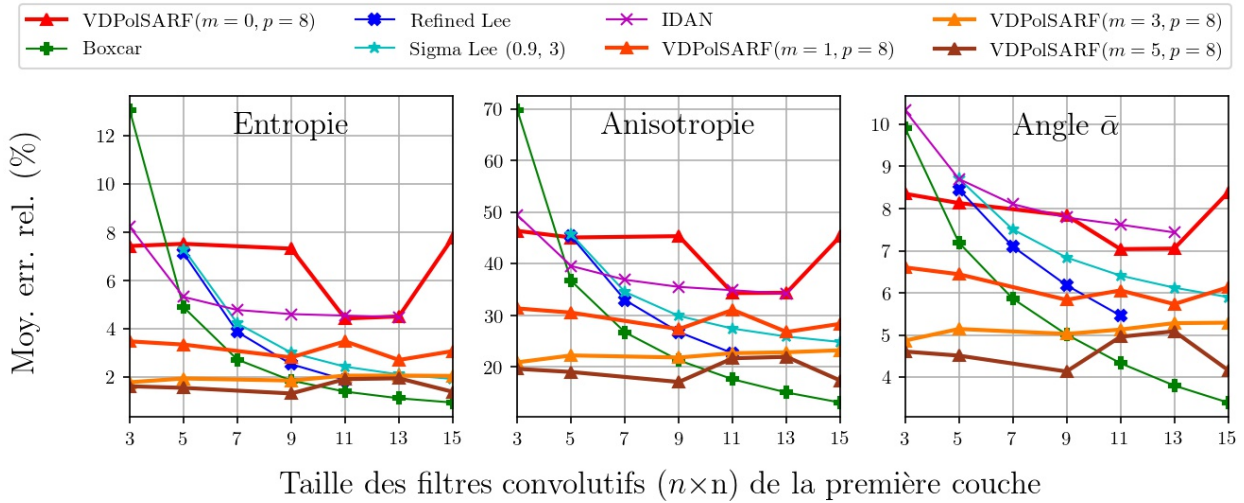
**Fig. A.91. Surfaces aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Surface aléatoires



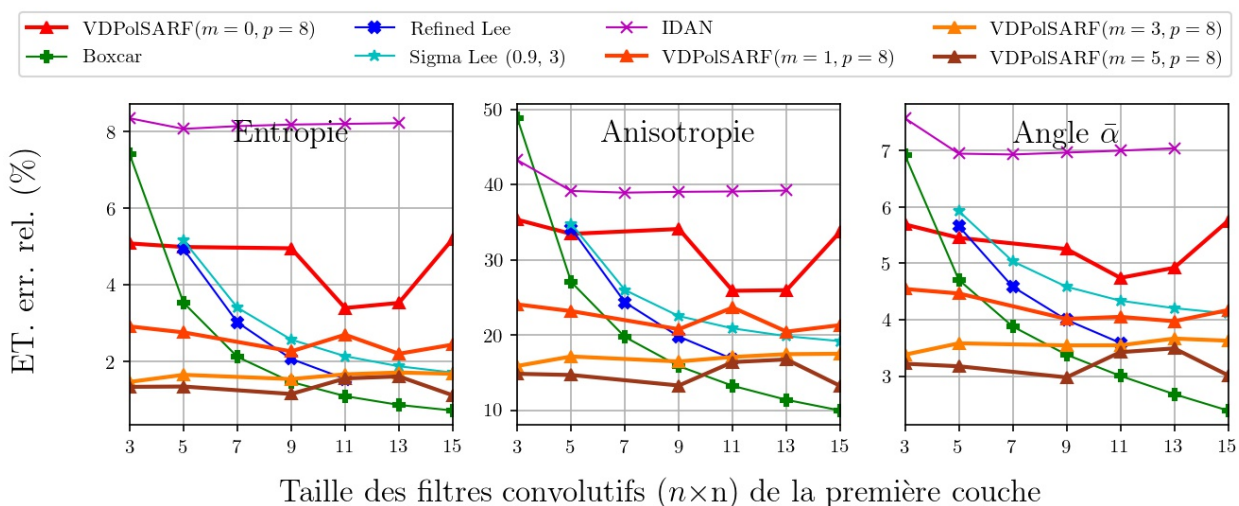
**Fig. A.92. Surfaces aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

### Structures Complexes



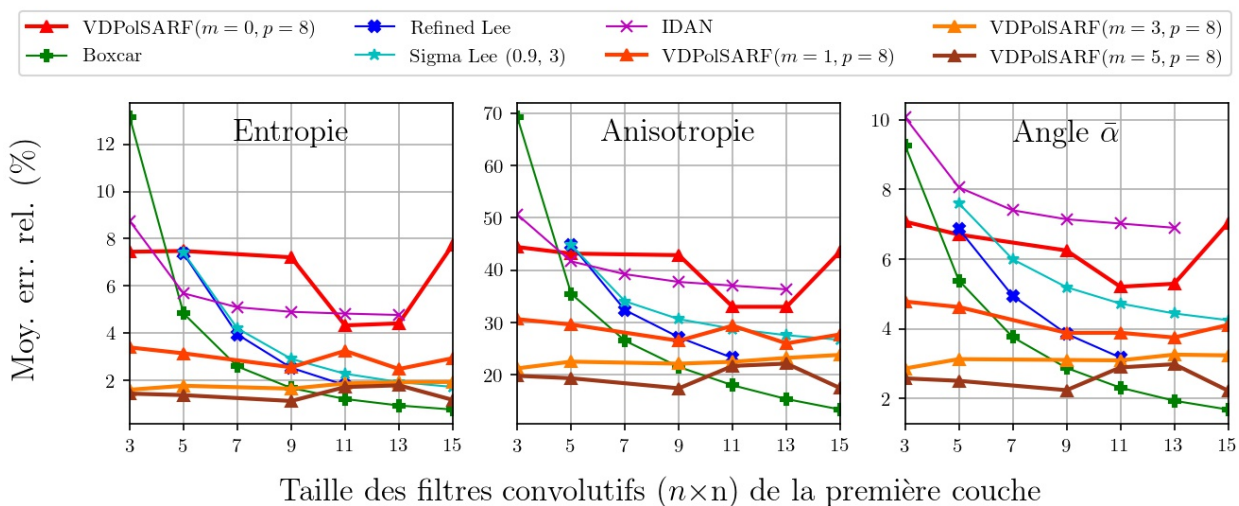
**Fig. A.93. Structures complexes.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Structures Complexes



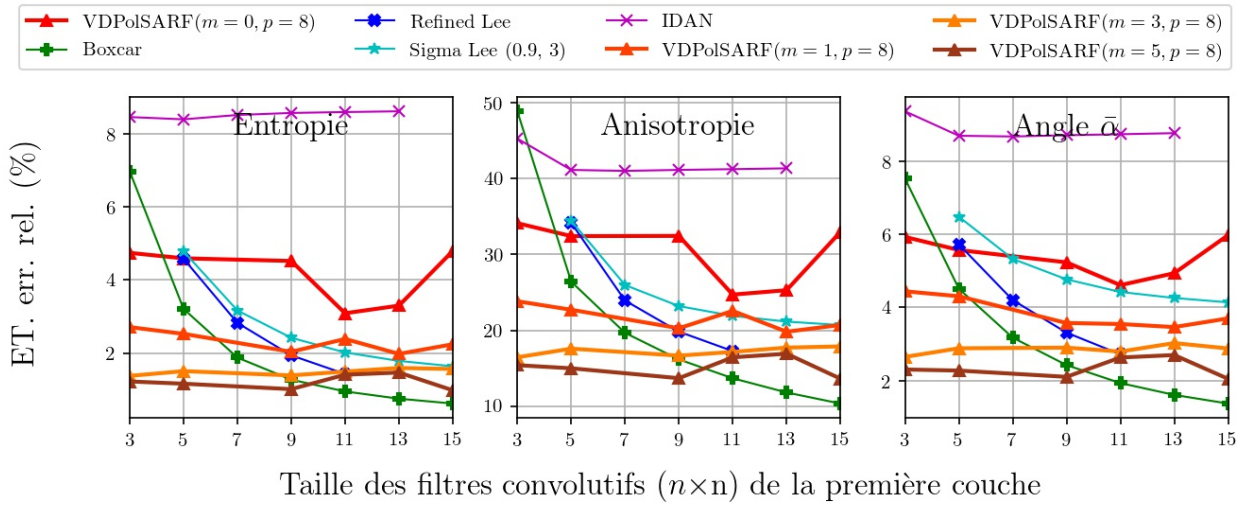
**Fig. A.94. Structures complexes.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



**Fig. A.95. Diffuseurs anisotropiques aléatoires.** Moyenne des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .

## Diffuseurs Anisotropiques aléatoires



**Fig. A.96. Diffuseurs anisotropiques aléatoires.** Écart-type des erreurs relatives (%) des estimés de la décomposition  $H/A/\bar{\alpha}$  en fonction de la taille ( $n$ ) des filtres de la première couche convolutive pour la famille des modèles  $m = \{0,1,3,5\}$  et  $p = 8$ .



