

# Statistiques sur générations WCRG et comparaisons avec d'autres modèles

Jean-Eric Campagne, Etienne Lempereur

3 octobre 2023

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Modélisation WCRG sur les données WL-1</b>	<b>3</b>
2.1	Phase d'apprentissage . . . . .	3
2.2	Phase de synthèse . . . . .	7
<b>3</b>	<b>Réduction d'information par l'usage de statistiques</b>	<b>8</b>
<b>4</b>	<b>Résultats concernant les données WL-1</b>	<b>10</b>
<b>5</b>	<b>Modèles WCRG et données WL-2 et <math>\phi_4</math></b>	<b>12</b>
<b>6</b>	<b>Discussion</b>	<b>18</b>
<b>7</b>	<b>Résumé</b>	<b>21</b>
	<b>Références</b>	<b>23</b>
<b>A</b>	<b>Plots de contrôle...</b>	<b>23</b>

# 1. Introduction

Dans cette note est exposée une série de statistiques calculées sur des images générées par une modélisation WCRG décrite dans la référence [Guth et al. \(2023\)](#) en les comparant aux mêmes statistiques calculées, non seulement avec les images ayant servies au training du modèle, mais aussi avec des images générées par un DGAN, et un modèle "micro-local". Brièvement, les données utilisées par la suite sont les suivantes:

WL-1 : il s'agit de cartes de Weak Lensing<sup>1</sup> issues d'un modèle Deep GAN (CosmoGAN) de la référence [Mustafa et al. \(2019\)](#). L'idée première de l'étude était de comparer les cartes générées par ce DGAN et celles d'un modèle WCRG. Les auteurs de l'article nous ont fourni, pour le training 100,000 images 128x128 pixels, tout comme un lot d'images générées par le réseau CosmoGAN. Ainsi, nous n'avons donc pas ré-entraîné un DGAN à cette occasion. Nous avons extrait 5,000 images<sup>2</sup> pour l'entraînement d'un modèle WCRG (noté par la suite WCRG-WL1). Nous détaillons un peu plus le traitement dans la section 2.

WL-2 &  $\phi_4$  : A la suite de la comparaison entre le modèle WCRG-WL1 et le DGAN, des questions se sont posées. Nous avons donc repris les images d'entraînement et synthétisées des modèles de la référence [Guth et al. \(2023\)](#) pour calculer les statistiques que ce soit pour des images de type WL ou de type  $\phi_4$  avec différentes valeurs de  $\beta = \{0.5, 0.68, 0.76\}$ . Cela est détaillé dans la section 5

Les statistiques utilisées sont principalement issues du code python *scattering\_transform* développé par Siho Cheng<sup>3</sup> et utilisé entres autres dans les articles [Cheng & Ménard \(2021\)](#); [Cheng et al. \(2023\)](#). Ces statistiques concernent, les power, bi et tri spectra ainsi que les coefficients de corrélation  $C_{00}$  et  $C_{01}$ . Concernant les données de type WL, la statistique du "comptage de pics" a également été utilisée. Ces statistiques sont décrites dans la section 3 et les résultats de leurs calculs avec les différents lots de données sont présentés dans les sections 4 et 5.

Concernant la modélisation WCRG, le code python utilisé est celui du repository mentionné dans la publication afférente<sup>4</sup>.

---

1. La référence utilise des résultats de la simulation N-corps GadGet2.

2. C'est la statistique utilisée dans la référence ([Guth et al. 2023](#))

3. [https://github.com/SihaoCheng/scattering\\_transform](https://github.com/SihaoCheng/scattering_transform) commit Fri Jun 2 11:14:53

4. <https://github.com/Elempereur/WCRG/>

## 2. Modélisation WCRG sur les données WL-1

### 2.1 Phase d'apprentissage

Avant l'entraînement du modèle WCRG, un traitement des données que l'on peut qualifier de "gaussianisation" a été effectué. En effet, sur la figure 1, on peut remarquer que la distribution des valeurs des pixels est mono-mode et peut être transformée en distribution normale par la méthode Cox-Box suivie d'une standardisation classique<sup>5</sup>. Le résultat de la transformation globale est illustré sur la figure 2. Cette transformation globale (invertible) est appliquée sur tous les lots de d'images (entraînement et génération) du DGAN.

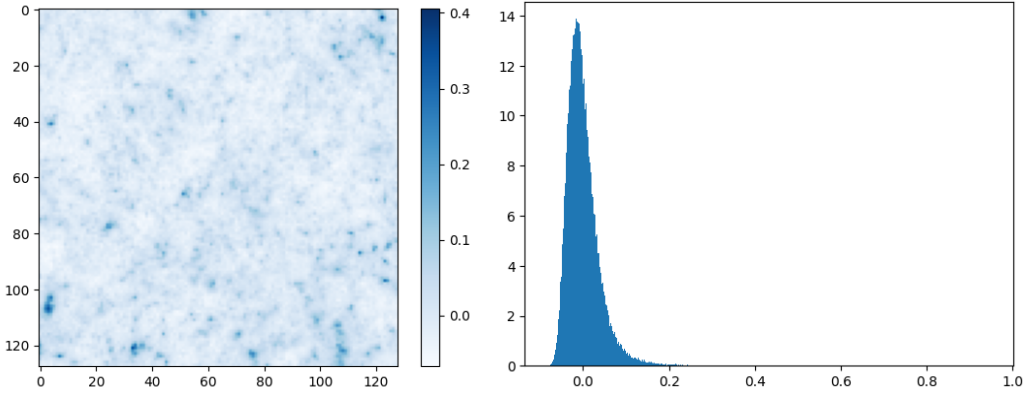


FIGURE 1 – Exemple d'une image du lot WL-1 (gauche) et la distribution afférente des valeurs de pixels (droite).

Concernant l'optimisation du modèle WCRG<sup>6</sup>, la démarche est très proche de celle documenté sur le repository du code, en particulier le notebook *Learning\_Models.ipynb*

---

5. Si  $x$  est la valeur d'un pixel original alors primo par l'optimisation Cox-Box on trouve  $\lambda^* \approx -0.22$ , et  $x' = (x^{\lambda^*} - 1)/\lambda^*$  suit une loi gaussienne, puis la standardisation classique permet d'obtenir une distribution normale  $x' \sim \mathcal{N}(0, 1)$ .

6. Notons que l'optimisation du modèle pourrait sans doute se passer de l'opération de gaussianisation préalable, mais l'idée était que cela pourrait aider *a priori* de ne pas avoir à traiter des queues de distributions trop asymétriques. D'un point de vue pratique, les queues de loi demandent un fine tuning fin des positions et largeurs des potentiels scalaires, qui ne sont pas très robustes au changement de queue (Il faut ajuster les potentiels scalaire selon les quantiles de la queue). Gaussianiser permet quasiment d'utiliser des potentiels uniformément positionnés et de moins fine-tuner.

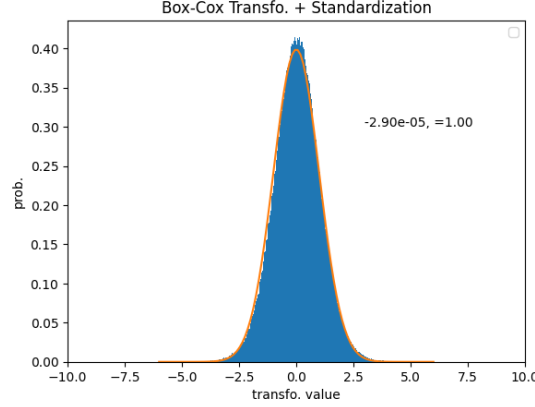


FIGURE 2 – Distribution des valeurs de pixels de la figure 1 après application d’une transformation Cox-Box de paramètre  $\lambda \approx -0.22$  et d’une standardisation classique.

moyennant quelques adaptations décrites ci-après<sup>7</sup>.

- Les ondelettes utilisées sont de la familles Debauchies d’ordre 4 avec un padding périodique.
- Pour l’ansatz  $E(x_J)$  ( $J = 7$ ,  $L = 1$ ) a été utilisée la fonction

```
def ANSATZ_NoCondi(L,centers,sigma,shifts,shifts_sym = False):
    """Non conditionnal ansatz for direct estimation of energy, a scalar
    potential (sigmoids) + a quadratic potential

    Parameters:
    L (int): system size = L*L
    centers (tensor): position of the centers of the sigmoids
    sigma (tensor) : width of the sigmoids
    shifts (list of tuples) : spatial shifts for the quadratic potential,
    carefull (0,0) is already taken into account, do not add here
    shifts_sym (Bool) : if True, the shifts are not symetrized
    """
```

avec 20 sigmoïdes régulièrement espacées entre le min et le max de la distribution des pixels  $\approx [-100, 100]$ , et par ailleurs `shifts = ()`). L’optimisation SGD est menée après avoir normalisé le hessien.

- A toutes les échelles  $L > 1$ , nous avons utilisé la fonction

```
def ANSATZ_Wavelet(W,L,centers,sigma,mode,shifts,shifts_sym = False):
    """ Conditionnal ansatz for conditonal Energy \bar E(\bar x_j\vert x_{j})
    estimation with a wavelet transform,with a scalar potential (sigmoids) + a
    quadratic potential
```

7. Nb. Même si la suite pourrait paraître fastidieuse, une certaine exhaustivité s’impose néanmoins, surtout que les codes n’ont pu être tournés sur notebooks partageables

```

Parameters:
W (Wavelet) : Wavelet to perform fast wavelet transform
L (int): system size ( of  $x_{j-1}$ ) =  $L \times L$ 
centers (tensor): position of the centers of the sigmoids
sigma (tensor) : width of the sigmoids
shifts (list of tuples) : spatial shifts for the quadratic potential,
carefull (0,0) is already taken into account, do not add here
shifts_sym (Bool) : if True, the shifts are not symetrized
"""

```

Tout comme pour  $E_J$ , l'optimisation SGD est menée après avoir normalisé le hes-sien.

- A l'échelle  $L = 2$ , 30 sigmoïdes régulièrement espacées dans l'intervalle  $[-75, 75]$  sont utilisées, tandis que `extend = 1.0`. Les autres paramètres sont `mode = 'All'` et `shifts = ()`.
- A l'échelle  $L = 4$ , 30 sigmoïdes sont également utilisées mais cette fois réparties uniformément selon les quantiles avec  $q_{min} = 10^{-5}$ ,  $q_{max} = 1.0$  et `extend = 1.0`. De plus `mode = 'All'` tandis que `shifts = ((1,0),(0,1),(1,1))` conjointement à `shifts_sym = True`.
- A l'échelle  $L = 8$ , on utilise 40 sigmoïdes réparties uniformément selon les quantiles avec  $q_{min} = 10^{-5}$ ,  $q_{max} = 1.0$  et `extend = 1.0`. Par ailleurs outre `mode = 'All'`, nous avons utilisé la fonction suivante pour définir les shifts tels que `shifts = shift_modif(L//4)`.

```

def shift_modif(n):
    shifts = []
    for i in range(-n,n):
        for j in range(-n,n):
            if i==0 and j==0:
                pass
            else:
                shifts.append((i,j))
    return shifts

```

- A l'échelle  $L = 16$ , se sont 40 sigmoïdes uniformément réparties sur l'intervalle  $[-30, 30]$  que l'on utilise avec `extend = 1.0`. Par ailleurs `mode = 'All'` et `shifts = shift_modif(L//4)`.
- A l'échelle  $L = 32$ , on utilise comme précédemment 40 sigmoïdes uniformément réparties sur l'intervalle  $[-20, 20]$  (`extend = 1.0`). Les paramètres `extend`, `mode` et `shifts` ont les mêmes valeurs qu'à l'échelle précédente.
- A l'échelle  $L = 64$ , on utilise 30 sigmoïdes réparties uniformément selon les quantiles avec  $q_{min} = 10^{-5}$ ,  $q_{max} = 1.0$  et `extend = 1.0`. De plus si `mode = 'All'` comme précédemment, par contre `shifts = shift_modif(L//8)`.
- Enfin à l'échelle  $L = 128$  (taille des images d'entraînement), on utilise 30 sigmoïdes

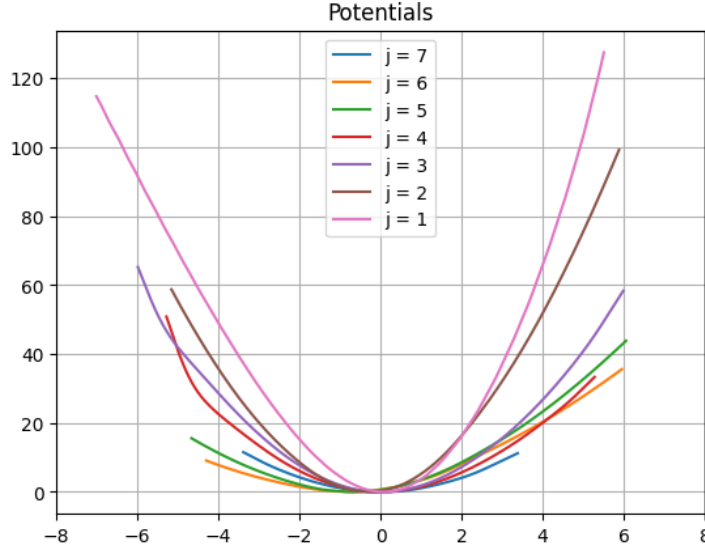


FIGURE 3 – Potentiels scalaires équivalents  $\bar{v}_j$  appris à chaque échelle  $j$  pour les images de WL-1.

uniformément réparties sur l'intervalle  $[-7.0, 5.5]$  (`extend = 0.25`), et si `mode = 'All'` comme précédemment, par contre `shifts = shift_modif(L//16)`.

- Les paramètres qui sont particulièrement délicats affectant la synthèse des images aux différentes échelles, sont ceux qui définissent les centres des sigmoïdes (l'intervalle  $[min, max]$  pour `linspace_centers`, ou  $[q_{min}, q_{max}]$  pour `quantile_centers`. La méthode `shift_modif` a été utilisée in fine mais ne change pas vraiment les résultats par rapport à l'usage de `shift_quad_Sym` utilisée de prime abord. Les optimisations SGD sont souvent reprises pour un peu tuner les paramètres tels que le nombre d'époques `num_epochs` et le learning rate `lr`. Parfois deux étapes d'optimisation ont été pratiquées, notamment avec une valeur de `lr` plus petite pour la seconde étape.

En pratique, le conditionnement de la loss de l'apprentissage dépend des potentiels scalaires. Ainsi, si ces potentiels ne se superposent pas trop le conditionnement est bon. En revanche, s'ils sont trop redondants le conditionnement diverge. Il faut alors être patient sur l'optimisation et diminuer le learning rate progressivement.

Concernant les paramètres qui définissent les sigmoïdes à toutes les échelles, ils ont été ajustés à la main largement pour que les histogrammes de contrôles lors de la phase de synthèse (voir la section suivante) montrent une bonne adéquation entre images d'origines et images de synthèse. La technique de sampling est en tout point identique à celle mise en œuvre pour l'article. Ce qui peut changer concerne le nombre de steps et la taille de

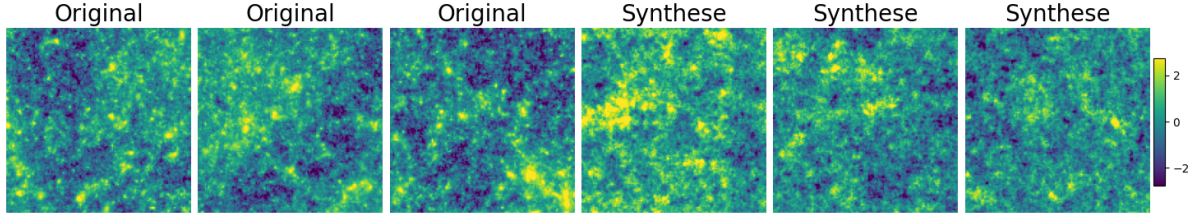


FIGURE 4 – Exemples de cartes WL-1 utilisées pour l’entraînement et synthétisées une fois le modèles WCRG optimisé.

chaque step. Notons que dans le cas d’une invariance parfaite d’échelle comme pour  $\phi_4$  au point critique, les hyper-paramètres sont les mêmes à chaque échelle (à normalisation par l’écart type de  $x_j$  près). Sans auto-similarité, il faut manuellement ajuster. On pourrait utiliser des règles ad-hoc satisfaisant pour fixer les hyperparamètres.

Une fois les paramètres "optimisés" (par ex. les sigmoïdes), les différents potentiels appris  $\bar{v}_j(x_{j-1})$  sont présentés sur la figure 3.

## 2.2 Phase de synthèse

Une fois le modèle appris, on peut regarder à chaque échelle la qualité des synthèses. Les histogrammes décrits ci-après sont dans l’appendix pour ne pas surcharger le corps du texte.

A l’échelle  $L = 1$ , la comparaison des distributions de valeurs de pixels entre données d’entraînement et synthétisées est montrée sur la figure 14. Aux échelles suivantes, outre la comparaison des distributions des valeurs de pixels des images originales et synthétisées, on peut également questionner la distribution des cartes de détails hautes fréquences (horizontales/verticales/diagonales) obtenues par décomposition en ondelettes. Les résultats aux échelles  $L \in [1, 128]$  sont présentés sur les figures de 15 à 21. Pour des raisons tenant à la taille mémoire des GPUs utilisés, le nombre de cartes générées jusqu’à l’échelle 16 incluse est de 5000, tandis que pour les échelles supérieures on ne peut disposer que de 500 cartes.

A l’échelle  $L = 128$ , on dispose de cartes synthétisées "finales" qui si tout se passe au mieux doivent ressembler comme deux gouttes d’eau aux cartes utilisées pour l’entraînement (ou cartes "originales"). La figure 4 donne quelques exemples de cartes originales et synthétisées. Il est indéniable que le modèle a appris quelque chose. Maintenant, dans l’article Guth et al. (2023), il est montré non seulement des cartes synthétisées, la distribu-

tion des valeurs de pixels<sup>8</sup> mais aussi la comparaison du spectre de puissance. Ce dernier point a motivé l'usage de statistiques afin de caractériser les distributions des pixels et de comparer ces statistiques obtenues sur les cartes originales et les synthétisées. Ceci est présenté dans la section suivante.

### 3. Réduction d'information par l'usage de statistiques

Un certain nombre de statistiques ont été utilisées afin d'aller au-delà de l'aspect visuel jugeant de la qualité de la synthèse des cartes de WL-1. Comme mentionné dans l'introduction, en premier lieu il a été utilisé le code développé par Siho Cheng afin de calculer:

- les classiques spectre de puissance, *bi*-spectre et *tri*-spectre;
- le coefficient de scattering  $S1_{iso}$ , ainsi que les coefficients composites  $s_{21}$  (sparsity) et  $s_{22}$  (shape) définis dans la référence [Cheng & Ménard \(2021\)](#);
- et les corrélations de scattering définies selon<sup>9</sup>

$$C_{01} = \langle (x * \psi_2)(|x * \psi_1| * \psi_2)^* \rangle / factor \quad (1)$$

$$C_{11} = \langle (|x * \psi_1| * \psi_3)(|x * \psi_2| * \psi_3)^* \rangle / factor \quad (2)$$

( $C_{01}$  et  $C_{11}$  sont des approximations quadratiques respectivement du bi-spectre et du tri-spectre ([Cheng et al. 2023](#))) avec  $x$  la carte de champ considéré (WL,  $\phi^4$ ) de training ou généré,  $\psi_i$  des ondelettes de Morlet à différentes échelles et orientations et où le choix de normalisation est défini par

$$factor = L2(x * \psi_1) \times L2(x * \psi_2) \quad (3)$$

Les définitions de ces facteurs sont décrites dans l'article [Cheng et al. \(2023\)](#):  $C_{01}$  et  $C_{11}$  correspondent respectivement à  $\tilde{S}_3$  et  $\tilde{S}_4$ , et la version "isotropique" prise par défaut opère une réduction sur les paires d'angles<sup>10</sup>.

Un point à noter concerne le type des ondelettes utilisées pour le modèle WCRG et pour le calcul des statistiques des images générées. Les représentations  $(\bar{x}_i, x_J)$  et  $(x * \psi_i, x * \phi_J)$  sont bien deux représentations en ondelettes, qui sélectionnent des bandes fréquences localisées spatialement, mais la première est réelle, orthogonale et utilise des ondelettes

---

8. nb. le modèle de l'article a été optimisé directement sur les cartes de WL-2 sans transformation préalable donc on ne peut comparer les modèles. Cela sera sans doute à revoir ultérieurement.

9.  $\langle x \rangle = \text{Ave}_u(x(u))$ . Considérant les échelles des ondelettes  $j_1 \leq j_2 \leq j_3$ .

10. Nb. nous avons vérifié que les effets que nous avons constatés et retranscrits ci-après sont également mis en évidence avec les versions *non-isotropiques*.



de Daubechies ?, la seconde est redondante, complexe et utilise des Morlets ?. Ainsi la première représentation est sous-échantillonnée, la seconde non.

L'information contenue dans la phase de  $x * \psi_j$  est une information spatiale ?. Schématiquement, pour un  $\bar{x}_j$  cosinus,  $x * \psi_j$  sera une exponentielle complexe de même période. L'information contenue dans l'amplitude de  $\bar{x}_j$  est dans la phase de  $x * \psi_j$ .

On pourra faire le parallèle entre  $\bar{x}_i$  et  $\text{Re}(x * \psi_i)$  et  $x_J$  et  $x * \phi_J$ , ces cartes couvrant les mêmes bandes fréquences avec une localisation spatiale similaire. Dans ce cas, le modèle WCRG impose quasiment la covariance de  $\text{Re}(x * \psi_i)$  et les histogrammes de  $x * \phi_j$ .

En pratique afin de fixer un peu les idées, l'obtention des différents spectres et coefficients s'effectue schématiquement selon les étapes suivantes (si "maps" correspond à un lot de cartes):

```
bins_pw = 30
Nmaps,M,N = maps.shape
J = int(np.log2(min(M,N))) - 1
pw, kr = scattering.get_power_spectrum(maps,
                                       k_range=np.logspace(0,np.log10(M/2*1.4),
                                       bins_pw+1))
bi_calc = scattering.Bispectrum_Calculator(M,N,
                                       k_range=np.logspace(0,np.log10(M/2*1.4), J-1))
tri_calc = scattering.Trispectrum_Calculator(M,N,
                                       k_range=np.logspace(0,np.log10(M/2*1.4), J-1))

st_calc = scattering.Scattering2d(M, N, J=6, L=4)

coeffs = st_calc.scattering_coef(maps)
s1 = np.log10(coeffs['S1_iso'])
s21 = np.log10(coeffs['s21'])
s22 = np.log10(coeffs['s22'])

cov_coef = st_calc.scattering_cov(maps)
select_and_index = get_scattering_index(J, L, normalization='P00', C11_criteria='j2>=
j1')
c01 = cov_coef['C01_iso'][:,select_and_index['select_2_iso']]
c11 = cov_coef['C11_iso'][:,select_and_index['select_3_iso']]
```

De plus, afin de satisfaire des contraintes de taille de mémoire, les spectres/coefficients ont été calculés sur des lots de cartes dont on en a tiré des moyennes et écarts-types (barres d'erreur dans les histogrammes).

Dans le cas des cartes de WL, on a également utilisé le comptage de "pics" au dessus d'un seuil. Il s'agit d'une statistique d'ordre supérieur utilisée pour la recherche de non-gaussianité qui s'est répandue récemment (voir la référence [Lanzieri et al. \(2023\)](#) et les

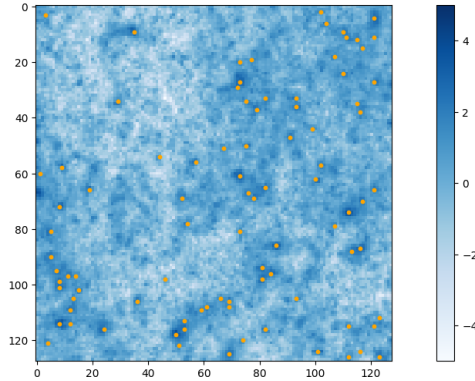


FIGURE 5 – Exemples de la recherche de pics au dessus d'un seuil ici pris à 2.

références incluses). Pour se faire la méthode détaillée dans l'article a été utilisée<sup>11</sup> avec: à la base la fonction `find_peaks2d_tf` du package `lenspack`<sup>12</sup> qui cherche la localisation des pics ainsi que leurs "hauteurs", suivie d'un lissage par un noyau gaussien. Le résultat est donc un histogramme de la distribution smoothée des hauteurs de pics d'une carte (ou d'un lot de cartes). Un exemple de recherche de pics est donné sur la figure 5.

## 4. Résultats concernant les données WL-1

Une fois le modèle WCRG obtenu selon la méthode détaillée à la section 2.1, on peut calculer les statistiques décrites dans la section précédente non seulement sur des cartes générées (Sec. 2.2) mais aussi sur des cartes ayant servi à l'apprentissage du modèle. Les résultats sont présentés sur la figure 6 sous les appellations "wcrG" (rouge) et "train" (bleu) (*nb. les autres modèles mentionnés dans l'introduction le DGAN et le "micro-canonique" sont également présentés par anticipation*). Il est assez remarquable que la quasi totalité des distributions issues de la génération de cartes par le modèle WCRG soient bien, voire très bien reproduites. Donc, il n'est pas étonnant que les différences que nous pointons par la suite n'aient pas été mises en évidence dans l'article Guth et al. (2023).

Le modèle WCRG contraint de façon **macro**-canonique le spectre, et des histogrammes à chaque échelle. Les statistiques utilisées pour la comparaison n'interviennent donc pas directement dans la modélisation, mais sont vraisemblablement incluses dans les potentiels scalaires. Ceux-ci mélangent non-linéairement les pixels, via les termes  $x_j$ , qui peuvent être vues comme des moyennes sur des voisinages locaux de tailles  $2^j$ .

11. <https://github.com/LSSTDESC/DifferentiableHOS.git>

12. <https://github.com/CosmoStat/lenspack>

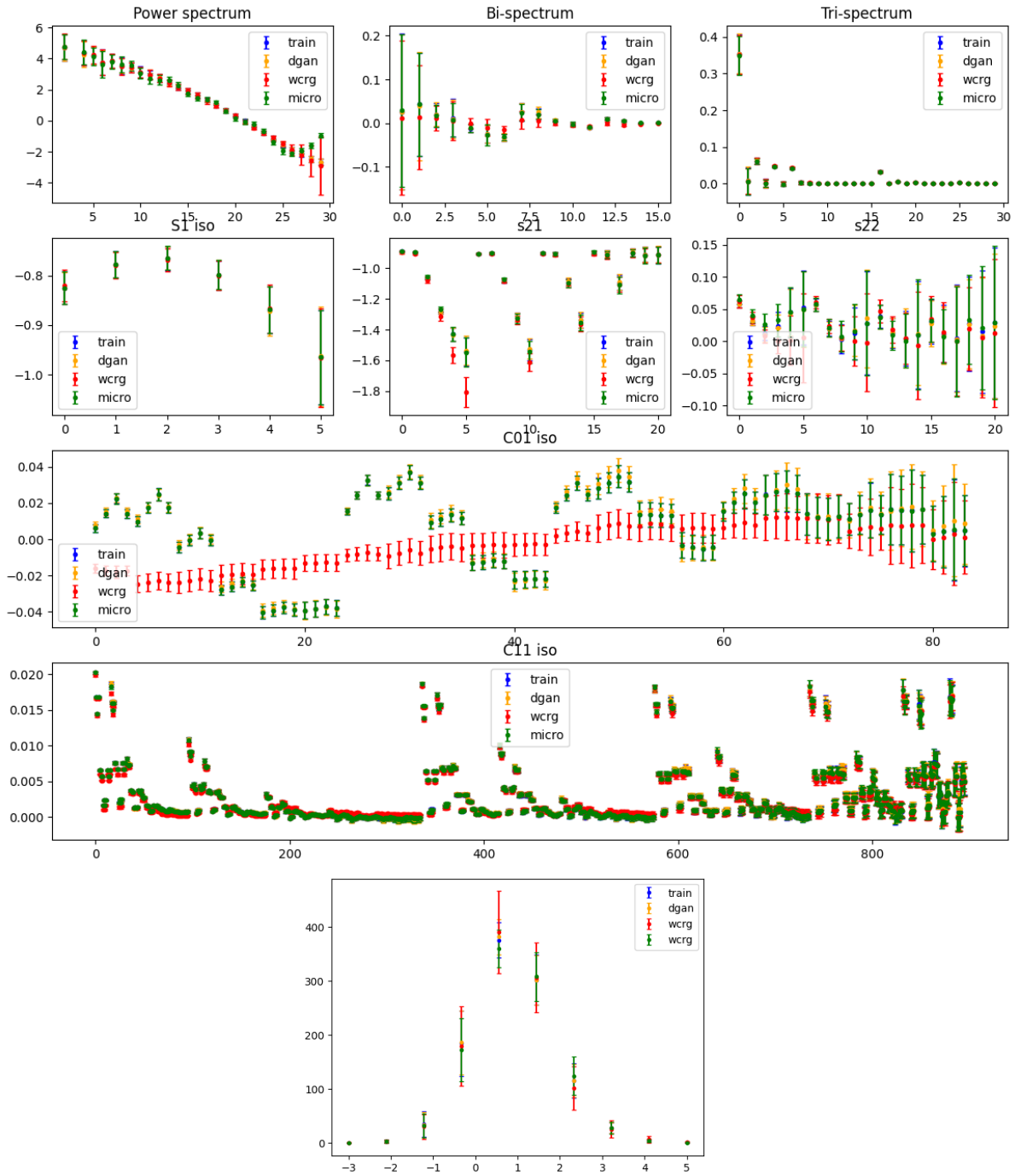


FIGURE 6 – Histogrammes des statistiques détaillées à la section 3. Il est a noté qu'excepté le comptage de pics, les distributions sont calculées avec 500 cartes.

Si on se penche sur les distributions qui dénotent, on constate que:

- en premier lieu ce qui questionne concerne la distribution  $C_{01}$  qui manifestement a un problème, alors que la distribution  $C_{11}$  est bien reproduite;
- dans une moindre mesure on constate aussi que le bi-spectre (resp.  $s_{21}$ ) est moins bien reproduit que le tri-spectre (resp.  $s_{22}$ ).

Ce sont ces deux points dont l'origine peut être commune qui a motivé les investigations des sections suivantes.

Notons que si l'on prend les cartes issues de la génération du réseau CosmoGAN mentionné en introduction, il y a un parfait accord de toutes les distributions "dgan" (orange) avec les distribution d'entraînement ("train") (Fig. 6). En ce sens, le GAN fait mieux sur ce critère<sup>13</sup>.

Nous avons également procédé à la synthèses de cartes en utilisant une modélisation **micro**-canonique (Cheng et al. 2023) à l'aide du code de Siho Cheng selon typiquement

```
st_calc.scattering_cov(...)['for_synthesis']
```

qui utilise les paramètres: le rapport  $\langle x \rangle / \sigma(x)$ ,  $L2(x * \psi_i)$ ,  $S_1$ , ainsi que les parties réelles et imaginaires des coefficients de corrélation  $C_{01}$  et  $C_{01}$ . Ce type de modélisation/synthèse notée "micro"(vert) sur la figure 6 reproduit très bien la quasi totalité des distributions exceptée<sup>14</sup> le spectre de puissance à grande valeur de  $k$  même si on note que les fluctuations des valeurs sont bien plus réduites que pour les autres modèles pourtant calculées avec le même nombre de cartes. Ceci dit, concernant les coefficients  $C_{01}$  l'accord est parfait, mais on s'en doutait puisqu'ils rentrent dans la liste des paramètres ayant servi pour définir la métrique à minimiser afin de produire le modèle micro-canonique.

Dans les sections suivantes, nous allons utiliser les cartes générées par les modèles WCRG ainsi que celles d'entraînement de type WL-2 et  $\phi_4$  décrites dans l'introduction, afin de savoir si ces modèles ayant été entraînés dans des conditions différentes performant mieux ou pas que celui envisagé dans cette section avec les données WL-1.

## 5. Modèles WCRG et données WL-2 et $\phi_4$

A la suite des résultats exposés à la section précédente, la question qui vient est de savoir pour quelles raisons la distribution  $C_{01}$  (et aussi bi-spectre et  $s_{21}$ ) est nettement

13. Rappel: à l'origine de l'étude était la comparaison "dgan", "wcrg", et dans ce contexte le nombre d'images ayant servi à l'entraînement du modèle WCRG est de 5,000 images alors que le DGAN en a utilisé 200,000 (Mustafa et al. 2019)

14. Si l'on ne porte que les distributions "train" et "micro", la remontée du spectre de puissance "micro" est bien visible.

moins bien reproduite par le modèle WCRG optimisé comme décrit à la section 2.1. Avant d’esquisser quelques remarques, nous avons utilisé directement (c’est-à-dire sans nouveau entraînement) les modèles et les données de l’article Guth et al. (2023) afin d’en calculer les statistiques de la section 3.

Concernant les données WL-2, on dispose de 500 cartes d’entraînement ("train") et de 209 cartes générées ("wcrgr"). Une note technique: nous avons procédé à un recalage de la distribution des valeurs de pixels des cartes générées afin que la moyenne soit égale à celle des cartes d’entraînement, mais nous n’avons pas opéré de gaussianisation de la section 2.1). Les résultats sont présentés sur la figure 7.

Par rapport à la figure 6 utilisant un modèle optimisé sur les données WL-1 (gaussianisée), on constate:

- primo que la distribution des  $C_{01}$  calculées avec les cartes générées ("wcrgr") suit les ondulations de la distribution calculée sur les données d’entraînement ("train"). Mais il y a un petit désaccord néanmoins mais bien plus petit.
- les petits désaccords sur les distribution bi-spectre et  $s_{21}$  sont de même ampleur.
- on note également un petit désaccord sur la distribution des  $C_{11}$ .

Ces données nous renseignent sur le fait qu’un modèle WCRG est capable de générer des cartes avec des coefficients  $C_{01}$  convenables en tous les cas bien plus que ceux de la figure 6.

Concernant les données  $\phi_4$ , on dispose de 3 lots obtenus avec les valeurs de  $\beta$  différentes (ie. 0.5, 0.68, 0.76) qui influencent la dynamique sous-jacente. Tout comme pour les cartes WL-2, on a procédé à l’accord des moyennes des distributions des valeurs de pixels entre les cartes générées et les cartes d’entraînement. Les différents résultats sont exposés sur les figures 8, 9, et 10.

Ce que l’on constate c’est que:

- les distributions obtenues avec les données  $\beta = 0.50$  (*haute température*) sont remarquablement concordantes; de plus les coefficients  $C_{01}$  sont quasi-nuls même sur les données d’entraînement.
- les désaccords entre les distributions avec les données  $\beta = 0.76$  (*basse température*) sont assez semblables à ceux observés sur les données WL-2.
- la figure des distributions des coefficients  $C_{01}$  avec les données  $\beta = 0.68$  (*point critique*) est particulière: les coefficients calculés avec les données générées sont quasi-nuls, or on constate une structuration de la distribution calculée avec les données d’entraînement. Notons que contrairement au cas WL la distribution du bi-spectre semble bien reproduite (aux erreurs statistiques près), et il en va de même avec la distribution  $s_{21}$ .

Ainsi, on dispose de plusieurs cas de figures et à part 1 cas ( $\phi_4$ ,  $\beta = 0.50$  soit à haute température) tous les autres exhibent des divergences concernant principalement les coeffi-

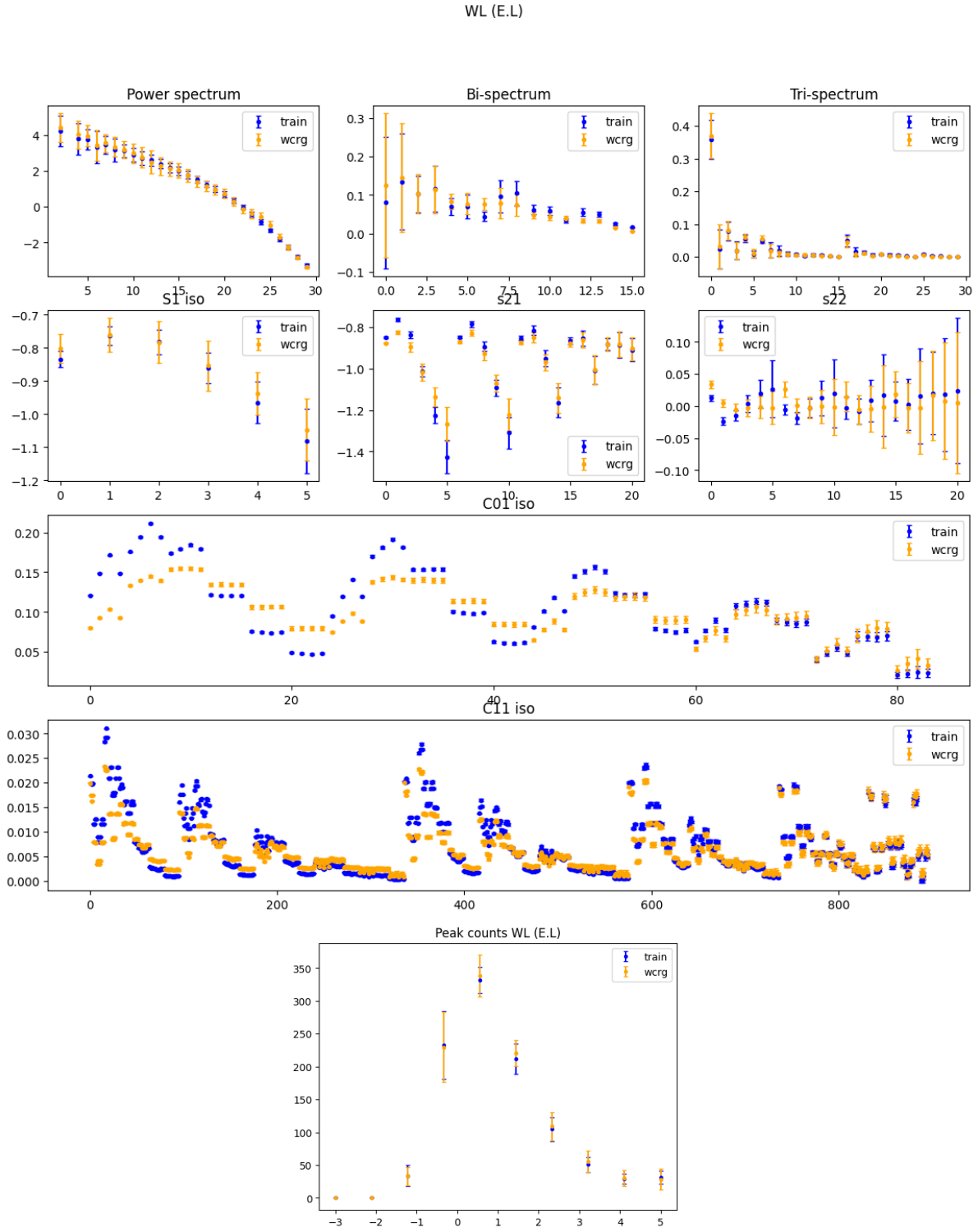


FIGURE 7 – Statistiques de la figure 6 mais concernant les données WL-2.

$$\phi_4 : \beta = 0.50$$

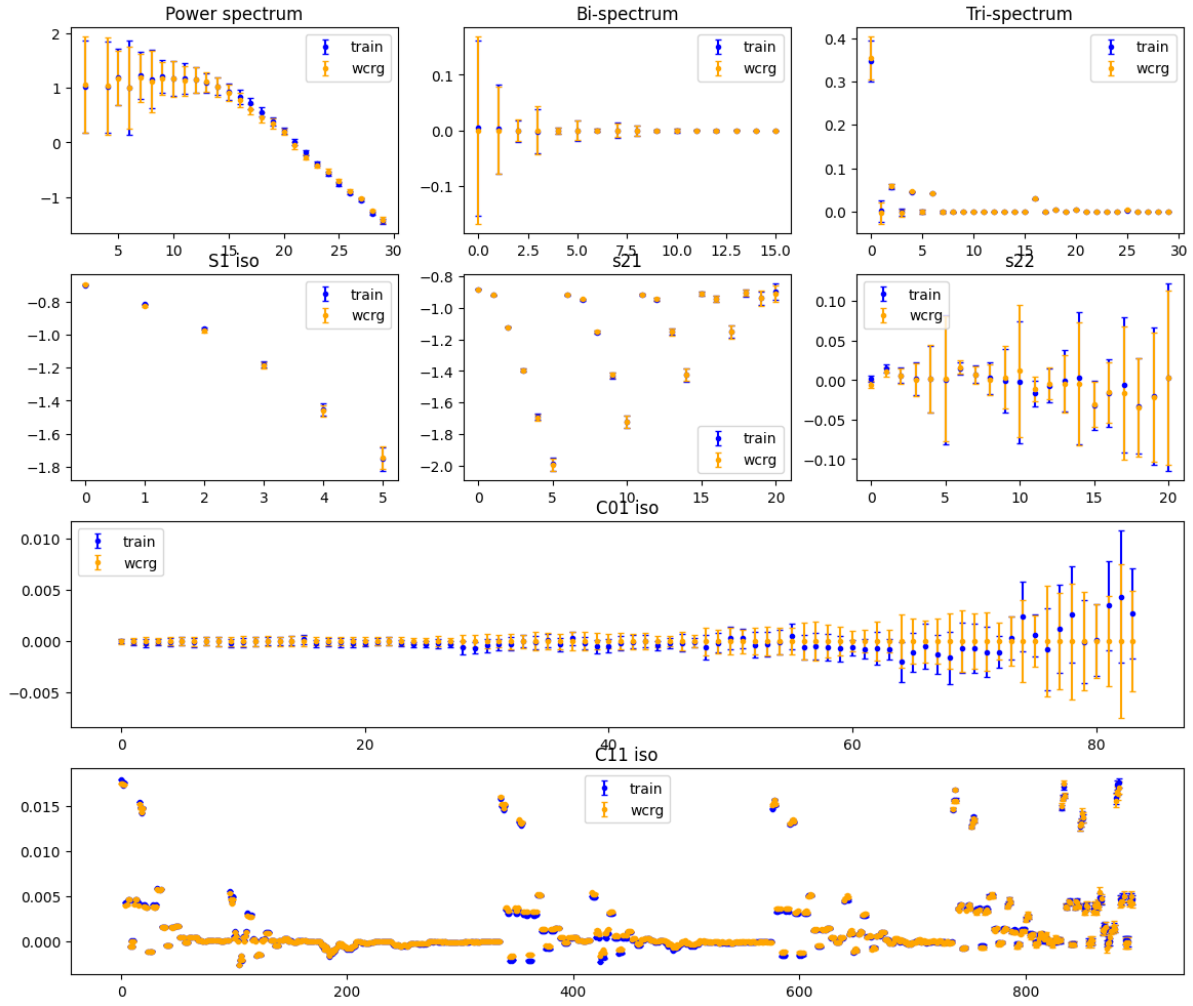


FIGURE 8 – Statistiques de la figure 6 mais concernant les données  $\phi_4$  ( $\beta = 0.50$ ). Notons que l'on dispose de 500 cartes "train" et de 400 cartes "wcr'g".

$$\varphi_4 : \beta = 0.68$$

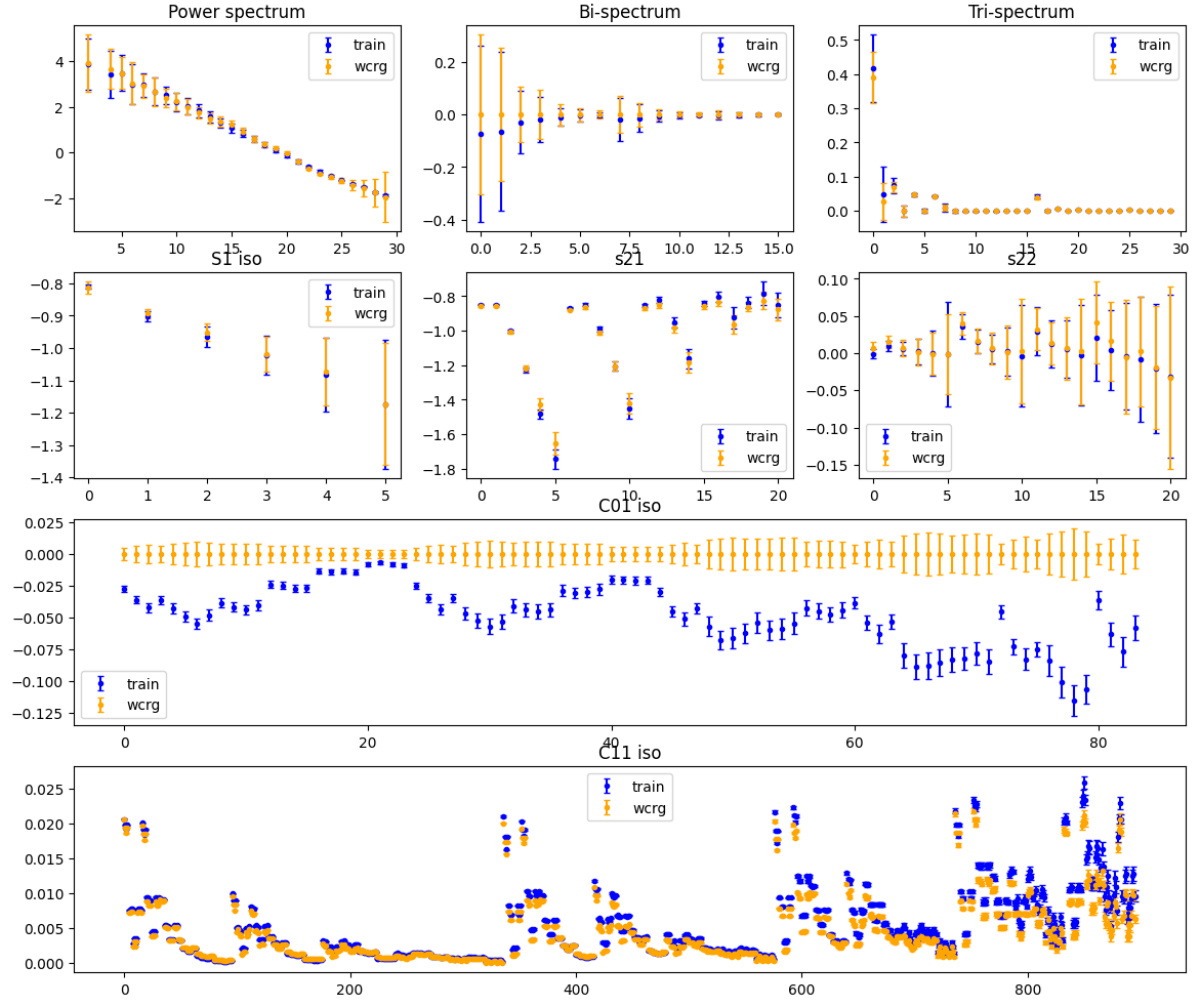


FIGURE 9 – Statistiques de la figure 6 mais concernant les données  $\phi_4$  ( $\beta = 0.68$ ). Notons que l'on dispose de 500 cartes "train" et de 500 cartes "wcrq".



$$\phi_4 : \beta = 0.76$$

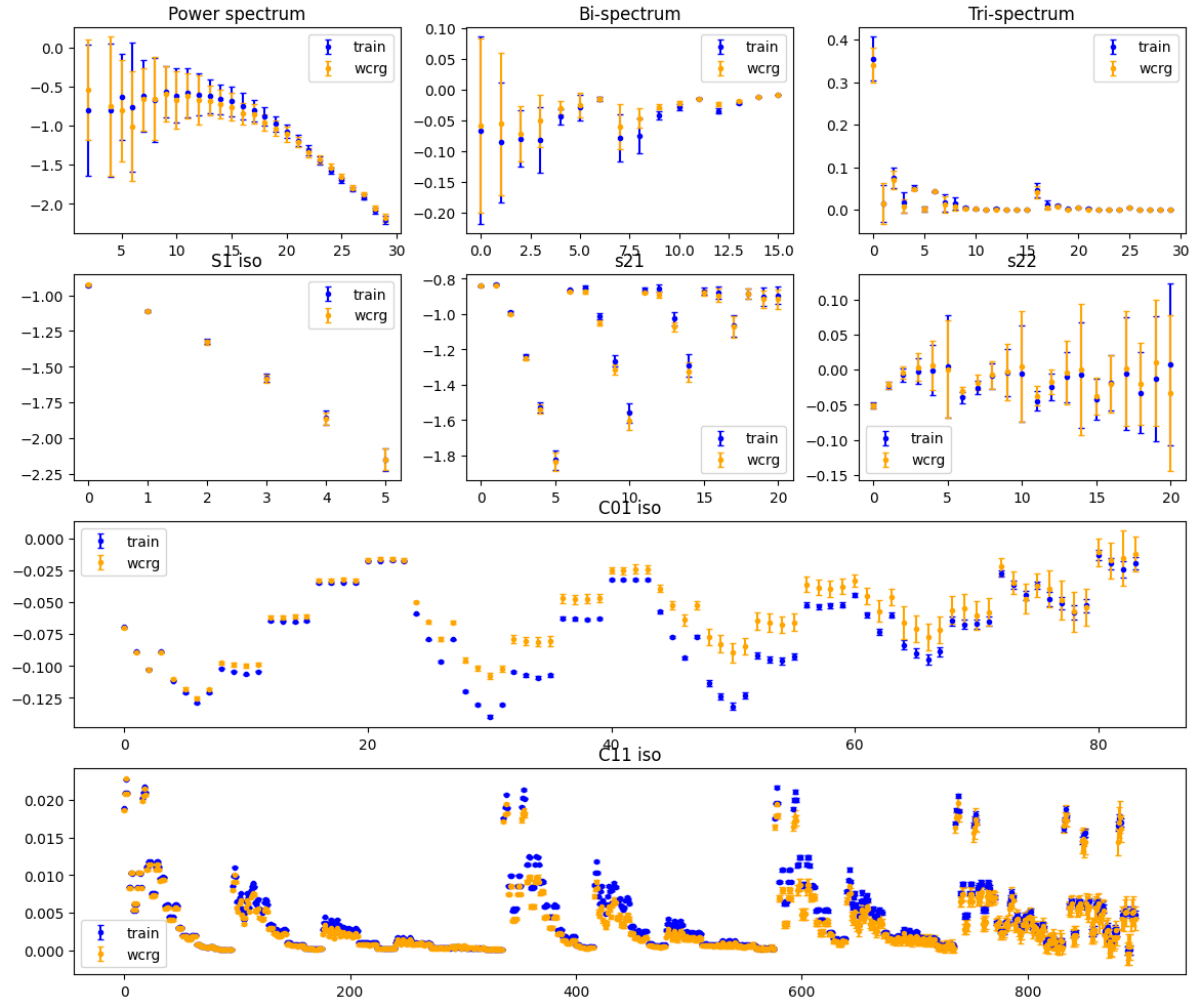


FIGURE 10 – Statistiques de la figure 6 mais concernant les données  $\phi_4$  ( $\beta = 0.76$ ). Notons que l'on dispose de 500 cartes "train" et de 50 cartes "wcr".

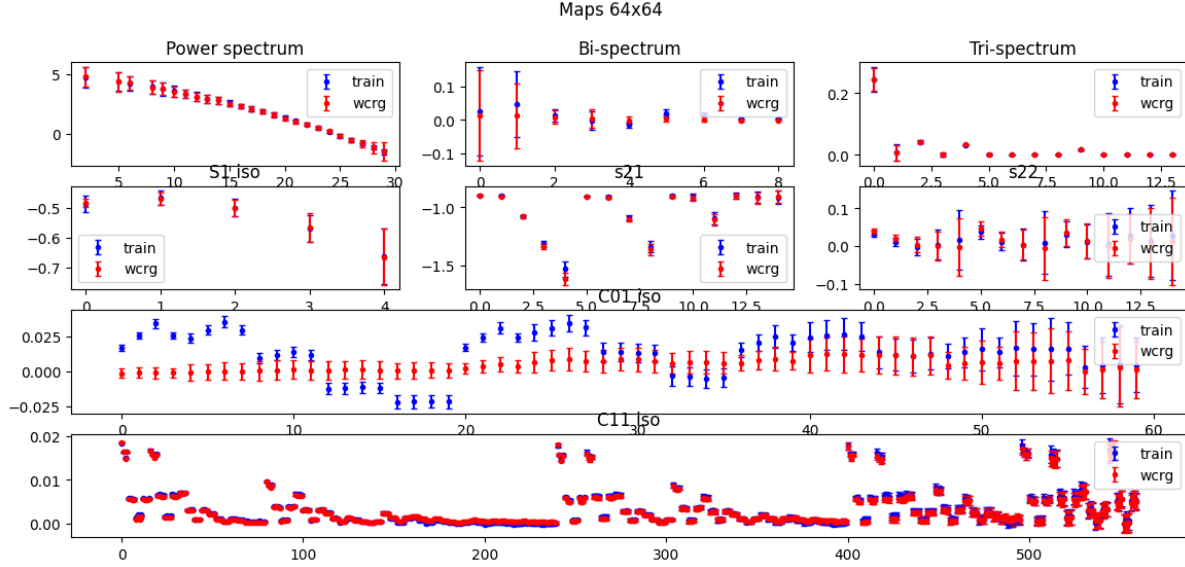


FIGURE 11 – Statistiques de la figure 6 mais concernant les données WL-1 à l'échelle  $L = 64$ . Notons que l'on dispose de 500 cartes "train" et de 50 cartes "wcrgr".

cients  $C_{01}$ . Même si on n'exclut pas un bug sous-jacent, les distributions de ces coefficients générés montrent tout de même des structures sauf précisément pour  $\phi_4, \beta = 0.50$ .

Deux expérimentations supplémentaires ont été menée pour essayer de comprendre la distribution des  $C_{01}$ :

- la première concerne les données WL-1 mais avec les cartes à l'échelle  $L = 64$ . Le résultat est présenté sur la figure 11. Les désaccords des distributions  $C_{01}$  sont présents. Notons que les autres distributions sont très bien reproduites.
- la seconde reprend la synthèse avec un modèle micro-canonique mais où l'on a volontairement fixé à 0 les coefficients  $C_{01}$ . Le résultat est illustré sur la figure 12. Au-delà de la distribution des  $C_{01}$ , l'hypothèse d'une corrélation des désaccords  $C_{01}$ , bi-spectrum et  $s_{21}$  peut sembler avoir un certain sens.

## 6. Discussion

Nous allons tenter de donner des éléments afin de comprendre le comportement des statistiques  $C_{01}$  et  $C_{11}$ .

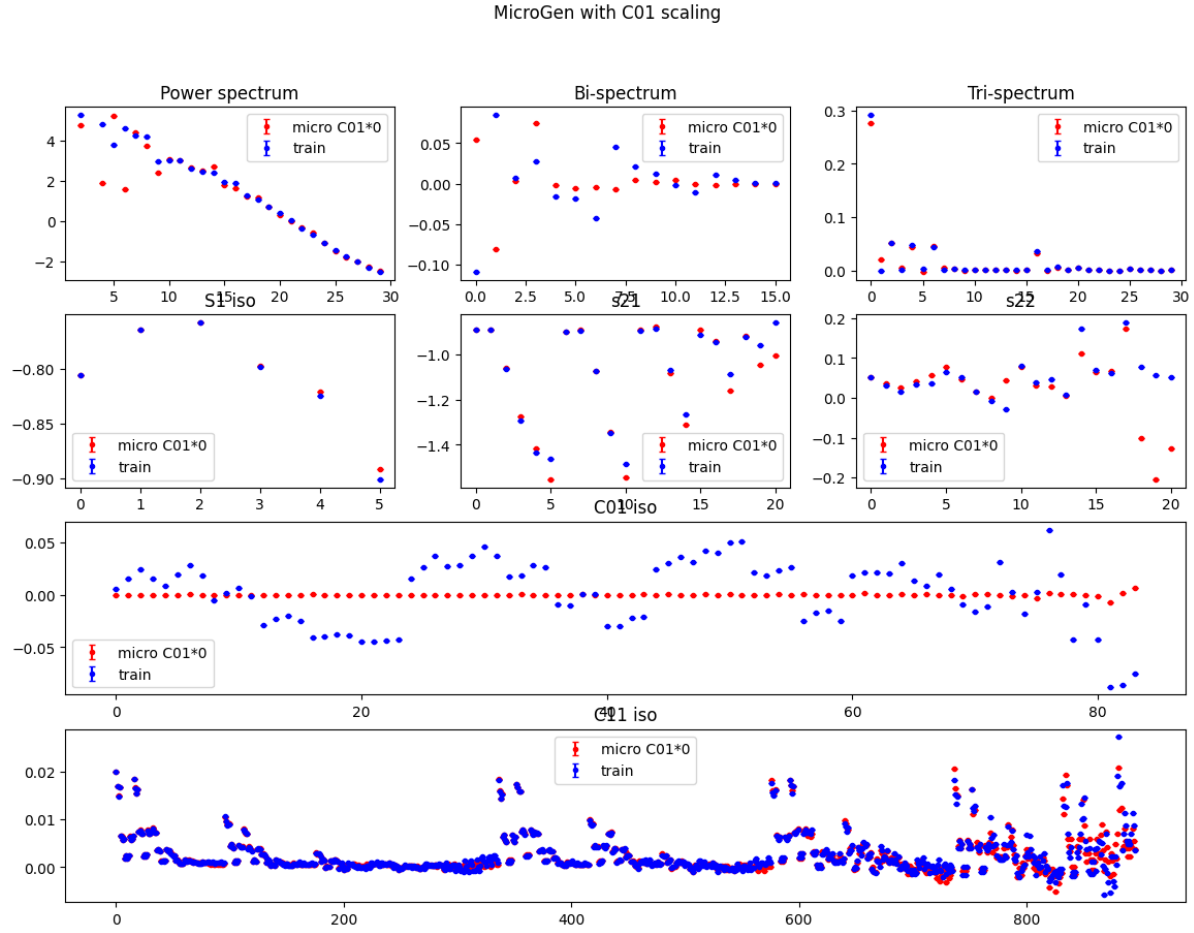


FIGURE 12 – Statistiques de la figure 6 mais concernant les données WL-1 et un modèle micro-canonique du type de celui discuté à la section 4 mais où les coefficients  $C_{01}$  sont volontairement mis à 0. Notons que l'on a utilisé qu'une unique carte pour calculer les distributions.

### $C_{01}$ et $C_{11}$ dans un cadre gaussien

Pour un processus  $x$  gaussien, les statistiques  $C_{01}$  et  $C_{11}$  sont contraintes par le spectre de puissance.

$C_{11}$  correspond à l'expression de la matrice de covariance  $\langle |x * \psi_{j_1}|(u) |x * \psi_{j_2}|^*(u + \tau) \rangle_\tau$  dans une base qui la quasi-diagonalise ?. Dans le cas gaussien, si  $\hat{\psi}_{j_1} \hat{\psi}_{j_2} = 0$ ,  $x * \psi_{j_1}$  et  $x * \psi_{j_2}$  sont indépendantes ?, et

$$\begin{aligned} \langle |x * \psi_{j_1}| |x * \psi_{j_2}| \rangle &= \mathbb{E}[|x * \psi_{j_1}|] \mathbb{E}[|x * \psi_{j_2}|] \\ &= \text{Cste} \sqrt{\mathbb{E}[|x * \psi_{j_1}|^2]}^{\frac{1}{2}} \sqrt{\mathbb{E}[|x * \psi_{j_2}|^2]}^{\frac{1}{2}}. \end{aligned}$$

Ainsi,  $C_{11}$  est contraint par le spectre de puissance (*nb. en physique le spectre de puissance fait plutôt référence au filtrage avec les exponentielles complexes à la place des ondelettes.*).

Le même argument donne que  $C_{01}$  vaut 0. En effet,

$$\langle (|x * \psi_{j_1}| * \psi_{j_2})(x * \psi_{j_2}) \rangle = \mathbb{E}[|x * \psi_{j_1}| * \psi_{j_2}] \mathbb{E}[x * \psi_{j_2}] = 0$$

. On peut donc s'attendre, en l'absence de contraintes non-gaussiennes, à ce que  $C_{01}$  soit nul.

Par ailleurs, pour le WCRG, les déviations non-gaussiennes de  $C_{01}$  et de  $C_{11}$  sont une conséquence du potentiel non-linéaire qui s'applique sur  $x_{j-1}$  conditionnellement à  $x_j$ , et donc conditionnellement à toutes les bandes fréquences inférieures à celle de  $\bar{x}_j$ .

### Non Gaussianité imposée par le potentiel scalaire

La partie non-gaussienne des potentiels est faite de potentiels scalaires non linéaires pixel par pixel sur les cartes  $x_j$ . Les coefficients des cartes  $\bar{x}_{j+1}$  sont mélangés non-linéairement conditionnellement à la basse fréquence  $x_{j+1}$ . Les potentiels scalaires imposent des conditions d'amplitude sur  $x_j$ , sans conditions spatiales. Cela impose par conséquent une condition d'amplitude sur  $\bar{x}_{j+1}$ , sans cohérence spatiale (autre que le conditionnement par la basse fréquence).

On peut extrapoler et conclure à une condition d'amplitude sur les cartes  $x * \psi_j$ , sans condition de phase, qui correspond à la cohérence spatiale. En effet, si on imagine que  $\bar{x}_j$  est une sinusoïde, contraindre uniquement sa distribution ne contraint pas sa phase relative. On contraindrait alors l'amplitude de  $x * \psi_j$ , sans contraindre sa phase.

Or,  $C_{11}$  est une mesure d'amplitude, et non pas de phase. La contrainte d'amplitude, conséquence du potentiel non-linéaire, peut alors permettre la bonne reproduction de  $C_{11}$ . Au contraire,  $C_{01}$  est une statistique de phases. Comme ces dernières ne sont pas contraintes, les synthèses ne devraient donc pas exhiber de caractère non-gaussien sur  $C_{01}$ .

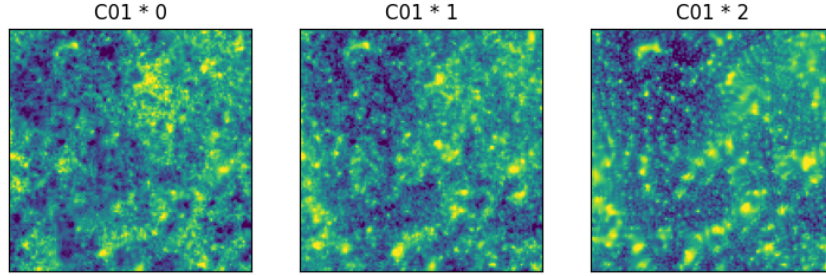


FIGURE 13 – Impact des statistiques  $C_{01}$  sur des synthèses micro-canoniques.

### Concordance expérimentale

On peut discuter séparément le cas concernant le système  $\varphi^4$  de celui des données de Weak Lensing.

- $\varphi^4$ : Dans le cas à haute température ( $\beta = 0.5$ ), les statistiques  $C_{01}$ , qui sont attendues d'être nulles, sont bien reproduites. Pour ce modèle, on perd la corrélation entre échelles, et par conséquent il s'agit d'un modèle quasiment marginal, bien reproduit par les potentiels scalaires.  
 Au point critique ( $\beta = 0.68$ ), la non-gaussianité très forte (car toutes les échelles sont corrélées) de  $C_{01}$  n'est pas capturée, et les statistiques reproduites sont nulles. Par contre,  $C_{11}$  est bien reproduite.  
 Enfin, A basse température ( $\beta = 0.76$ ), la corrélation entre échelles est plus faible mais non nulle, la non-gaussianité est sous évaluée par  $C_{01}$ , qui est plus petit sur les synthèses que sur les données.
- Weak Lensing: Pour WL2,  $C_{01}$  souffre de sous-évaluation, comme dans le cas précédent  $\varphi^4$   $\beta = 0.76$ . Concernant WL1,  $C_{01}$  est quasiment nul, sans structuration, contrairement aux cartes d'entraînement. Il serait envisageable de refaire un entraînement sans procéder à la gaussianisation des pixels pour lever un doute sur d'entraînement. Dans les deux cas de Weak Lensing, les  $C_{11}$  sont bien reproduits.

### Conséquence sur la physique apprise

On peut se poser la question de savoir ce que cela implique sur la physique. On argumente qu'augmenter  $C_{01}$  concentrerait le champ autour de pics et inversement les mettre à 0, diminue les pics.

On voit figure 13 qu'augmenter le coefficient  $C_{01}$ , à spectre de puissance fixé, favorise le clustering à petite échelle spatiale, et donc symétriquement le modèle WCRG tel qu'il est actuellement défini a tendance à diluer les corrélations à courte portée.

$C_{01}$  est proportionnel à  $\langle (x * \psi_2)(|x * \psi_1| * \psi_2)^* \rangle$ . A spectre de puissance fixé, si ce coefficient est grand, c'est que les champs  $x * \psi_2$  et  $|x * \psi_1| * \psi_2$  ont leurs énergies qui sont

non seulement spatialement localisées aux mêmes endroits, mais avec une phase relative nulle. On s'attend à des interférences constructives en ces points, et donc, plus ces stats sont grandes, plus les pics seront forts. Au contraire quand  $C_{01}$  est nul, les phases des différentes échelles ne s'alignent pas et ne se combinent pas positivement.

## 7. Résumé

Dans cette note, nous avons exposé les résultats de calculs de statistiques permettant d'aller au-delà de l'aspect visuel des cartes générées et le matching des spectres de puissance, afin d'apprécier la qualité de synthèse des modèles WCRG dans le cas de données de Weak Lensing et  $\phi_4$ . **Ce qui en ressort à l'heure de l'écriture de cette note, c'est que si globalement les distributions d'ordre supérieur sont bien reproduites, ce qui est un résultat nouveau très encourageant, il en reste une qui fait défaut à savoir celle des coefficients de corrélation  $C_{01}$ .** Le cas particulier où  $C_{01}$  est bien reproduit concerne le système  $\phi_4$  avec  $\beta = 0.50$  (haute température), mais il se trouve que dans ce cas précis on s'attend à ce que  $C_{01}$  soit nul.

Notons qu'un réseau DGAN optimisé sur les données WL-1 donne des cartes dont les statistiques sont en parfait accord avec celles obtenues sur les données d'entraînement. Il en va de même avec des modèles micro-canoniques mais qui par nature sont optimisés de telles distributions.

Donc, il "reste" à comprendre comment on peut améliorer la modélisation des potentiels WCRG afin d'obtenir de meilleurs résultats. Une piste a été élaborée dans la discussion de la section 6: elle concernerait les phases relatives des différentes bandes fréquences, qui ne seraient pas contraintes par le modèle.

## Références

- Cheng, S. & Ménard, B. 2021, arXiv e-prints, arXiv:2112.01288
- Cheng, S., Morel, R., Allys, E., Ménard, B., & Mallat, S. 2023, arXiv e-prints, arXiv:2306.17210
- Guth, F., Lempereur, E., Bruna, J., & Mallat, S. 2023, arXiv e-prints, arXiv:2306.00181
- Lanzieri, D., Lanusse, F., Modi, C., et al. 2023, arXiv e-prints, arXiv:2305.07531
- Mustafa, M., Bard, D., Bhimji, W., et al. 2019, Computational Astrophysics and Cosmology, 6, 1

## A. Plots de contrôle...

Les histogrammes de cette section sont issues de la phase de synthèse du modèle WCRG entraîné sur les données WL-1 (Sec. 2.2).

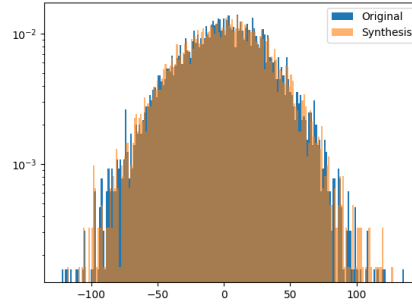


FIGURE 14 – Comparaison des valeurs de pixels à l'échelle  $L = 1$ .

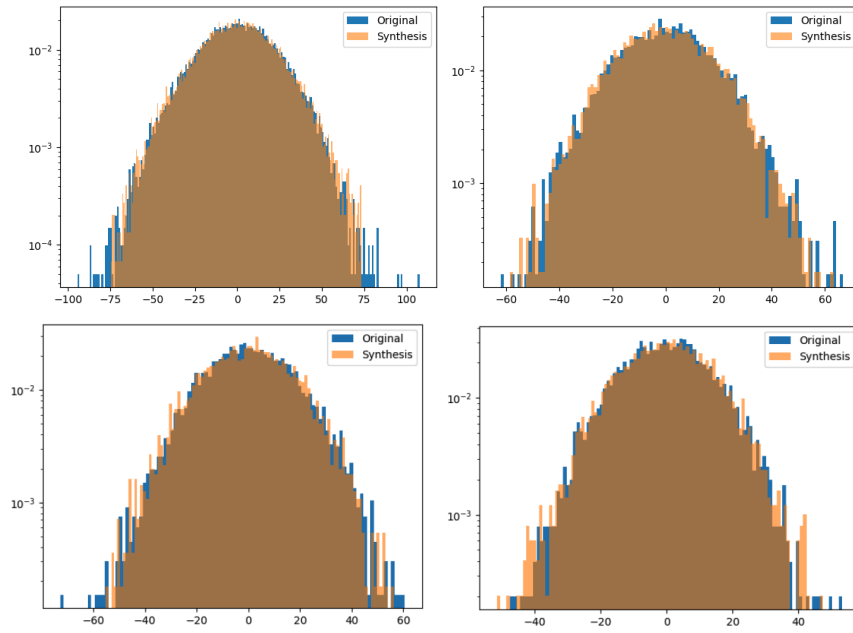


FIGURE 15 – Comparaison à l'échelle  $L = 2$  entre cartes de type "originale/entraînement" et de type "synthétisée": distribution des valeurs de pixels de cartes "totales" (haut-gauche), de détails "H" (haut-droite), de détails "V" (bas-gauche) et de détails "D" (bas-droite).



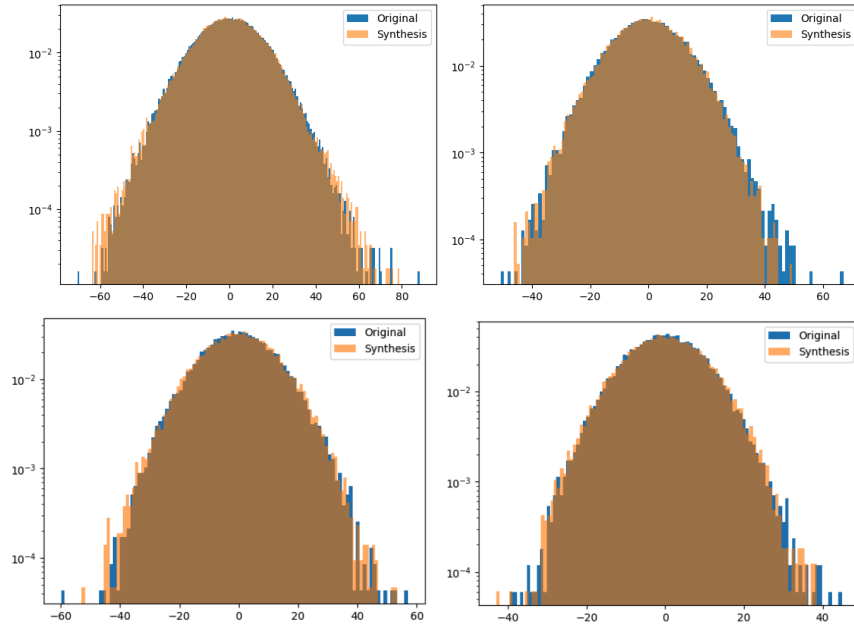


FIGURE 16 – Echelle  $L = 4$ : légende identique à la figure 15.

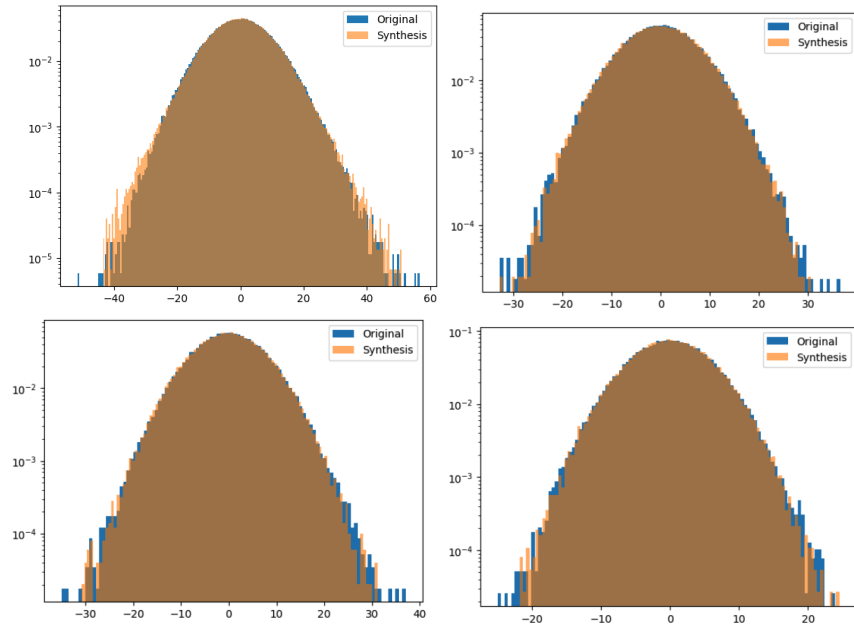


FIGURE 17 – Echelle  $L = 8$ : légende identique à la figure 15.

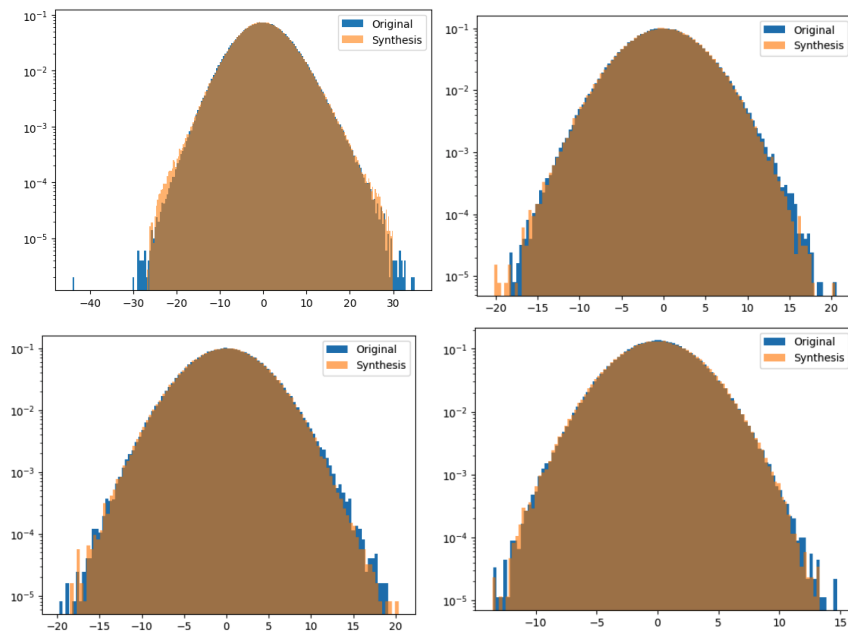


FIGURE 18 – Echelle  $L = 16$ : légende identique à la figure 15.

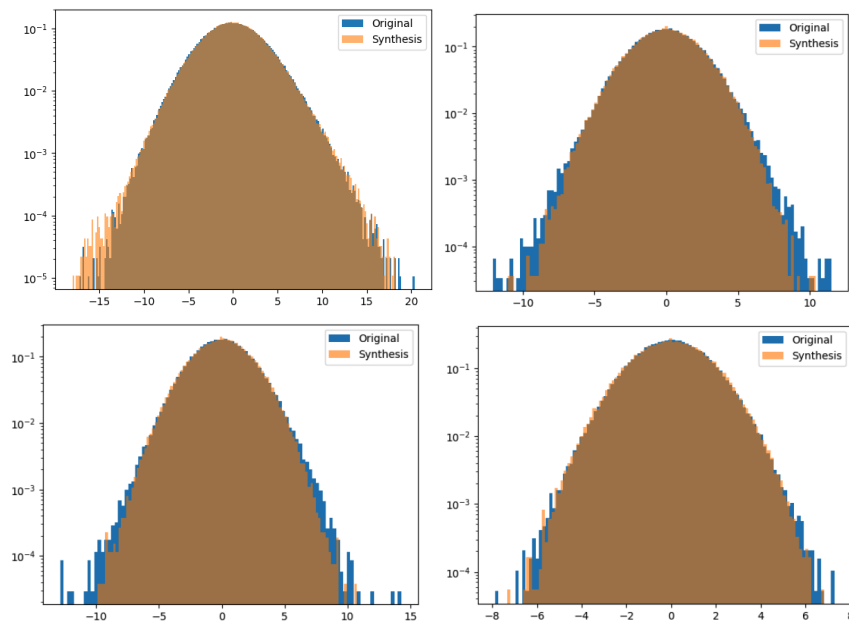


FIGURE 19 – Echelle  $L = 32$ : légende identique à la figure 15.

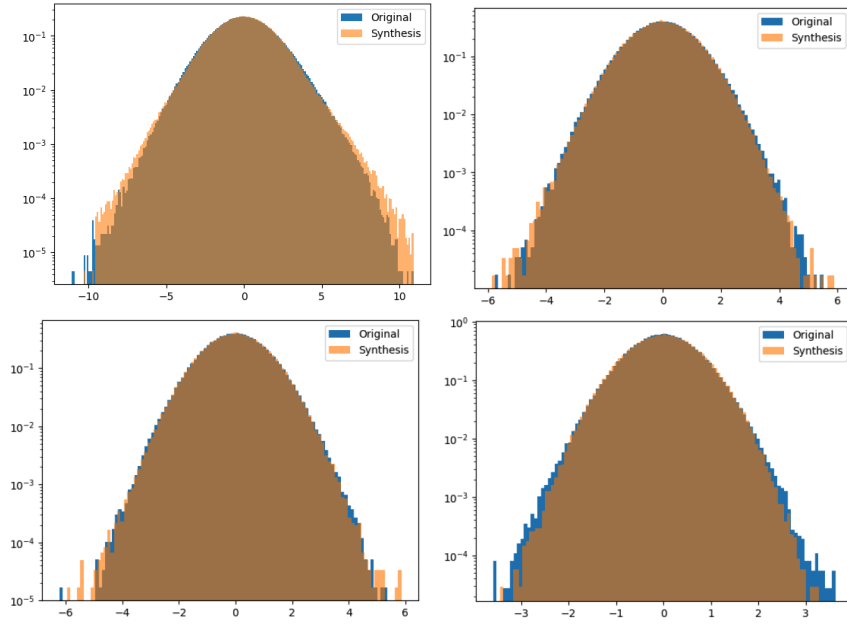


FIGURE 20 – Echelle  $L = 64$ : légende identique à la figure 15.

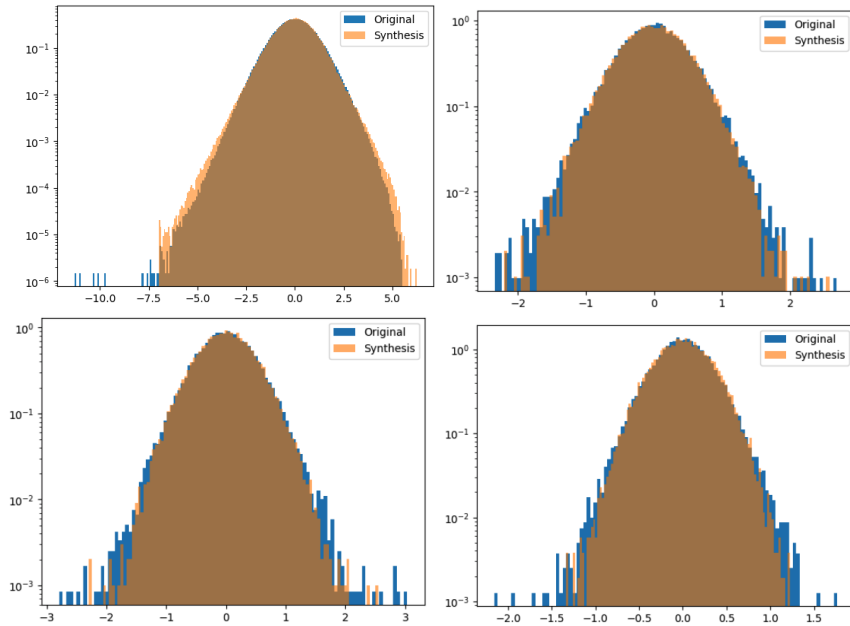


FIGURE 21 – Echelle  $L = 128$  (la plus grande possible): légende identique à la figure 15.