Latent Autoregression via Gaussian-Process Priors in Variational Autoencoders

Yves Ruffenach
Conservatoire National des Arts et Métiers
yves.ruffenach.auditeur@lecnam.net
yves@ruffenach.eu
ORCID: 0009-0009-4737-0555

Abstract

We study a class of sequential latent-variable models in which temporal dependence is carried entirely by a Gaussian-process prior on a continuous latent trajectory. Given a finite index set $\{1, \dots, L\}$ and a Gaussian process on $[0, 1]$ with covariance kernel k, we consider the induced joint Gaussian law of the latent variables $(z_1, \dots, z_L)$ and use its canonical factorization

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

to define a purely latent autoregressive process. In this construction, causality is a probabilistic property of the latent path, obtained by Gaussian conditioning, rather than the result of an explicit recurrence in the observation space. We give an explicit form for the Gaussian conditionals $p(z_t \mid z_{<t})$ and discuss regularity properties of the induced latent process, in particular its covariance structure and its relation to Markov and AR(p) processes.

We then couple this latent process with a non-autoregressive observation model and derive a variational formulation in the spirit of variational autoencoders: the correlated Gaussian-process prior $p(z_{1:L})$ plays the role of a structured prior on paths, the approximate posterior $q_\phi(z_{1:L} \mid x)$ is amortized by an encoder, and the objective function is a regularized evidence lower bound (ELBO) in which the Kullback-Leibler term measures the deviation from the latent autoregressive prior. We analyse the resulting probabilistic model from the viewpoint of stochastic processes and approximate Bayesian inference, emphasizing the interplay between the Gaussian–process geometry on latent paths and the variational approximation.

As an illustration, we implement this framework with finite-dimensional Gaussian-process priors and a non-autoregressive decoder, and we report numerical results on a standard sequence dataset. These experiments are not the main contribution of the paper: they serve to show that the latent autoregressive scheme induced by the Gaussian-process prior can be trained stably and exploited in practice in a constrained proof-of-concept regime.

Keywords. Gaussian processes; latent autoregression; variational autoencoders; stochastic processes; sequential generative models; Bayesian inference.

1 Introduction

A central theme in modern probability and statistics is the construction of flexible stochastic processes on high-dimensional spaces that remain amenable to inference. In sequential settings, this often takes the form of latent-variable models in which an unobserved process $(Z_t)_{t=1}^{L}$ carries the temporal dependence, while observations $(X_t)_{t=1}^{L}$ are obtained through a (possibly nonlinear) emission mechanism. Classical examples include Markov chains, autoregressive (AR) processes, and state-space models.

In this work we study a different type of latent dynamics, in which temporal structure is imposed by a Gaussian process (GP) prior on a *continuous* latent trajectory. More precisely, we consider a finite collection of latent variables $(z_1, \dots, z_L)$ in $\mathbb{R}^{d_z}$ and assume that they arise as evaluations of an underlying Gaussian process at ordered time points $0 < t_1 < \cdots < t_L \leq 1$. The induced joint law is multivariate Gaussian, so that

$$(z_1, \dots, z_L) \sim \mathcal{N}(0, K),$$

for a covariance matrix K determined by the kernel. From a probabilistic point of view, this construction is classical.

Our contribution is to make explicit and exploit the *causal factorization* of this joint distribution. By writing

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

we obtain a purely latent autoregressive process in which causality is induced by Gaussian conditioning. Unlike Markov models, the conditional law of z_t depends on the entire history $z_{<t}$, and unlike classical autoregressive models on observations, the temporal structure is entirely confined to the latent space. This leads to a family of GP-based latent autoregressive processes governed by the covariance structure of the Gaussian process.

We then couple this latent process with a non-autoregressive observation model, in the spirit of variational autoencoders (VAEs). The Gaussian-process prior plays the role of a correlated prior on latent paths, while an amortized encoder yields an approximate posterior. This gives rise to a variational objective in which the Kullback-Leibler term measures the deviation from the latent autoregressive Gaussian prior. The resulting model can be viewed as a probabilistic alternative to autoregressive sequence models, where temporal dependence is expressed analytically at the latent level and the decoder is fully parallel.

The main contributions of this paper are as follows:

- we formalize a class of latent autoregressive processes obtained by causal factorization of a Gaussian-process prior on a finite grid;
- we derive explicit Gaussian conditional distributions $p(z_t \mid z_{<t})$ and discuss basic properties (existence, covariance structure, relation to Markov and AR(p) processes);
- we embed this latent process in a variational framework and obtain an ELBO-type objective in which the KL term is computed against the correlated prior;
- we illustrate, on a standard sequence dataset, that this construction can be implemented in practice and trained stably within a constrained proof-of-concept setup.

2 Related Work

2.1 Variational Inference and Latent Variable Models

Variational methods have become a standard tool for approximate Bayesian inference in latent-variable models. In its simplest form, a variational approximation replaces an intractable posterior distribution $p_\theta(z \mid x)$ by a tractable distribution $q_\phi(z \mid x)$ minimizing the Kullback-Leibler divergence. The resulting objective, often referred to as the evidence lower bound (ELBO), reads

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

and provides a tractable surrogate for the marginal likelihood $\log p_\theta(x)$.

Within this general framework, variational autoencoders (VAEs) [9] constitute a class of latent-variable models in which the approximate posterior q_ϕ and the likelihood model p_θ are

parameterized by neural networks. Many structural variants exist: scaled Kullback-Leibler regularization (β-VAE [6]), hierarchical priors [17], and sequential latent structures [3, 8]. From a probabilistic perspective, these models can be viewed as flexible non-linear state-space models, in which variational inference plays the role of an amortized filtering or smoothing procedure.

A key theme in this literature is the construction of latent spaces that retain meaningful global structure beyond what is directly observable. This aspect is central to few-shot and one-shot generalization [16], where latent regularity plays the same role as prior smoothness in classical Bayesian non-parametrics.

2.2 Gaussian Processes and Correlated Latent Priors

Gaussian processes (GPs) [15] provide a canonical class of stochastic processes indexed by continuous time or space, and form the basis for a long tradition in Bayesian non-parametrics. Given a mean function m and a covariance kernel k, a GP satisfies

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

and any finite collection $(f(t_1), \dots, f(t_L))$ is jointly Gaussian with covariance matrix $K = (k(t_i, t_j))_{i,j}$.

When GPs are used as priors for latent trajectories, their covariance structure can regularize or constrain the evolution of the latent variables. This idea has been studied under various forms in the machine learning literature. In GP-VAEs [1, 2, 13], the latent variables of a VAE are endowed with a correlated Gaussian prior $z \sim \mathcal{N}(0, K)$, which induces a form of temporal or spatial continuity. Such models produce smooth latent paths rather than independent latent points, but the induced correlations are symmetric and do not encode causal direction. Consequently, these approaches yield latent continuity but not latent autoregression in the sense of a factorization $p(z_{1:L}) = \prod_{t=1}^{L} p(z_t \mid z_{<t})$.

2.3 Latent Dependence, Conditioning, and One-Shot Behavior

Several works have emphasized that latent correlations can substantially improve generalization when data are scarce. Rezende et al. [16] demonstrated that correlated latent priors combined with variational inference can produce coherent samples from single examples. More recent work, such as dense GP layers [19], shows that embedding GP-like structures into deep networks enhances contextual coherence.

The ability of Gaussian processes to provide analytically tractable conditional distributions plays a central role in these constructions. For a GP evaluated at $\{t_1, \dots, t_L\}$, the conditional distribution of z_t given $z_{<t}$ is Gaussian:

$$p(z_t \mid z_{<t}) = \mathcal{N}\Big(k_{12}^\top K_{11}^{-1} z_{<t}, K_{22} - k_{12}^\top K_{11}^{-1} k_{12}\Big),$$

where K_{11} is the covariance of $z_{<t}$ and k_{12} their cross-covariance with z_t. This conditional representation is a classical result of multivariate Gaussian theory, and forms the probabilistic foundation of the latent autoregressive model developed in the present work.

2.4 Sequential Latent Models and the Lack of Full Causality

Sequential latent-variable models are well documented. Stochastic recurrent VAEs [3], deep Kalman-like models [8], and continuous-time latent SDE/SSM models [21, 11] provide increasingly expressive families of latent processes, typically of the form

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

or more elaborate Markovian structures. However, these models rely on parametric state transitions whose temporal dependence is learned through neural weights. They do not exploit the analytic conditional structure of Gaussian processes, nor do they provide a full causal factorization with dependence on the entire latent history.

Inverse autoregressive flows [10] manipulate latent densities using autoregressive maps, but the induced dependence is parametric and does not correspond to a causal stochastic dynamic. Similarly, existing GP-based approaches impose correlations but not direction.

To our knowledge, no prior work combines:

- a correlated Gaussian-process prior on a latent trajectory,
- a fully causal factorization $p(z_{1:L}) = \prod_{t=1}^{L} p(z_t \mid z_{<t})$ obtained analytically from GP conditioning,
- and a non-autoregressive likelihood model.

This combination produces a latent autoregressive process in which causal structure arises from probabilistic geometry rather than learned transition weights.

2.5 Computational Considerations

Inference with Gaussian processes is limited by the cost of inverting covariance matrices, nominally $O(L^3)$. Several scalable numerical methods have been proposed, such as inducing-point approximations [18] and matrix-free linear solvers implemented via structured or stochastic approximations. In practice, we make use of the BBMM method of [4], which permits quasi-quadratic GP inference in $O(L^2)$ operations and enables joint optimization of kernel hyperparameters with variational objectives.

The kernel choice (RBF, Matérn [5], or spectral mixture [20]) determines the regularity of the latent process, in the classical sense of sample-path smoothness, and influences the conditional means and variances appearing in the latent autoregressive factorization.

2.6 Summary

The present work lies at the intersection of three bodies of literature:

- variational inference and latent-variable models,
- Gaussian-process priors on latent trajectories,
- sequential latent models and nonparametric stochastic processes.

While prior approaches have used Gaussian processes to induce latent correlations, they have not exploited the analytic causal factorization available for finite GP evaluations. The construction developed here extends this direction by placing temporal causality in the latent space, yielding a latent autoregressive process governed entirely by the covariance structure of the Gaussian process. We view this as a probabilistic alternative to learned state transitions, and as a conceptual step toward generative models in which sequential structure is expressed analytically rather than parametrically.

3 Mathematical Preliminaries

We briefly recall the basic Gaussian-process (GP) and matrix-analytic facts used throughout the paper. The goal is to make the probabilistic assumptions and the linear-algebraic structure fully explicit.

3.1 Gaussian Processes on a Finite Grid

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space and consider a centred Gaussian process

$$f \sim \mathcal{GP}(0, k)$$

indexed by $[0,1]$, where $k : [0,1] \times [0,1] \to \mathbb{R}$ is a positive-definite kernel.

(H1) **Kernel regularity.** The kernel k is continuous on $[0,1]^2$ and positive definite in the sense that, for any finite family of pairwise distinct points $0 < t_1 < \cdots < t_L \leq 1$, the Gram matrix

$$K_{tt} \;=\; \big(k(t_i, t_j)\big)_{1 \leq i,j \leq L}$$

is symmetric positive definite.

(H2) **Latent dimensionality.** The latent dimension $d_z \in \mathbb{N}$ is fixed, while the sequence length L may vary. We write

$$z_t \in \mathbb{R}^{d_z}, \qquad z_{1:L} = (z_1, \ldots, z_L) \in \mathbb{R}^{L d_z}.$$

Under (H1), for any choice of $0 < t_1 < \cdots < t_L \leq 1$ we can define the finite-dimensional Gaussian vector

$$z_{1:L} = \big(f(t_1), \ldots, f(t_L)\big),$$

which is distributed according to

$$z_{1:L} \sim \mathcal{N}(0, K_{tt}).$$

In the latent-variable model considered in this work, we use a d_z-dimensional GP, implemented as d_z i.i.d. copies of f. This yields the block-structured covariance

$$\mathrm{Cov}(z_{1:L}) = K_{tt} \otimes I_{d_z},$$

where I_{d_z} denotes the $d_z \times d_z$ identity matrix and $\otimes$ is the Kronecker product. The Kronecker structure will be used explicitly in the computation of KL divergences and log-densities.

3.2 Gaussian Conditioning

We recall the classical conditioning formulas for multivariate Gaussians in the notation used later for the causal factorization.

Let (Y_1, Y_2) be a centred Gaussian vector with

$$\begin{pmatrix} Y_1 \\ Y_2 \end{pmatrix} \sim \mathcal{N}\left(0, \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}\right),$$

where Σ_{11} is invertible. Then the conditional distribution of Y_2 given $Y_1 = y_1$ is Gaussian with

$$\mathbb{E}[Y_2 \mid Y_1 = y_1] = \Sigma_{21}\Sigma_{11}^{-1} y_1, \tag{1}$$

$$\mathrm{Cov}(Y_2 \mid Y_1 = y_1) = \Sigma_{22} - \Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}. \tag{2}$$

Applied to the GP evaluation vector $z_{1:L}$, this yields the sequential conditioning formulas used to construct the latent autoregressive factorization. More precisely, for each $t \in \{2, \ldots, L\}$ we partition

$$z_{1:L} = (z_{<t}, z_t), \qquad z_{<t} = (z_1, \ldots, z_{t-1}),$$

and the covariance matrix accordingly as

$$\Sigma_{11} = K_{(<t,<t)} \otimes I_{d_z}, \quad \Sigma_{12} = k_{(<t,t)} \otimes I_{d_z}, \quad \Sigma_{22} = k_{(t,t)} \otimes I_{d_z},$$

where $K_{(<t,<t)}$ is the Gram matrix restricted to indices $\{1,\dots,t-1\}$ and $k_{(<t,t)}$ is the column vector of cross-covariances between t and $\{1,\dots,t-1\}$.

Using (1)–(2) we obtain conditional means μ_t and covariances Σ_t as

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

with explicit formulas given in Section 4.3.

3.3 Kronecker Products and Tensor Structures

We recall the basic identities used throughout the paper for Kronecker products. For matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{n\times n}$,

$$A \otimes B \in \mathbb{R}^{mn\times mn}$$

is defined blockwise by

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1m}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mm}B \end{bmatrix}.$$

In our setting:

- $K_{tt} \in \mathbb{R}^{L\times L}$ encodes temporal correlations;
- $I_{d_z} \in \mathbb{R}^{d_z\times d_z}$ acts on the latent coordinates;
- the joint covariance of $z_{1:L} \in \mathbb{R}^{Ld_z}$ is $K_{tt} \otimes I_{d_z}$.

We use the standard properties:

$$\det(K_{tt} \otimes I_{d_z}) = \det(K_{tt})^{d_z},$$
$$(K_{tt} \otimes I_{d_z})^{-1} = K_{tt}^{-1} \otimes I_{d_z},$$

whenever K_{tt} is invertible. These identities are used implicitly in the computation of Gaussian log-densities and KL divergences between diagonal posteriors and the GP prior.

Standing assumptions. Unless otherwise stated, all probabilistic statements are made under hypotheses (H1)–(H3) of Sections 3 and 4.1. In particular:

- the kernel k is continuous and positive definite on $[0,1]^2$;
- the latent dimension d_z is fixed while the sequence length L may vary (within the regime $L \le T_{\max}$ used in practice);
- all Gram matrices K_{tt} arising from such sequences are numerically well conditioned, with a uniform lower bound on their smallest eigenvalue, enforced by a jitter term εI.

Assumption (H4) is purely numerical and is only invoked when discussing the BBMM complexity estimates in Section 4.1. Under these conditions, the causal Gaussian factorization in Proposition 4.1 is well defined and the complexity considerations of Section 4.1 apply.

4 Probabilistic Framework and Methodology

4.1 Complexity of Gaussian-Process Latent Dynamics

We briefly summarize the computational complexity associated with the latent Gaussian-process prior and state the numerical assumptions under which the proposed scheme is implemented.

Exact complexity. For a sequence of length T and a kernel k evaluated on $0 < t_1 < \cdots < t_T \leq 1$, the temporal covariance is $K_{tt} \in \mathbb{R}^{T \times T}$. Exact GP computations (log-density and sampling) require a Cholesky factorization

$$K_{tt} = LL^\top,$$

with cost $O(T^3)$ operations and $O(T^2)$ memory. In the latent model considered here, the latent dimension d_z is fixed and the covariance of $z_{1:T}$ is $K_{tt} \otimes I_{d_z}$, so that the dominant cost still comes from the $T \times T$ part; Kronecker structure removes only the dependence on d_z in the factorization.

Approximate complexity. To avoid the cubic dependence on T, we rely on matrix-free GP inference, in particular BBMM-style methods based on conjugate gradients (CG) and stochastic trace estimation [4]. In this regime:

- the cost of a single CG iteration for solving $K_{tt}v = b$ is $O(T^2 d_z)$, assuming dense kernels and no additional structure;
- the total cost is therefore $O\big(T^2 d_z\, n_{\text{iter}}\big)$, where n_{iter} is the number of CG iterations required to reach a prescribed tolerance ε_{CG};
- classical CG theory implies that, for fixed ε_{CG}, one typically has $n_{\text{iter}} = O\big(\sqrt{\kappa(K_{tt})}\log(1/\varepsilon_{\text{CG}})\big)$, where $\kappa(K_{tt})$ denotes the condition number of K_{tt}.

Under the spectral bound (H3) and a fixed tolerance ε_{CG}, the quantity n_{iter} remains bounded over the finite set of covariance matrices considered, so that the effective scaling observed in our experiments is close to $O(T^2 d_z)$.

Numerical assumptions. The next two assumptions are purely numerical: they are only used to justify the complexity claims for BBMM-style GP inference and play no role in the measure-theoretic existence and uniqueness results of Section 3.

(H3) **Uniform spectral bounds on the experimental grid.** There exists $T_{\max} \in \mathbb{N}$ and constants $0 < \underline{\lambda} \leq \overline{\lambda} < \infty$ such that, for all sequence lengths $T \leq T_{\max}$ considered in our experiments and all kernel hyperparameters of interest, the eigenvalues of K_{tt} satisfy

$$\underline{\lambda} \leq \lambda_{\min}(K_{tt}) \leq \lambda_{\max}(K_{tt}) \leq \overline{\lambda}.$$

In practice, we add a small jitter term εI to K_{tt}, which enforces $\lambda_{\min}(K_{tt}) \geq \varepsilon > 0$ on the finite set of covariance matrices actually used.

(H4) **Controlled CG tolerance.** CG iterations in BBMM are stopped when the residual norm falls below a fixed tolerance $\varepsilon_{\text{CG}} > 0$. We assume that this tolerance is chosen such that the resulting approximation of K_{tt}^{-1} is sufficiently accurate for the purposes of computing KL divergences and ELBO estimates; no probabilistic statement in this paper depends on the exact value of ε_{CG}.

Latent vs. observation-level complexity. The key reduction in complexity comes from the fact that temporal dependence is modeled on a latent sequence $z_{1:L}$ with $L \ll N$ and $d_z \ll d$, rather than on the full observation sequence $x_{1:N}$. Under the assumptions above, the GP-related cost scales as $O(L^2 d_z)$, to be compared with $O(N^2 d)$ for a typical self-attention layer on the observation tokens.

In other words, the computational gain is not purely due to numerical approximations of GP inference, but also to a change in representation: temporal dependence is encoded once in the covariance structure of the latent process, rather than recomputed layer by layer in the observation space.

4.2 Latent Architecture and Observation Model

The model architecture is organized around three components: an amortized encoder, a correlated latent process, and an observation model. This separation is typical of variational formulations and is particularly important here, as it isolates the temporal dynamics within the latent space.

Encoder. The encoder maps an observed sequence $x_{1:N}$ to a family of approximate posteriors over latent trajectories. We write

$$q_\phi(z_{1:L} \mid x_{1:N}),$$

and in the present work take q_ϕ to be Gaussian, with means and covariances parameterized by a temporal network (for instance a causal convolutional architecture). The precise parametric form of the encoder is not essential for the probabilistic formulation; its role is to provide an amortized approximation to the intractable posterior $p_\theta(z_{1:L} \mid x_{1:N})$.

Latent process. The core of the model is a Gaussian-process prior on the latent sequence $z_{1:L}$ endowed with a causal factorization. Formally, we assume that $(z_1, \dots, z_L)$ arises from a GP evaluated at ordered time points, and we use the induced Gaussian conditionals to define

$$p_\theta(z_{1:L}) = \prod_{t=1}^{L} p_\theta(z_t \mid z_{<t}), \qquad p_\theta(z_t \mid z_{<t}) = \mathcal{N}(m_t, \Sigma_t),$$

where (m_t, Σ_t) are the predictive mean and covariance obtained from the Gaussian conditioning formulas. The covariance kernel $k_\psi(t, t')$ determines the structure of dependence: for instance, a squared-exponential or Matérn kernel [5] enforces temporal continuity, while spectral kernels [20] can encode approximate periodicities.

Observation model (decoder). Given a latent trajectory $z_{1:L}$, observations are generated by an observation model

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

which is conditionally independent across positions given the entire latent sequence. This structure permits fully parallel generation in the observation space: all x_n are sampled simultaneously once $z_{1:L}$ is known.

The functional form of $p_\theta(x_n \mid z_{1:L})$ can be chosen to match the nature of the data: Gaussian likelihoods for continuous signals, categorical likelihoods for discrete symbols, etc. From the probabilistic standpoint, the decoder is simply a family of conditional distributions indexed by the latent path; its parametrization (via neural networks or otherwise) is orthogonal to the latent autoregressive construction.

This separation highlights an important conceptual distinction between classical autoregressive models and the present latent-variable approach. In a standard autoregressive model one specifies

$$p(x_{1:N}) = \prod_{t=1}^{N} p(x_t \mid x_{<t}),$$

whereas in the GP-based latent framework one works with

$$p(x_{1:N}) = \int p_\theta(x_{1:N} \mid z_{1:L})\, p_\theta(z_{1:L})\, \mathrm{d}z_{1:L},$$

so that temporal dependence is encoded in $p_\theta(z_{1:L})$ rather than directly in the observation conditionals.

4.3 Purely Latent Autoregression

Proposition 4.1 (Causal Gaussian Factorization)**.** *Assume* (H1)-(H2) *hold and let* $0 < t_1 < \cdots < t_L \leq 1$ *be fixed. Consider the centred Gaussian vector*

$$z_{1:L} \sim \mathcal{N}(0, K_{tt} \otimes I_{d_z}).$$

Then:

1. *For each* $t \in \{1, \ldots, L\}$ *there exists a unique Gaussian conditional distribution*

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

where μ_t *is affine in* $z_{<t}$ *and* Σ_t *is a symmetric positive-definite matrix in* $\mathbb{R}^{d_z \times d_z}$.

2. *The joint law admits the unique causal factorization*

$$p(z_{1:L}) = \prod_{t=1}^{L} p(z_t \mid z_{<t}), \tag{3}$$

where, for $t \geq 2$*, the mean and covariance are given by the Gaussian-conditioning formulas*

$$\mu_t(z_{<t}) = \big(k_{(t,<t)}^\top K_{(<t,<t)}^{-1}\big) \otimes I_{d_z}\ z_{<t}, \tag{4}$$

$$\Sigma_t = \big(k_{(t,t)} - k_{(t,<t)}^\top K_{(<t,<t)}^{-1} k_{(t,<t)}\big)\, I_{d_z}, \tag{5}$$

with the convention that $p(z_1) = \mathcal{N}(0, k(t_1, t_1) I_{d_z})$.

Proof. By (H1), K_{tt} is symmetric positive definite, hence invertible, so the finite-dimensional Gaussian measure $\mathcal{N}(0, K_{tt} \otimes I_{d_z})$ on $\mathbb{R}^{Ld_z}$ is non-degenerate. In particular, for each $t \in \{2, \ldots, L\}$ the pair $(z_{<t}, z_t)$ is a centred Gaussian vector taking values in the Polish space $\mathbb{R}^{(t-1)d_z} \times \mathbb{R}^{d_z}$.

The existence and uniqueness (up to $p(z_{<t})$-null sets) of regular conditional distributions $p(z_t \mid z_{<t})$ then follows from the general disintegration theorem for probability measures on Polish spaces; see for instance Kallenberg [7, Theorem 6.3]. In the Gaussian case, these regular conditional laws are themselves Gaussian and are given explicitly by the conditioning formulas (1)–(2).

For each $t \geq 2$ we partition

$$z_{1:L} = (z_{<t}, z_t), \qquad z_{<t} \in \mathbb{R}^{(t-1)d_z},\ z_t \in \mathbb{R}^{d_z},$$

and the covariance as

$$\Sigma_{11} = K_{(<t,<t)} \otimes I_{d_z}, \quad \Sigma_{12} = k_{(<t,t)} \otimes I_{d_z}, \quad \Sigma_{22} = k_{(t,t)} \otimes I_{d_z},$$

where $K_{(<t,<t)} \in \mathbb{R}^{(t-1)\times(t-1)}$ and $k_{(<t,t)} \in \mathbb{R}^{t-1}$. Hence $\Sigma_{11} \in \mathbb{R}^{(t-1)d_z \times (t-1)d_z}$, $\Sigma_{12} \in \mathbb{R}^{(t-1)d_z \times d_z}$ and $\Sigma_{22} \in \mathbb{R}^{d_z \times d_z}$, so that the Kronecker expressions in (4)–(5) are dimensionally consistent. Applying (1)–(2) to this partition yields the claimed formulas for μ_t and Σ_t.

Finally, since $p(z_{1:L})$ admits a Lebesgue density on $\mathbb{R}^{Ld_z}$, the factorization (3) follows from the chain rule for densities, and its uniqueness from the uniqueness (up to null sets) of the conditional Gaussian laws $p(z_t \mid z_{<t})$. □

We now formalize the notion of purely latent autoregression alluded to in the introduction. The setting is that of a latent process $(z_1, \ldots, z_L)$, with $z_t \in \mathbb{R}^{d_z}$, endowed with a joint distribution that factorizes causally in latent space.

4.3.1 General principle

A model exhibits purely latent autoregression if the joint law of $z_{1:L}$ admits a factorization

$$p(z_{1:L}) = \prod_{t=1}^{L} p(z_t \mid z_{<t}), \tag{6}$$

where each conditional $p(z_t \mid z_{<t})$ depends only on the past latents and not on the observations. In the Gaussian-process-based construction considered here, these conditionals are Gaussian:

$$p(z_t \mid z_{<t}) = \mathcal{N}(\mu_t, \Sigma_t), \tag{7}$$

$$\mu_t = k_{(t,<t)}^{\top} K_{(<t,<t)}^{-1} z_{<t}, \tag{8}$$

$$\Sigma_t = k_{tt} - k_{(t,<t)}^{\top} K_{(<t,<t)}^{-1} k_{(t,<t)}, \tag{9}$$

where $K_{(<t,<t)}$ is the covariance of $(z_1, \ldots, z_{t-1})$ and $k_{(t,<t)}$ their cross-covariance with z_t. These formulas follow from classical multivariate Gaussian conditioning and show that the latent dynamics are governed by the covariance structure of the GP rather than by parametric transition weights.

In this sense the model can be interpreted as a Bayesian autoregression: memory of the past is transmitted through correlations in the prior rather than through explicitly learned recurrence. A trajectory $z_{1:L}$ is first sampled according to (6), and observations are then generated conditionally on this latent path.

4.3.2 Comparison with Markov and observation-level autoregression

The factorization (6) should be contrasted with the first-order Markov property

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

in which the conditional law at time t depends only on the immediate predecessor z_{t-1}. Markov models are often sufficient for physical processes with short-range dependencies, but they are limited in their ability to express long-range temporal structure without additional hierarchy.

On the other hand, classical autoregressive models at the observation level specify

$$p(x_{1:N}) = \prod_{t=1}^{N} p(x_t \mid x_{<t}),$$

and therefore place causality directly on the observables. This formulation permits rich dependence but typically entails sequential generation and sensitivity to local errors, as each x_t conditions on all preceding observations.

The present model occupies an intermediate position. It retains full latent causality (each z_t depends on $z_{<t}$) but operates in a continuous latent space where the covariance structure enforces smooth, probabilistic coherence. Temporal dependence is thus decoupled from the symbolic or observed sequence, which is recovered in a second step via the observation model.

4.3.3 Formal definition

For future reference, we state the definition adopted in this paper.

Definition 4.2 (Latent autoregressive process)**.** *Let $(z_1, \ldots, z_L)$ be random variables with values in $\mathbb{R}^{d_z}$, defined on a probability space $(\Omega, \mathcal{F}, \mathbb{P})$. We say that $(z_1, \ldots, z_L)$ forms a* latent autoregressive process *if:*

1. *The joint law of $z_{1:L}$ admits a density $p(z_{1:L})$ with respect to the Lebesgue measure on $\mathbb{R}^{Ld_z}$.*

2. *There exist Borel-measurable functions $m_t : (\mathbb{R}^{d_z})^{t-1} \to \mathbb{R}^{d_z}$ and symmetric positive-definite matrices $\Sigma_t \in \mathbb{R}^{d_z \times d_z}$ such that*

$$p(z_{1:L}) = p(z_1) \prod_{t=2}^{L} \varphi(z_t; m_t(z_{<t}), \Sigma_t), \tag{10}$$

where $\varphi(\cdot; m, \Sigma)$ denotes the Gaussian density with mean m and covariance Σ.

In the GP-based model developed in this paper, m_t and Σ_t are not free parameters: they are given uniquely by the Gaussian conditioning formulas (4)–(5) associated with the kernel k and the evaluation times $(t_1, \dots, t_L)$.

Remark 1 (On densities vs. regular conditional laws). In Definition 4.2 we formulate latent autoregression at the level of Lebesgue densities. In the GP-based construction of Section 4.3, hypothesis (H1) implies that $K_{tt} \otimes I_{d_z}$ is strictly positive definite, so the joint law of $z_{1:L}$ is a non-degenerate Gaussian measure on $\mathbb{R}^{Ld_z}$ and therefore admits such a density.

If one wishes to include degenerate Gaussian priors (for instance in the limit of a vanishing nugget), the same notion can be reformulated in terms of regular conditional probabilities $\mathbb{P}(z_t \in \cdot \mid z_{<t})$ without reference to densities, using standard disintegration results on Polish spaces; see again Kallenberg [7, Theorem 6.3].

In the GP-based model developed here, f_θ and Σ_t are determined analytically by the covariance kernel and the conditioning formulas. When f_θ depends only on z_{t-1} one recovers a Markov structure; when it depends on the full history $z_{<t}$ one obtains full latent autoregression. This definition emphasizes that the causal structure is entirely internal to the latent process and distinct from any autoregression at the observation level.

4.4 Variational Formulation and Objective

We now recall the variational formulation used for inference and learning in the model. Let $x_{1:N}$ denote an observed sequence and $z_{1:L}$ a latent trajectory. The joint distribution factorizes as

$$p_\theta(x_{1:N}, z_{1:L}) = p_\theta(x_{1:N} \mid z_{1:L})\, p_\theta(z_{1:L}), \tag{11}$$

where $p_\theta(z_{1:L})$ is the latent autoregressive GP prior described above, and $p_\theta(x_{1:N} \mid z_{1:L})$ the observation model.

The marginal likelihood of the data is

$$\log p_\theta(x_{1:N}) = \log \int p_\theta(x_{1:N}, z_{1:L})\, \mathrm{d}z_{1:L}, \tag{12}$$

which is typically intractable. A standard variational approximation introduces an auxiliary posterior $q_\phi(z_{1:L} \mid x_{1:N})$ and uses Jensen's inequality to obtain the evidence lower bound

$$\log p_\theta(x_{1:N}) \;\geq\; \mathbb{E}_{q_\phi(z_{1:L}|x_{1:N})}\big[\log p_\theta(x_{1:N} \mid z_{1:L})\big] - D_{\mathrm{KL}}(q_\phi(z_{1:L} \mid x_{1:N}) \,\|\, p_\theta(z_{1:L})). \tag{13}$$

The first term encourages $p_\theta(x_{1:N} \mid z_{1:L})$ to place mass on the observations when $z_{1:L}$ is sampled from the variational posterior; the second term regularizes the variational posterior toward the correlated GP prior. The variational parameters ϕ and the model parameters θ are learned by maximizing this lower bound over data.

In practice, it is often convenient to introduce a scale parameter $\beta > 0$ on the divergence term:

$$\mathcal{L}_\beta(\theta, \phi) = \mathbb{E}_{q_\phi(z_{1:L}|x_{1:N})}\big[\log p_\theta(x_{1:N} \mid z_{1:L})\big] - \beta\, D_{\mathrm{KL}}(q_\phi(z_{1:L} \mid x_{1:N}) \,\|\, p_\theta(z_{1:L})), \tag{14}$$

which controls the trade-off between fidelity to the observations and adherence to the latent autoregressive prior. The case $\beta = 1$ corresponds to the usual ELBO; values $\beta \neq 1$ interpolate between stronger regularization and more flexible reconstructions.

4.4.1 Notation and assumptions

For clarity, we summarize the main notation:

- $x_{1:N}$: observed sequence;
- $z_{1:L}$: latent sequence, with L not necessarily equal to N;
- $p_\theta(z_{1:L})$: latent autoregressive GP prior;
- $q_\phi(z_{1:L} \mid x_{1:N})$: variational posterior (encoder);
- $p_\theta(x_{1:N} \mid z_{1:L})$: observation model (decoder);
- k_ψ: covariance kernel, parameterized by hyperparameters ψ;
- θ, ϕ, ψ: collections of parameters for the prior, variational family, and kernel.

Under these assumptions, the model defines a fully differentiable generative system in which temporal dependence is embedded in the latent prior, and the interaction with data is mediated by the observation model and the variational posterior.

5 Experiments - Empirical Validation and Limitations of the Latent Autoregressive Scheme

This chapter provides an empirical validation of the purely latent autoregressive scheme described in the previous sections. The aim is not to claim any form of asymptotic or large-scale optimality, but rather to demonstrate, in a controlled and reproducible regime, that the proposed probabilistic construction behaves as expected.

More precisely, we seek to verify the following points:

- a GP-VAE endowed with fully latent causality is trainable and numerically stable;
- the correlated latent prior is effectively used, in a way that cannot be reduced to a mere *KL-capping* artefact;
- the two sampling schemes (sequential vs. parallel) are empirically consistent with the same joint law;
- in the constrained regime considered here, the model surpasses a minimal autoregressive baseline in the observation space.

Throughout this chapter, all conclusions should be read as a *local* validation of the latent-autoregressive scheme under a restricted computational budget, rather than as a large-scale benchmark.

5.1 Experimental Objectives and Tested Hypotheses

We consider a corpus $\mathcal{D} = \{x^{(i)}_{1:N_i}\}_{i=1}^{M}$ and models of the form

$$p_\theta(x, z) = p_\theta(x \mid z)\, p_\theta(z),$$

with a correlated, causal latent prior $p_\theta(z)$ as defined previously. We evaluate the following hypotheses.

H1 - Trainability. There exists a set of hyperparameters (θ, ϕ, ψ) such that, on WikiText-2 [12],

$$\text{ELBO/token} \quad \text{converges to a finite limit and remains numerically stable,} \tag{15}$$

without divergence in the GP covariance inversion nor in the decoder gradients.

H2 - Effective Use of the Correlated Latent Space. Let $p_\theta^{\text{GP}}(z)$ denote the GP-AR prior and $p_\theta^{\text{iso}}(z)$ an isotropic Gaussian prior with $K = \sigma^2 I$. We define the token-averaged Kullback-Leibler term:

$$\text{KL/token} \;=\; \frac{1}{T}\,\text{KL}\big(q_\phi(z_{1:T} \mid x) \,\|\, p_\theta(z_{1:T})\big).$$

We monitor:

- the raw KL/token and its capped version under a threshold `kl_cap`;
- the dependence of the KL/token on `kl_cap` and on the final value β_{final};
- ablations where $p_\theta(z)$ is replaced by $p_\theta^{\text{iso}}(z)$.

Hypothesis H2 is considered supported if:

1. performance degrades when replacing $p_\theta^{\text{GP}}(z)$ by $p_\theta^{\text{iso}}(z)$,
2. KL/token remains significantly above the degenerate regime associated with latent collapse.

H3 - Consistency Between Lambda-SEQ and Lambda-PARA. Let $p_\theta^{\text{SEQ}}(z_{1:T})$ and $p_\theta^{\text{PARA}}(z_{1:T})$ denote the distributions implicitly realized by Lambda-SEQ (sequential GP conditioning) and Lambda-PARA (parallel Cholesky sampling). Theoretically, both schemes approximate the same joint law:

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

up to numerical precision. We test whether empirical metrics

$$\text{ELBO/token}, \quad \text{NLL(cont)}, \quad \text{PPL(cont)}, \quad \text{KL/token}$$

remain statistically indistinguishable (within the variance induced by optimization and numerical approximations).

The continuous negative log-likelihood NLL(cont) is estimated as

$$\text{NLL(cont)} \;=\; -\frac{1}{|\mathcal{D}_{\text{cont}}|} \sum_{(x_{1:T}) \in \mathcal{D}_{\text{cont}}} \log p_\theta(x_{1:T} \mid \text{prompt}),$$

where p_θ is evaluated on the logits (pre-softmax scores). The continuous perplexity is then defined as

$$\text{PPL(cont)} \;=\; \exp\big(\text{NLL(cont)}\big).$$

H4 - Minimal Comparison with an Autoregressive Baseline. We compare the Lambda family to a small Transformer baseline acting directly on tokens:

$$p_\theta^{\text{AR}}(x_{1:N}) = \prod_{t=1}^{N} p_\theta^{\text{AR}}(x_t \mid x_{<t}).$$

The objective is purely relative: to position the GP-VAE in terms of perplexity under a minimal autoregressive configuration, without any claim of fairness or exhaustive tuning in favor of the baseline.

5.2 Experimental Protocol

5.2.1 Corpus and Tokenization

All experiments are conducted on WikiText-2 [12] using the official train/validation/test splits. Text is tokenized with the GPT-2 tokenizer [14], yielding sequences

$$x_{1:T} \in \{1, \dots, V\}^T,$$

with a fixed sequence length $T = 64$. This choice bounds the cubic GP cost $O(T^3)$ at a level compatible with a single-GPU setup while preserving non-trivial temporal structure.

5.2.2 Task and Evaluation Metrics

The task is an autoregressive-style language modeling objective over $x_{1:T}$. We report:

- the validation perplexity PPL(val), defined as

$$\mathrm{PPL}(\mathrm{val}) = \exp\left(-\frac{1}{|\mathcal{D}_{\mathrm{val}}|} \sum_{x \in \mathcal{D}_{\mathrm{val}}} \frac{1}{T} \log p_\theta(x) \right);$$

- the continuation perplexity PPL(cont), computed on a standardized prompt+completion protocol (fixed prompt length, free continuation length).

In all cases, PPL is the exponential of an average negative log-likelihood per token, so that lower values correspond to better predictive performance.

5.2.3 Implementation of the Latent Autoregressive Scheme

We briefly summarize how the latent autoregressive mechanism is instantiated in the experiments.

(a) TCN encoder and diagonal temporal posterior. The encoder is a causal TCN that maps tokens $x_{1:T}$ to a factorized Gaussian posterior

$$q_\phi(z_{1:T} \mid x_{1:T}) = \prod_{t=1}^{T} \mathcal{N}(z_t \,;\, \mu_t, \mathrm{diag}(\sigma_t^2)),$$

with

$$\mu, \log \sigma^2 \in \mathbb{R}^{B \times T \times d_z} \quad \text{for a batch of size } B.$$

Sampling is performed via the standard reparameterization trick:

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

(b) Correlated GP prior over the latent trajectory. Temporal dependence in the latent space is induced by a Gaussian Process indexed by normalized times $t \in [0, 1]$. For a sequence of length T, we define

$$K_{tt}(i, j) = \sigma^2 \exp\!\Big(-\tfrac{(t_i - t_j)^2}{2\ell^2} \Big) + \sigma^2 \,\mathrm{nugget} \cdot \delta_{ij}, \quad 1 \le i, j \le T,$$

with learnable hyperparameters ℓ (length-scale), σ^2 (variance) and a relative nugget term. The latent prior is then

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

In practice, ℓ and σ^2 are parameterized via unconstrained variables passed through a *softplus* non-linearity, and an additional jitter term εI is added to K_{tt} to stabilize Cholesky decompositions.

Crucially, this joint Gaussian prior is not only used as a correlated regularizer: it is explicitly factorized into conditionals $p_\theta(z_t \mid z_{<t})$ in order to induce latent causality and support the Lambda-SEQ and Lambda-PARA sampling schemes.

(c) Global KL between diagonal posterior and GP prior. The regularization term is the trajectory-level divergence

$$\mathrm{KL}(q_\phi(z_{1:T} \mid x) \,\|\, p_\theta(z_{1:T})),$$

computed as a multivariate Gaussian KL and then averaged per token:

$$\mathrm{KL/token} = \frac{1}{T}\,\mathrm{KL}(q_\phi(z_{1:T} \mid x) \,\|\, p_\theta(z_{1:T})).$$

This quantity measures how strongly the encoder is constrained by the GP-induced temporal geometry.

(d) Non-autoregressive token decoder. The decoder receives the full latent trajectory $z_{1:T}$, adds positional information, and outputs token logits in parallel. Formally, the generative model takes the form

$$p_\theta(x_{1:T} \mid z_{1:T}) = \prod_{t=1}^{T} \mathrm{Cat}(x_t;\, \pi_\theta(z_{1:T})_t),$$

where $\pi_\theta(z_{1:T})_t$ denotes the softmax-normalized logits at position t.

The implementation is summarized by:

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

No causal mask or token-level recursion is used: all positions are decoded simultaneously from $z_{1:T}$. All sequential structure is therefore carried by the latent process; the decoder implements a parallel projection from continuous trajectories to discrete token distributions.

(e) Unconditional generation (Lambda-SEQ). Unconditional sampling realizes the factorization

$$p_\theta(z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t \mid z_{<t})$$

via a latent loop:

```
@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
    Dz, B = self.cfg.d_latent, batch_size
    z = torch.zeros(B, T, Dz, device=device)

    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
    )
```

At each step t, the function `_gp_conditional_step` performs Gaussian conditioning to sample from $p_\theta(z_t \mid z_{<t})$.

(f) Prompt-conditioned generation (Lambda-SEQ). Conditioned generation proceeds by first encoding a prompt into a latent prefix, then continuing autoregressively in the latent space:

```
@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

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

    mu, logvar = self.encoder(x_in[:, :T0])
    std = torch.exp(0.5 * logvar)
    z_prompt = mu + std * torch.randn_like(std)

    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

    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
```

Lambda-PARA replaces the latent loop with a block sampling scheme based on a Cholesky factor of K_{tt}, but targets the same joint law $p_\theta(z_{1:T})$.

5.2.4 Objective: ELBO Per Token

Training maximizes a token-averaged ELBO with additional regularization.

Theoretical ELBO. The canonical variational objective is

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

In code, this corresponds to

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

where `ll_0 + ll_multi` is the token-averaged log-likelihood and `kl_tok_raw_t` the raw KL/token.

Optimized training objective. The practically optimized objective is

$$\mathcal{L}_{\text{train}} = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \beta \,\text{KL}_{\text{cap}} - \lambda_{\text{emb}} \,\text{Reg}_{\text{emb}},$$

where:

- β follows a warm-up schedule before adapting around a KL/token target;
- KL_{cap} is a capped version of the KL/token, with threshold `kl_cap`;
- Reg_{emb} is an embedding-regularization term.

The hyperparameters (`kl_cap`, $\beta_{\max}$, adaptation schedule) are explored systematically in the ablations of Section 4.6.2.

5.3 Variants Lambda-SEQ and Lambda-PARA

We consider two latent sampling schemes:

- Lambda-SEQ: sequential sampling via GP conditionals $p_\theta(z_t \mid z_{<t})$;
- Lambda-PARA: parallel block sampling using a Cholesky factor of K_{tt}.

Both variants share the same encoder and decoder architectures. Hence, any difference in performance can be attributed to the numerical realization of the latent dynamics rather than to architectural changes.

5.4 Global Results (Lambda-SEQ vs Lambda-PARA)

On WikiText-2, we observe:

- no numerical divergence over the full training horizon;
- a convergent and stable KL/token plateau;
- very close metrics between Lambda-SEQ and Lambda-PARA.

The KL/token stabilizes around 12 nats. Unlike in preliminary experiments, Section 6.6.1 shows that this plateau depends on *kl_cap* and disappears when the GP structure is removed, which confirms the active use of the correlated prior. Training curves are not exactly superposed, but discrepancies remain in the range expected for two distinct numerical approximations of the same joint law.

5.5 Quantitative Comparison Lambda-SEQ vs Lambda-PARA

The following table reports results under identical hyperparameters:

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

Table 1: Results for Lambda-SEQ vs Lambda-PARA on WikiText-2.

Both variants use the same *kl_cap* (KL/token ≈ 12) and a final KL weight $\beta_{\text{final}} \approx 0.126$ (not shown in the table).

Interpretation. The two schemes yield very close metrics. Lambda-PARA slightly improves continuation perplexity, at the cost of negligible differences in throughput at $T = 64$. Given the absence of multi-seed analysis and the numerical differences between sampling procedures, the two variants can reasonably be regarded as empirically consistent approximations of the same latent process.

5.6 Extended Analysis: Variants, Ablations, and Stability

5.6.1 GP-VAE-TCN vs Transformer Performance (Lambda-SEQ-X Family)

We evaluate several Lambda-SEQ-X variants (with $X \in \{I, B, C, J, K\}$) sharing the same GP-VAE backbone but differing in regularization settings:

Model	PPL(val)	NLL(cont)	PPL(cont)	tok/s	Quality
Lambda-SEQ-I	3.35	0.4773	1.61	~ 9000	Excellent
Lambda-SEQ-B	3.27	0.5455	1.73	~ 8900	Very good
Lambda-SEQ-C	3.34	0.5077	1.66	~ 9000	Very good
Lambda-SEQ-J	3.03	0.5635	1.76	~ 9000	Good
Lambda-SEQ-K	3.05	0.6005	1.82	~ 9000	Good
Transformer	326.94	5.7898	326.94	~ 15700	Poor

Table 2: Lambda-SEQ-X variants and Transformer baseline on WikiText-2.

Analysis. All Lambda-SEQ-X variants obtain PPL(cont) $\in [1.61, 1.82]$, compatible with a compact, well-regularized GP-VAE. The Transformer baseline is deliberately underpowered and only used as a numerical reference: it does not reflect the capabilities of a tuned autoregressive model at comparable scale.

5.6.2 Critical Ablations

Isotropic diagonal prior. We replace the GP prior by $p_\theta^{\text{iso}}(z_{1:T}) = \mathcal{N}(0, \sigma^2 I)$, thereby eliminating all temporal correlation. Under this ablation:

- KL/token drops to about 3 nats, indicating partial latent collapse;
- PPL(cont) degrades by +0.15 to +0.30;
- discourse-level coherence and calibration deteriorate sharply.

Sequential/parallel consistency also disappears, confirming that the GP structure is not redundant.

Variation of *kl_cap*. We vary *kl_cap* to probe the sensitivity of the latent dynamics to KL regularization. Empirically:

- *kl_cap* too small (e.g. 8) leads to latent collapse, reduced KL/token and degraded PPL(cont);
- *kl_cap* too large (e.g. ≥ 20) yields stable but more costly models;
- intermediate values around 12 provide a robust compromise between expressivity and stability.

These observations confirm that the activation of latent causality depends directly on maintaining an appropriate KL budget.

5.6.3 Lambda-2-SEQ-X Series: Stability and Collapse

The Lambda-2-SEQ-X series explores regularization sensitivity in a finer grid:

Model	NLL(cont)	PPL(cont)	PPL(val)	ELBO/tok	tok/s	Quality
Lambda-2-SEQ-T	0.2883	1.33	3.18	-9.058	~ 8600	Excellent
Lambda-2-SEQ-S	0.3097	1.36	3.05	-8.927	~ 8500	Very good
Lambda-2-SEQ-G	0.4606	1.59	3.19	-8.501	~ 8500	Good
Lambda-2-SEQ-H	0.4790	1.61	3.29	-9.958	~ 8600	Good
Transformer	5.7898	326.94	326.94	-6.105	15700	Very poor

Table 3: Examples of Lambda-2-SEQ-X models vs Transformer.

Interpretation. Well-regularized models (T, S, G, H) achieve PPL(cont) $\in [1.33, 1.61]$. By contrast, under-regularized configurations (not listed) exhibit complete collapse, with unstable KL and poor continuation, which confirms the central role of KL regularization in supporting latent autoregression.

5.7 Transformer Baseline

For completeness, we summarize the Transformer baseline:

Model	PPL(val)	NLL(cont)	PPL(cont)	tok/s
Transformer	326.94	5.7898	326.94	~ 15700

Table 4: Minimal Transformer baseline.

This model is intentionally minimal and not heavily tuned; it serves only as a coarse reference scale for perplexity and throughput.

5.8 Synthesis

The empirical results can be summarized as follows:

- Lambda-SEQ and Lambda-PARA are both trainable and numerically stable on WikiText-2 under a causal GP prior.
- Ablations confirm that the GP structure is actively used: removing it or over-constraining the KL leads to measurable degradation in PPL(cont) and in qualitative coherence.
- Lambda-PARA is empirically consistent with Lambda-SEQ, supporting the idea that parallel latent sampling can approximate the same GP-AR process.
- GP-VAEs outperform the minimal Transformer baseline in the considered regime, providing a lower bound on the practical capacity of the latent-autoregressive scheme.
- KL regularization is a critical control parameter: insufficient regularization induces collapse, while an appropriately tuned *kl_cap* maintains non-trivial latent dynamics.

5.9 Limitations and Perspectives

Limitations. The present study has several limitations:

- GP computations remain at least quadratic in T; although BBMM empirically behaves near $O(T^2)$, no dedicated scaling law is reported.
- All results are single-seed; variance across random initializations is not explored.
- The Transformer baseline is minimal and not tuned for competitive performance; it serves as a numerical anchor rather than a fair opponent.

- The TCN encoder is deliberately less expressive than a Transformer; this bias reflects a design choice favoring architectural simplicity in the proof-of-concept.
- No systematic qualitative analysis of generated samples is included; we focus on quantitative metrics and leave a detailed study of generative behavior to future work.

Perspectives. Several extensions are natural:

- more scalable GP approximations (structured kernels, inducing schemes) could allow significantly longer sequences;
- richer decoders (e.g. attention with restricted span) could be combined with latent causality without reverting to token-level autoregression;
- broader prompting protocols and conditional tasks would provide a more complete characterization of latent autoregression;
- the same framework extends directly to continuous-time signals and time-series data, which constitute promising application domains for GP-based latent dynamics.

5.10 Code and Reproducibility

All code used for the experiments in this work (GP hyperparameter estimation, latent autoregressive implementation, Lambda-SEQ and Lambda-PARA variants, training and generation scripts, and associated configurations) is available at:

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

The repository contains:

- the full implementation of the latent-autoregressive GP-VAE;
- training, evaluation, and generation scripts;
- configuration files for reproducing the Lambda-SEQ and Lambda-PARA series;
- exact hyperparameter settings used to produce the tables of Section 4 (including `kl_cap` values and KL schedules);
- a minimal reproducibility guide.

This release is intended to make the proposed scheme inspectable, reproducible, and extensible, in particular for exploring alternative kernels, encoder architectures, or latent sampling strategies.

6 Discussion

This study has examined, in a controlled and small-scale setting, the feasibility and behavior of an autoregressive scheme located entirely in latent space. Concretely, the Lambda model and its two variants, Lambda-SEQ and Lambda-PARA, instantiate a family of models of the form

$$p_\theta(x, z) = p_\theta(x \mid z)\, p_\theta(z),$$

where $p_\theta(z)$ is a correlated, causal latent prior and $p_\theta(x \mid z)$ is a fully parallel decoder. These variants provide a test bed for assessing:

- training stability under a Gaussian-process prior on z;
- effective exploitation of the correlated latent space (as measured by KL/token and ablations);
- coherence between two distinct sampling strategies that target the same joint latent law.

6.1 Exploiting the Correlated Latent Structure

Empirically, a GP-VAE equipped with a correlated prior $p_\theta(z)$ can be trained stably on a standard corpus under moderate regularization. Let

$$\text{KL/token} = \frac{1}{T} D_{\text{KL}}(q_\phi(z_{1:T} \mid x) \,\|\, p_\theta(z_{1:T}))$$

denote the token-averaged KL term. The experiments show that KL/token reliably approaches its target cap `kl_cap` across runs, which indicates that:

- the latent representation is not in a collapsed regime (the encoder uses the GP prior);
- the GP covariance K_{tt} effectively shapes the latent trajectories $z_{1:T}$.

In this proof-of-concept setting, the objective is not to quantify strong or long-range latent causality per se, but to verify that a correlated latent scheme produces a non-degenerate internal trajectory and that the GP prior is actually used by the model rather than acting as a purely nominal regularizer.

6.2 Consistency Between Sequential and Parallel Generation

Let $p_\theta^{\text{SEQ}}(z_{1:T})$ and $p_\theta^{\text{PARA}}(z_{1:T})$ denote the latent distributions implicitly realized by Lambda-SEQ and Lambda-PARA, respectively. Both variants are designed to approximate the same joint Gaussian law

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

but via different sampling procedures:

- Lambda-SEQ uses step-by-step Gaussian conditioning $p_\theta(z_t \mid z_{<t})$;
- Lambda-PARA uses block sampling from a Cholesky factor of K_{tt}.

The measured metrics (ELBO/token, NLL(cont), PPL(cont), KL/token) are very close across the two variants. This is consistent with the theory of multivariate Gaussians: sequential conditioning and parallel sampling are two equivalent procedures for drawing from the same joint distribution.

From a methodological standpoint, this observation supports the idea that, under a GP prior, the *sampling order* does not determine predictive quality. The temporal structure is encoded in the prior $p_\theta(z)$, not in the algorithmic details of the sampler, provided that both samplers are faithful to the same covariance structure.

6.3 Latent Dynamics vs. Symbolic Sequentiality

A central conceptual point of this work is the distinction between:

- *procedural sequentiality*, associated with the temporal loop of step-by-step sampling (Lambda-SEQ);
- *probabilistic structure*, encoded by the analytic factorization of the GP prior:

$$p_\theta(z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t \mid z_{<t}).$$

The two Lambda variants show that it is possible:

- to represent temporal dependence entirely within the latent space via the factorization of $p_\theta(z_{1:T})$;

- while keeping the symbolic projection $p_\theta(x_{1:T} \mid z_{1:T})$ fully parallel in the observation space.

Formally, the decoder implements

$$p_\theta(x_{1:T} \mid z_{1:T}) = \prod_{t=1}^{T} p_\theta(x_t \mid z_{1:T}),$$

so that all tokens are generated in a single pass once $z_{1:T}$ is given. This stands in contrast to classical autoregressive architectures, which specify

$$p_\theta^{\mathrm{AR}}(x_{1:T}) = \prod_{t=1}^{T} p_\theta^{\mathrm{AR}}(x_t \mid x_{<t}),$$

and thus construct temporal structure through symbolic recursion.

The empirical results do not aim to demonstrate dominance over established autoregressive models, but they do highlight a qualitatively different regime: temporal dependence is defined analytically in latent space and only then projected to symbols, rather than being built directly at the token level.

6.4 Scaling Perspectives

The present work is deliberately restricted to a reduced configuration: short sequences ($T = 64$), a compact architecture, and a medium-sized corpus. Within this regime, the GP prior remains computationally tractable and the effect of latent autoregression is observable.

Several natural scaling directions follow:

- extending to larger T and richer kernels k_ψ, in order to probe long-range latent dependence;
- exploring more expressive decoders (e.g. restricted attention, deeper architectures) while preserving parallel generation in the observation space;
- transferring the same latent-autoregressive scheme to other modalities (time series, continuous signals, multimodal data).

Each of these directions must be considered under the scalability constraints of Gaussian processes: even with BBMM and inducing schemes, the cost of handling large covariance matrices remains at least quadratic in sequence length and requires careful numerical design.

6.5 Limitations and Points of Attention

The experimental setting highlights several critical limitations and conditions of validity:

- **Computational cost.** GP computations scale at least quadratically in T; the proof-of-concept remains constrained to moderate sequence lengths and a single-GPU budget.
- **Hyperparameter sensitivity.** Performance depends on kernel hyperparameters (ℓ, σ^2, nugget), as well as on KL-related hyperparameters (`kl_cap`, $\beta_{\max}$); poorly tuned settings can cause latent collapse or instability.
- **Non-incremental parallel decoding.** The fully parallel decoder does not support token-by-token streaming: the model prioritizes global coherence over incremental generation.
- **Single-seed evaluation.** All reported results are single-seed; variance across random initializations is not quantified and remains an open point for more systematic studies.

These limitations delimit the regime in which the observed trends should be interpreted and underline the need for more extensive experimentation before drawing broader conclusions.

6.6 Discussion Summary

Taken together, the results show that a latent autoregressive scheme based on:

- an analytically specified covariance (Gaussian process prior),
- an explicit causal factorization in the latent,
- and a fully parallel decoder in the observation space,

is trainable, numerically stable, and coherent across its two sampling variants Lambda-SEQ and Lambda-PARA.

Without claiming superiority over established autoregressive architectures, this proof of concept suggests that, in certain regimes, temporal structure can be *relocated* into latent space:

$$p_\theta(x_{1:T}) = \int p_\theta(x_{1:T} \mid z_{1:T})\, p_\theta(z_{1:T})\, \mathrm{d}z_{1:T},$$

so that the decoder acts as a projection of a already-structured latent dynamic onto a symbolic sequence.

This shift opens a methodological space of interest: that of sequential models where the dynamics are defined analytically in the latent prior, rather than constructed through stacks of recurrent or attentional operations at the observation level.

6.7 Latent Autoregression vs. Asymmetric Kernels

Finally, it is useful to situate the proposed latent autoregressive scheme with respect to earlier attempts at introducing temporal directionality into Gaussian processes.

Some previous works have considered asymmetric or "causal" kernels (for instance, Wiener-type constructions) to encode an orientation in the covariance structure. Although such kernels can introduce a directional bias, they remain fundamentally metric: they modulate correlations as a function of relative positions in time, but do not, by themselves, define an explicit sequential dynamic.

The present approach differs in nature. Rather than inferring causality indirectly from a kernel bias, we impose it *structurally* through the latent factorization

$$p_\theta(z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t \mid z_{<t}),$$

which encodes an explicit directed dependence between latent states. This factorization resolves ambiguities inherent in purely asymmetric kernels (for instance when several points have comparable kernel distances) by providing a genuine mechanism for sequential arbitration in the latent.

It is also important to distinguish this explicit causal factorization from Bayesian GP extensions that place priors over kernel hyperparameters. While such hierarchical models enrich the distribution over covariances, they do not, by themselves, create temporal directionality: the uncertainty concerns the shape of the kernel, not a sequential relation between states. Latent autoregressivity, as used here, arises from the decomposition of the latent prior into conditionals, not from the Bayesian nature of the GP. The two mechanisms may interact in more elaborate models, but they remain conceptually distinct.

By combining:

- correlated geometry, provided by the GP covariance K_{tt},
- and explicit causal progression, provided by the factorization $p_\theta(z_{1:T}) = \prod_t p_\theta(z_t \mid z_{<t})$,

the Lambda framework goes beyond the limitations of purely asymmetric kernels and establishes a more robust latent dynamic tailored to sequential modeling.

7 Conclusion

7.1 Empirical Scope

This work has presented a deliberately restricted proof of concept. The experimental regime is intentionally small-scale:

- a compact Lambda architecture;
- short sequences ($T = 64$);
- a minimal autoregressive baseline for numerical anchoring.

Within this controlled setting, we have shown that a GP-VAE equipped with:

$$p_\theta(x, z) = p_\theta(x \mid z)\, p_\theta(z), \qquad p_\theta(z) = \mathcal{N}(0, K_{tt} \otimes I_{d_z}),$$

and endowed with a purely latent causal factorization,

$$p_\theta(z_{1:T}) = \prod_{t=1}^{T} p_\theta(z_t \mid z_{<t}),$$

is trainable, numerically stable, and coherent across its two sampling variants (Lambda-SEQ and Lambda-PARA). No divergence is observed in the GP computations, and token-level metrics remain consistent across sampling strategies that approximate the same joint latent law.

7.2 Conceptual Scope

Beyond the restricted empirical domain, the Lambda scheme can be interpreted as a bridge between two modelling traditions:

- continuous Bayesian state-space models governed by Gaussian processes;
- neural language models driven by a symbolic decoder.

The sequential dynamics are encoded analytically by the GP prior, inference follows a variational Bayesian principle, and linguistic realization is delegated to a non-autoregressive observation model:

$$p_\theta(x_{1:T}) = \int p_\theta(x_{1:T} \mid z_{1:T})\, p_\theta(z_{1:T})\, \mathrm{d}z_{1:T}.$$

Although the present study does not attempt to formulate a general geometric theory of linguistic structure, the results indicate that several properties usually associated with token-level autoregression—directionality, memory, global coherence—can emerge directly from the covariance geometry of the latent space. In this sense, latent autoregression provides an analytically grounded alternative to symbolic recursion.

7.3 Perspectives

The limitations identified throughout the study (quadratic GP cost, kernel sensitivity, lack of multiseed evaluation, and absence of incremental decoding for Lambda-PARA) define the conditions under which the observed phenomena should be interpreted. They also point to several natural directions for future research.

1. **Scaling up.** Extending the framework to longer sequences, richer kernels k_ψ, and more expressive decoder architectures, while maintaining tractable GP computations.

2. **Stronger autoregressive baselines.** A more rigorous comparison with tuned autoregressive models would clarify the precise contribution of the correlated latent component to perplexity and continuation quality.

3. **Multimodal and continuous domains.** The formulation extends naturally to time series, continuous signals, or physical trajectories, where GP priors already play a central role.

4. **Conditional generation and latent prompting.** Given a prompt $x_{1:T_0}$ with latent encoding $q_\phi(z_{1:T_0} \mid x_{1:T_0})$, one can construct a prefix

$$z_{1:T_0},$$

then generate the continuation via the latent autoregressive mechanism:

$$z_t \sim p_\theta(z_t \mid z_{<t}), \qquad t = T_0 + 1, \ldots, T,$$

before projecting the full trajectory through the decoder. This provides a principled form of latent prompting for completion or instruction-style tasks.

5. **Hierarchical Bayesian extensions.** Kernel hyperparameters can themselves be made context-dependent via a learned function g, yielding hierarchical priors of the form

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

thus enriching covariance structure without altering the causal factorization.

Taken together, these directions suggest that sequential models grounded in latent geometry—rather than symbolic recursion—constitute a credible line of research for developing compact, stable, and interpretable language models. The present proof of concept does not make claims of scale or optimality, but it demonstrates that explicit latent autoregression combined with GP covariance offers an analytically coherent framework worthy of further exploration.

References

[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] Olav Kallenberg. *Foundations of Modern Probability*. Springer, 2nd edition, 2002.

[8] Maximilian Karl, Maximilian Soelch, Justin Bayer, and Patrick van der Smagt. Deep variational bayes filters. *arXiv preprint*, arXiv:1605.06432, 2017.

[9] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint*, arXiv:1312.6114, 2014.

[10] 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.

[11] 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.

[12] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint*, arXiv:1609.07843, 2016.

[13] Tim Pearce, Michael Smith, Stefan Zohren, and Alexandra Brintrup. Bayesian autoencoders with gaussian process priors. *Proceedings of the 37th International Conference on Machine Learning*, 118:2321–2330, 2020. arXiv:1906.02511.

[14] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

[15] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[16] 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.

[17] 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.

[18] Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *AISTATS*, pages 567–574, 2009.

[19] 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.

[20] Andrew Gordon Wilson and Ryan Prescott Adams. Gaussian process kernels for pattern discovery and extrapolation. In *ICML*, pages 1067–1075, 2013. arXiv:1302.4245.

[21] 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.