# Latent Stochastic Interpolants for Probabilistic Time Series Forecasting

**Max Bourgeat** [*1 2] **Sobihan Surendran** [*3 4]

## Abstract

Deep learning has achieved strong performance in time series forecasting, yet probabilistic forecasting remains challenging due to the need to represent uncertainty, multimodality, and temporal dependence across future trajectories. Motivated by the success of latent-space generative models in domains such as image and text generation, we investigate latent generative modeling for probabilistic time series forecasting. We propose Latent Stochastic Interpolants, a conditional latent generative framework that learns expressive latent dynamics using stochastic interpolants. Our method combines a recurrent encoder that summarizes historical observations and covariates, a conditional latent stochastic interpolant, and a probabilistic decoder to model conditional future trajectories. We show that the corresponding conditional stochastic interpolant objective recovers the optimal conditional velocity field, providing a principled justification for the proposed method. Empirically, the proposed approach achieves competitive results compared with existing generative forecasting methods on standard benchmark datasets.

## 1. Introduction

Time series forecasting plays a central role in many real-world applications, including energy systems (Hong et al., 2020), transportation networks (Bai et al., 2020), and finance (Sezer et al., 2020). Beyond accurate point forecasts, many applications require predictive distributions that quantify uncertainty, capture multimodality, and support decision-making under uncertainty (Gneiting & Raftery, 2007). While deep neural architectures such as recurrent networks (Salinas et al., 2020) and Transformers (Lim et al., 2021; Zhou et al., 2021) have significantly improved forecasting accuracy, modeling complex uncertainty in sequential data remains a challenging problem.

A common approach to probabilistic forecasting relies on autoregressive likelihood-based models, which parameterize the predictive distribution at each time step conditioned on past observations and optimize the likelihood of observed data (Salinas et al., 2020; Lim et al., 2021). While effective in many settings, these methods often struggle to represent rich, multimodal future trajectories, as they rely on relatively simple output distributions or deterministic hidden representations. To address this limitation, latent-variable models introduce stochastic representations that capture uncertainty beyond deterministic states. Latent models have proven effective across diverse domains, including image generation (Rombach et al., 2022), text generation (Bowman et al., 2016), logistics (Surendran et al., 2025), and molecule design (Gómez-Bombarelli et al., 2018). They have also been explored for time series modeling (Krishnan et al., 2017; Chung et al., 2015), enabling more expressive predictive distributions. However, these approaches often prescribe the form of the latent distribution in advance, limiting their ability to learn flexible conditional latent dynamics adapted to the observed history.

In parallel, generative modeling has recently undergone significant advances with the development of score-based generative models (Song et al., 2021; Ho et al., 2020) and flow-based approaches (Lipman et al., 2023). These methods learn flexible generative processes by transforming simple noise distributions into complex data distributions through learned continuous dynamics. While they have achieved remarkable success in domains such as image and text generation, their application to time series forecasting remains relatively limited. Recent works have begun to explore these approaches for time series forecasting (Rasul et al., 2021; Tashiro et al., 2021; Chen et al., 2024b;a), but they often struggle to capture temporal dependence and structured uncertainty in sequential data.

In this work, we bridge these two lines of research by introducing a *conditional latent stochastic interpolant model*

[1]Department of Computer Engineering and Software Engineering, Polytechnique Montréal, Montréal, Canada [2]CIRRELT, Montréal, Canada [3]Sorbonne Université and Université Paris Cité, CNRS, Laboratoire de Probabilités, Statistique et Modélisation, F-75005 Paris, France [4]LOPF, Califrais' Machine Learning Lab, Paris, France. Correspondence to: Max Bourgeat <max.bourgeat@polymtl.ca>.

*Proceedings of the ICML Workshop on Forecasting as a New Frontier of Intelligence*, Seoul, South Korea, 2026. Copyright 2026 by the author(s).

for probabilistic time series forecasting. Our approach uses a recurrent encoder to summarize past observations and covariates into a deterministic state, and learns stochastic latent dynamics through conditional stochastic interpolants in latent space. By transporting simple base noise through a learned conditional dynamical system, the model captures flexible latent uncertainty adapted to the observed history. A probabilistic decoder then maps the resulting latent samples to predictive distributions over future observations. This design combines the structure of latent state-space models with the flexibility of continuous-time generative models, enabling expressive conditional latent dynamics while preserving temporal conditioning through the recurrent state.

More precisely, our contributions are summarized as follows.

- We propose a novel *conditional latent stochastic interpolant* framework for probabilistic time series forecasting, combining recurrent encoders with flexible latent generative dynamics.
- We introduce a training objective that integrates stochastic interpolant matching in latent space with probabilistic reconstruction and latent regularization. We also show that the corresponding conditional interpolant objective recovers the optimal conditional velocity field.
- We provide empirical evaluations and ablations on standard forecasting benchmarks, demonstrating that expressive latent dynamics learned through stochastic interpolants improve uncertainty modeling and forecasting performance compared to existing approaches, including generative models.

## 2. Notation and Background

### 2.1. Notation

We use $\pi$ to denote probability distributions and $p$ to denote their corresponding densities with respect to the Lebesgue measure. For $x, y \in \mathbb{R}^d$, we denote by $\langle x, y\rangle$ the standard inner product of $\mathbb{R}^d$, by $\|\cdot\|$ the Euclidean norm for vectors and its induced operator norm for matrices. We denote the time derivative of a path $f(s)$ by $\dot{f}(s) := \frac{\mathrm{d}}{\mathrm{d}s} f(s)$.

### 2.2. Stochastic Interpolants

Stochastic Interpolants define a tractable framework for generative modeling by learning continuous-time dynamics that transport samples from a simple base distribution to a complex target distribution (Albergo et al., 2025). Let $\varepsilon \sim p_0$ denote a sample from a base distribution and $z_{\mathrm{data}} \sim p_1$ a sample from the target distribution. A stochastic interpolant is a continuous family of random variables $\{z(s)\}_{s\in[0,1]}$ such that $z(0) = \varepsilon$ and $z(1) = z_{\mathrm{data}}$. A common choice is the linear stochastic interpolant defined for all $s \in [0, 1]$,

$$z(s) = \alpha(s)\varepsilon + \beta(s)z_{\mathrm{data}} + \gamma(s)\xi, \qquad \xi \sim \mathcal{N}(0, \mathbf{I}_d),$$

where $\alpha, \beta, \gamma : [0, 1] \to \mathbb{R}$ are deterministic functions satisfying $\alpha(0) = 1$, $\beta(0) = 0$, $\alpha(1) = 0$, $\beta(1) = 1$, and $\gamma(0) = \gamma(1) = 0$. When $\gamma(s) = 0$, the interpolant is deterministic, while suitable noise schedules yield stochastic paths governed by Fokker–Planck dynamics, connecting stochastic interpolants to diffusion-based models (Albergo et al., 2025). The associated pathwise velocity is

$$\dot{z}^\star(s) = \alpha'(s)\varepsilon + \beta'(s)z_{\mathrm{data}} + \gamma'(s)\xi.$$

Since the optimal velocity field $v^\star(z, s) = \mathbb{E}[\dot{z}^\star(s) \mid z(s) = z]$ depends on the data distribution, it is generally not available in closed form. We therefore approximate it with a neural velocity field $v_\theta : \mathbb{R}^d \times [0, 1] \to \mathbb{R}^d$, parameterized by $\theta \in \Theta$, and train it using the flow-matching objective (Lipman et al., 2023)

$$\mathbb{E}_{s\sim\mathcal{U}([0,1])}\mathbb{E}_{(\varepsilon,z_{\mathrm{data}},\xi)}\left[\|v_\theta(z(s), s) - \dot{z}^\star(s)\|^2\right]. \quad (1)$$

In practice, exact sampling from the learned continuous-time dynamics is generally intractable, so samples are obtained by discretizing the dynamics using numerical ODE/SDE solvers (Song et al., 2021; Ho et al., 2020).

## 3. Our Method

### 3.1. Model

We propose a conditional latent-variable model for probabilistic time series forecasting. Given an observation history and known covariates, the model constructs a deterministic recurrent state $h_{t+1}$ that summarizes the available context. A stochastic latent variable $z_{t+1}$ then captures residual uncertainty about the future observation. The architecture consists of three components: a recurrent encoder, a conditional latent generator based on stochastic interpolants, and a probabilistic decoder.

**Recurrent encoder and inference model.** Let $x_{0:t}$ denote the observation history and $c_{t+1}$ denote optional covariates available at prediction time. A recurrent encoder maintains a hidden state $h_t \in \mathbb{R}^{d_h}$ defined as

$$h_{t+1} = f_w(h_t, x_t, c_{t+1}),$$

where $f_w$ is a recurrent network. The state $h_{t+1}$ summarizes past information and conditions both the latent dynamics and the decoder. To model uncertainty in the next observation, we introduce a latent variable $z_{t+1}$ and define the predictive distribution as

$$p(x_{t+1} \mid h_{t+1}, c_{t+1}) = \int p_\theta(x_{t+1} \mid z_{t+1}, h_{t+1}, c_{t+1}) \times p(z_{t+1} \mid h_{t+1})\,\mathrm{d}z_{t+1}. \quad (2)$$

Since the true posterior is intractable, we introduce a variational distribution $q_\phi(z_{t+1} \mid x_{t+1}, h_{t+1})$ to obtain a tractable training objective by maximizing the evidence lower bound (ELBO) (Kingma & Welling, 2014).

**Latent stochastic interpolant.** We model the conditional latent distribution $p_\psi(z_{t+1}|h_{t+1})$ implicitly using a stochastic interpolant. During training, we sample a posterior latent variable $\tilde{z}_{t+1} \sim q_\phi(z_{t+1}|x_{t+1}, h_{t+1})$, a base variable $\varepsilon \sim \mathcal{N}(0, \mathbf{I}_d)$, and independent noise $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$. For $s \in [0, 1]$, we define

$$z_{t+1}(s) = \alpha(s)\varepsilon + \beta(s)\tilde{z}_{t+1} + \gamma(s)\xi, \tag{3}$$

with pathwise velocity

$$\dot{z}^\star_{t+1}(s) = \alpha'(s)\varepsilon + \beta'(s)\tilde{z}_{t+1} + \gamma'(s)\xi. \tag{4}$$

We train a conditional velocity field $v_\psi(z, s, h_{t+1})$ to match this velocity. The learned dynamics induces the implicit conditional distribution $p_\psi(z_{t+1}|h_{t+1})$.

**Probabilistic decoder.** The decoder defines the conditional likelihood $p_\theta(x_{t+1}|z_{t+1}, h_{t+1}, c_{t+1})$, mapping the latent variable, recurrent state, and covariates to a distribution over the next observation. Marginalizing over $z_{t+1}$ gives

$$p(x_{t+1} \mid h_{t+1}, c_{t+1}) = \int p_\theta(x_{t+1} \mid z_{t+1}, h_{t+1}, c_{t+1}) \times p_\psi(z_{t+1} \mid h_{t+1})\, \mathrm{d}z_{t+1}. \tag{5}$$

### 3.2. Training

The model is trained by combining a stochastic interpolant objective in latent space with a reconstruction term and a regularization term. The overall objective is

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x,h)}\left[\log p_\theta(x \mid z, h)\right] - \lambda_{\text{reg}}\, \mathrm{KL}\big(q_\phi(z \mid x, h) \,\|\, p_\psi(z \mid h)\big) + \lambda_{\text{SI}}\, \mathcal{L}_{\text{SI}}, \tag{6}$$

where $\mathcal{L}_{\text{SI}}$ denotes the conditional SI loss defined as

$$\mathbb{E}_{s\sim\mathcal{U}([0,1])}\mathbb{E}_{\varepsilon,\, \tilde{z}_{t+1},\, \xi}\left[\left\|v_\psi(z_{t+1}(s), s, h_{t+1}) - \dot{z}^\star_{t+1}(s)\right\|^2\right],$$

with $\tilde{z}_{t+1} \sim q_\phi(\cdot|x_{t+1}, h_{t+1})$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I}_d)$, and $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$. The interpolant $z_{t+1}(s)$ and target velocity $\dot{z}^\star_{t+1}(s)$ are defined in (3) and (4). The SI objective learns a conditional velocity field that transports a simple base distribution to a flexible latent distribution adapted to $h$. The reconstruction term ensures consistency with observed data, while the regularization term controls the complexity of the inference model.

Unlike standard approaches that impose a fixed-form latent distribution, the interpolant loss enables learning a flexible conditional latent distribution. From a variational perspective (Hoffman & Johnson, 2016), the KL term can be interpreted as controlling an upper bound on the conditional mutual information between $x$ and $z$, while allowing the marginal latent distribution to adapt to the data. This flexibility is important for capturing multimodal uncertainty in forecasting. The overall training procedure is summarized in Algorithm 1.

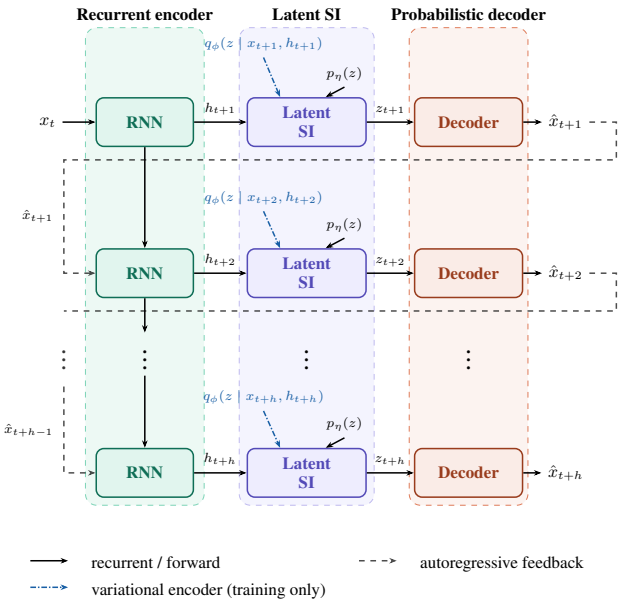


*Figure 1.* Autoregressive rollout of the proposed latent stochastic interpolant model. At each step $k$, a recurrent encoder updates the hidden state $h_{t+k}$, a latent stochastic interpolant transports Gaussian noise through learned dynamics to produce the latent sample $z_{t+k}$, which the probabilistic decoder maps to the forecast $\hat{x}_{t+k}$. The generated forecast is fed back as input for the next step. During training, the encoder $q_\phi$ provides posterior latent samples.

### 3.3. Sampling and Prediction

At test time, we first compute the recurrent state $h_{t+1}$ from the observed history and covariates, then sample a latent vector $z_{t+1} \sim p_\psi(z_{t+1}|h_{t+1})$ by integrating the learned stochastic interpolant dynamics from Gaussian noise. Given each latent sample, the decoder $p_\theta$ produces a distribution over the next observation. The predictive distribution is approximated by Monte Carlo sampling:

$$p(x_{t+1}|h_{t+1}, c_{t+1}) \approx \frac{1}{M}\sum_{m=1}^{M} p_\theta(x_{t+1}|z^{(m)}_{t+1}, h_{t+1}, c_{t+1}),$$

where $z^{(m)}_{t+1} \sim p_\psi(z_{t+1}|h_{t+1})$. A sampling step is summarized in Algorithm 2; repeating it $M$ times yields forecasts.

### 3.4. Conditional Velocity Matching

We characterize the optimal velocity field induced by the conditional interpolant objective. For a fixed conditioning state $h$, define the expected conditional interpolant loss

$$\mathcal{L}_{\mathrm{SI}}^{h}(v) = \int_0^1 \mathbb{E}\left[\|v(z(s), s, h) - \dot{z}(s)\|^2 \middle| h\right] ds. \quad (7)$$

**Proposition 3.1.** *Assume that $\alpha, \beta, \gamma$ are continuously differentiable on $[0, 1]$ and satisfy the endpoint conditions in (3), and that*

$$\mathbb{E}[\|\varepsilon\|^2 + \|\tilde{z}\|^2 + \|\xi\|^2 \mid h] < \infty$$

*for almost every $h$. Then, for almost every $h$, the unique minimizer of $\mathcal{L}_{\mathrm{SI}}^{h}(v)$ in $L^2$ is given by*

$$v^\star(z, s, h) = \mathbb{E}[\dot{z}(s) \mid z(s) = z, h].$$

*Moreover, this velocity field also minimizes the global objective $\mathcal{L}_{\mathrm{SI}}(v)$ for almost every $(z, s, h)$.*

Proposition 3.1 shows that the optimal velocity field is the conditional expectation of the target velocity, extending flow-matching results to our conditional latent setting. Thus, minimizing the interpolant objective provides a principled way to learn $v_\psi(z, s, h)$, which can approximate the optimal conditional velocity in $L^2$ with a sufficiently expressive model class.

## 4. Experiments

**Evaluation metrics.** We evaluate probabilistic forecasts using the Continuous Ranked Probability Score (CRPS-sum), which measures the quality of the predictive distribution (Koochali et al., 2022). All metrics are computed using the `GluonTS` package (Alexandrov et al., 2019).

**Datasets.** We evaluate our approach on four widely used real-world multivariate time series datasets adopted in probabilistic forecasting benchmarks: Exchange, Electricity, Solar, and Traffic. Each time series is normalized using instance-wise statistics computed on the context window. These datasets are widely used in prior work on probabilistic time series forecasting, enabling fair comparison with existing methods. Dataset details are provided in Appendix C.

**Baselines.** We compare our approach against representative generative forecasting baselines: Vec-LSTM, Denoising Diffusion Probabilistic Models (DDPM), Flow Matching (FM), Score-based Generative Models (SGM), and Stochastic Interpolants (SI). Following Chen et al. (2024a), all methods are trained until convergence and tuned over comparable hyperparameter ranges to ensure a fair comparison.

*Table 1.* CRPS-sum performance across datasets (lower is better).

| Method | Exchange | Solar | Traffic | Electricity |
|---|---|---|---|---|
| Vec-LSTM | 0.008±0.001 | 0.391±0.017 | 0.087±0.041 | – |
| DDPM | 0.007±0.001 | 0.396±0.103 | 0.099±0.022 | 0.033±0.005 |
| FM | 0.008±0.002 | 0.503±0.214 | 0.083±0.015 | **0.008±0.002** |
| SGM | 0.375±0.162 | 0.932±0.018 | 0.666±0.063 | 0.560±0.034 |
| SI | 0.007±0.001 | 0.465±0.099 | 0.083±0.005 | 0.324±0.093 |
| Latent SI | **0.006±0.001** | **0.283±0.018** | **0.042±0.004** | 0.026±0.003 |

**Implementation details.** For our approach, we use stabilization techniques, including KL balancing (Hafner et al., 2020), free bits (Kingma et al., 2016), and KL annealing (Bowman et al., 2016). All experiments are conducted on a single NVIDIA A100 GPU. Further implementation details and hyperparameters are provided in Appendix C.

**Results.** Latent SI achieves the best performance on Exchange, Solar, and Traffic, and remains competitive on Electricity. These results indicate that learning stochastic interpolants in latent space improves the modeling of complex predictive uncertainty in multivariate time series. The improvement is particularly pronounced on Traffic, the highest-dimensional dataset, suggesting that the latent formulation provides a more scalable representation of structured temporal uncertainty. In contrast, standard diffusion-based methods (DDPM, SGM) show either degraded performance or high variability across datasets, highlighting the difficulty of directly modeling high-dimensional temporal data in observation space. Flow Matching performs strongly on Electricity but is less consistent across the remaining benchmarks. Overall, these results suggest that combining stochastic interpolants with latent representations yields a more stable, scalable, and expressive framework for probabilistic time series forecasting.

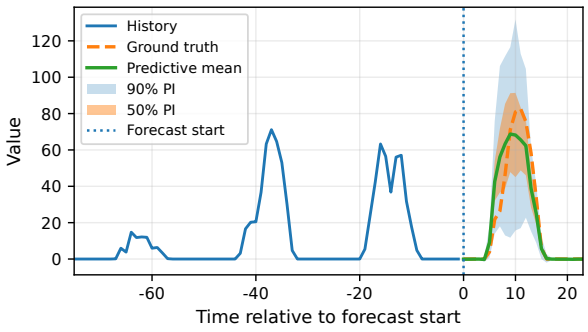


*Figure 2.* Probabilistic forecasting example on the Solar dataset. The figure shows the observed history, ground-truth future trajectory, predictive mean, and 50%/90% prediction intervals for a representative test window. The vertical dotted line indicates the forecast start, with time measured relative to this point.

## 5. Discussion

In this paper, we proposed Latent Stochastic Interpolants, a conditional latent generative framework for probabilistic time series forecasting. The proposed approach combines recurrent temporal conditioning with stochastic interpolant dynamics in latent space, allowing the model to generate flexible predictive distributions over future trajectories. Empirical results on standard forecasting benchmarks show that our method achieves competitive or improved probabilistic forecasting performance compared with existing generative baselines. Since our current implementation relies on a relatively simple architecture, an interesting direction for future work is to explore more modern architectures, such as Transformers. Further work will also focus on improving sampling efficiency and studying the calibration and sharpness of the learned predictive distributions.

## Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

## References


Albergo, M., Boffi, N. M., and Vanden-Eijnden, E. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26 (209):1–80, 2025.

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., et al. Gluonts: Probabilistic time series models in python. *arXiv preprint arXiv:1906.05264*, 2019.

Bai, L., Yao, L., Li, C., Wang, X., and Wang, C. Adaptive graph convolutional recurrent network for traffic forecasting. In *Advances in Neural Information Processing Systems*, volume 33, pp. 17804–17815, 2020.

Bowman, S., Vilnis, L., Vinyals, O., Dai, A., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. In *Proceedings of the 20th SIGNLL conference on computational natural language learning*, pp. 10–21, 2016.

Chen, Y., Biloš, M., Mittal, S., Deng, W., Rasul, K., and Schneider, A. Recurrent interpolants for probabilistic time series prediction. In *NeurIPS 2024 Third Table Representation Learning Workshop*, 2024a.

Chen, Y., Goldstein, M., Hua, M., Albergo, M. S., Boffi, N. M., and Vanden-Eijnden, E. Probabilistic forecasting with stochastic interpolants and föllmer processes. In *International Conference on Machine Learning*, pp. 6728–6756. PMLR, 2024b.

Chung, J., Kastner, K., Dinh, L., Goel, K., Courville, A., and Bengio, Y. A recurrent latent variable model for sequential data. In *Advances in Neural Information Processing Systems*, 2015.

Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.

Gómez-Bombarelli, R., Wei, J. N., Duvenaud, D., Hernández-Lobato, J. M., Sánchez-Lengeling, B., Sheberla, D., Aguilera-Iparraguirre, J., Hirzel, T. D., Adams, R. P., and Aspuru-Guzik, A. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

Hafner, D., Lillicrap, T., Norouzi, M., and Ba, J. Mastering atari with discrete world models. *arXiv preprint arXiv:2010.02193*, 2020.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

Hoffman, M. D. and Johnson, M. J. Elbo surgery: yet another way to carve up the variational evidence lower bound. In *NeurIPS Workshop*, 2016.

Hong, T., Pinson, P., Wang, Y., Weron, R., Yang, D., and Zareipour, H. Energy forecasting: A review and outlook. *IEEE Open Access Journal of Power and Energy*, 7:376–388, 2020.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In *International Conference on Learning Representations*, 2014.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., and Welling, M. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Koochali, A., Schichtel, P., Dengel, A., and Ahmed, S. Random noise vs. state-of-the-art probabilistic forecasting methods: A case study on crps-sum discrimination ability. *Applied Sciences*, 12(10):5104, 2022.

Krishnan, R. G., Shalit, U., and Sontag, D. Structured inference networks for nonlinear state space models. In *AAAI Conference on Artificial Intelligence*, 2017.

Lim, B., Arik, S. Ö., Loeff, N., and Pfister, T. Temporal fusion transformers for interpretable multi-horizon time series forecasting. In *International Journal of Forecasting*, 2021.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Rasul, K., Seward, C., Schuster, I., and Vollgraf, R. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *International Conference on Machine Learning*, 2021.

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., and Ommer, B. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022.

Ronneberger, O., Fischer, P., and Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and*

*computer-assisted intervention*, pp. 234–241. Springer, 2015.

Salinas, D., Flunkert, V., Gasthaus, J., and Januschowski, T. Deepar: Probabilistic forecasting with autoregressive recurrent networks. In *International Journal of Forecasting*, 2020.

Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M. Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. *Applied soft computing*, 90:106181, 2020.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021.

Surendran, S., Fermanian, A., and Corff, S. L. Latent guided sampling for combinatorial optimization. *arXiv preprint arXiv:2506.03672*, 2025.

Tashiro, Y., Song, J., Song, Y., and Ermon, S. Csdi: Conditional score-based diffusion models for probabilistic time series imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.

Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., and Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

# A. Training and Sampling Details

**Algorithm 1** Training Conditional Latent Stochastic Interpolants

**Require:** Dataset $\{(x_{0:T}, c_{0:T})\}$, encoder $f_\omega$, inference model $q_\phi$, velocity field $v_\psi$, decoder $p_\theta$, schedules $\alpha, \beta, \gamma$, weights $\lambda_{\text{interp}}, \lambda_{\text{rec}}, \lambda_{\text{reg}}$, number of latent samples $K$

**while** not converged **do**

Sample $(x_{0:T}, c_{0:T})$ and $t \sim \mathcal{U}\{0, \dots, T-1\}$

Compute recurrent state $h_{t+1} \leftarrow f_\omega(x_{\leq t}, c_{\leq t+1})$

Sample $\{\tilde{z}_{t+1}^{(k)}\}_{k=1}^{K} \sim q_\phi(\cdot \mid x_{t+1}, h_{t+1})$

$k^\star \leftarrow \arg\min_k \left[ -\log p_\theta(x_{t+1} \mid \tilde{z}_{t+1}^{(k)}, h_{t+1}, c_{t+1}) \right]$

$\tilde{z}_{t+1} \leftarrow \tilde{z}_{t+1}^{(k^\star)}$

Sample $s \sim \mathcal{U}([0,1])$, $\xi \sim \mathcal{N}(0, \mathbf{I}_d)$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I}_d)$

Form latent stochastic interpolant and target velocity:

$$z_{t+1}(s) \leftarrow \alpha(s)\varepsilon + \beta(s)\tilde{z}_{t+1} + \gamma(s)\xi, \qquad \dot{z}_{t+1}^\star(s) \leftarrow \alpha'(s)\varepsilon + \beta'(s)\tilde{z}_{t+1} + \gamma'(s)\xi$$

Compute the loss

$$\begin{aligned}\mathcal{L} \leftarrow & -\log p_\theta(x_{t+1} \mid \tilde{z}_{t+1}, h_{t+1}, c_{t+1}) + \lambda_{\text{reg}} \mathrm{KL}(q_\phi(z_{t+1} \mid x_{t+1}, h_{t+1}) \,\|\, p_0(z_{t+1})) \\ & + \lambda_{\text{SI}} \| v_\psi(z_{t+1}(s), s, h_{t+1}) - \dot{z}_{t+1}^\star(s) \|^2\end{aligned}$$

Update $(\omega, \phi, \psi, \theta)$ using $\nabla \mathcal{L}$

**end while**

**Algorithm 2** Conditional Sampling with Euler Discretization

**Require:** History $(x_{\leq t}, c_{\leq t+1})$, encoder $f_\omega$, velocity field $v_\psi$, decoder $p_\theta$, number of Euler steps $N$

Compute recurrent state $h_{t+1} \leftarrow f_\omega(x_{\leq t}, c_{\leq t+1})$

Sample $\varepsilon \sim \mathcal{N}(0, I)$ and set $z_{t+1}^{(0)} \leftarrow \varepsilon$

Set $\Delta s \leftarrow 1/N$

**for** $j = 0, \dots, N-1$ **do**

  $s_j \leftarrow j/N$

  $z_{t+1}^{(j+1)} \leftarrow z_{t+1}^{(j)} + \Delta s \, v_\psi(z_{t+1}^{(j)}, s_j, h_{t+1})$

**end for**

Set $z_{t+1} \leftarrow z_{t+1}^{(N)}$

Sample $\hat{x}_{t+1} \sim p_\theta(\cdot \mid z_{t+1}, h_{t+1}, c_{t+1})$

# B. Proof of Proposition 3.1

*Proof.* Fix $h$ and $s \in [0,1]$. For any measurable velocity field $v$, define

$$v^\star(z(s), s, h) := \mathbb{E}[\dot{z}(s) \mid z(s), h].$$

We decompose

$$v(z(s), s, h) - \dot{z}(s) = \big(v(z(s), s, h) - v^\star(z(s), s, h)\big) + \big(v^\star(z(s), s, h) - \dot{z}(s)\big).$$

Taking the squared norm and conditioning on $h$, we obtain

$$\begin{aligned}\mathbb{E}\big[\|v(z(s), s, h) - \dot{z}(s)\|^2 \mid h\big] = & \, \mathbb{E}\big[\|v(z(s), s, h) - v^\star(z(s), s, h)\|^2 \mid h\big] + \mathbb{E}\big[\|v^\star(z(s), s, h) - \dot{z}(s)\|^2 \mid h\big] \\ & + 2\mathbb{E}[\langle v(z(s), s, h) - v^\star(z(s), s, h), v^\star(z(s), s, h) - \dot{z}(s)\rangle \mid h].\end{aligned}$$

The cross term is zero. Indeed, since $v(z(s), s, h) - v^\star(z(s), s, h)$ is measurable with respect to $z(s)$ and $h$, the tower property gives

$$\begin{aligned}&\mathbb{E}[\langle v(z(s), s, h) - v^\star(z(s), s, h), v^\star(z(s), s, h) - \dot{z}(s)\rangle \mid h]\\ &= \mathbb{E}[\langle v(z(s), s, h) - v^\star(z(s), s, h), \mathbb{E}[v^\star(z(s), s, h) - \dot{z}(s) \mid z(s), h]\rangle \mid h]\,.\end{aligned}$$

By definition of $v^\star$,

$$\mathbb{E}[v^\star(z(s), s, h) - \dot{z}(s) \mid z(s), h] = v^\star(z(s), s, h) - \mathbb{E}[\dot{z}(s) \mid z(s), h] = 0.$$

Therefore,

$$\mathbb{E}\big[\|v(z(s), s, h) - \dot{z}(s)\|^2 \mid h\big] = \mathbb{E}\big[\|v(z(s), s, h) - v^\star(z(s), s, h)\|^2 \mid h\big] + \mathbb{E}\big[\|v^\star(z(s), s, h) - \dot{z}(s)\|^2 \mid h\big]\,. \tag{8}$$

The second term does not depend on $v$, while the first term is nonnegative. Hence, for each fixed $s$, the loss is minimized when $v(z(s), s, h) = v^\star(z(s), s, h)$ almost surely under the conditional law of $z(s)$ given $h$. Equivalently,

$$v^\star(z, s, h) = \mathbb{E}[\dot{z}(s) \mid z(s) = z, h].$$

Integrating (8) over $s \in [0, 1]$ gives

$$\mathcal{L}^h_{\mathrm{SI}}(v) = \mathcal{L}^h_{\mathrm{SI}}(v^\star) + \int_0^1 \mathbb{E}\big[\|v(z(s), s, h) - v^\star(z(s), s, h)\|^2 \mid h\big]\, ds. \tag{9}$$

Thus $v^\star$ is the unique minimizer of $\mathcal{L}^h_{\mathrm{SI}}$ in $L^2$. Finally, since the global objective is obtained by averaging the conditional losses over $h$,

$$\mathcal{L}_{\mathrm{SI}}(v) = \mathbb{E}_h\left[\mathcal{L}^h_{\mathrm{SI}}(v)\right],$$

the same velocity field also minimizes $\mathcal{L}_{\mathrm{SI}}(v)$ for almost every $(z, s, h)$. □

## C. Experiments Details

### C.1. CRPS-sum evaluation metrics

Let $y_t \in \mathbb{R}^d$ denote the ground truth at time $t$, and let $\hat{y}_t^{(m)}$ denote the $m$-th sample from the predictive distribution, with $m = 1, \ldots, M$.

To evaluate the full predictive distribution, we use the Continuous Ranked Probability Score (CRPS). For a univariate variable, the CRPS at time $t$ is defined as:

$$\mathrm{CRPS}(F_t, y_t) = \int_{-\infty}^{+\infty} \left(F_t(z) - \mathbf{1}_{\{y_t \le z\}}\right)^2 dz, \tag{10}$$

where $F_t$ is the predictive cumulative distribution function.

In practice, given samples, CRPS can be estimated as:

$$\mathrm{CRPS}(y_t) = \frac{1}{M}\sum_{m=1}^{M} |\hat{y}_t^{(m)} - y_t| - \frac{1}{2M^2}\sum_{m=1}^{M}\sum_{m'=1}^{M} |\hat{y}_t^{(m)} - \hat{y}_t^{(m')}|. \tag{11}$$

For multivariate time series, we compute CRPS on the aggregated (summed) time series:

$$y_t^{\mathrm{sum}} = \sum_{i=1}^{d} y_{t,i}, \quad \hat{y}_t^{(m),\mathrm{sum}} = \sum_{i=1}^{d} \hat{y}_{t,i}^{(m)}. \tag{12}$$

The final CRPS-sum metric is:

$$\text{CRPS-sum} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{CRPS}\big(y_t^{\mathrm{sum}}\big). \tag{13}$$

### C.2. Datasets

**Exchange-Rate.** This dataset consists of daily exchange rates of multiple currencies. It is relatively low-dimensional but exhibits complex temporal dependencies and non-stationary behavior, making it a standard benchmark for evaluating forecasting models.

**Electricity.** The electricity dataset contains hourly electricity consumption measurements from $d = 370$ clients. It is a high-dimensional dataset with strong seasonal patterns (daily and weekly) and significant cross-series dependencies.

**Solar.** The solar dataset includes solar power production measurements from $d = 137$ photovoltaic plants. It exhibits strong periodic patterns as well as high variability due to weather conditions, making probabilistic modeling essential.

**Traffic.** The traffic dataset contains road occupancy rates collected from $d = 963$ sensors in a transportation network. It is highly multivariate, with strong spatial and temporal correlations across variables.

### C.3. Models

**Baselines** All these methods can be interpreted within a unified framework where a neural network is trained to approximate a time-dependent field. More specifically, each model learns a function that takes as input a state vector $x_t$ (living in the same space as the data), a time variable $t$, and a conditioning embedding $c$, and outputs a quantity defining the generative dynamics.

The nature of this learned field depends on the model: in DDPM, the network predicts the noise term added to the data; in SGM, it estimates the score function $\nabla \log p_t(x)$; in FM, it models a vector field defining a probability flow; in SI, it jointly models the velocity field and the score function associated with the interpolant dynamics. In our experiments we use a $\gamma = \sqrt{2s(1-s)}$ and a trigonometric interpolant as proposed by (Chen et al., 2024a)

To ensure a fair comparison, we use the same neural network architecture across all models. Differences in performance therefore stem solely from the underlying training objectives and generative formulations, rather than from architectural choices.

In the time series forecasting experiments, the history encoder is implemented as a single-layer RNN with a latent dimensionality of 128. The field function is parameterized using a U-Net-like architecture (Ronneberger et al., 2015) composed of 8 residual blocks, each with a hidden size of 64.

To further improve training stability, we adopt paired sampling when training the stochastic interpolant models.

The baseline models are trained with 200 epochs and 64 batch sizes with a learning rate $10^{-3}$ except the SI model which is trained with 100 epochs and 128 batch sizes with a learning rate $10^{-4}$.

**Our approach** We adapt the neural network architecture and the hyperparameters to the dataset. For all datasets, we use a $\gamma = \sqrt{2s(1-s)}$ and a trigonometric, 100 epochs, batch size of 128, learning rate of $10^{-4}$. The differences between datasets can be found in Table 2.

| Hyperparameters | Exchange Rate | Electricity | Solar | Traffic |
|---|---|---|---|---|
| hidden size | 128 | 2048 | 256 | 512 |
| VAE | CNN | MLP | MLP | CNN |
| #layers VAE | 2 | 16 | 16 | 4 |
| latent size | 16 | 128 | 16 | 16 |

*Table 2.* Hyperparameters