Quantification d'incertitudes pour l'assimilation de données à partir d'un modèle d'apprentissage profond

Anthony FRION[1] David S. GREENBERG[1]

[1]Helmholtz-Zentrum Hereon, 1 Max-Planck Straße, 21502 Geesthacht, Allemagne

Résumé – L'assimilation de données via des méthodes d'apprentissage automatique permet d'identifier des conditions initiales pour les simulations géoscientifiques. Des avancées récentes permettent de réaliser l'assimilation directement à partir de données bruitées et incomplètes, sans accès à l'état réel du système, mais ne peuvent pas quantifier l'incertitude associée à leurs prédictions. Ainsi, nous proposons et comparons ici différentes extensions de ces méthodes, permettant de produire des prédictions probabilistes.

Abstract – Data assimilation, for which many recently proposed methods are based on machine learning, can identify initial conditions for geoscientific simulations. Recent advances allow the assimilation to be learned directly from noisy and incomplete data with no access to a groundtruth state, but cannot quantify the uncertainty associated with their predictions. Thus, we propose and compare several extensions of these methods, enabling to produce probabilistic predictions.

1 Introduction

Dans les géosciences, il est courant de devoir estimer l'état complet d'un système dynamique en se basant sur un ensemble d'observations incomplètes et/ou bruitées. On peut alors recourir à l'assimilation de données en combinant cette source d'information avec une connaissance a priori du système, souvent sous la forme d'un modèle dynamique. Formellement, un tel problème peut être défini via un modèle espace-état :

$$\mathbf{x}_{t+1} = \mathcal{M}(\mathbf{x}_t), \tag{1}$$

$$\mathbf{y}_t = \mathcal{H}_t(\mathbf{x}_t) + \boldsymbol{\epsilon}_t, \quad \boldsymbol{\epsilon}_t \sim \mathcal{N}(0, \boldsymbol{\Sigma}), \tag{2}$$

où $\mathbf{x}_t \in \mathbb{R}^n$ représente un vecteur d'état, $\mathbf{y}_t$ une observation liée à $\mathbf{x}_t$ via l'opérateur d'observation $\mathcal{H}_t$ (consistant souvent en un masque binaire) avec un bruit $\boldsymbol{\epsilon}_t$ que l'on suppose généralement gaussien avec une moyenne nulle et une covariance $\boldsymbol{\Sigma}$ indépendante du temps t. Étant donné un ensemble d'observations $\mathbf{y}_{1:T}$, il est généralement impossible de reconstituer avec certitude l'état $\mathbf{x}_{1:T}$ du système étudié au cours du temps, et il est donc important de pouvoir mesurer l'incertitude associée à un état assimilé. Plus généralement, on peut étudier la distribution de probabilité suivie par l'état $\mathbf{x}$ du système sachant un ensemble d'observations $\mathbf{y}$, c'est-à-dire $p(\mathbf{x}_{1:T}|\mathbf{y}_{1:T})$.

Ici, nous nous basons sur le modèle CODA proposé par [12]. Ce modèle permet une estimation directe de l'état le plus probable du système dynamique en fonction d'un ensemble parcimonieux d'observations $\mathbf{y}_{t-w:t+w}$ disponibles dans une fenêtre temporelle autour de l'état reconstitué au temps t, soit :

$$G_\theta(\mathbf{y}_{t-w:t+w}) = \hat{\mathbf{x}}_t \approx \arg\max_{\mathbf{x}\in\mathbb{R}^n} p(\mathbf{x}_t = \mathbf{x}|\mathbf{y}_{t-w:t+w}). \tag{3}$$

avec w déterminant la taille de la fenêtre d'observations. Cependant, il s'agit d'une estimation ponctuelle, qui ne permet pas le calcul d'une incertitude, ni a fortiori une estimation de la distribution suivie par l'état du système sachant les données observées. Dans les sections suivantes, nous étudierons différentes extensions de ce modèle permettant une quantification de l'incertitude ainsi que leurs performances respectives. Ainsi, on substituera à l'équation (3) une estimation complète de la distribution $p(\mathbf{x}_t|\mathbf{y}_{t-w:t+w})$ de l'état au temps t.

2 Contexte

2.1 Le modèle CODA déterministe

Commençons par une recontextualisation du modèle CODA. Vis-à-vis de la plupart des approches alternatives d'apprentissage automatique pour l'assimilation de données, la méthode de [12] permet d'entraîner le modèle qui réalise l'assimilation directement à partir d'observations bruitées et incomplètes, sans jamais devoir comparer ses prédictions à une vérité terrain $\mathbf{x}$ complète et sans bruit. En contrepartie, la dynamique $\mathcal{M}$ du système est supposée complètement différentiable et connue avec un certain degré d'exactitude. Dans le travail ici présenté, on supposera qu'elle l'est de manière exacte, mais le modèle peut aussi être adapté pour apprendre, par exemple, un paramètre inconnu de la dynamique. Le modèle CODA prend en entrée une fenêtre d'observations $\mathbf{y}_{t-w:t+w}$ accompagnée d'un masque binaire $\mathbf{H}_{t-w:t+w}$ indiquant quelles variables sont observées à chaque pas de temps, et renvoie l'estimation de l'équation (3), qui approxime le maximum a posteriori de la distribution réelle de $\mathbf{x}_t$. Pour ce faire, les auteurs de [12] introduisent l'architecture 1.5D Unet, qui adapte l'architecture Unet de [10] à un cas où l'entrée comporte 2 dimensions (la dimension spatiale et la dimension temporelle de $\mathbf{y}_{t-w:t+w}$) tandis que la sortie comporte seulement la dimension spatiale de $\mathbf{x}_t \in \mathbb{R}^n$. La dernière couche du réseau est une couche de convolution 1D, liant un état latent $\mathbf{z}_t \in \mathbb{R}^{n\times c}$ (avec c la "profondeur", au même titre que les 3 couleurs d'un pixel pour une convolution 2D sur une image) à un état prédit $\hat{\mathbf{x}}_t \in \mathbb{R}^{n\times 1}$ (pour le système Lorenz-96, mais on peut généraliser à des systèmes comportant plusieurs variables par emplacement spatial). Le modèle CODA est entraîné via la fonction de coût

$$L(\theta) = \mathbb{E}_t\Big[\sum_{i=0}^{w} ||\mathcal{H}_{t+i} \circ \mathcal{M}^{(i)}(\hat{\mathbf{x}}_t) - \mathbf{y}_{t+i}||^2 + \alpha||\hat{\mathbf{x}}_{t+w} - \mathcal{M}^{(w)}(\hat{\mathbf{x}}_t)||^2\Big]. \tag{4}$$

Le premier terme caractérise la fidélité aux données observées, tandis que le second encourage le modèle à être consistent vis-à-vis des prédictions réalisées à un autre pas de temps avec

la fenêtre décalée des observations $\mathbf{y}_{t:t+2w}$. Le paramètre α définit le poids relatif de ces deux termes.

2.2 Différentes méthodes pour obtenir une quantification de l'incertitude

S'il est difficile d'accéder à la distribution réelle $p(\mathbf{x}_t|\mathbf{y}_{t-w:t+w})$, il est désirable d'approximer au moins sa variance, de manière à pouvoir produire une quantification de l'incertitude associée à une prédiction ponctuelle réalisée pour un ensemble donné d'observations. Notons que les sources d'incertitude associées à la prédiction d'un modèle d'apprentissage machine sont multiples. On peut considérer qu'une part de l'incertitude est due à l'incapacité de ce modèle à tirer entièrement parti de ses données d'entraînement, tandis qu'une autre partie est due au caractère incomplet de ces données vis-à-vis du problème que l'on cherche à résoudre. Pour une discussion plus détaillée sur les différents types d'incertitude, on pourra se référer par exemple à [5]. Passons maintenant en revue quelques méthodes classiquement utilisées pour adapter un réseau de neurones représentant un modèle déterministe afin de réaliser une quantification d'incertitude pour ses prédictions. Les méthodes ici discutées sont présentées plus en détail dans [5].

L'une des méthodes les plus simples pour quantifier l'incertitude est d'entraîner séparément plusieurs modèles déterministes, qui constituent les membres d'un ensemble. Ainsi, la variance entre les prédictions des membres de l'ensemble étant donnée une même entrée peut être considérée comme une mesure de l'incertitude de l'ensemble. Les approches basées sur les ensembles, telles que le filtre de Kalman d'ensemble [2] et les forêts d'arbres décisionnels [9], sont en effet populaires pour la quantification d'incertitudes. En ce qui concerne les réseaux de neurones, [7] suggère d'entraîner plusieurs instances d'un même modèle, avec différentes initialisations des paramètres, sur le même jeu de données.

Les ensembles de réseaux de neurones sont simples à implémenter mais coûteux à entraîner, et ne permettent qu'une modélisation rudimentaire de la distribution de l'état comme une somme de Dirac, généralement équiprobables. Une approche alternative consiste à recourir au *dropout*, c'est-à-dire à la désactivation aléatoire de certains neurones du modèle. En effet, les auteurs de [3] proposent d'utiliser le dropout pendant l'entraînement mais aussi l'inférence du modèle, de manière à obtenir une sortie probabiliste, et relient cette approche aux réseaux de neurones bayésiens.

Une dernière méthode que nous étudierons ici est l'approche paramétrique, qui consiste à faire en sorte que la sortie d'un réseau de neurones contienne les paramètres d'une distribution de probabilité dont on peut tirer des échantillons de manière différentiable. Un exemple classique est de paramétrer une distribution gaussienne conditionnée à l'entrée du modèle, avec une matrice de covariance diagonale. Les paramètres du réseau de neurones peuvent alors être entraînés via le *reparameterization trick* proposé par [6].

2.3 Évaluation de la qualité des incertitudes

Dans un contexte où l'on s'intéresse à une prédiction probabiliste et non ponctuelle, la plupart des métriques usuelles, telles que l'erreur quadratique moyenne, ne permettent plus d'entraîner ni d'évaluer correctement les modèles. Dans notre cas, la distribution de probabilité réelle de l'état étant donné un ensemble d'observations ne peut pas être facilement obtenue, et on se contentera donc de comparer une vérité terrain ponctuelle $\mathbf{x}_* \in \mathbb{R}^n$ avec une distribution de probabilité prédite $\mathbf{x}$. Pour ce faire, nous recourrons à 2 approches : le diagramme *spread-skill* [1] avec les métriques associées, et le *continuous ranked probability score* (CRPS) [4]. Nous effectuons ci-après une description rapide de ces notions, et référons le lecteur intéressé à [5] pour une discussion plus détaillée.

Le diagramme *spread-skill*, ou dispersion-exactitude, consiste à étudier le ratio entre la dispersion (spread) et l'exactitude (skill) des prédictions d'un modèle. La dispersion est définie comme l'écart-type des prédictions, tandis que l'exactitude correspond à la racine de l'erreur quadratique moyenne de la prédiction moyenne vis-à-vis de la vérité terrain. Pour tracer un diagramme à partir d'un jeu de données comportant un grand nombre d'exemples, on peut réaliser un histogramme des prédictions en fonction des dispersions associées, et pour chaque rectangle de cet histogramme calculer l'exactitude moyenne des prédictions. Ceci permet de tracer sur un diagramme la courbe de l'exactitude en fonction de la dispersion. Idéalement, cette courbe doit rester aussi proche que possible de la fonction identité, dans la mesure où la distribution de probabilité réelle des états comporte elle-même, par définition, une exactitude correspondant à sa dispersion en moyenne. En plus du diagramme lui-même, on peut utiliser 2 métriques pertinentes. Le ratio dispersion-exactitude (spread-skill ratio, SSRAT) calculé sur tous les exemples du jeu de données, a une valeur idéale de 1. Un ratio inférieur à 1 traduit des prédictions trop confiantes (sous-dispersées) tandis qu'une valeur supérieure à 1 traduit des prédictions trop peu confiantes (trop dispersées). La fiabilité dispersion-exactitude (spread-skill reliability, SSREL) calcule la somme des écarts en valeur absolue entre la courbe du diagramme et la fonction identité, pondérée par l'histogramme, et a une valeur optimale de 0.

Le CRPS se définit comme suit, à partir de la fonction de répartition F de la prédiction d'un modèle et d'une vérité terrain ponctuelle x_* :

$$\text{CRPS}(F, x_*) = \int_{-\infty}^{\infty} [F(x) - \mathbb{1}_{x \geq x_*}]^2 \mathrm{d}x, \tag{5}$$

où $\mathbb{1}_{x \geq x_*}$ vaut 0 quand $x < x_*$ et 1 sinon. Le CRPS est une généralisation de l'erreur absolue moyenne à des prédictions probabilistes, comme le montre son expression pour une prédiction consistant en un ensemble de M valeurs ponctuelles équiprobables $(x_i)_{1 \leq i \leq M}$, établie par [4] :

$$\text{CRPS} = \frac{1}{M} \sum_{i=1}^{M} |x_* - x_i| - \frac{1}{2} \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} |x_i - x_j|. \tag{6}$$

3 Expériences

Ici, nous étudions différentes manières concrètes d'étendre le modèle CODA, présenté en section 2.1, via des méthodes décrites en section 2.2. Similairement à [12], nous considérons comme cas d'étude le système dynamique Lorenz-96 [8] avec $n = 40$ variables, intégré numériquement avec un pas de temps $\Delta t = 0.01$ et un paramètre de forçage $F = 8$. Les données d'entraînement sont obtenues en masquant aléatoirement chaque variable de chaque pas de temps avec une probabilité de 75%, et en leur appliquant un bruit d'une amplitude $\sigma = 1$.

Pour rappel, l'entraînement est réalisé uniquement à partir de ces données masquées et bruitées, sans aucun accès aux trajectoires originelles.

On considère deux cas différents : l'un où les modèles sont entraînés sur une série temporelle de 10^4 pas de temps, soit une trajectoire de 100 secondes, et l'autre avec une série temporelle de 3×10^5 pas de temps, soit 3000 secondes. Pour chacun de ces cas, on entraîne quatre modèles probabilistes différents :

- Un ensemble de 5 modèles déterministes, chacun entraîné avec les hyperparamètres proposés par [12] avec différentes initialisations des paramètres. Lors de l'inférence, l'écart-type des prédictions de ces 5 membres étant donnée une même entrée est utilisé comme une quantification de l'incertitude de l'ensemble.

- Un modèle similaire aux modèles déterministes, auquel on ajoute un dropout de probabilité $p = 0.2$ à la couche de convolution finale du modèle. L'application de ce dropout au cours de l'inférence permet la définition implicite d'un ensemble de prédictions possibles.

- Un modèle paramétrique définissant une distribution gaussienne de covariance diagonale. Ce modèle est obtenu en doublant la taille de la couche de convolution finale du modèle. De cette manière, étant donnée une fenêtre d'observations $\mathbf{y}_{t-w:t+w}$, le modèle calcule $\boldsymbol{\mu}_t \in \mathbb{R}^n$, $\boldsymbol{\sigma}_t \in \mathbb{R}^n$, et la distribution de la sortie du modèle est alors définie comme $p(\mathbf{x}_t|\mathbf{y}_{t-w:t+w}) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, où $\boldsymbol{\Sigma}_t$ est une matrice diagonale dont les coefficients diagonaux sont donnés par $\boldsymbol{\sigma}_t$.

- Un modèle paramétrique définissant une distribution gaussienne de faible rang $r = 2$. La sortie du modèle contient alors $\boldsymbol{\mu}_t \in \mathbb{R}^n$, $\mathbf{D}_t \in \mathbb{R}^n$ et $\boldsymbol{L}_t \in \mathbb{R}^{n \times r}$. La matrice de covariance est définie par $\boldsymbol{\Sigma}_t = \text{diag}(\mathbf{D}_t) + \mathbf{L}_t\mathbf{L}_t^\intercal$, et finalement $p(\mathbf{x}_t|\mathbf{y}_{t-w:t+w}) \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$.

Les méthodes d'ensembles basées sur plusieurs modèles déterministes ou sur du dropout ne nécessitent pas de changement de leur fonction de coût vis-à-vis de l'approche déterministe de l'équation (4) (même si on pourrait envisager de les entraîner avec une métrique probabiliste comme le CRPS). Pour les méthodes paramétriques en revanche, un entraînement naïf via cette fonction de coût mènerait à une variance tendant vers 0 pour réduire le risque d'erreur. On remplace donc la fonction de coût de l'équation (4) par la nouvelle fonction de coût :

$$L(G) = \mathbb{E}_{t,\hat{\mathbf{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)} \Big[\sum_{i=0}^{w} ||(\mathcal{H}_{t+i} \circ \mathcal{M}^{(i)}(\hat{\mathbf{x}}_t) - \mathbf{y}_{t+i})||^2 + \alpha \ \log p(\mathcal{M}^{(w)}(\hat{\mathbf{x}}_t)|\boldsymbol{\mu}_{t+w}, \boldsymbol{\Sigma}_{t+w}) \Big]. \quad (7)$$

En pratique, le premier terme - caractérisant la fidélité aux données observées - reste inchangé par rapport à l'équation (4), si ce n'est qu'il est calculé en propageant dans le temps un échantillon de la distribution estimée $p(\mathbf{x}_t|\mathbf{y}_{t-w:t+w})$ plutôt qu'une prédiction ponctuelle. Sous certaines hypothèses sur le bruit des observations, ce terme peut être interprété comme la log-vraisemblance des observations vis-à-vis de la distribution prédite de l'état. Pour le deuxième terme - favorisant la cohérence des prédictions du modèle -, on substitue l'erreur quadratique moyenne de l'équation (4) par la log-vraisemblance de l'échantillon $\hat{\mathbf{x}}_t \sim \mathcal{N}(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t)$, avancé par la dynamique

TABLE 1 : Performances d'assimilation probabiliste des modèles ensembliste, dropout et gaussiens avec covariance diagonale ou covariance de faible rang, sur le système Lorenz-96. Les cellules "L96-10K" et "L96-300K" désignent respectivement des modèles entraînés sur une série temporelle comportant 10^4 et 3×10^5 pas de temps.

	Modèle	Ensemble	Dropout	Diagonale	Faible rang
L96-10K	SSRAT	0.736	**0.780**	0.731	0.587
	SSREL	**0.102**	0.103	0.105	0.162
	CRPS	0.288	**0.283**	0.295	0.306
	MSE	**0.228**	0.240	0.264	0.266
L96-300K	SSRAT	0.550	1.040	**1.035**	0.640
	SSREL	0.118	0.074	**0.012**	0.125
	CRPS	0.215	0.206	**0.195**	0.264
	MSE	**0.112**	0.129	0.124	0.199

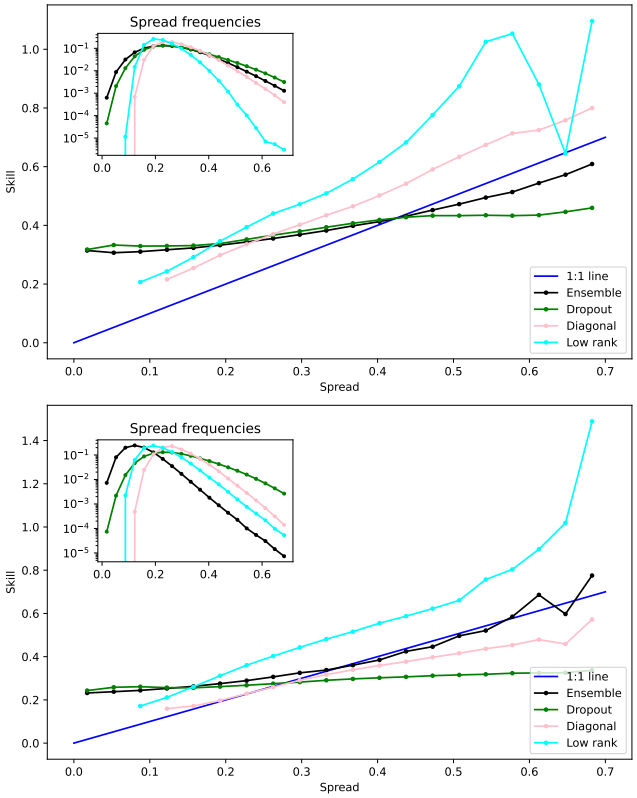

FIGURE 1 : Diagrammes dispersion-exactitude pour les modèles étudiés, avec des données d'entraînement comportant 10^4 pas de temps en haut et 3×10^5 pas de temps en bas.

connue $\mathcal{M}^{(w)}$, selon la distribution $\mathcal{N}(\boldsymbol{\mu}_{t+w}, \boldsymbol{\Sigma}_{t+w})$, dont les moments sont calculés par le même modèle à partir de la fenêtre d'observations $\mathbf{y}_{t:t+2w}$ centrée sur le temps $t + w$. Il est à noter que, avec une simple erreur quadratique moyenne, le modèle ne serait pas encouragé à produire des variances significatives, et produirait donc des prédictions très sous-dispersées. Néanmoins, notre choix basé sur un critère de vraisemblance n'est pas le seul possible, et on pourrait par exemple utiliser une estimation du CRPS à la place.

Les résultats quantitatifs des 4 modèles sur la base des critères SSRAT, SSREL, CRPS (décrits en section 2.3) et MSE (erreur quadratique moyenne) sont rapportés dans le

tableau 1. Les diagrammes dispersion-exactitude associés sont représentés sur la figure 1. Rappelons que la valeur idéale du SSRAT est de 1, tandis que le SSREL, le CRPS et la MSE ont une valeur idéale de 0. Parmi ces métriques, le CRPS nous semble être la plus complète pour décrire la qualité d'une prédiction probabiliste, puisqu'il rend compte à la fois du biais des prédictions et de la quantification des incertitudes. Les résultats rapportés pour chaque méthode utilisent donc les hyperparamètres minimisant en particulier le CRPS. La MSE est calculée avec une prédiction centrale de chaque modèle : la moyenne des 5 membres pour l'ensemble, la prédiction en activant tous les neurones pour le modèle dropout, et la moyenne prédite pour les modèles paramétriques.

Les résultats du tableau 1 mettent en lumière des performances dépendant fortement de la quantité de données utilisées pour l'entraînement. Premièrement, les métriques CRPS et MSE, qui rendent compte des biais des modèles, s'améliorent significativement avec le passage de 10^4 à 3×10^5 pas de temps dans les données d'entraînement, quel que soit le modèle. En outre, la méthode ensembliste obtient la meilleure MSE dans tous les cas, ce qui est attendu puisque cette méthode est souvent utilisée pour réduire les biais, et comporte plus de paramètres. En revanche, cette méthode ne permet pas un contrôle précis des incertitudes, ce qui se traduit par des résultats sous-optimaux en termes de SSRAT, SSREL et CRPS. Les méthodes basées sur une couche de dropout et sur une distribution gaussienne diagonale permettent pour leur part une calibration plus facile de l'incertitude, respectivement via la proportion de dropout ou le paramètre α de l'équation (7), ce qui se traduit par de meilleurs résultats de SSRAT, SSREL et CRPS dans le cas d'une grande quantité de données d'entraînement. En particulier, la méthode basée sur la paramétrisation d'une distribution gaussienne de covariance diagonale permet de très bonnes performances dans la métrique SSREL avec beaucoup de données, ce qui se traduit clairement dans le diagramme dispersion-exactitude du bas de la figure 1. Cependant, la distribution gaussienne paramétrée avec une matrice de covariance de rang faible obtient les moins bonnes performances dans les 2 configurations, bien qu'elle soit plus expressive que la gaussienne de variance diagonale et qu'elle obtienne un meilleur coût de validation pendant l'entraînement. Cela tend à suggérer que la fonction de coût de l'équation (7) pourrait ne pas être parfaitement alignée avec la qualité des distributions prédites, notamment quand la covariance n'est pas diagonale.

4 Conclusion

Dans cet article, nous avons traité du problème de l'approximation de la distribution des états complets d'un système dynamique lorsque seules des observations partielles et bruitées de ces états sont disponibles. En particulier, nous avons présenté un modèle d'apprentissage profond résolvant un problème d'assimilation de manière déterministe, et proposé plusieurs extensions de ce modèle visant à rendre ses prédictions probabilistes. Les qualités des distributions prédites par ces extensions ont été comparées via différentes métriques, pour le système Lorenz-96 avec différentes quantités de données d'entraînement. Parmi les nombreuses extensions possibles de ce travail, on adaptera dans de futurs travaux nos méthodes à des contextes plus difficiles où la dynamique du système n'est pas entièrement connue.

5 Bibliographie

Références

[1] Luca DELLE MONACHE, F Anthony ECKEL, Daran L RIFE, Badrinath NAGARAJAN et Keith SEARIGHT : Probabilistic weather prediction with an analog ensemble. *Monthly Weather Review*, 141(10):3498–3516, 2013.

[2] Geir EVENSEN : The ensemble kalman filter : Theoretical formulation and practical implementation. *Ocean dynamics*, 53:343–367, 2003.

[3] Yarin GAL et Zoubin GHAHRAMANI : Dropout as a bayesian approximation : Representing model uncertainty in deep learning. *In international conference on machine learning*, pages 1050–1059. PMLR, 2016.

[4] Tilmann GNEITING et Adrian E RAFTERY : Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

[5] Katherine HAYNES, Ryan LAGERQUIST, Marie MCGRAW, Kate MUSGRAVE et Imme EBERT-UPHOFF : Creating and evaluating uncertainty estimates with neural networks for environmental-science applications. *Artificial Intelligence for the Earth Systems*, 2(2):220061, 2023.

[6] Diederik P KINGMA, Max WELLING *et al.* : Auto-encoding variational bayes, 2013.

[7] Balaji LAKSHMINARAYANAN, Alexander PRITZEL et Charles BLUNDELL : Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

[8] Edward N LORENZ : Predictability : A problem partly solved. *In Proc. Seminar on predictability*, volume 1, pages 1–18. Reading, 1996.

[9] Nicolai MEINSHAUSEN et Greg RIDGEWAY : Quantile regression forests. *Journal of machine learning research*, 7(6), 2006.

[10] Olaf RONNEBERGER, Philipp FISCHER et Thomas BROX : U-net : Convolutional networks for biomedical image segmentation. *In Medical image computing and computer-assisted intervention–MICCAI 2015 : 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015.

[11] Nitish SRIVASTAVA, Geoffrey HINTON, Alex KRIZHEVSKY, Ilya SUTSKEVER et Ruslan SALAKHUTDINOV : Dropout : a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[12] Vadim ZINCHENKO et David S GREENBERG : Combined optimization of dynamics and assimilation with end-to-end learning on sparse observations. *arXiv preprint arXiv :2409.07137*, 2024.