



GEOLEARNING
CHAIRE // Data Science for the Environment

Méthodes efficaces pour l'inférence de modèles spatio-temporels

Journée geolearning

8 avril 2026

Alexandre Loret

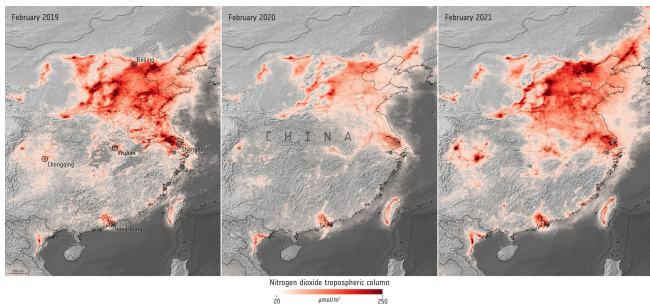
Thomas Romary, Lucia Clarotto et Nicolas Desassis

1 Contexte

Phénomène spatio-temporel

Un **phénomène spatio-temporel** peut par exemple être :

- La pollution de l'air
- L'ensoleillement
- Des déformations dans une galerie



Source : contains modified Copernicus Sentinel data (2019-21), processed by ESA

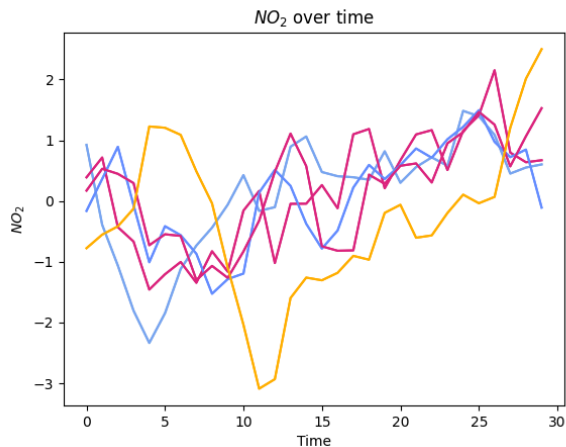
Dans ce cas, les observations sont :

- Réparties de manière irrégulière dans l'espace (des points sur une carte)
- Discontinues mais régulières dans le temps (par exemple toutes les heures)

Phénomène spatio-temporel (ii)

Illustration par **données synthétiques 1D-spatiales et 1D temporelles**.

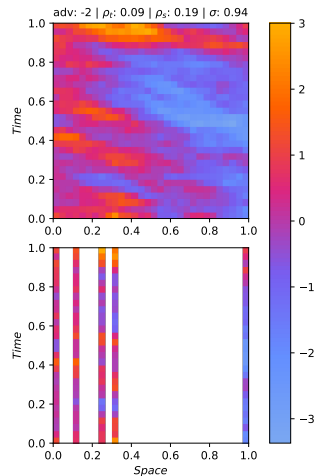
→ Concentration de NO_2 mesurée par des capteurs disposés le long de la N20



Phénomène spatio-temporel (iii)

Illustration par **données synthétiques 1D-spatiales et 1D temporelles**.

→ Concentration de NO_2 mesurée par des capteurs disposés le long de la N20



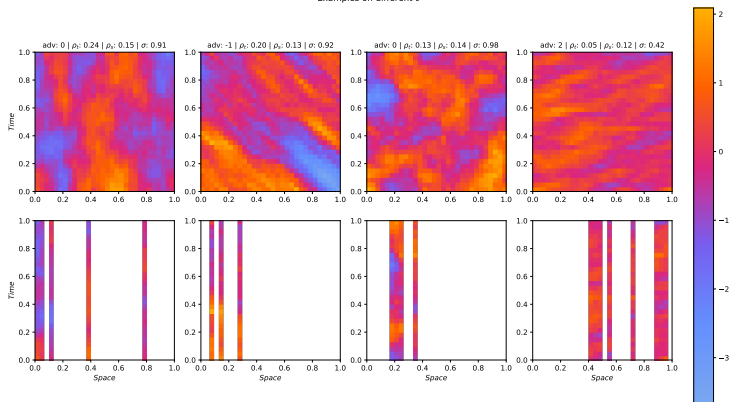
Phénomène spatio-temporel (iv)

6 / 19

On suppose qu'il y a un **modèle** derrière les phénomènes spatio-temporels
Le modèle a des paramètres qui nous sont inconnus

→ Objectif : Trouver les paramètres à partir des observations

Exemples on different θ



Ainsi, un **modèle** permettrait de **prédire** n'importe où, n'importe quand

2 Modèle spatio-temporel

Données

Observations bruitées $y(\mathbf{s}_i, t_j)$ à certains points dans l'espace $(\mathbf{s}_1, \dots, \mathbf{s}_n)$ de $\mathcal{D} \subset \mathbb{R}^d$ et pas de temps (t_1, \dots, t_m) de $\mathcal{T} \subset \mathbb{R}$

Modèle

Pour chaque $i \in \{1, \dots, n\}$ et $j \in \{1, \dots, m\}$, on suppose que les observations $y(\mathbf{s}_i, t_j)$ sont des réalisations de variables aléatoires $Y(\mathbf{s}_i, t_j)$

$$Y(\mathbf{s}_i, t_j) = \mu(\mathbf{s}_i, t_j) + Z(\mathbf{s}_i, t_j) + \tau \varepsilon_{i,j}, \quad \tau > 0$$

- μ l'**effet fixe**
- Z un champs gaussien centré stationnaire de **fonction de covariance**

$$C(\mathbf{h}, u) = \text{Cov}(Y(\mathbf{s} + \mathbf{h}, t + u), Y(\mathbf{s}, t))$$

- $\varepsilon_{i,j} \sim \mathcal{N}(0, 1)$ iid

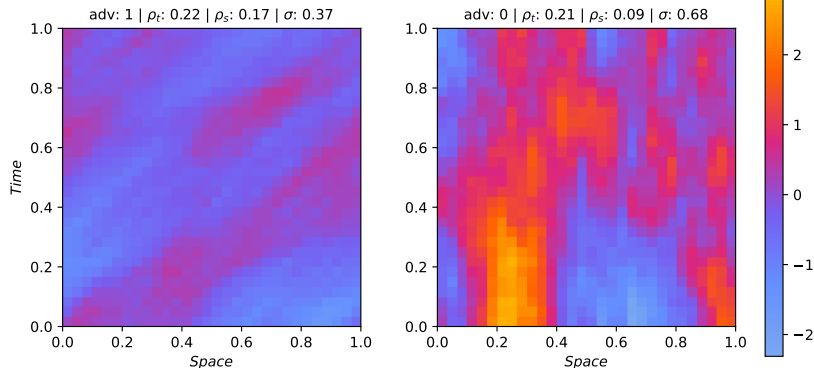
Fonction de covariance paramétrique

On se place dans un contexte paramétrique sans effet-fixe :

$$Y(\mathbf{s}, t; \theta) = Z(\mathbf{s}, t; \theta) + \tau \varepsilon_{i,j}, \quad \tau > 0$$

Cela implique de choisir une **fonction de covariance** C_θ

Exemples on different θ



$$\theta = (\text{adv}, \rho_t, \rho_s, \sigma)$$

- adv = Advection
- ρ_t = portée temporelle
- ρ_s = portée spatiale
- σ = écart type

3 Estimation

→ Objectif : Trouver θ_{obs}

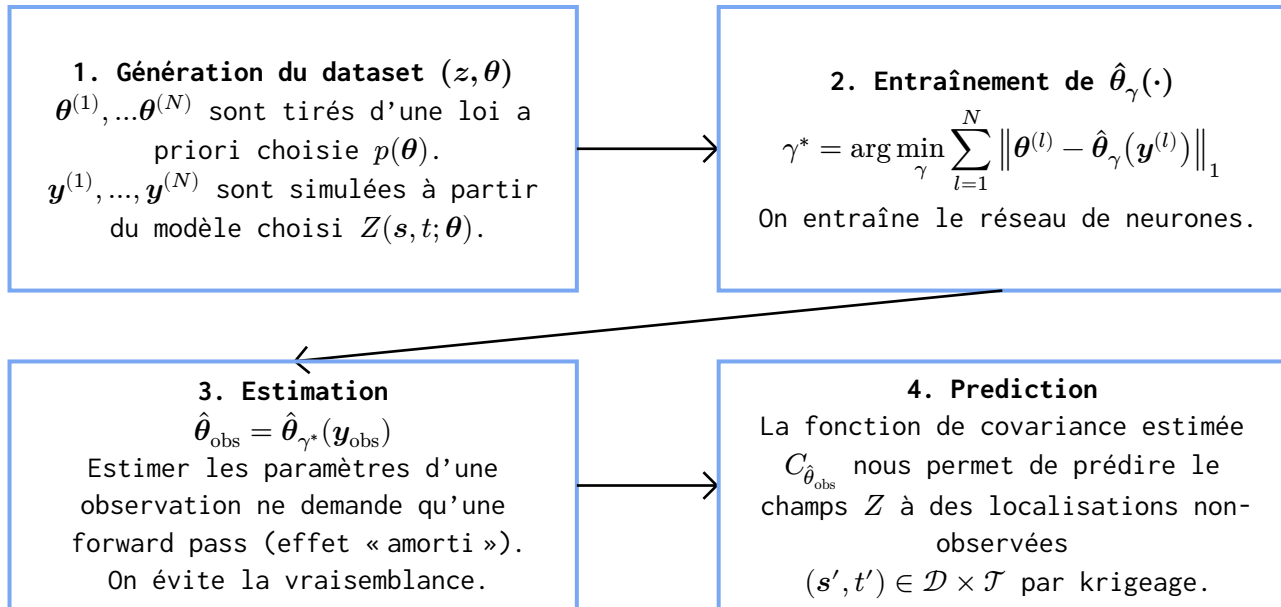
L'Estimation du Maximum de vraisemblance (MLE) est une méthode classique. Mais la méthode devient très coûteuse lorsque que le nombre de données augmente

$$\mathcal{L}(\theta \mid \mathbf{y}_{\text{obs}}) = -\frac{nm}{2} \log(2\pi) - \frac{1}{2} \log \det(\Sigma_{\theta}) - \frac{1}{2} \|\mathbf{y}_{\text{obs}}\|_{\Sigma_{\theta}^{-1}}^2$$

avec Σ_{θ} la matrice de covariance qui associe la covariance C_{θ} entre chaque paire de points observés

De la littérature existe pour simplifier/accélérer ce calcul

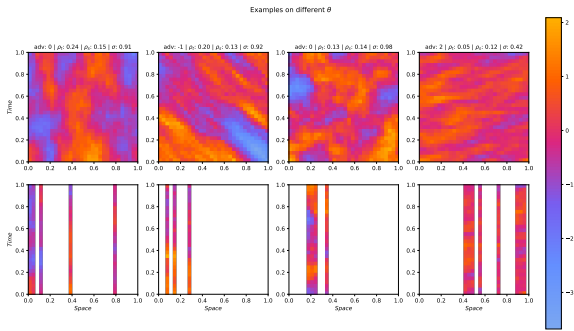
→ Mais le calcul de la vraisemblance reste long...



Détails simulations

Génération du dataset :

- Tirage des $\theta^{(1)}, \dots, \theta^{(N)}$ selon une loi uniforme
- Pour chaque θ , on simule une **grille entière**
- A chaque époque de l'entraînement
 - On génère un masque de même taille et on met les valeurs sous le masque à 0
 - Une observation prise en entrée par le réseau est de la forme (données, masque)



1D spatial \times 1D temporel :

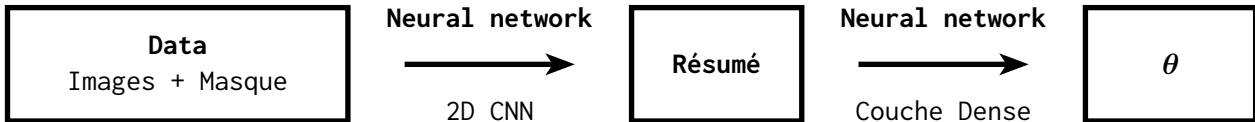
- Plus de clarté sur des slides
- Plus rapide à simuler/entraîner

Choix Réseau de neurones

On utilise un CNN pour traiter les grilles de données

- 2D CNN : **Conv2D** → **Conv2D** → **MaxPooling**

Sur le cas 1D spatial + 1D temporel, les convolutions 2D sont capables de **capturer la dépendance** entre le temps et l'espace. Cette dépendance est très marquée dans les cas avec advection

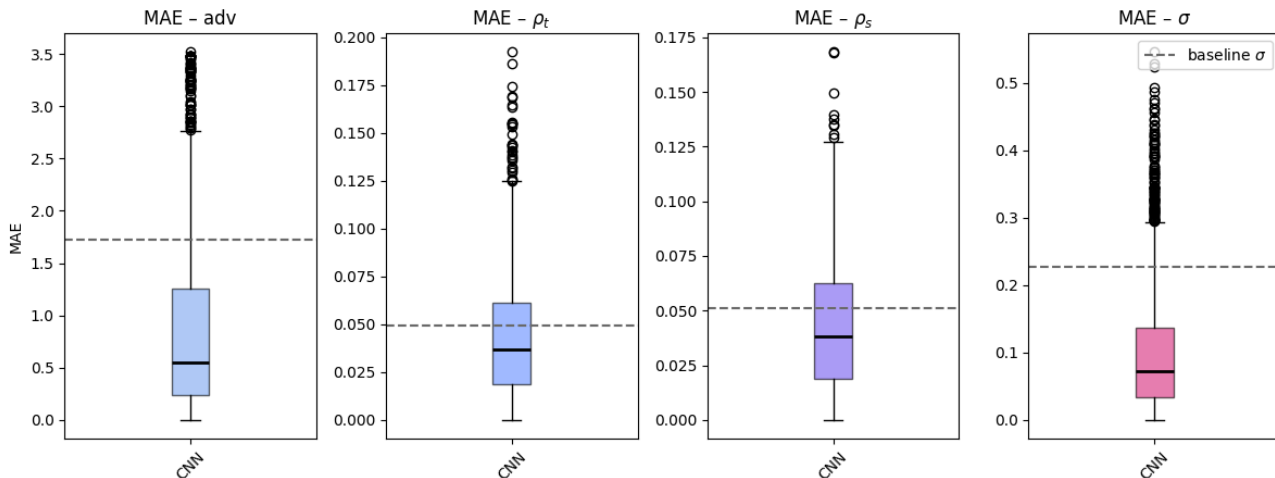


→ Après entraînement, ce réseau permet de d'**estimer les paramètres** de n'importe quel échantillon/simulation prise en entrée.

Choix Réseau de neurones (ii)

Résultat sur un jeu de test

Test error - ~80% missing values



Prédiction par krigeage

- On peut estimer les valeurs à des localisations non mesurées (s'_i, t'_j) en fonction de valeurs connues aux localisations (s_i, t_j) à proximité

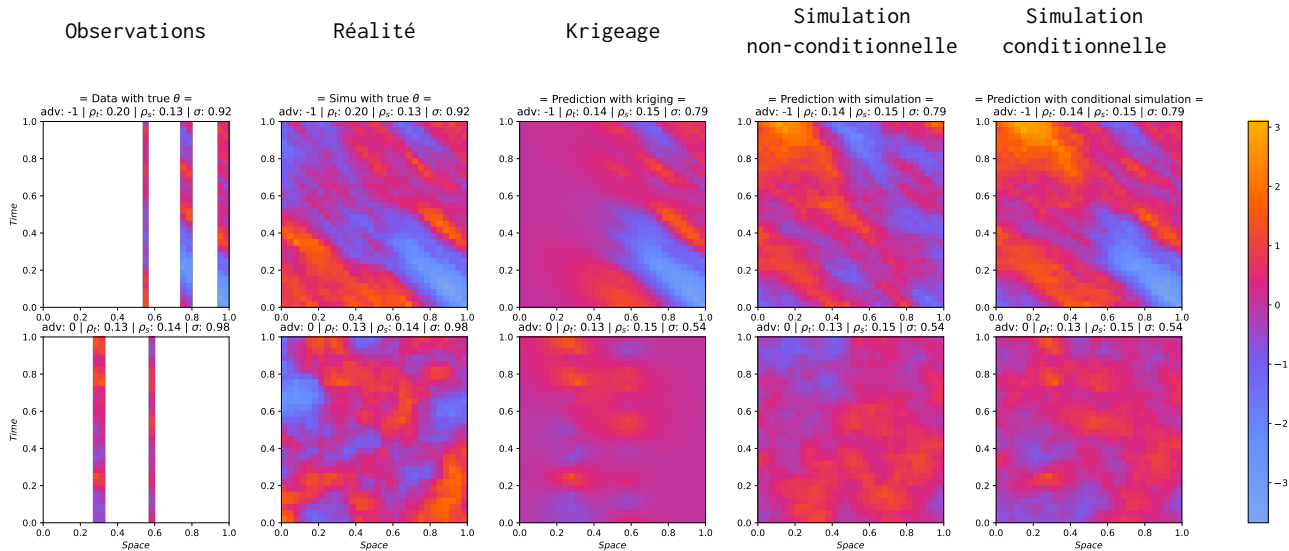
Simulations non-conditionnelles

- On peut générer des simulations à partir des paramètres estimés
- Nos observations réelles ne seront pas incluses dans ces nouvelles simulations

Simulations conditionnelles

- On peut générer des simulations qui respectent les valeurs observées aux points de mesure
- Une simulation conditionnelle est donnée par

simu non-conditionnelle – krigeage de simu non-conditionnelle + krigeage



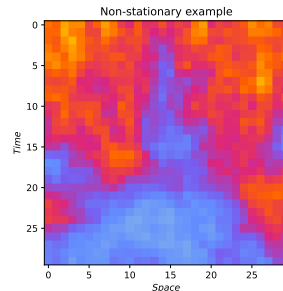
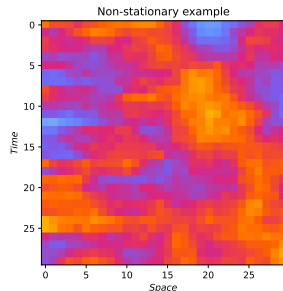
4 Conclusion

Pour l'instant

- On est capable d'estimer un modèle spatio-temporel avec advection
- Par Simulation-Based Inference
 - ▶ Le dataset d'entraînement est simulable rapidement sur grilles complètes
 - ▶ L'estimation par CNN capture la dépendance spatio-temporelle
 - ▶ L'inférence est presque instantanée

Suite

- Estimer une loi des paramètres
- Passer à des modèles non-stationnaires



Annexes

Exemple de modèle

Un champs gaussien aléatoire avec covariance de Matérn 3D (2D spatial + 1D temporel).

$$C_{\theta}(\mathbf{h}, u) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \sqrt{\left(\frac{h_1}{\rho_s}\right)^2 + \left(\frac{h_2}{\rho_s}\right)^2 + \left(\frac{u}{\rho_t}\right)^2} \right)^{\nu} K_{\nu} \left(\sqrt{2\nu} \sqrt{\left(\frac{h_1}{\rho_s}\right)^2 + \left(\frac{h_2}{\rho_s}\right)^2 + \left(\frac{u}{\rho_t}\right)^2} \right)$$

avec Γ la fonction Gamma et K_{ν} la fonction de Bessel modifiée du second ordre. ν est fixé à 1. Une **advection** est ajoutée à la main par dessus.

Cela donne 4 paramètres à estimer : adv , ρ_s , ρ_t et σ

note : cas 1D spatial + 1D temporel

note : isotropie si $\text{adv} = 0$

Détails Champs Gaussien Aléatoire

Écriture matricielle

$$\mathbf{Z} = \begin{pmatrix} \mathbf{Z}_{t_1} \\ \mathbf{Z}_{t_2} \\ \dots \\ \mathbf{Z}_{t_m} \end{pmatrix} \text{ avec } \mathbf{Z}_{t_j} = \begin{pmatrix} Z(\mathbf{s}_1, t_j) \\ Z(\mathbf{s}_2, t_j) \\ \dots \\ Z(\mathbf{s}_n, t_j) \end{pmatrix}$$

$$\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma}) \text{ avec } \mathbf{\Sigma} = \begin{pmatrix} \ddots & & & & \ddots \\ \dots & \text{Cov}(Z(\mathbf{s}_i, t_j), Z(\mathbf{s}_{i'}, t_{j'})) & \dots & & \dots \\ \ddots & & \dots & & \ddots \end{pmatrix}$$

Définitions :

- Stationnaire : Dépend uniquement de la distance entre les points
- Isotrope : $C(\mathbf{h}, u) = C(-\mathbf{h}, u) = C(\mathbf{h}, -u) = C(-\mathbf{h}, -u)$

Prédiction

Krigeage

- Utilisé pour estimer les valeurs à des localisations non mesurées (s'_i, t'_j) en fonction de valeurs connues aux localisations (s_i, t_j) à proximité
- On suppose $z(s_i, t_j)$ réalisation de

$$Z(s_i, t_j; \theta^*) + \tau \varepsilon_{i,j}$$

- On veut prédire $z(s'_i, t'_j)$ « réalisation » de

$$Z(s'_i, t'_j; \theta^*)$$

- Au format matriciel, grâce à C_{θ^*} , on construit :

$$(\mathbf{Z}, \mathbf{Z}') \sim \mathcal{N}(0, \begin{pmatrix} \mathbf{V}_{\text{obs-obs}} & \boldsymbol{\Sigma}_{\text{obs-pred}} \\ \boldsymbol{\Sigma}_{\text{pred-obs}} & \boldsymbol{\Sigma}_{\text{pred-pred}} \end{pmatrix}) \text{ avec } \mathbf{V}_{\text{obs-obs}} = \boldsymbol{\Sigma}_{\text{obs-obs}} + \tau^2 \text{id}$$

- Espérance conditionnelle :

$$\mathbf{z}_{\text{pred}} = \mathbb{E}[\mathbf{Z}' \mid \mathbf{Z} = \mathbf{z}_{\text{obs}}] = \boldsymbol{\Sigma}_{\text{pred-obs}} \mathbf{V}_{\text{obs-obs}}^{-1} (\mathbf{z}_{\text{obs}})$$

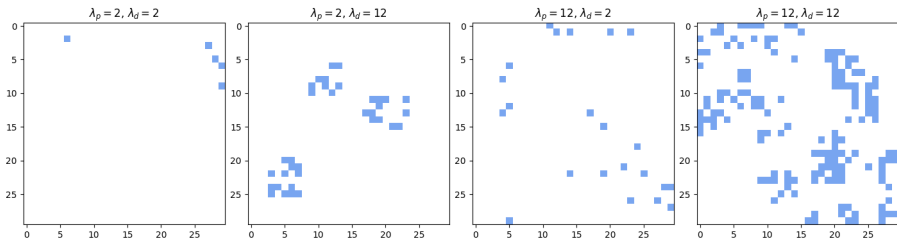
Choix du masque

On choisi un nombre de parents aléatoire que l'on place aléatoirement dans l'espace.

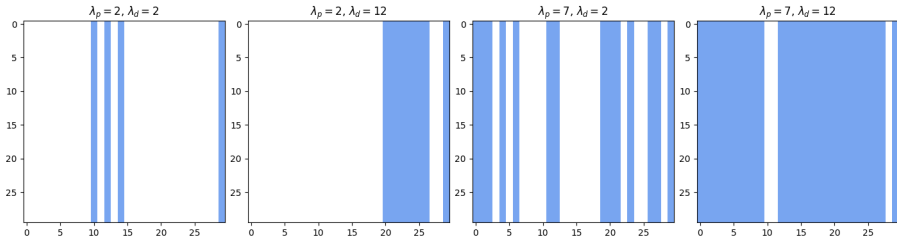
Pour chaque parent :

- On choisi un nombre de points aléatoire que l'on place aléatoirement autour de lui.

2D spatial
(une frame)



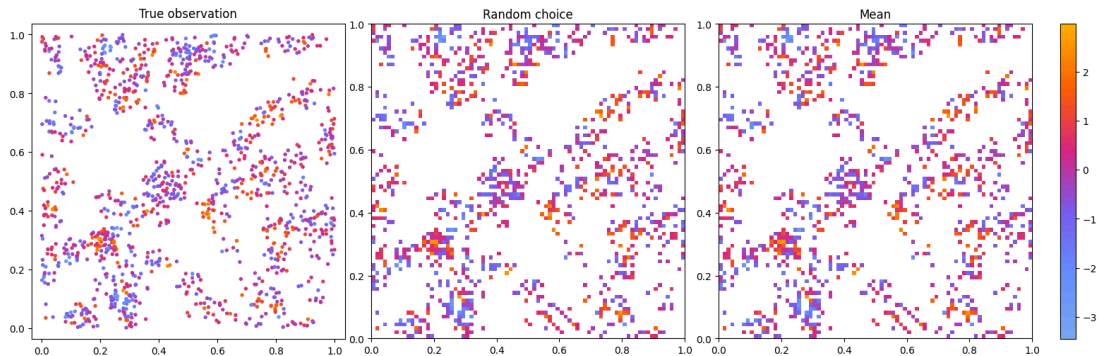
1D spatial
1D temporel



Observations vers grille

Cela implique de trouver une façon de mettre des observations sur des grilles.

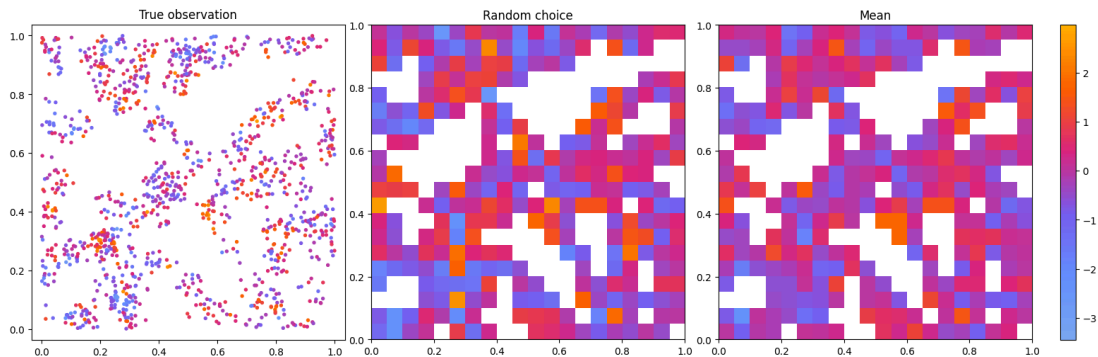
- Les pas de temps sont une dimension de la grille
- Pour l'espace :
 - ▶ Choisir une taille de grille
 - ▶ Si plusieurs points dans une cellule, en choisir un aléatoirement



Observations vers grille (ii)

Cela implique de trouver une façon de mettre des observations sur des grilles.

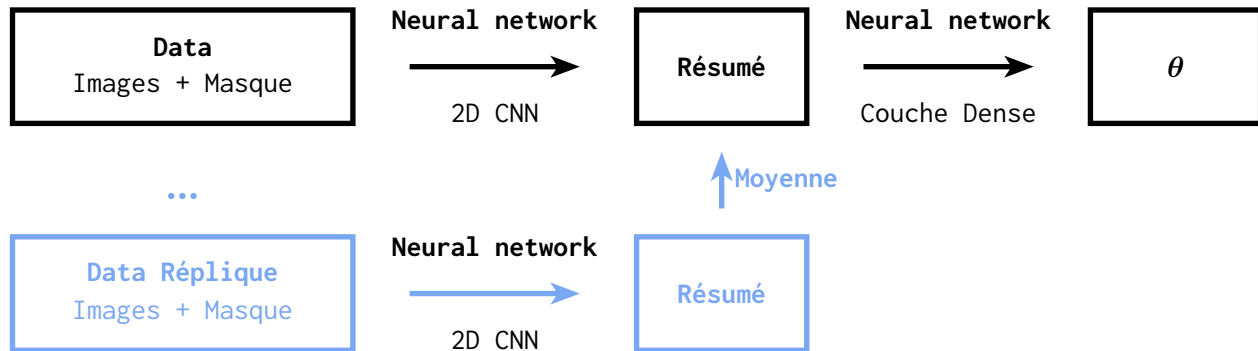
- Les pas de temps sont une dimension de la grille
- Pour l'espace :
 - Choisir une taille de grille
 - Si plusieurs points dans une cellule, en choisir un aléatoirement



Deepset

Architecture deepset :

Si on a des données répliquées (on les sait générées sur un même θ), on veut les utiliser pour améliorer l'estimation du modèle. Dans le train, le réseau apprend qu'un θ peut avoir plusieurs simus (ou l'inverse).

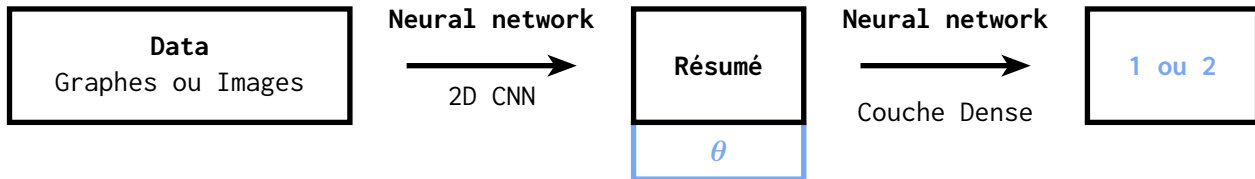


NLS

Il a été montré que la vraisemblance peut être approchée de la façon suivante :

$$L(\theta|y) = \frac{h(y, \theta)}{1 - h(y, \theta)}$$

avec $h(y, \theta)$ un classifieur qui associe $C = 1$ si la réalisation y et le paramètre θ sont dépendant, $C = 2$ si indépendants.



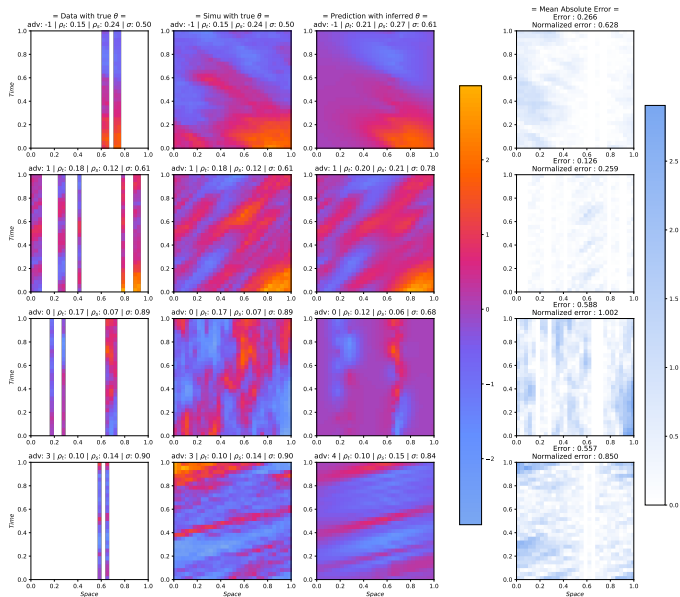
Ce classifieur permet de trouver la vraisemblance pour un couple (donnée, θ). Pour une donnée, le meilleur θ est trouvé par gridsearch ou optimisation.

Choix Réseau de neurones

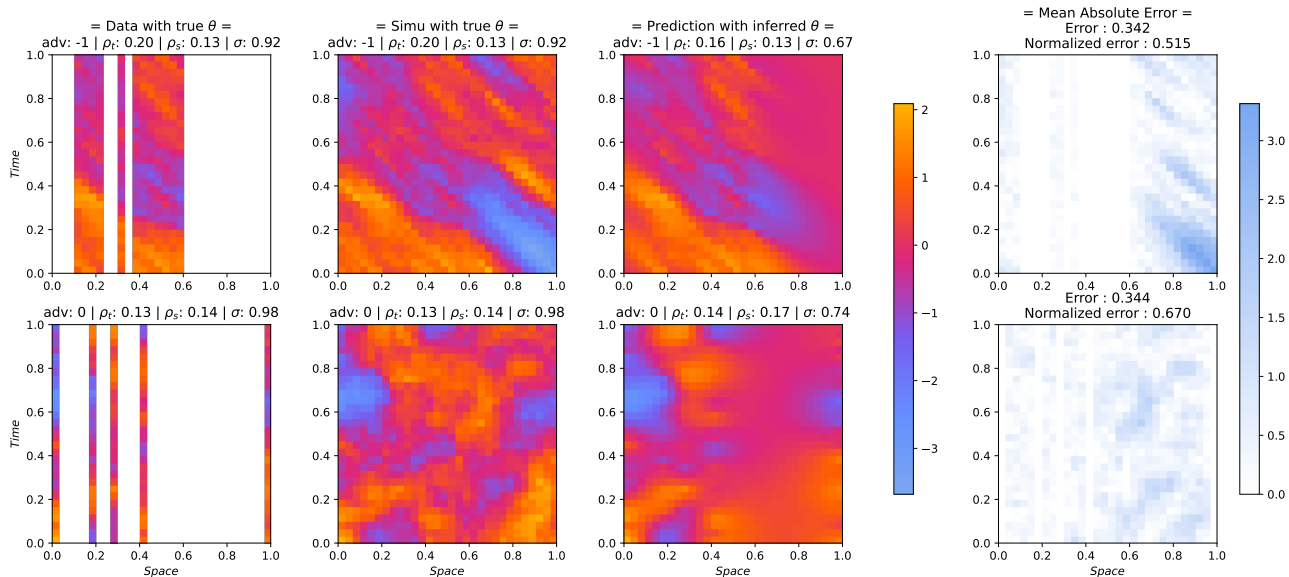
Layer Type	Output Shape	Kernel	Activation	Parameters
Conv2d (1)	$(\mathcal{H}, \mathcal{G}' \times \mathcal{T}')$	5×5	ReLU	1,632
Conv2d (2)	$(\mathcal{H}, \mathcal{G}'' \times \mathcal{T}'')$	3×3	ReLU	9,248
AdaptiveMaxPool	\mathcal{H}	-	-	0
Linear (1)	\mathcal{H}	-	ReLU	1,056
Linear (2)	$\dim \Theta$	-	-	132

Tableau 1. - Architecture Réseau de neurones

Résultats

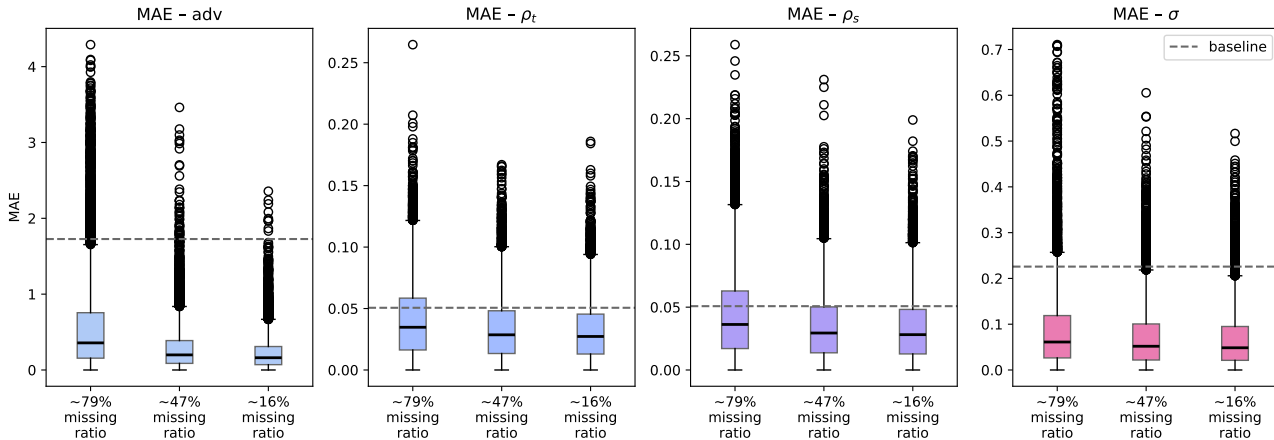


Résultats (ii)



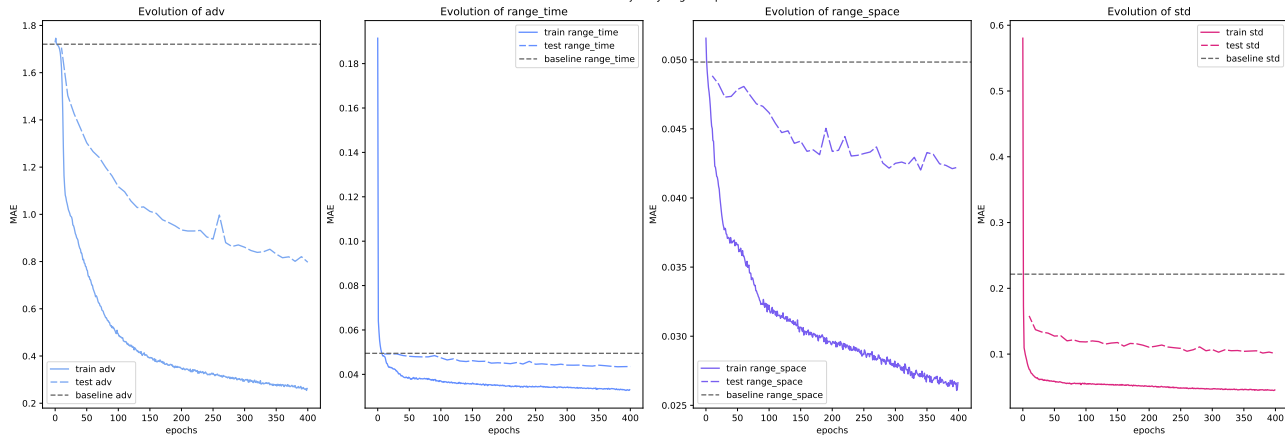
Résultats (iii)

Test error comparison - jointly max 1D-Space CNN

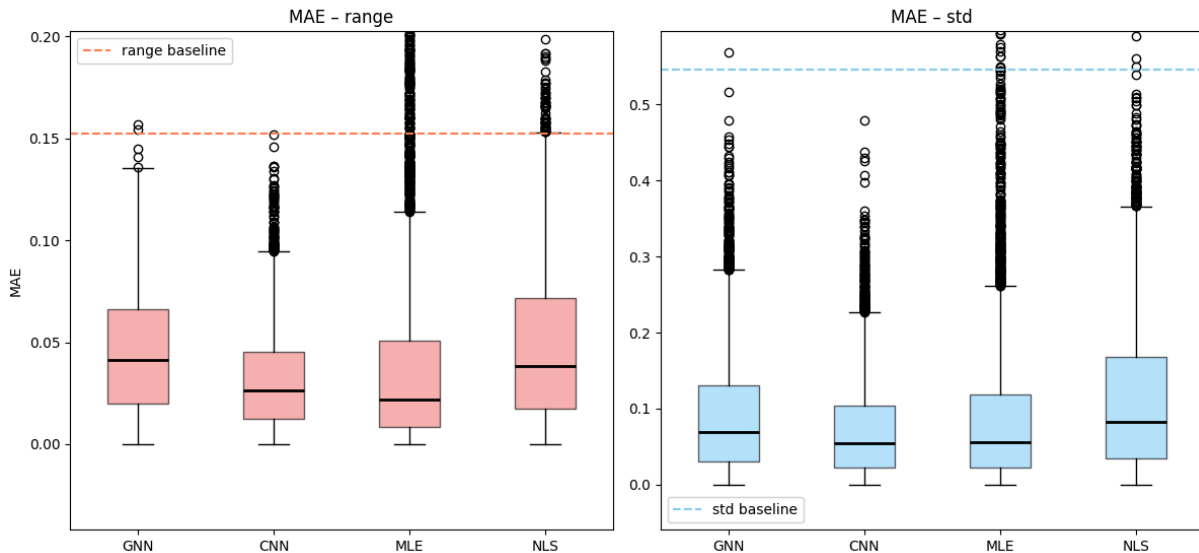


Résultats (iv)

Loss evolutions of jointly avg 1D-Space CNN

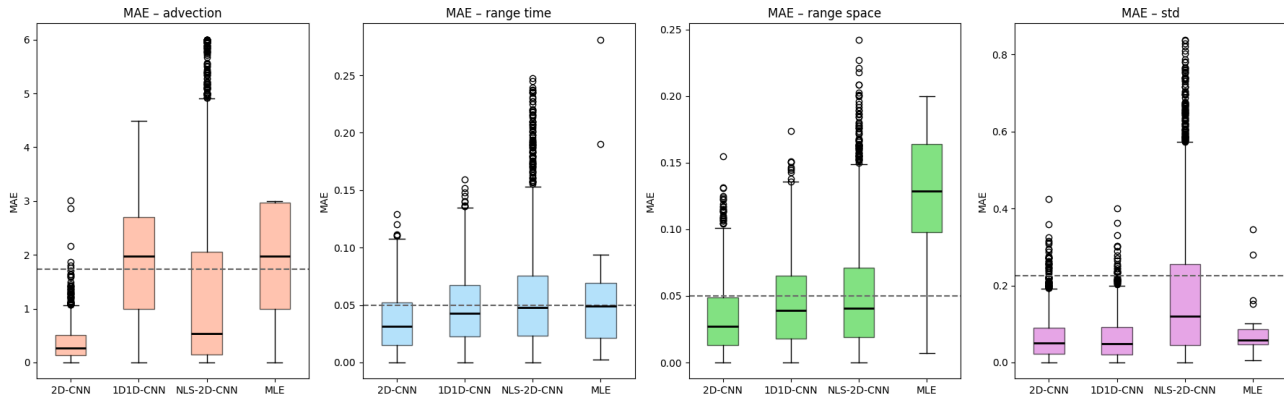


Test error comparison cropped



MLE (ii)

Test error comparison



MLE (iii)

