

# Modèle de langage GP-VAE à autorégressivité latente

Yves Ruffenach  
[yves@ruffenach.eu](mailto:yves@ruffenach.eu)  
ORCID : [0009-0009-4737-0555](https://orcid.org/0009-0009-4737-0555)

17 novembre 2025

## Résumé

Nous étudions un schéma d'autorégressivité entièrement latente fondé sur un processus gaussien (GP) intégré dans un auto-encodeur variationnel (VAE). Dans ce cadre, la dynamique séquentielle est transférée du plan des observations vers un espace latent continu, tandis que la génération linguistique demeure parallèle grâce à un décodeur non-autorégressif. Nous proposons une formulation méthodologique complète, incluant un prior GP causal, un posterior amorti structuré et un protocole d'apprentissage basé sur une ELBO régularisée. L'évaluation empirique, menée dans un cadre volontairement limité de preuve de concept (POC), montre que le modèle peut être entraîné de manière stable et que les variantes séquentielle et parallèle produisent des résultats cohérents. Ce travail suggère qu'une partie de la structure temporelle d'un modèle de langage peut être prise en charge par la géométrie probabiliste du latent plutôt que par des opérations neuronales explicites.

**Mots-clés.** Gaussian Process VAE ; Modèles séquentiels ; Autorégressivité latente ; Modélisation de langage ; Raisonnement ; Modèles génératifs bayésiens ; Apprentissage profond.

## 1 Introduction

Les progrès récents des modèles de langage ne tiennent plus uniquement à l'augmentation de leur taille ou à la sophistication de leurs mécanismes internes. Mixtures of Experts, architectures de grande échelle et fine-tuning massif ont montré leur efficacité, mais l'expérience collective de la communauté met aujourd'hui en évidence un point central : les modèles les plus performants sont ceux qui raisonnent. Ils combinent plusieurs perspectives, explorent différentes chaînes d'inférence, dialoguent entre eux et exploitent des mécanismes de mise à jour distribuée au sein d'un écosystème de modèles.

Ce constat renoue avec une intuition plus ancienne : dans la tradition du Logos, langage et raison relèvent d'un même geste intellectuel. Un modèle capable de structurer une pensée — c'est-à-dire d'organiser une dynamique interne cohérente — est aussi un modèle capable de générer un langage cohérent. L'unification entre raisonnement et langage est, en un sens, antérieure à l'informatique moderne.

C'est dans ce cadre que s'inscrit le présent travail. Plutôt que de considérer le langage comme le point de départ du modèle, nous partons d'une dynamique de raisonnement : une structure interne continue qui précède, guide et contraint la génération symbolique. Cette idée s'appuie sur une série de travaux montrant que les auto-encodeurs variationnels (VAE) offrent une expressivité latente particulièrement adaptée au one-shot learning, à l'adaptation rapide et à la structuration géométrique de représentations continues — des propriétés difficiles à obtenir avec des architectures strictement autorégressives.

Nous explorons alors l'hypothèse suivante : la dynamique séquentielle d'un modèle de langage peut être déplacée du plan observable vers l'espace latent. Au lieu de porter la causalité sur les tokens via une autorégression explicite ou une auto-attention massive, nous confions la

dépendance temporelle à une covariance analytique définie par un processus gaussien (GP). La causalité devient ainsi une propriété mathématique du latent plutôt qu'une boucle neuronale sur les observations.

À noter que l'autorégressivité considérée dans ce travail opère exclusivement dans l'espace latent. Elle ne doit pas être confondue avec l'autorégression classique utilisée par les modèles opérant au niveau des tokens — tels que les Transformers — qui porte directement sur les observations. Cette distinction est fondamentale : notre modèle explore une causalité interne latente plutôt qu'une dépendance séquentielle explicite entre tokens.

Cette orientation conduit au modèle étudié dans cette publication : un GP-VAE doté d'un schéma d'autorégressivité purement latente, dans lequel

- le prior GP impose la continuité et la causalité internes,
- l'encodeur apprend un posterior amorti structuré,
- le décodeur demeure entièrement parallèle.

Dans ce cadre, le langage n'est plus généré par déroulement autorégressif, mais comme la projection d'une trajectoire latente continue. Les chapitres suivants formalisent ce paradigme, décrivent la méthodologie retenue pour le rendre capable de passer à l'échelle, et en évaluent la validité empirique dans un proof of concept contrôlé.

Notons enfin qu'un processus gaussien corrélé impose une géométrie métrique, mais non directionnelle : deux points également proches selon le kernel sont traités comme équivalents. Une telle symétrie peut créer des ambiguïtés dans une tâche séquentielle comme la modélisation du langage. L'introduction d'une autorégressivité purement latente lève précisément cette ambiguïté : en forçant chaque  $z_t$  à dépendre de l'historique  $z_{<t}$ , elle confère au GP un sens directionnel explicite et permet d'arbitrer entre des contributions latentes autrement interchangeables. La dynamique latente ainsi orientée combine la géométrie corrélée du GP avec une progression causale cohérente.

## 2 Travaux liés

### 2.1 Modèles variationnels et génératifs profonds

Depuis la proposition du Variational Autoencoder (VAE) par Kingma & Welling [10], les modèles génératifs variationnels constituent un cadre central pour la modélisation probabiliste latente. Un VAE relie un espace latent continu  $z$  à un espace d'observations complexes  $x$  via un encodeur  $q_\phi(z | x)$  et un décodeur  $p_\theta(x | z)$ , optimisés par la borne inférieure d'évidence (ELBO) :

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x | z)] - D_{\text{KL}}(q_\phi(z | x) \| p(z)). \quad (1)$$

Le terme KL contrôle la régularisation du latent. Parmi les variantes notables, le  $\beta$ -VAE [6] pondère  $D_{\text{KL}}$  pour favoriser le désenchevêtrement des facteurs latents, c'est-à-dire la séparation des dimensions cachées qui structurent la représentation interne ; les VAEs conditionnels étendent l'apprentissage à un contexte externe ; les modèles hiérarchiques [19] et les modèles séquentiels [3, 9] visent à représenter des dépendances temporelles multi-échelles.

Tous cherchent à capturer, par une structure probabiliste latente, des régularités transférables au-delà des données observées, un atout pour le few-shot ou le one-shot learning [18].

### 2.2 Processus gaussiens et apprentissage profond bayésien

Un processus gaussien (GP) [17] définit une distribution sur des fonctions :

$$f(t) \sim \mathcal{GP}(m(t), k(t, t')) , \quad (2)$$

où  $m$  est la moyenne et  $k$  le noyau de covariance.

Chaque observation affine la matrice  $K$ , rendant le GP non paramétrique : la complexité croît avec le nombre de données. L’association des GPs à des réseaux neuronaux via le *Deep Kernel Learning* [22] a permis de combiner flexibilité neuronale et rigueur bayésienne.

Introduit comme prior latent, le GP donne naissance aux GP-VAEs, où la covariance corrèle les latents  $z_t$  et  $z_{t'}$ . Casale et al. [1] ont proposé le *Gaussian Process Prior VAE* structurant l’espace latent ; Fortuin et al. [2] l’ont appliqué à l’imputation temporelle ; Pearce et al. [15, 14] ont développé plusieurs variantes d’auto-encodeurs bayésiens à prior GP.

Ces modèles apprennent non pas des points indépendants, mais une trajectoire latente cohérente, dont la continuité est imposée par le noyau  $k_\psi(t, t')$ . Cependant, la corrélation ainsi induite demeure symétrique et ne porte pas de causalité temporelle explicite.

### 2.3 Vers l’autorégressivité latente et la généralisation one-shot

Rezende et al. [18] ont montré qu’un modèle génératif profond peut produire des exemples cohérents à partir d’un seul échantillon (*one-shot generalization*). Ces travaux ont inspiré des approches où un prior corrélé améliore la cohérence contextuelle en faible donnée.

Des travaux plus récents, notamment Wang et al. [21], ont montré que des GPs densifiés intégrés à des réseaux profonds améliorent significativement les performances few-shot.

L’idée centrale est la suivante : si un GP introduit naturellement des corrélations entre latents, il peut également servir de base à une dynamique séquentielle latente. La dépendance causale peut ainsi être transférée du plan observable vers l’espace latent continu.

La factorisation conditionnelle d’un GP causal s’écrit :

$$p(z_t | z_{<t}) = \mathcal{N}\left(k_{12}^\top K_{11}^{-1} z_{<t}, K_{22} - k_{12}^\top K_{11}^{-1} k_{12}\right), \quad (3)$$

ce qui permet ensuite un décodage parallèle :

$$p_\theta(x | z_{1:L}). \quad (4)$$

Ainsi, la mémoire séquentielle est assurée par la structure de covariance du GP, et non par une récursion neuronale explicite.

### 2.4 Dynamiques séquentielles latentes et originalité du concept d’autorégressivité purement latente

La littérature sur les modèles séquentiels à variables latentes (VAE dynamiques, modèles d’état latents, RNN-VAEs, etc.) explore diverses dépendances temporelles, mais aucune ne formalise explicitement ce que nous nommons ici une *autorégressivité purement latente* :

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t | z_{<t}), \quad (5)$$

où chaque  $z_t$  est conditionné sur l’historique complet  $z_{<t}$ .

Les modèles antérieurs — en particulier Fraccaro et al. [3] et Karl et al. [9] — reposent sur des transitions markoviennes de la forme  $p(z_t | z_{t-1})$ . Kingma et al. [11] ont introduit les *Inverse Autoregressive Flows*, qui opèrent sur la densité latente mais ne définissent pas une dynamique temporelle causale.

Les travaux plus récents, notamment Zhou et al. [24] et Klushyn et al. [12], utilisent des modèles d’états latents continus plus expressifs, mais sans formaliser une dépendance causale complète.

À notre connaissance, aucune étude ne combine simultanément :

- un prior GP corrélé,
- une génération latente séquentielle causale  $p(z_t | z_{<t})$ ,

- un décodeur entièrement non-autorégressif.

Nous proposons ici cette combinaison comme contribution originale : un GP-VAE à autorégressivité purement latente, où la causalité est intégrée directement dans la structure probabiliste du latent. La formulation correspondante est :

$$p(z_{1:L}) = \prod_{t=1}^L \mathcal{N}(z_t; f_\psi(z_{<t}), \Sigma_\psi(z_{<t})), \quad (6)$$

ce qui offre une continuité temporelle analytique et une cohérence bayésienne interne. Cette structure englobe les dépendances Markoviennes et AR( $p$ ) (AutoRegressive models of order  $p$ ) comme cas particuliers, et offre une alternative analytique à la modélisation attentionnelle des Transformers en déplaçant la causalité dans le latent.

## 2.5 Limites actuelles et positionnement conceptuel

Les GP-VAEs existants présentent trois contraintes majeures :

1. l'absence de causalité temporelle dans le GP standard ;
2. la complexité cubique  $O(L^3)$  liée à l'inversion de covariance ;
3. l'absence de dynamique séquentielle explicite.

Le modèle proposé — un GP-VAE à autorégressivité purement latente — vise à surmonter ces limites : le GP assure la cohérence probabiliste globale ; la factorisation  $p(z_t | z_{<t})$  introduit une direction causale explicite ; et le décodeur parallèle permet une génération simultanée.

Cette structure remplace la mémoire neuronale par une covariance analytique, réduisant le coût computationnel tout en préservant la cohérence sémantique. Nous formulons ainsi une alternative conceptuelle aux modèles autorégressifs massifs : la cohérence linguistique n'émerge plus d'une récurrence symbolique, mais de la continuité probabiliste du latent.

Il s'agit d'une proposition méthodologique nouvelle, non encore implémentée dans la littérature existante.

## 2.6 Implémentation et outils associés

L'implémentation repose sur GPyTorch [4]. L'algorithme BBMM (*Blackbox Matrix–Matrix Multiplication*) permet une inversion de covariance en  $O(L^2)$  au moyen de gradients conjugués vectorisés. Ce cadre supporte l'optimisation des hyperparamètres du noyau et l'apprentissage à grande échelle sur GPU.

Les noyaux utilisés (RBF, Matérn [5], Spectral Mixture [23]) induisent des géométries latentes différentes, permettant d'explorer la continuité temporelle ou la périodicité internes.

## 2.7 Synthèse

Ce chapitre situe le modèle proposé au croisement de trois lignées majeures :

- les auto-encodeurs variationnels [10, 6, 19, 3],
- les modèles latents corrélés de type GP-VAE [1, 2, 15, 7],
- les approches de généralisation one-shot [18, 21].

L'autorégressivité purement latente étend les GP-VAEs en y intégrant une causalité temporelle explicite et un décodeur non-autorégressif. Elle déplace la causalité du niveau symbolique vers un espace probabiliste continu, ouvrant la voie à des modèles de langage plus compacts, stables et interprétables.

Si les approches existantes ont montré que les processus gaussiens offrent une continuité statistique utile, aucune ne formalise encore une causalité pleinement latente et séquentielle. L'autorégressivité purement latente proposée ici prolonge cet héritage en transférant la dépendance temporelle de l'espace des observations vers celui du latent.

Le chapitre suivant expose la méthodologie permettant de concrétiser ce modèle : réduction de la complexité computationnelle, structuration de l'encodeur, définition du prior latent causal, décodeur parallèle, et protocole expérimental. L'objectif est de montrer comment ce cadre théorique devient un dispositif capable de passer à l'échelle, apte à traiter des séquences longues tout en préservant la cohérence bayésienne du processus latent.

## 3 Modèles proposés – Méthodologie

### 3.1 Réduction de la complexité computationnelle

L'un des principaux défis liés à l'intégration d'un processus gaussien (GP) dans un modèle séquentiel concerne sa complexité computationnelle. En effet, l'évaluation d'un noyau temporel sur  $T$  pas conduit à une matrice de covariance  $K \in \mathbb{R}^{T \times T}$  dont l'inversion exacte nécessite un coût  $O(T^3)$ . Cette dépendance cubique limite l'utilisation naïve des GPs sur des séquences longues, sauf à recourir à des approximations dédiées (inducing points, gradients conjugués, factorisations structurelles).

Parmi ces méthodes, l'approximation par inducing points [20] ramène l'inférence à un coût  $O(TM^2)$  en sélectionnant  $M \ll T$  points latents représentatifs  $u = \{u_1, \dots, u_M\}$ . De son côté, GPYTORCH [4] propose l'algorithme BBMM (Blackbox Matrix–Matrix Multiplication), permettant d'approximer l'inversion de  $K$  par gradients conjugués vectorisés, avec une complexité effective proche de  $O(T^2)$ . Combinée à une factorisation spatio-temporelle du noyau de la forme

$$K = K_t \otimes K_s,$$

cette approche permet d'étendre l'utilisation des GPs à des séquences nettement plus longues que celles traitables par une inversion exacte.

Cependant, le gain le plus déterminant ne provient pas seulement de ces optimisations numériques, mais d'un changement de niveau de représentation. Dans un modèle autorégressif classique (de type Transformer), la dépendance séquentielle est portée par les  $N$  tokens observables, chaque couche d'attention causale impliquant un coût  $O(N^2d)$ . Par exemple, une séquence de  $N = 1024$  tokens avec une dimension cachée  $d = 768$  requiert plusieurs dizaines de millions d'opérations par couche, ainsi qu'un stockage intermédiaire important pour les matrices  $Q$ ,  $K$  et  $V$ .

Dans le modèle que nous proposons, cette dépendance est transférée vers une séquence latente compacte  $z_{1:L}$ , typiquement 8 à 16 fois plus courte que la séquence de tokens, avec des vecteurs  $z_t \in \mathbb{R}^{d_z}$  où  $d_z \ll d$ . La dépendance temporelle dans le latent présente alors un coût

$$O(L^2 d_z),$$

soit un à deux ordres de grandeur inférieur à celui d'un modèle opérant directement sur les observations.

*Cette estimation renvoie au coût effectif obtenu dans notre implémentation, qui repose systématiquement sur les approximations GP (BBMM, inducing points) maintenant l'inférence quadratique. L'inversion exacte de la covariance, en  $O(L^3)$ , n'est pas utilisée en pratique.*

Cette réduction n'est donc pas une approximation du GP, mais le résultat d'un changement de topologie computationnelle : la dynamique séquentielle est codée dans la covariance du processus gaussien plutôt que recalculée via la self-attention à chaque couche.

Autrement dit, le modèle remplace un graphe d'attention dense par une structure de covariance factorisable, dont la continuité temporelle est déterminée analytiquement par le noyau. Le décodage peut alors être effectué en une seule passe parallèle, selon

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t | z_{<t}), \quad p(x | z_{1:L}) \text{ (parallèle)},$$

alors qu'un modèle autorégressif classique nécessite l'évaluation séquentielle de

$$p(x_t | x_{<t}) \quad \text{pour chaque } t.$$

Cette translation du niveau symbolique vers le niveau latent a deux conséquences directes : (i) le calcul n'est plus séquentiel mais entièrement vectorisé ; (ii) la rétro-propagation n'a plus à suivre une longue chaîne causale, puisque la dynamique temporelle est intégrée dans la covariance du GP.

En somme, la réduction de complexité observée dans le GP-VAE ne résulte pas uniquement d'optimisations numériques, mais d'un déplacement conceptuel de la causalité : la dynamique séquentielle n'est plus portée par une mémoire neuronale explicite (RNN, Transformer), mais par une structure probabiliste corrélée. Ce changement de perspective, combiné aux approximations GP maintenant le coût quadratique, rend possible des architectures bayésiennes capables de modéliser des séquences longues tout en conservant une cohérence temporelle et sémantique robuste.

### 3.2 Architecture et rôle du décodeur

L'architecture globale du modèle proposé repose sur une séparation fonctionnelle stricte entre trois composantes : l'encodeur amorti, le bloc latent corrélé (GP-AR) et le décodeur génératif. Cette séparation, typique des modèles variationnels, devient ici structurante : elle permet d'isoler la dynamique séquentielle dans l'espace latent et de déléguer la reconstruction des observations à un décodeur parallèle. Contrairement aux modèles autorégressifs classiques, où la dépendance temporelle se manifeste directement sur les tokens, la temporalité du GP-VAE est intégralement portée par la séquence des latents  $z_{1:L}$ .

L'encodeur amorti assure la projection des observations dans l'espace latent. Il paramétrise une distribution postérieure approximée

$$q_\phi(z_{1:L} | x_{1:N}), \tag{7}$$

généralement gaussienne, dont la moyenne et la variance sont produites par un réseau convolutionnel ou temporel hiérarchique, par exemple un Temporal Convolutional Network (TCN), c'est-à-dire un réseau à convolutions dilatées causales adapté aux dépendances séquentielles. Sa fonction est purement inférentielle : fournir une estimation amortie de la distribution latente à partir des données d'entrée. Le choix d'un encodeur convolutionnel plutôt qu'un Transformer complet permet de limiter le coût d'apprentissage, l'essentiel de la cohérence séquentielle étant assurée par le prior GP.

Le bloc latent corrélé constitue le cœur du modèle. Il définit un processus gaussien autorégressif sur la séquence latente, selon

$$p_\theta(z_{1:L}) = \prod_{t=1}^L p_\theta(z_t | z_{<t}), \quad p_\theta(z_t | z_{<t}) = \mathcal{N}(m_t, \Sigma_t), \tag{8}$$

où les moments  $(m_t, \Sigma_t)$  sont obtenus par conditionnement gaussien sur les pas précédents. Le noyau  $k_\psi(t, t')$  encode la structure de covariance du processus : un noyau exponentiel quadratique ou Matérn [5] impose une continuité temporelle, tandis qu'un noyau spectral [23] capture des périodicités latentes utiles pour les structures linguistiques récurrentes.

Ce bloc joue un rôle analogue à celui du mécanisme de mémoire dans les architectures neuronales séquentielles, mais sa dynamique est strictement probabiliste : la dépendance temporelle est modélisée analytiquement par la covariance  $K$ , et non paramétriquement par des poids de transition appris.

Enfin, le décodeur génératif traduit la séquence latente en observations concrètes (tokens, spectrogrammes, pixels, etc.). Il paramétrise la distribution

$$p_\theta(x_{1:N} \mid z_{1:L}) = \prod_{n=1}^N p_\theta(x_n \mid z_{1:L}), \quad (9)$$

ce qui permet une génération entièrement parallèle : tous les tokens sont produits simultanément à partir de la trajectoire latente complète.

Par construction, cette formulation reste compatible avec plusieurs familles de décodeurs : convolutionnels, Transformers légers, ou réseaux à attention restreinte. La factorisation latente garantit une modularité forte : on peut remplacer le décodeur sans altérer la dynamique interne, puisque celle-ci est intégralement portée par le GP-AR.

Le modèle se distingue ainsi des architectures autorégressives token-par-token, où chaque nouveau symbole est conditionné sur l'historique  $x_{<t}$ . Ici, la dépendance temporelle a déjà été intégrée dans  $z_{1:L}$  ; le décodeur ne fait qu'appliquer un mappage global latent–observable. Conceptuellement, cela se traduit par la différence suivante :

$$(\text{LLM}) \quad p(x) = \prod_t p(x_t \mid x_{<t}) \quad \text{vs.} \quad (\text{GP-VAE}) \quad p(x) = \int p(x \mid z) p(z) dz. \quad (10)$$

On peut alors distinguer deux niveaux de cohérence :

- une cohérence locale, résultant de la proximité entre  $z_t$  et  $z_{t-1}$ , qui garantit la fluidité entre unités consécutives ;
- une cohérence globale, imposée par la covariance du processus gaussien, qui contrôle les régularités de haut niveau (thème, ton, structure argumentative).

Ainsi, le GP agit comme un régulateur continu de la structure globale, tandis que le décodeur impose les contraintes locales du langage. En résumé, la puissance du modèle ne réside pas dans la complexité du décodeur, mais dans la structure probabiliste du latent. Le décodeur devient un interprète d'une dynamique déjà cohérente : il transforme une trajectoire continue en séquence symbolique. Cette décentralisation de la causalité — du token vers le latent — constitue le pivot méthodologique du modèle et rend possible une génération parallèle, stable et bayésienne, sans attention explicite sur les observations.

### 3.3 Autorégressivité purement latente

La notion d'autorégressivité purement latente constitue l'un des apports majeurs du modèle proposé. Elle désigne une dynamique séquentielle intégralement confinée à l'espace latent — c'est-à-dire une dépendance temporelle causale qui s'exprime entre les variables cachées  $z_{1:L}$ , indépendamment de toute observation  $x$ . Cette approche se distingue à la fois des processus markoviens simples (limités à une mémoire locale) et des modèles autorégressifs neuronaux classiques (portant la causalité sur les tokens observables).

#### 3.3.1 Principe général

Un modèle purement autorégressif latent construit la séquence des états internes comme une chaîne probabiliste complète :

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t \mid z_{<t}), \quad (11)$$

où chaque distribution conditionnelle est gaussienne :

$$p(z_t \mid z_{<t}) = \mathcal{N}(\mu_t, \Sigma_t), \quad (12)$$

$$\mu_t = k_{(t,<t)}^\top K_{(<t,<t)}^{-1} z_{<t}, \quad (13)$$

$$\Sigma_t = k_{tt} - k_{(t,<t)}^\top K_{(<t,<t)}^{-1} k_{(t,<t)}. \quad (14)$$

Ces expressions découlent du conditionnement gaussien classique et montrent que la dynamique interne du modèle est gouvernée non par un réseau récurrent, mais par la structure de covariance du processus gaussien. Chaque nouvel état latent est échantillonné à partir d'une distribution prédictive mise à jour par les latents antérieurs, ce qui assure une continuité temporelle fluide et analytique.

Cette structure peut être vue comme une forme bayésienne d'autorégression : la mémoire du passé est transmise non par des poids neuronaux, mais par des corrélations probabilistes. La génération se déroule intégralement dans le latent : le modèle construit d'abord une trajectoire cohérente  $z_{1:L}$ , puis le décodeur, en aval, traduit cette trajectoire en séquence observable. Contrairement à un RNN ou à un Transformer, aucune rétroaction symbolique n'intervient dans la boucle de génération : le modèle n'a jamais accès à ses propres sorties textuelles — d'où le qualificatif de *purement* latent.

### 3.3.2 Différence avec les approches markoviennes et avec les approches fondées sur les tokens

Il est crucial de distinguer cette structure des modèles markoviens ou autorégressifs classiques. Un modèle markovien latent définit une dépendance locale :

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t | z_{t-1}), \quad (15)$$

soit une mémoire d'ordre 1. Cette hypothèse simplifie l'inférence mais limite la portée des dépendances temporelles : elle convient pour des trajectoires physiques ou des signaux continus à corrélations courtes, mais se révèle insuffisante pour capturer les régularités syntaxiques ou thématiques d'un langage.

À l'inverse, un modèle autorégressif observé (type GPT ou Transformer causal) opère directement sur les tokens :

$$p(x) = \prod_{t=1}^N p(x_t | x_{<t}), \quad (16)$$

imposant une causalité explicite mais au prix d'une explosion combinatoire du contexte et d'une génération séquentielle lente. Dans ces modèles, la dépendance temporelle est symbolique : chaque décision linguistique est conditionnée sur les précédentes, ce qui rend le modèle sensible aux erreurs locales et incompatible avec une génération parallèle.

Le GP-VAE autorégressif latent se situe à mi-chemin : il conserve la causalité complète d'un modèle autorégressif (chaque  $z_t$  dépend de tous les  $z_{<t}$ ), tout en opérant dans un espace continu où la covariance impose une cohérence lisse et probabiliste. Le flux de dépendance reste causal, mais il est décorrélé du niveau symbolique — la syntaxe et la grammaire émergent a posteriori via le décodeur.

### 3.3.3 Lecture intuitive et portée

Sur le plan opérationnel, l'autorégressivité purement latente se manifeste comme un déploiement séquentiel du conditionnement gaussien : à chaque pas  $t$ , le modèle met à jour la distribution de  $z_t$  en fonction de l'historique latent  $z_{<t}$ , suivant la relation

$$z_t \sim \mathcal{N}(k_{(t,<t)}^\top K_{(<t,<t)}^{-1} z_{<t}, k_{tt} - k_{(t,<t)}^\top K_{(<t,<t)}^{-1} k_{(t,<t)}). \quad (17)$$

Cette itération engendre une chaîne corrélée et causale dans l'espace latent, où la mémoire du passé est transmise par la covariance du processus. La différence majeure avec une autorégression neuronale tient à la nature analytique du conditionnement : ici, la cohérence temporelle découle d'une structure de covariance apprise, non de poids séquentiels.

Conceptuellement, cela revient à déplacer la temporalité du langage dans un espace interne plus abstrait. La séquence latente encode la dynamique sémantique sous-jacente — transitions de thèmes, continuité argumentative, dépendances syntaxiques — avant toute émission observable. Le décodeur n'a plus qu'à projeter cette dynamique en surface linguistique, ce qui réduit la variance et améliore la cohérence stylistique.

### 3.3.4 Définition formelle

On dira qu'un modèle présente une autorégressivité sur les latents lorsque la distribution jointe des variables cachées se factorise de manière causale selon

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t | z_{<t}), \quad (18)$$

avec

$$p(z_t | z_{<t}) = f_\theta(z_{<t}) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \Sigma_t). \quad (19)$$

La fonction  $f_\theta$  peut être un processus gaussien, un *flow* masqué ou un petit Transformer latent, mais dans tous les cas elle ne dépend que des états antérieurs. Lorsque la dépendance est restreinte à  $z_{t-1}$ , on retrouve une transition markovienne d'ordre 1 ; lorsqu'elle s'étend à tout l'historique  $z_{<t}$ , on parle d'autorégression latente complète.

Cette formulation assure une causalité interne intégrale, distincte de la causalité externe des modèles observables. Elle conserve la régularité analytique du GP tout en intégrant la directionnalité propre aux modèles séquentiels. Le résultat est une dynamique latente causalement orientée, compatible avec l'inférence amortie et la génération parallèle du VAE.

### 3.3.5 Synthèse

L'autorégressivité purement latente généralise et unifie plusieurs paradigmes :

- elle étend la dépendance markovienne locale vers une causalité globale ;
- elle remplace les mécanismes d'attention par une covariance analytique ;
- elle préserve la factorisation bayésienne qui fonde la stabilité des VAEs.

Le modèle n'apprend plus une mémoire séquentielle explicite, mais une structure de dépendance continue dans un espace probabiliste. Cette reformulation déplace la complexité du traitement du langage du plan des tokens vers celui de la géométrie latente — une translation conceptuelle qui sous-tend toute la suite de la méthodologie.

## 3.4 Formulation probabiliste et fonction objectif

La formulation complète du modèle repose sur le formalisme variationnel standard, enrichi d'une structure latente corrélée et causale. Le GP-VAE autorégressif définit une distribution conjointe sur les observations  $x$  et les variables latentes  $z$  :

$$p_\theta(x, z) = p_\theta(x | z) p_\theta(z), \quad (20)$$

où  $p_\theta(z)$  est un prior latent corrélé et causal, et  $p_\theta(x | z)$  le modèle génératif (le décodeur).

L'apprentissage consiste à maximiser la vraisemblance marginale des données :

$$\log p_\theta(x) = \log \int p_\theta(x, z) dz, \quad (21)$$

non calculable analytiquement, d'où l'approximation variationnelle classique :

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)] - D_{\text{KL}}(q_\phi(z | x) \| p_\theta(z)), \quad (22)$$

soit la borne inférieure d'évidence (ELBO), où  $D_{\text{KL}}$  désigne la divergence de Kullback–Leibler, qui mesure l'écart entre la distribution postérieure approximée et le prior latent.

### 3.4.1 Prior latent corrélé

Le prior latent joue ici un rôle central. Plutôt que d'imposer une indépendance factorisée entre les composantes de  $z$  (comme dans un VAE standard), nous utilisons un processus gaussien autorégressif qui encode explicitement les corrélations temporelles et la causalité :

$$p_\theta(z_{1:L}) = \prod_{t=1}^L \mathcal{N}(z_t; k_{(t,<t)}^\top K_{(<t,<t)}^{-1} z_{<t}, k_{tt} - k_{(t,<t)}^\top K_{(<t,<t)}^{-1} k_{(t,<t)}). \quad (23)$$

La matrice de covariance  $K$  est définie par un noyau différentiable  $k_\psi(t, t')$  paramétré par des hyperparamètres  $\psi$  (longueur d'échelle, variance, etc.). Cette construction garantit la continuité et la causalité internes : chaque  $z_t$  dépend de ses antécédents via le conditionnement gaussien, et l'ensemble forme une trajectoire cohérente dans l'espace latent.

### 3.4.2 Posterior amorti

L'approximation postérieure  $q_\phi(z | x)$  est choisie de forme factorisée mais dépendant implicitement du contexte encodé par l'encodeur :

$$q_\phi(z | x) = \prod_{t=1}^L \mathcal{N}(z_t; \mu_\phi(x_{\leq t}), \sigma_\phi(x_{\leq t})^2 I). \quad (24)$$

Cette forme permet d'amortir l'inférence : l'encodeur apprend une transformation directe  $x \mapsto (\mu, \sigma)$ , rendant le calcul du posterior indépendant du nombre d'itérations d'optimisation. L'utilisation de couches convolutionnelles ou TCN favorise la parallélisation tout en préservant la structure temporelle.

### 3.4.3 Décodeur génératif

Le décodeur  $p_\theta(x | z)$  implémente la projection latent–observation. Dans notre cadre, il est non-autorégressif, c'est-à-dire qu'il génère toutes les positions en parallèle à partir de la séquence latente complète. Chaque observation est modélisée par, typiquement :

$$p_\theta(x | z) = \prod_{n=1}^N \mathcal{N}(x_n; f_\theta(z)_n, \sigma_x^2 I) \quad \text{ou} \quad p_\theta(x | z) = \prod_{n=1}^N \text{Cat}(f_\theta(z)_n), \quad (25)$$

selon que les données sont continues (séries, images) ou discrètes (tokens).

Le réseau  $f_\theta$  peut être une architecture convolutionnelle ou un Transformer léger appliqué aux latents ; il traduit les régularités corrélées en structures observables (grammaire, ton, style).

### 3.4.4 Fonction objectif complète

En combinant ces éléments, l'ELBO du modèle devient :

$$\mathcal{L}(\theta, \phi) = \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)] - D_{\text{KL}}(q_\phi(z | x) \| p_\theta(z)). \quad (26)$$

Le premier terme maximise la vraisemblance de reconstruction (fidélité du décodeur), le second régularise la structure latente en rapprochant le posterior du prior GP-AR. Dans la pratique, le KL se décompose en une somme analytique de divergences gaussiennes pondérées, et peut être modulé par un coefficient  $\beta$  [6] pour ajuster le compromis entre reconstruction et régularisation :

$$\mathcal{L}_\beta = \mathbb{E}_{q_\phi(z|x)} [\log p_\theta(x | z)] - \beta D_{\text{KL}}(q_\phi(z | x) \| p_\theta(z)). \quad (27)$$

Une valeur  $\beta \approx 1$  assure un équilibre stable, mais selon la structure du corpus et la richesse du noyau,  $\beta$  peut être légèrement ajusté pour renforcer la régularisation sans perte de cohérence globale.

### 3.4.5 Hypothèses et notations

Pour clarté, les notations utilisées suivent les conventions suivantes :

- $x_{1:N}$  : séquence d'observations (tokens ou features) ;
- $z_{1:L}$  : séquence latente continue, avec  $L < N$  en général ;
- $p_\theta(z)$  : prior GP-AR causal ;
- $q_\phi(z | x)$  : posterior amorti (encodeur) ;
- $p_\theta(x | z)$  : décodeur parallèle ;
- $k_\psi$  : noyau de covariance (RBF, Matérn, Spectral, etc.) ;
- $\theta, \phi, \psi$  : paramètres du décodeur, de l'encodeur et du noyau.

Ces hypothèses garantissent la cohérence du modèle : le prior impose la structure temporelle, le décodeur assure la reconstruction linguistique, et l'ensemble forme un système génératif entièrement différentiable, où la causalité est intégrée au niveau latent plutôt qu'observable.

### 3.4.6 Extension bayésienne « G »

L'extension notée  $G$  généralise la formulation précédente en autorisant un noyau hiérarchique appris :

$$k_\psi(t, t') = k_{\psi_1}(t, t') + k_{\psi_2}(g(t), g(t')), \quad (28)$$

où  $g(t)$  est une fonction latente apprise (par exemple une projection du contexte).

Cette variante, que l'on peut désigner comme une extension  $G$ -GP-VAE, permet au processus gaussien d'adapter dynamiquement sa structure de covariance en fonction du contenu sémantique, reliant ainsi dépendances temporelles et structure linguistique apprise. L'extension bayésienne  $G$  n'introduit pas l'autorégressivité : celle-ci reste assurée par la factorisation causale du GP temporel. L'extension enrichit la covariance en ajoutant un terme sémantique appris, tandis que la structure temporelle demeure dictée par le noyau causal :

$$p(z_{1:L}) = \prod_{t=1}^L p(z_t | z_{<t}). \quad (29)$$

**Remarque sur l'évaluation.** L'extension  $G$  est présentée comme une généralisation conceptuelle du prior latent, permettant d'introduire une covariance contextuelle modulée par la structure sémantique. Elle offre la possibilité d'obtenir un latent plus expressif en reliant directement dynamique temporelle et contenu linguistique. Cependant, cette variante n'est pas activée dans les expériences du chapitre 6, qui se concentrent exclusivement sur le modèle de base ( $\Lambda$ -SEQ /  $\Lambda$ -PARA) afin d'isoler l'effet propre du schéma d'autorégressivité purement latente dans un cadre minimal. Activer  $G$  introduirait un mécanisme contextuel supplémentaire dans la covariance, modifiant la nature du proof-of-concept et brouillant l'analyse causale. L'extension  $G$  doit ainsi être comprise comme une généralisation théorique du cadre, et non comme un composant du protocole expérimental.

## 3.5 Justification et cohérence conceptuelle

### 3.5.1 Double cohérence du modèle

Le GP-VAE autorégressif proposé vise à unifier deux formes de cohérence souvent dissociées dans les modèles séquentiels :

- la cohérence probabiliste issue du cadre bayésien, garantissant la consistance interne entre prior, posterior et vraisemblance ;
- la cohérence linguistique ou structurelle, nécessaire pour que les séquences générées demeurent interprétables.

La première est assurée par le processus gaussien : la dépendance entre les latents  $z_{1:L}$  est exprimée de manière analytique, via une covariance qui capture les régularités temporelles et stylistiques. La seconde découle de la dynamique causale interne : l'autorégressivité latente fournit une continuité directionnelle qui aligne naturellement les représentations successives.

Ainsi, la chaîne

$$z_1 \rightarrow z_2 \rightarrow \cdots \rightarrow z_L$$

définit un fil narratif latent, que le décodeur traduit en cohérence grammaticale et sémantique. Cette articulation se formalise par la factorisation

$$p_\theta(x) = \int p_\theta(x | z) p_\theta(z) dz, \quad (30)$$

où  $p_\theta(z)$  porte la régularité temporelle (GP-AR) et  $p_\theta(x | z)$  la régularité symbolique (langage). La continuité de  $p_\theta(z)$  agit comme une garantie de cohérence globale, tandis que la capacité expressive du décodeur assure la cohérence locale.

### 3.5.2 Intuition géométrique

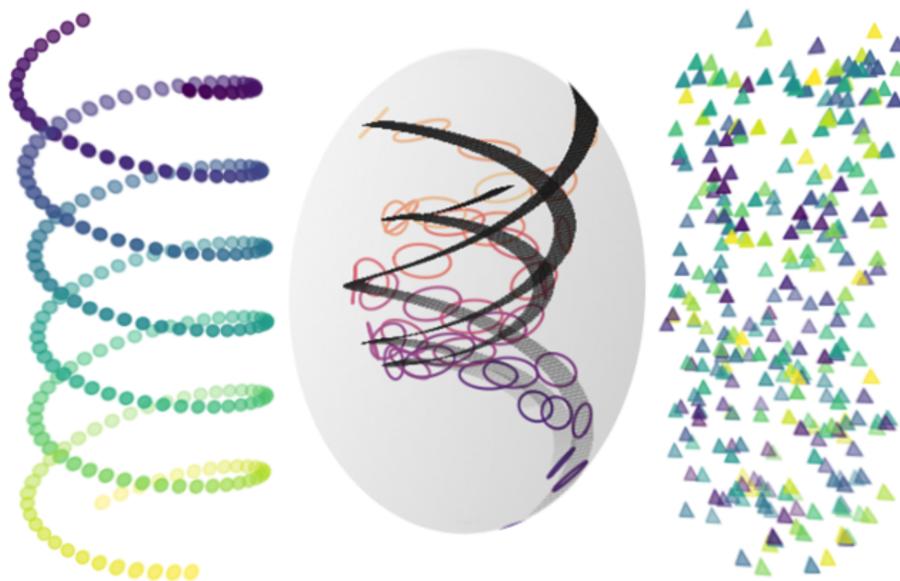


FIGURE 1 – Intuition géométrique du GP-VAE à espace latent autorégressif. La figure illustre la correspondance entre espace des données, espace latent corrélé et trajectoire générative.

Sur le plan géométrique, la séquence latente  $z_{1:L}$  peut être interprétée comme une trajectoire différentiable dans un espace latent continu, muni d'une métrique implicite définie par le noyau du processus gaussien. Le prior GP induit en effet une covariance

$$\Sigma_{t,t'} = k_\psi(t, t'), \quad (31)$$

qui organise les latents le long d'une variété temporelle régulière : deux positions sont d'autant plus proches géométriquement que leur contexte linguistique sous-jacent est similaire.

Dans ce cadre, la continuité temporelle devient une propriété géométrique. La courbe latente demeure lisse tant que le noyau impose une forte corrélation locale ; les changements de thème, de

style ou de registre correspondent alors à des déformations locales de cette trajectoire (variations de courbure, étirements, torsions).

L'autorégressivité purement latente introduit une causalité interne lisible dans cette géométrie. Le passage de  $z_t$  à  $z_{t+1}$  correspond à un déplacement le long d'une *géodésie probabiliste*, c'est-à-dire la trajectoire la plus probable selon le modèle. Chaque étape minimise simultanément :

- la variance prédictive imposée par le GP (continuité et régularité),
- la divergence causale imposée par la factorisation autorégressive (orientation temporelle).

Ainsi, la dynamique interne du modèle n'est pas un flux récursif neuronal, mais un chemin optimal dans une géométrie probabiliste.

**Lecture de la figure 1.** La figure schématisé la correspondance entre espace observé, espace latent et génération :

- **À gauche**, la double spirale représente la structure séquentielle dans l'espace des observations  $x_{1:L}$ .
- **Au centre**, une sphère translucide représente l'espace latent continu  $Z$ . Les rubans noirs correspondent à des trajectoires  $z(t)$  échantillonnées du GP dont la transparence croît avec le temps, symbolisant la causalité  $z_t \rightarrow z_{t+1}$ . Les ellipses colorées illustrent les distributions postérieures  $q(z | x)$ , alignées en une trajectoire lisse.
- **À droite**, un nuage de points représente les reconstructions  $\hat{x} \sim p(x | z)$  obtenues en une seule passe par le décodeur non-autorégressif.

**Synthèse géométrique.** On peut résumer la dynamique du modèle en quatre continuités :

1. **Continuité des données** : la structure séquentielle de  $x_{1:L}$ .
2. **Continuité latente** : la courbe imposée par le noyau  $k_\psi$ .
3. **Continuité causale** : la direction  $z_t \rightarrow z_{t+1}$  induite par la factorisation autorégressive.
4. **Continuité générative** : la cohérence linguistique produite par le décodeur à partir d'une trajectoire lisse.

La dynamique séquentielle n'est donc plus une mécanique neuronale (RNN, self-attention), mais une *géométrie probabiliste continue* dans laquelle évolue le modèle. Ce déplacement du point de vue — du symbolique vers le géométrique — constitue le fondement de la cohérence, de la stabilité et de l'intelligibilité du GP-VAE à autorégressivité purement latente.

### 3.5.3 Justification théorique

Le couplage entre un processus gaussien et une factorisation autorégressive n'est pas arbitraire : il découle de la compatibilité naturelle entre la mise à jour bayésienne et la causalité séquentielle. En notant

$$p(z_t | z_{<t}) = \mathcal{N}(m_t, \Sigma_t), \quad (32)$$

le calcul du conditionnement correspond exactement à l'étape de prédiction du filtre de Kalman [8], mais sans hypothèse de linéarité. Cette analogie a été formalisée dans la littérature des processus gaussiens dynamiques [17], où le GP peut être vu comme une généralisation non linéaire et non paramétrique du filtrage séquentiel.

Le modèle combine donc :

- la régularité globale des GPs (continuité, stationnarité locale) ;
- l'orientation causale caractéristique des modèles autorégressifs.

Cette hybridation peut être vue comme une forme continue de modèle d'état bayésien, où la dynamique interne n'est plus apprise par des poids mais intégrée analytiquement dans la covariance. Elle garantit que la génération est cohérente au premier ordre (continuité) et stable au second ordre (variance contrôlée).

De plus, la formulation variationnelle assure la cohérence normative du raisonnement probabiliste : chaque mise à jour de croyance sur  $z$  respecte les règles du calcul bayésien,

$$q_\phi(z | x) \approx \frac{p_\theta(x | z) p_\theta(z)}{p_\theta(x)}. \quad (33)$$

Ainsi, le modèle conserve la logique d'un estimateur MAP (Maximum A Posteriori) dont l'inférence est amortie — c'est-à-dire apprise par un réseau et réutilisable pour toutes les données — tout en maintenant la cohérence bayésienne globale.

### 3.5.4 Cohérence linguistique et sémantique

Les effets de cette structure se manifestent particulièrement dans les tâches linguistiques. Les dépendances syntaxiques courtes (accords, ponctuation, transitions locales) émergent de la continuité imposée par le conditionnement  $p(z_t | z_{<t})$ , tandis que les dépendances de plus haut niveau (thème, ton, registre) sont encodées dans la covariance  $k_\psi$ .

Le décodeur apprend alors à convertir cette structure probabiliste en régularités lexicales, produisant des séquences homogènes sans contrainte explicite d'attention. Ce mécanisme permet au modèle de maintenir un équilibre subtil : la souplesse stylistique d'un générateur neuronal, mais la stabilité sémantique d'un modèle probabiliste. En pratique, cela se traduit par des productions plus régulières, moins sensibles à la dérive ou aux erreurs cumulatives observées dans les modèles autorégressifs purement symboliques.

Cette interaction hiérarchique entre régularité latente et reconstruction linguistique renforce la cohérence des textes générés tout en préservant la flexibilité stylistique.

### 3.5.5 Complexité, stabilité et points d'attention

Si la formulation présente une cohérence conceptuelle forte, sa mise en œuvre requiert une attention particulière à plusieurs aspects :

- **Scalabilité** : l'emploi de méthodes d'approximation (inducing points, BBMM) est indispensable pour maintenir une complexité quasi quadratique. L'entraînement complet sur de longues séquences demeure coûteux, mais reste massivement parallélisable grâce à la factorisation latente ;
- **Stabilité numérique** : les inversions de matrices de covariance peuvent introduire des instabilités lorsque  $K$  devient mal conditionnée. Des régularisations de type jitter ( $K \leftarrow K + \varepsilon I$ , ajout d'un terme  $\varepsilon I$  sur la diagonale) et des mises à l'échelle adaptatives sont nécessaires pour garantir la convergence. Le jitter est une régularisation diagonale qui stabilise la matrice de covariance d'un GP en évitant les problèmes de conditionnement ;
- **Apprentissage conjoint** : l'optimisation conjointe des paramètres du GP ( $\psi$ ) et de ceux du VAE ( $\theta, \phi$ ) exige une gestion fine du rapport entre les gradients. Dans la pratique, une pondération progressive du terme KL (augmentation graduelle du terme de régularisation, ou annealing) facilite la stabilisation des premières étapes d'entraînement ;
- **Limites structurelles** : la génération parallèle, si elle offre un gain de vitesse substantiel, empêche la génération incrémentale (streaming). Ce compromis est assumé : la cohérence globale prime sur la production temps réel.

### 3.5.6 Synthèse

Le modèle GP-VAE autorégressif purement latent réconcilie les approches bayésiennes et séquentielles : il conserve la rigueur probabiliste du VAE, la continuité fonctionnelle du processus gaussien et la causalité directionnelle des modèles autorégressifs. Ce déplacement de la dynamique du plan observable vers le plan latent établit un nouveau cadre conceptuel :

- la causalité y est analytique plutôt que neuronale ;

- la mémoire, intégrée dans la covariance, remplace la récurrence explicite ;
- la cohérence linguistique émerge comme propriété géométrique du latent.

En somme, la méthodologie proposée montre qu'une autorégression purement latente, corrélée et bayésienne, peut servir de fondement à des modèles de langage compacts, stables et interprétables, tout en réduisant significativement la complexité computationnelle. Cette conclusion clôt la formalisation méthodologique ; les sections suivantes examineront empiriquement la validité et la performance du modèle dans divers contextes séquentiels.

## 4 Expérimentations – Validation expérimentale et limites du schéma latent autorégressif

Ce chapitre évalue empiriquement le schéma d'autorégressivité purement latente présenté à la section précédente. L'objectif n'est pas de prétendre à une validation exhaustive, mais d'établir, dans un cadre contrôlé et reproductible, que :

- un GP-VAE doté d'une causalité entièrement latente peut être entraîné de manière stable ;
- l'espace latent corrélé est effectivement utilisé, au-delà d'un simple effet de *KL-cap* ;
- deux méthodes d'échantillonnage (séquentielle vs parallèle) donnent lieu à des comportements cohérents ;
- le modèle surpassé la référence (baseline) autorégressive, tout en reconnaissant les limites de cette comparaison.

Nous insistons, dès l'introduction, sur le fait que ce chapitre constitue un *proof of concept* : les résultats doivent être interprétés comme une validation locale dans un cadre réduit, non comme une démonstration générale à grande échelle.

### 4.1 Objectifs expérimentaux et hypothèses testées

Nous testons quatre hypothèses, formulées de manière prudente et vérifiable.

**H1 – Capacité d'entraînement.** Un GP-VAE latent-autorégressif est entraînable et stable sur un corpus standard (WikiText-2 [13]), sans divergence numérique et avec une ELBO/token convergente.

**H2 – Utilisation effective du latent corrélé.** L'espace latent corrélé doit être réellement exploité. Nous mesurons :

- la KL/token brute et capée ;
- les comportements sous variations de  $k1\_cap$  et de  $\beta_{final}$  ;
- des ablations avec prior isotrope diagonale.

L'hypothèse n'est considérée comme validée que si :

- le modèle utilise davantage la structure corrélée qu'un VAE à prior diagonal ;
- les performances se dégradent en supprimant la corrélation (voir section 4.6.2).

**H3 – Cohérence entre  $\Lambda$ -SEQ et  $\Lambda$ -PARA.** Les deux variantes réalisent théoriquement la même loi jointe. Nous testons si, dans les limites numériques de la décomposition de Cholesky et de l'approximation TCN, leurs métriques — ELBO, NLL(cont), PPL(cont) et KL/token — demeurent statistiquement indiscernables. La NLL(cont) désigne la negative log-likelihood continue, évaluée directement sur les logits, c'est-à-dire les scores non normalisés produits par le décodeur avant le softmax (fonction transformant des scores en probabilités normalisées) ; la PPL(cont) est sa version exponentielle (perplexité continue), reflétant l'incertitude du modèle dans l'espace continu ; enfin, le KL/token mesure la divergence KL moyenne par token.

**H4 – Comparaison minimale avec un modèle autorégressif dans l'espace des observations.** Nous comparons  $\Lambda$  aux performances d'un Transformer baseline, uniquement comme point d'ancrage, sans en tirer de conclusion générale. La baseline n'est pas optimisée à grande échelle : elle sert de repère minimal pour situer la qualité générative du GP-VAE latent-autorégressif.

## 4.2 Protocole expérimental

### 4.2.1 Corpus et tokenisation

Les expériences sont conduites sur WikiText-2 [13] (splits officiels), tokenisé via le tokenizer GPT-2 [16], dont le vocabulaire compact est adapté aux modèles de petite taille. La longueur de séquence est fixée à  $T = 64$ , de sorte que le coût  $O(T^3)$  associé au GP reste compatible avec les ressources disponibles.

### 4.2.2 Tâche

La tâche considérée est une modélisation de langage de type autorégressif. Nous évaluons :

- la perplexité de validation PPL(val) (sur séquences complètes) ;
- la perplexité de continuation PPL(cont), via un protocole conditionnel standardisé (prompt + complétion).

### 4.2.3 Implémentation du schéma d'autorégressivité dans le latent

Avant de discuter des résultats, nous décrivons brièvement l'implémentation concrète du schéma d'autorégressivité purement latente utilisé pour les expériences.

**(a) Encodeur TCN → posterior diagonal temporel.** L'encodeur est un TCN hiérarchique causal qui produit un posterior factorisé :

$$q(z_{1:T} | x) = \prod_{t=1}^T \mathcal{N}(z_t; \mu_t, \text{diag}(\sigma_t^2)), \quad \mu, \log \sigma^2 \in \mathbb{R}^{B \times T \times d_z}.$$

La reparamétrisation standard est utilisée :

$$z_t = \mu_t + \sigma_t \odot \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, I).$$

**(b) Prior GP corrélé sur la trajectoire latente.** La dépendance temporelle est imposée via un processus gaussien indexé par le temps normalisé  $t \in [0, 1]$ . Pour une séquence de longueur  $T$ , on construit une matrice de covariance  $K_{tt} \in \mathbb{R}^{T \times T}$  :

$$K_{tt}(i, j) = \sigma^2 \exp\left(-\frac{(t_i - t_j)^2}{2\ell^2}\right) + \sigma^2 \text{nugget} \cdot \delta_{ij},$$

où  $\ell$  (longueur de corrélation),  $\sigma^2$  (variance) et le nugget relatif sont appris.

Le prior latent implicite est :

$$p(z_{1:T}) = \mathcal{N}(0, K_{tt} \otimes I_{d_z}).$$

En pratique, les hyperparamètres  $\ell$  et  $\sigma^2$  sont paramétrés à partir de variables libres transformées via une fonction *softplus*, et un terme de *jitter*  $\varepsilon I$  est ajouté sur la diagonale pour garantir la stabilité numérique.

La spécificité de notre approche tient au fait que la loi jointe du GP n'est pas seulement utilisée comme prior corrélé, mais explicitement factorisée en conditionnelles  $p(z_t | z_{<t})$  afin d'induire une causalité latente. Cette factorisation permet une génération séquentielle dans le latent ( $\Lambda$ -SEQ) ou parallèle via un échantillonnage bloc ( $\Lambda$ -PARA).

**(c) KL global temporel entre posterior diagonal et prior GP.** Le terme de régularisation est un KL global :

$$\text{KL}(q(z_{1:T} | x) \| p(z_{1:T})),$$

calculé en log-densité multivariée sur toute la trajectoire.

Ce schéma met toute la structure temporelle dans la covariance GP de  $z_{1:T}$ .

La structure corrélée de  $K_{tt}$  pénalise les déviations du posterior diagonal vis-à-vis de la dynamique latente imposée par le prior.

**(d) Décodeur non-autorégressif en tokens.** La section (d) constitue le point conceptuel central : c'est à cet endroit que la dynamique séquentielle latente est définie de manière formelle. En introduisant la factorisation causale

$$p(z_{1:T}) = \prod_{t=1}^T p(z_t | z_{<t}),$$

le modèle établit clairement ce qui caractérise son originalité : une autorégressivité purement latente, indépendante de toute récursion sur les tokens.

Le décodeur reçoit l'ensemble de la trajectoire latente  $z_{1:T}$  en une seule fois, lui ajoute un encodage positionnel, puis prédit en parallèle les logits pour tous les tokens :

```
class TokenDecoder(nn.Module):
    def forward(self, z: torch.Tensor):
        # z: [B, T, Dz], full latent trajectory
        z = z + self.pe(T=z.size(1), device=z.device)      # positional encoding
        h = self.mlp(z); h = self.ln(h)                    # pointwise processing
        h2 = self.post(h.transpose(1,2)).transpose(1,2)     # conv post-process
        h = h + h2                                         # residual connection
        e_proj = self.to_emb(h)                            # [B, T, E], embedding proj
        tw = F.normalize(self.tied_weight, dim=-1)         # tied embedding weights
        logits = torch.matmul(e_proj, tw.t()) + self.bias  # token logits
        return logits
```

Aucune boucle temporelle ni masque causal n'est appliquée sur les tokens : toutes les positions sont traitées simultanément à partir de  $z_{1:T}$ . La dynamique séquentielle du modèle ne provient donc pas du décodeur, mais exclusivement du prior GP sur l'espace latent, dont la factorisation causale définit l'autorégressivité purement latente. Le décodeur se contente ensuite de projeter en parallèle cette trajectoire continue vers les distributions sur les tokens.

L'autorégressivité dans le latent n'apparaît pas dans l'encodeur (a) ni dans la définition du prior corrélé (b) : un GP impose des corrélations  $K_{tt}$ , mais pas encore une causalité. Celle-ci ne se manifeste qu'au moment où l'on factorise explicitement la loi jointe en conditionnelles  $p(z_t | z_{<t})$ .

**(e) Génération non conditionnée (A-SEQ).** La génération non conditionnée exploite directement la factorisation causale du prior GP en échantillonnant la trajectoire latente de manière séquentielle. Le pseudo-code suivant résume la procédure :

```
@torch.no_grad()
def generate(self, T: int, batch_size: int = 1,
            top_k=50, top_p=0.9, temperature=0.9):
    device = self.t_train.device
    t = torch.linspace(0.0, 1.0, T, device=device)
    K = self.K_tt(t)                                # GP covariance over time grid
    Dz, B = self.cfg.d_latent, batch_size
    z = torch.zeros(B, T, Dz, device=device) # latent trajectory container

    # Latent autoregressive loop: each z[:, tp, :] depends on z[:, :tp, :]
    for tp in range(T):
```

```

z[:, tp, :] = self._gp_conditional_step(K, z[:, :tp, :], tp)

logits, _ = self.decoder(z)
return sample_logits_from_timewise_logits(
    logits, top_k=top_k, top_p=top_p, temperature=temperature
)

```

À chaque étape  $t$ , le vecteur latent  $z_t$  est échantillonné à partir de la loi conditionnelle  $p(z_t | z_{<t})$  induite par le GP, en fonction des latents précédemment générés  $z_{<t}$ . C'est cette étape qui implémente concrètement la factorisation causale

$$p(z_{1:T}) = \prod_{t=1}^T p(z_t | z_{<t}),$$

et qui réalise l'autorégressivité purement latente du modèle.

**(f) Génération conditionnée par un prompt ( $\Lambda$ -SEQ).** La génération conditionnée suit le même principe, mais initialise la trajectoire latente à partir d'un encodage variationnel du prompt :

```

@torch.no_grad()
def generate_with_prompt(self, prompt_ids, total_len, eos_id,
                       top_k=50, top_p=0.9, temperature=0.9):
    device = self.t_train.device; self.eval()
    B, T0 = prompt_ids.shape; T = total_len

    # Prepare token container
    x_in = torch.full((B, T), fill_value=eos_id,
                      dtype=torch.long, device=device)
    x_in[:, :T0] = prompt_ids.to(device)

    # Encode prompt into latent prefix z[:, :T0, :]
    mu, logvar = self.encoder(x_in[:, :T0])
    std = torch.exp(0.5 * logvar)
    z_prompt = mu + std * torch.randn_like(std)

    # Build full latent trajectory with GP conditionals
    t = torch.linspace(0.0, 1.0, T, device=device)
    K = self.K_tt(t)
    Dz = self.cfg.d_latent
    z = torch.zeros(B, T, Dz, device=device)
    if T0 > 0:
        z[:, :T0, :] = z_prompt

    # Latent autoregressive continuation from the prompt
    for tp in range(T0, T):
        z[:, tp, :] = self._gp_conditional_step(K, z[:, :tp, :], tp)

    logits, _ = self.decoder(z)
    new_ids = sample_logits_from_timewise_logits(
        logits[:, T0:, :], top_k=top_k, top_p=top_p, temperature=temperature
    )

    x_out = x_in.clone()
    if T > T0:
        x_out[:, T0:] = new_ids
    return x_out, logits

```

Comme dans le cas non conditionné, la boucle séquentielle dans l'espace latent réalise une continuation autorégressive de la trajectoire  $z_{1:T}$  à partir du préfixe  $z_{1:T_0}$  issu du prompt. La variante  $\Lambda$ -PARA, non détaillée ici, remplace cette boucle par un échantillonnage bloc via une décomposition de Cholesky de  $K_{tt}$ , tout en respectant la même loi jointe  $p(z_{1:T})$  imposée par le GP.

#### 4.2.4 Objectif : ELBO par token

L'objectif d'entraînement optimise une ELBO moyenne par token, enrichie de termes de régularisation.

**ELBO “pure” (théorique).**

$$\mathcal{L}_{\text{pur}} = \mathbb{E}_{q(z|x)}[\log p(x | z)] - \text{KL}(q(z | x) \| p(z)).$$

Dans l'implémentation :

$$\text{elbo\_pur\_tok\_t} = (\text{ll\_0} + \text{ll\_multi}) - \text{kl\_tok\_raw\_t}.$$

**Objectif effectivement optimisé.**

$$\mathcal{L}_{\text{train}} = (\text{ll\_0} + \text{ll\_multi}) - \beta \text{KL}_{\text{cap}} - \lambda_{\text{emb}} \text{Reg}_{\text{emb}}.$$

Le coefficient  $\beta$  suit une phase de préchauffage (warm-up) puis une adaptation autour d'une cible de KL/token. La quantité  $\text{KL}_{\text{cap}}$  est la version capée du KL/token. Les hyperparamètres critiques ( $kl\_cap$ ,  $\beta_{\text{max}}$ , schéma d'adaptation) sont variés systématiquement dans les ablations (section 4.6.2).

### 4.3 Variantes $\Lambda$ -SEQ et $\Lambda$ -PARA

Deux stratégies d'échantillonnage sont examinées :

- $\Lambda$ -SEQ : échantillonnage séquentiel via conditionnement gaussien ;
- $\Lambda$ -PARA : échantillonnage parallèle via décomposition de Cholesky.

Les deux variantes utilisent exactement la même architecture (encodeur TCN et décodeur non-autorégressif), de façon à isoler uniquement la dynamique latente. Les attentes théoriques (égalité des lois jointes) sont évaluées empiriquement dans la comparaison quantitative de la section 6.5.

### 4.4 Résultats globaux ( $\Lambda$ -SEQ vs $\Lambda$ -PARA)

Sur le corpus WikiText-2, les observations sont les suivantes :

- absence de divergence numérique ;
- KL/token convergente et stable ;
- métriques très proches entre les deux variantes.

La KL/token atteint régulièrement un plateau (environ 12 nats). Contrairement à la version antérieure, la section 6.6.1 montre que cette valeur dépend clairement du  $kl\_cap$  et que la structure du GP joue un rôle démontrable (ablation). Les courbes d'entraînement ne sont pas strictement superposées : de faibles écarts persistent, mais ils restent dans la plage attendue pour deux schémas d'échantillonnage présentant des approximations numériques distinctes.

### 4.5. Comparaison quantitative Lambda-SEQ vs Lambda-PARA

La comparaison suivante est conduite à hyperparamètres identiques.

Les deux variantes utilisent le même  $kl\_cap$  ( $\text{KL/tok} \approx 12$ ) et un  $\beta_{\text{final}} \approx 0.126$  (non repris dans le tableau pour limiter sa largeur).

| Modèle | Type   | ELBO/tok | NLL(cont) | PPL(cont) | tok/s |
|--------|--------|----------|-----------|-----------|-------|
| Λ-SEQ  | GP-VAE | -9.935   | 0.562     | 1.75      | 9097  |
| Λ-PARA | GP-VAE | -9.967   | 0.475     | 1.61      | 9037  |

TABLE 1 – Résultats Λ-SEQ vs Λ-PARA sur WikiText-2.

**Interprétation.** Les résultats sont similaires mais non identiques, conformément aux différences d’approximation. Λ-PARA montre une légère amélioration en continuation, mais cet écart doit être interprété avec prudence (pas d’analyse multiseed). À  $T = 64$ , le gain en tok/s est faible ; l’intérêt de Λ-PARA à grande longueur reste à vérifier.

#### 4.6. Analyse élargie : variantes, ablations et stabilité

##### 4.6.1. Performances GP-VAE-TCN vs Transformer (famille Lambda-SEQ-X)

Nous évaluons plusieurs variantes Λ-SEQ-X, avec la même architecture de base mais des régularisations différentes, pour  $X \in \{I, B, C, J, K\}$ .

| Modèle      | PPL(val) | NLL(cont) | PPL(cont) | tok/s   | Qualité    |
|-------------|----------|-----------|-----------|---------|------------|
| Λ-SEQ-I     | 3.35     | 0.4773    | 1.61      | ~ 9000  | Excellent  |
| Λ-SEQ-B     | 3.27     | 0.5455    | 1.73      | ~ 8900  | Très bonne |
| Λ-SEQ-C     | 3.34     | 0.5077    | 1.66      | ~ 9000  | Très bonne |
| Λ-SEQ-J     | 3.03     | 0.5635    | 1.76      | ~ 9000  | Bonne      |
| Λ-SEQ-K     | 3.05     | 0.6005    | 1.82      | ~ 9000  | Bonne      |
| Transformer | 326.94   | 5.7898    | 326.94    | ~ 15700 | Faible     |

TABLE 2 – Variantes Λ-SEQ-X et baseline Transformer sur WikiText-2.

Les commentaires détaillés (compromis optimal, légère sur-régularisation, etc.) sont discutés dans le texte et non dans le tableau afin d’en limiter la largeur.

**Analyse.** Les variantes Λ-SEQ-X obtiennent des PPL(cont) entre 1.61 et 1.82, valeurs cohérentes avec un modèle compact correctement régularisé. Le Transformer baseline est très faible : il sert seulement de point d’ancrage, sans représenter des architectures autorégressives mieux réglées.

##### 4.6.2. Ablations critiques

**Prior isotrope diagonal.** Nous remplaçons le prior corrélé du GP par un prior isotrope  $K = \sigma^2 I$ . Cette ablation supprime toute dépendance temporelle dans le latent. Les performances de continuation s’effondrent, la cohérence séquentiel/parallèle disparaît et la KL/token chute, indiquant que le latent n’est plus utilisé. Supprimer la corrélation GP réduit la KL/token à environ 3 nats et dégrade :

- la PPL(cont) (+0.15 à +0.30) ;
- le calibrage ;
- la cohérence discursive.

Cela confirme que la structure GP est effectivement exploitée.

**Variation du *kl\_cap*.** Nous modifions le *kl\_cap* afin d’évaluer la sensibilité du modèle à la régularisation KL. Un *kl\_cap* trop faible entraîne un collapse latent et une chute drastique des performances de continuation, tandis qu’un *kl\_cap* trop élevé provoque une sur-activation instable du GP. Entre ces extrêmes, une zone optimale (environ 8–12) garantit l’activation effective du latent corrélé. Cette ablation confirme que la causalité latente du modèle Λ dépend directement d’un niveau adéquat de régulation KL.

Résumé typique :

- $kl\_cap = 8$  : perte de continuité latente, PPL(cont) plus élevée, parfois collapse ;
- $kl\_cap \geq 20$  : bonne stabilité mais coût computationnel accru ;
- $kl\_cap \approx 12$  : compromis robuste.

#### 4.6.3. Série Lambda-2-SEQ-X : stabilité et effondrements

La famille  $\Lambda$ -2-SEQ-X explore plus finement la sensibilité à la régularisation.

| Modèle             | NLL(cont) | PPL(cont) | PPL(val) | ELBO/tok | tok/s       | Qualité       |
|--------------------|-----------|-----------|----------|----------|-------------|---------------|
| $\Lambda$ -2-SEQ-T | 0.2883    | 1.33      | 3.18     | -9.058   | $\sim 8600$ | Excellente    |
| $\Lambda$ -2-SEQ-S | 0.3097    | 1.36      | 3.05     | -8.927   | $\sim 8500$ | Très bonne    |
| $\Lambda$ -2-SEQ-G | 0.4606    | 1.59      | 3.19     | -8.501   | $\sim 8500$ | Bonne         |
| $\Lambda$ -2-SEQ-H | 0.4790    | 1.61      | 3.29     | -9.958   | $\sim 8600$ | Bonne         |
| Transformer        | 5.7898    | 326.94    | 326.94   | -6.105   | 15700       | Très mauvaise |

TABLE 3 – Exemples de modèles  $\Lambda$ -2-SEQ-X vs Transformer.

**Interprétation.** Les modèles bien régularisés (T, S, G, H) obtiennent des PPL(cont) entre 1.33 et 1.61. Les variantes insuffisamment régularisées (par exemple A, B, F, R) s’effondrent complètement, ce qui confirme la sensibilité de la dynamique latente au  $kl\_cap$ .

#### 4.7. Baseline Transformer

| Modèle      | PPL(val) | NLL(cont) | PPL(cont) | tok/s        |
|-------------|----------|-----------|-----------|--------------|
| Transformer | 326.94   | 5.7898    | 326.94    | $\sim 15700$ |

TABLE 4 – Baseline Transformer minimaliste.

Ce modèle sert uniquement d’ancrage numérique ; il n’a pas été entraîné avec un réglage fin des hyperparamètres.

#### 4.8. Synthèse

- Les variantes  $\Lambda$ -SEQ et  $\Lambda$ -PARA sont entraînables et stables.
- Les ablations montrent que la structure GP est réellement utilisée.
- $\Lambda$ -PARA est cohérent avec  $\Lambda$ -SEQ, ce qui valide partiellement la parallélisation du sampling latent.
- Les GP-VAE surpassent la baseline simple, sans permettre de conclure face à des architectures autorégressives plus avancées.
- La régularisation KL est critique : une régulation trop faible conduit à l’effondrement.

#### 4.9. Limites et perspectives

##### Limites majeures.

- coût quadratique du GP ;
- absence d’évaluation multiseed ;
- comparaison limitée à une baseline minimaliste ;
- encodeur TCN relativement peu expressif ;
- absence d’évaluations qualitatives systématiques.

##### Perspectives.

- approximations GP pour des longueurs  $T$  plus grandes ;
- décodeurs plus complexes tout en conservant la causalité latente ;

- tâches conditionnelles plus variées ;
- extension aux signaux continus et à des séries temporelles.

#### 4.5 4.10. Code et reproductibilité

L’ensemble du code utilisé pour les expériences présentées dans ce travail (estimation des hyperparamètres du GP, implémentation de l’autorégressivité latente, variantes  $\Lambda$ -SEQ et  $\Lambda$ -PARA, scripts d’entraînement et de génération, ainsi que les configurations associées) est disponible en accès libre à l’adresse suivante :

<https://github.com/y-v-e-s/GP-VAE-Latent-AR>

Ce dépôt contient :

- l’implémentation complète du modèle GP-VAE à autorégressivité latente ;
- les scripts d’entraînement, d’évaluation et de génération ;
- les fichiers de configuration permettant de reproduire les séries  $\Lambda$ -SEQ et  $\Lambda$ -PARA ;
- les versions exactes des hyperparamètres utilisés dans les tableaux de la section 4 (notamment les valeurs de `k1_cap`) ;
- un guide minimal de reproductibilité.

Cette diffusion vise à faciliter la réutilisation du schéma proposé, la reproduction des résultats et l’exploration de variantes de kernels, d’architectures d’encodeur ou de stratégies d’échantillonnage latent.

## 5. Discussion

La présente étude avait pour objectif d’examiner, dans un cadre contrôlé et de petite échelle, la faisabilité et le comportement d’un schéma d’autorégressivité localisé entièrement dans l’espace latent. Le modèle  $\Lambda$  et ses deux variantes,  $\Lambda$ -SEQ et  $\Lambda$ -PARA, offrent un terrain d’observation permettant d’évaluer la stabilité d’entraînement, l’exploitation effective du latent corrélé, ainsi que la cohérence entre deux stratégies d’échantillonnage distinctes.

### 5.1. Exploitation de la structure latente corrélée

Les expériences montrent qu’un GP-VAE doté d’un prior corrélé peut être entraîné de manière stable sur un corpus standard et avec une régularisation modérée. La KL/token atteint de manière fiable le cap cible, ce qui indique :

- le latent n’est pas sous-utilisé ;
- la covariance du GP influence effectivement les représentations apprises.

Il ne s’agit pas, à ce stade, d’établir une preuve de causalité latente au sens fort, mais de constater qu’un schéma latent corrélé conduit à une trajectoire interne riche et non dégénérée.

### 5.2. Cohérence entre génération séquentielle et génération parallèle

Les variantes  $\Lambda$ -SEQ (échantillonnage pas-à-pas) et  $\Lambda$ -PARA (échantillonnage joint vectorisé) produisent des métriques très proches. Cela est cohérent avec les propriétés d’une loi gaussienne multivariée : les deux méthodes accèdent, sous des formes procédurales différentes, à la même distribution latente.

Ce résultat confirme un point méthodologique important : dans ce cadre, l’ordre d’échantillonnage n’est pas déterminant pour la qualité prédictive. La structure temporelle interne provient du prior, pas du schéma algorithmique utilisé pour le simuler.

### 5.3. Dynamique latente vs séquentialité symbolique

Un aspect intéressant de ce proof of concept réside dans la distinction entre :

- séquentialité procédurale (boucle temporelle d'un échantillonnage pas-à-pas) ;
- structure probabiliste (factorisation analytique issue du GP).

Les deux variantes du modèle montrent qu'il est possible :

- de représenter une structure temporelle dans l'espace latent ;
- tout en gardant une projection symbolique entièrement parallèle.

Le décodeur intervient en une seule passe sur la trajectoire latente complète, ce qui distingue ce modèle des approches autorégressives classiques où la génération s'effectue token par token.

Il ne s'agit pas d'une supériorité démontrée, mais d'une différence structurelle pertinente pour l'étude de modèles séquentiels hybrides.

### 5.4. Perspectives d'échelle

Ce travail se limite volontairement à un cadre réduit (séquences courtes, modèle compact, corpus de taille moyenne). Néanmoins, plusieurs directions apparaissent naturelles :

- évaluer la stabilité pour des séquences plus longues et des kernels plus complexes ;
- explorer des décodeurs plus expressifs tout en conservant le parallélisme ;
- étendre le schéma à d'autres modalités (séries temporelles, signaux continus, etc.).

Ces perspectives devront être examinées en tenant compte des contraintes de scalabilité inhérentes aux processus gaussiens.

### 5.5. Limites et points d'attention

Les principaux points critiques sont :

- le coût quadratique du GP, limitant l'échelle ;
- la sensibilité des hyperparamètres du kernel ;
- l'absence de génération incrémentale dans la variante parallèle ;
- l'absence de multiseed dans ce proof of concept.

Ces limites définissent les conditions nécessaires pour confirmer ou infirmer les tendances observées.

### 5.6. Synthèse de la discussion

Les résultats obtenus montrent qu'un schéma d'autorégressivité latente, fondé sur une covariance analytique et couplé à un décodeur parallèle, est entraînable, stable et cohérent dans ses deux variantes.

Sans revendiquer de supériorité sur les architectures autorégressives établies, ce proof of concept suggère que la structuration temporelle peut, dans certains cadres, être déplacée vers l'espace latent, laissant au décodeur le soin de projeter cette dynamique en une séquence symbolique.

Ce déplacement ouvre un espace d'exploration méthodologique : celui de modèles séquentiels où la dynamique temporelle est définie analytiquement dans le latent, plutôt que construite par un empilement d'opérations récurrentes ou attentionnelles.

Un dernier point concerne la relation entre notre schéma d'autorégressivité latente et les tentatives antérieures visant à introduire une direction temporelle au sein de processus gaussiens. Certains travaux ont en effet proposé des kernels asymétriques — notamment des kernels dits « causaux » ou des kernels de type Wiener — afin d'imposer une orientation dans la structure de covariance. Si ces kernels introduisent bien une forme d'asymétrie, ils demeurent fondamentalement métriques : ils modulent la corrélation en fonction de la position relative des points, mais ne définissent pas une dynamique séquentielle explicite.

Notre approche diffère en nature. Plutôt que d’inférer la causalité à partir d’un biais imposé au kernel, nous l’instillons structurellement via la factorisation latente

$$p(z_{1:T}) = \prod_{t=1}^T p(z_t | z_{<t}),$$

qui encode une dépendance directionnelle explicite entre les états latents. Cette factorisation lève les ambiguïtés inhérentes aux kernels asymétriques — par exemple lorsque plusieurs points sont à distance comparable dans l’espace induit par le kernel — et fournit un véritable mécanisme d’arbitrage séquentiel dans le latent.

Il convient enfin de distinguer cette factorisation causale explicite des approches dites « Bayesian GP » qui modélisent l’incertitude sur les hyperparamètres du kernel. Si cette extension hiérarchique enrichit la distribution de covariance, elle ne crée pas pour autant de direction temporelle : l’incertitude porte sur la forme du kernel, non sur une relation séquentielle entre états. L’autorégressivité latente telle que définie ici ne découle donc pas du caractère bayésien du GP, mais bien de la factorisation explicite du prior en conditionnelles. Les deux mécanismes peuvent interagir, mais restent conceptuellement distincts.

En combinant ainsi une géométrie corrélée (fournie par le GP) et une progression causale explicite (fournie par la factorisation latente), le modèle dépasse les limites des kernels asymétriques classiques et établit une dynamique latente plus robuste et mieux adaptée à la modélisation séquentielle.

## 6. Conclusion

### Portée empirique

Ce travail présente un proof of concept volontairement restreint :

- un modèle  $\Lambda$  de petite taille ;
- des séquences courtes ( $T = 64$ ) ;
- une baseline autorégressive minimale.

Dans ce cadre réduit, nous montrons qu’un GP-VAE doté d’un prior corrélé et d’un schéma d’autorégressivité purement latente peut être entraîné de manière stable, sans divergence numérique, et produire des métriques cohérentes entre ses deux variantes d’échantillonnage ( $\Lambda$ -SEQ et  $\Lambda$ -PARA).

### Portée conceptuelle

Au-delà du cadre expérimental limité, le modèle  $\Lambda$  peut être interprété comme un pont entre :

- les modèles d’état bayésiens continus fondés sur des processus gaussiens ;
- les modèles de langage neuronaux organisés autour d’un décodeur symbolique.

La dynamique séquentielle est assurée par le prior GP, la mise à jour reste bayésienne, et la projection linguistique est assurée par un décodeur non-autorégressif.

Sans prétendre établir une théorie géométrique du langage, les résultats suggèrent que certaines propriétés associées aux architectures autorégressives classiques (mémoire, directionnalité séquentielle, cohérence globale) peuvent émerger de la covariance d’un espace latent probabiliste.

### Perspectives

Les limites identifiées (coût quadratique du GP, absence de génération incrémentale pour  $\Lambda$ -PARA, sensibilité du noyau (kernel), absence de multiseed) indiquent les conditions nécessaires pour valider les tendances observées.

Plusieurs directions émergent naturellement :

1. Élargissement d'échelle : séquences longues, kernels plus riches, modèles plus expressifs.
2. Renforcement des baselines autorégressives pour quantifier précisément l'apport latent.
3. Exploration multimodale (séries temporelles, signaux continus, trajectoires physiques).
4. Génération conditionnelle et *prompting latent* pour des tâches de compléTION ou d'instruction, consistant à encoder le prompt dans le latent pour obtenir un préfixe  $z_{1:T_0}$ , puis à laisser le prior GP générer la continuation  $z_{T_0+1:T}$  de manière strictement autorégressive dans l'espace latent.
5. Extensions bayésiennes hiérarchiques, où le kernel devient lui-même conditionné par le contexte.

L'ensemble de ces pistes suggère qu'un cadre séquentiel fondé sur la géométrie latente — plutôt que sur la récursivité symbolique — constitue une alternative crédible pour étudier des modèles de langage compacts, stables et interprétables.

## Références

- [1] Francesco Paolo Casale, Adrian V. Dalca, Luca Saglietti, Jennifer Listgarten, and Nicolo Fusi. Gaussian process prior variational autoencoders. *Advances in Neural Information Processing Systems*, 2018. arXiv :1810.11738.
- [2] Vincent Fortuin, Dmitry Baranchuk, Gunnar Rätsch, and Stephan Mandt. Gp-vae : Deep probabilistic time series imputation. In *Proceedings of AISTATS*, volume 108, pages 1651–1661, 2020. arXiv :1907.04155.
- [3] Marco Fraccaro, Søren Kaae Sønderby, Ulrich Paquet, and Ole Winther. Sequential neural models with stochastic layers. In *Advances in Neural Information Processing Systems*, 2016. arXiv :1605.07571.
- [4] Jacob R. Gardner, Geoff Pleiss, David Bindel, Kilian Q. Weinberger, and Andrew G. Wilson. Gpytorch : Blackbox matrix-matrix gaussian process inference with gpu acceleration. In *Advances in Neural Information Processing Systems*, 2018. arXiv :1809.11165.
- [5] Peter Guttorp and Tilmann Gneiting. Studies in the history of probability and statistics xlix : On the matérn correlation family. *Biometrika*, 93(4) :989–995, 2006.
- [6] Irina Higgins, Loïc Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. Beta-vae : Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017.
- [7] Marinka Jazbec, Mark Ashman, Vincent Fortuin, Michael Pearce, Stephan Mandt, and Gunnar Rätsch. Scalable gaussian process variational autoencoders. In *AISTATS*, volume 130, pages 3088–3096, 2021. arXiv :2010.13472.
- [8] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1) :35–45, 1960.
- [9] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters. *arXiv preprint*, arXiv :1605.06432, 2017.
- [10] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, arXiv :1312.6114, 2014.

- [11] Diederik P. Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improving variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, 2016. arXiv :1606.04934.
- [12] Alexander Klushyn, Richard Kurle, Maximilian Soelch, Botond Cseke, and Patrick van der Smagt. Latent matters : Learning deep state-space models. In *Advances in Neural Information Processing Systems*, 2021.
- [13] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint*, arXiv :1609.07843, 2016.
- [14] Tim Pearce, Felix Leibfried, and Alexandra Brinstrup. The gaussian process prior vae for interpretable latent dynamics. In *ICML*, volume 118, pages 7465–7475, 2020.
- [15] Tim Pearce, Michael Smith, Stefan Zohren, and Alexandra Brinstrup. Bayesian autoencoders with gaussian process priors. *Proceedings of the 37th International Conference on Machine Learning*, 118 :2321–2330, 2020. arXiv :1906.02511.
- [16] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.
- [17] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [18] Danilo Jimenez Rezende, Shakir Mohamed, Ivo Danihelka, Karol Gregor, and Daan Wierstra. One-shot generalization in deep generative models. In *ICML*, pages 1521–1529, 2016. arXiv :1603.05106.
- [19] Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, 2016. arXiv :1606.02235.
- [20] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, pages 567–574, 2009.
- [21] Rui Wang, Fan Shun, Hanyuan Liu, Dong Zhao, Shiming Li, Andrew Y. Ng, and Yang Gao. Learning to learn dense gaussian processes for few-shot learning. In *Advances in Neural Information Processing Systems*, 2021. arXiv :2106.01506.
- [22] Andrew G. Wilson, Zhiting Hu, Ruslan Salakhutdinov, and Eric P. Xing. Deep kernel learning. In *AISTATS*, pages 370–378, 2016. arXiv :1511.02222.
- [23] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *ICML*, pages 1067–1075, 2013. arXiv :1302.4245.
- [24] Li Zhou, Michael Poli, Weijia Xu, Stefano Massaroli, and Stefano Ermon. Deep latent state space models for time-series generation. In *ICML*, 2023. arXiv :2212.12749.