Workshop at the 7th Symposium on Advances in Approximate Bayesian Inference (non-archival), 2025 1–19

Inference-Time Prior Adaptation in Simulation-Based Inference via Guided Diffusion Models

Paul E. Chang* PAUL.CHANG@HELSINKI.FI
University of Helsinki

Severi Rissanen SEVERI.RISSANEN@AALTO.FI
Aalto University

Nasrulloh Loka NASRULLOH.LOKA@HELSINKI.FI
University of Helsinki

Daolang Huang DAOLANG.HUANG@AALTO.FI
Aalto University

Luigi Acerbi LUIGI.ACERBI@HELSINKI.FI
University of Helsinki

Abstract

Amortized simulator-based inference has emerged as a powerful framework for tackling inverse problems and Bayesian inference in many computational sciences by learning the reverse mapping from observed data to parameters. Once trained on many simulated parameter-data pairs, these methods afford parameter inference for any particular dataset, yielding high-quality posterior samples with only one or a few forward passes of a neural network. While amortized methods offer significant advantages in terms of efficiency and reusability across datasets, they are typically constrained by their training conditions – particularly the prior distribution of parameters used during training. In this paper, we introduce *PriorGuide*, a technique that enables on-the-fly adaptation to arbitrary priors at inference time for diffusion-based amortized inference methods. Our method allows users to incorporate new information or expert knowledge at runtime without costly retraining.

1. Introduction

Simulation-based inference has become a fundamental tool across computational sciences, enabling parameter estimation in complex systems where the forward model (simulator) is available but its likelihood is intractable (Cranmer et al., 2020). In a Bayesian framework, we express prior beliefs about parameters as distributions and update them given observations (Robert, 2007). While traditional inference methods such as Markov Chain Monte Carlo (MCMC) are the gold standard with tractable likelihoods (Gelman et al., 2014), recent neural network approaches can directly learn the inverse mapping from observations to posterior distributions over model parameters (Greenberg et al., 2019; Radev et al., 2020). These methods are typically *amortized*, enabling efficient inference after training and facilitating meta-learning across related problems (Brown et al., 2020). In this context, 'inference' takes on a unified meaning: the neural network's forward pass directly produces a posterior estimate.

* Corresponding author. This is a workshop paper and preliminary work.

Modern generative modeling techniques such as transformers (Vaswani et al., 2017), flow-matching (Lipman et al., 2023), and diffusion models (Ho et al., 2020; Song et al., 2021) have proven particularly effective for this inverse modeling task, with recent work demonstrating state-of-the-art performance in simulation-based inference (Wildberger et al., 2024; Gloeckler et al., 2024; Chang et al., 2024). These methods learn the inverse mapping by generating training data – (model parameters, data) pairs – through simulation, typically using a uniform training distribution over parameters, equivalent to the prior, to ensure broad coverage of the parameter space.

However, this approach faces key limitations in practice. First, practitioners often possess domain-specific knowledge that could improve inference if incorporated as prior beliefs. Second, researchers may need to conduct prior sensitivity analysis to understand how their modeling assumptions affect conclusions (Elsemüller et al., 2024). Current methods either require retraining with new priors or offer only limited solutions. As the field moves toward larger foundation models for amortized inference (Hollmann et al., 2025), retraining becomes increasingly impractical.

While recent work has proposed techniques for prior specification at inference time (Elsemüller et al., 2024; Chang et al., 2024; Whittle et al., 2025), these amortized approaches are restricted to specific family of priors considered during training – from factorized histograms to Gaussian mixture models. While some of these families are very flexible in principle, training over the space of all meaningful runtime priors becomes rapidly infeasible. Diffusion interval guidance offers runtime prior specification, but limited to simple range constraints (Gloeckler et al., 2024). A general solution for incorporating arbitrary priors at runtime remains an open challenge.

Contributions. We introduce *PriorGuide*, a method that enables flexible incorporation of arbitrary prior beliefs at inference time for diffusion-based amortized inference models. Our approach requires no modifications to the base diffusion model's training procedure and supports more complex priors than previously explored methods. Our method works with existing diffusion-based inference models by implementing the prior as a guidance term. We demonstrate PriorGuide's effectiveness on synthetic examples and a challenging inverse problem. See Fig. A.1 for an illustration of our method.

2. Background

Diffusion models are a powerful framework for generative modeling that transforms samples from arbitrary to simple distributions and vice versa through a gradual noising and denoising process (Sohl-Dickstein et al., 2015). In the forward process, starting from a distribution $p(\boldsymbol{\theta}_0)$, Gaussian noise is progressively added to the samples until, at the end of the process ($t = 1$), the distribution converges to a simple terminal distribution (typically Gaussian). The forward process can be described as:

$$p(\boldsymbol{\theta}_t) = \int \mathcal{N}(\boldsymbol{\theta}_t | \boldsymbol{\theta}_0, \sigma(t)^2 \mathbf{I}) p(\boldsymbol{\theta}_0) \mathrm{d}\boldsymbol{\theta}_0, \tag{1}$$

where $\sigma(t)$ defines the noise variance schedule as a function of time (typically increasing with t), and $\boldsymbol{\theta}_t$ represents the noisy samples at time t. The corresponding reverse process reconstructs the original sample distribution from noise, and can be formulated as either a

stochastic differential equation (SDE) or an ordinary differential equation (ODE). For the Variance Exploding (VE) SDE (Song et al., 2021; Karras et al., 2022), the reverse process takes the form:

$$\text{Reverse SDE:} \quad \mathrm{d}\boldsymbol{\theta}_t = -2\dot{\sigma}(t)\sigma(t)\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t)\mathrm{d}t + \sqrt{2\dot{\sigma}(t)\sigma(t)}\,\mathrm{d}\omega_t, \tag{2}$$

where $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t)$ is the *score function* (gradient of the log-density), $\mathrm{d}\omega_t$ is a Wiener process representing Brownian motion (noise), and $\dot{\sigma}(t)$ is the time derivative of the variance schedule.

Learning the Score Function. The score function $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}_t)$ can be approximated using a neural network $s(\boldsymbol{\theta}_t, t)$, trained to minimize the denoising score matching loss (Hyvärinen and Dayan, 2005; Vincent, 2011; Song et al., 2021):

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{t\sim\mathcal{U}(0,1)}\mathbb{E}_{\boldsymbol{\theta}_0\sim p(\boldsymbol{\theta}_0)}\mathbb{E}_{\boldsymbol{\theta}_t\sim\mathcal{N}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0,\sigma(t)^2\mathbf{I})} \left\| s(\boldsymbol{\theta}_t, t) - \nabla_{\boldsymbol{\theta}_t} \log p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0) \right\|_2^2. \tag{3}$$

Once trained, the network $s(\boldsymbol{\theta}_t, t)$ approximates the gradient of the log-probability density of noised distributions and affords sampling through the reverse SDE (Eq. (2)). Starting from a sample $\boldsymbol{\theta}_t \sim \mathcal{N}(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0, \sigma_{\max}^2\mathbf{I})$ for $t = 1$ with sufficiently large $\sigma_{\max}$, integrating the reverse process backward in time approximately reconstructs the original distribution $p(\boldsymbol{\theta}_0)$.

Tweedie's Formula. Tweedie's formula provides a key connection between the posterior mean of $\boldsymbol{\theta}_0$ given $\boldsymbol{\theta}_t$ and the score function:

$$\mathbb{E}[\boldsymbol{\theta}_0|\boldsymbol{\theta}_t] = \mu_{0|t}(\boldsymbol{\theta}_t) = \boldsymbol{\theta}_t + \sigma(t)^2\nabla_{\boldsymbol{\theta}_t} \log p(\boldsymbol{\theta}_t). \tag{4}$$

This relationship enables direct estimation of the posterior mean at any noise level and establishes an equivalence between $\mu_{0|t}(\boldsymbol{\theta}_t)$ and $s(\boldsymbol{\theta}_t, t)$.

The diffusion framework's flexibility stems largely from its ability to incorporate guidance mechanisms, which afford steering the sampling process toward desired outcomes by including additional information or constraints. Notable examples include classifier guidance (Dhariwal and Nichol, 2021) and classifier-free guidance (Ho and Salimans, 2022), which afford controlled generation without retraining the model. For inverse problems, this guidance framework has been extended to incorporate likelihood information, particularly for Gaussian likelihoods (Chung et al., 2023; Song et al., 2023a).

For the inverse problems in this work, we learn a score function to approximate the conditional mapping $\nabla_{\boldsymbol{\theta}_t} \log p(\boldsymbol{\theta}_t|\mathbf{x})$ using the direct conditional training approach of Gloeckler et al. (2024). In this framework, the observation $\mathbf{x}$ is provided directly to the score network $s(\boldsymbol{\theta}_t, t, \mathbf{x})$, similar to the context in conditional neural processes (Garnelo et al., 2018). While our experiments in this paper use this direct approach, we note PriorGuide can also be applied to models using joint training with in-painting guidance (Lugmayr et al., 2022). In either case, PriorGuide adapts the guidance framework to transform the trained prior into an arbitrary prior at inference time.

3. PriorGuide

Consider an inverse problem where we observe data $\mathbf{x}$ and aim to infer parameters $\boldsymbol{\theta}$. Standard diffusion models for inverse problems are trained to approximate $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}|\mathbf{x})$ via a learned score function $s(\boldsymbol{\theta}_t, t, \mathbf{x})$, and sampling from the model produces posterior samples $p(\boldsymbol{\theta}|\mathbf{x})$

that are anchored to the training distribution (prior) $p(\boldsymbol{\theta})$. This constraint limits flexibility when new prior information becomes available, as incorporating it would traditionally require retraining the score model.

Given a diffusion model trained to sample from posterior $p(\boldsymbol{\theta}|\mathbf{x})$ with prior $p(\boldsymbol{\theta})$, our goal is to sample from a modified posterior $q(\boldsymbol{\theta}|\mathbf{x})$ that incorporates a new prior $q(\boldsymbol{\theta})$ without retraining. PriorGuide affords prior modification at sampling time by leveraging a basic statistical relationship:

Proposition 1 *Let the posterior under the original prior be given as $p(\boldsymbol{\theta}|\mathbf{x}) \propto p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})$, and let the posterior under the new prior be $q(\boldsymbol{\theta}|\mathbf{x}) \propto q(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta})$. Then, sampling from $q(\boldsymbol{\theta}|\mathbf{x})$ is equivalent to sampling from $\rho(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x})$ with $\rho(\boldsymbol{\theta}) \equiv \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}$ the new-over-old prior ratio.*

Proof We can rewrite the new posterior $q(\boldsymbol{\theta}|\mathbf{x})$ as

$$q(\boldsymbol{\theta}|\mathbf{x}) \propto q(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\, p(\boldsymbol{\theta})p(\mathbf{x}|\boldsymbol{\theta}) \propto \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}\, p(\boldsymbol{\theta}|\mathbf{x}) = \rho(\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathbf{x}),$$

where the prior ratio $\rho(\boldsymbol{\theta}) \equiv \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}$ takes the role of an importance weighing function. ■

Modified Posterior Score. Prop. 1, combined with the properties of diffusion models, allows us to express the score of the modified posterior at any time t as:

$$q(\boldsymbol{\theta}_t|\mathbf{x}) \propto \int \rho(\boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0|\mathbf{x})p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0)\mathrm{d}\boldsymbol{\theta}_0 \tag{5}$$

$$\nabla_{\boldsymbol{\theta}_t} \log q(\boldsymbol{\theta}_t|\mathbf{x}) = \nabla_{\boldsymbol{\theta}_t} \log \int \rho(\boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0|\mathbf{x})p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0, \mathbf{x})\mathrm{d}\boldsymbol{\theta}_0 \tag{6}$$

$$= \nabla_{\boldsymbol{\theta}_t} \log \int \rho(\boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t, \mathbf{x})p(\boldsymbol{\theta}_t|\mathbf{x})\mathrm{d}\boldsymbol{\theta}_0 \tag{7}$$

$$= \nabla_{\boldsymbol{\theta}_t} \log \int \rho(\boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t, \mathbf{x})\mathrm{d}\boldsymbol{\theta}_0 + \nabla_{\boldsymbol{\theta}_t} \log p(\boldsymbol{\theta}_t|\mathbf{x}) \tag{8}$$

where in Eq. (5) we write the modified posterior as an integral over $\boldsymbol{\theta}_0$ by noting that $q(\boldsymbol{\theta}_0|\mathbf{x}) \propto \rho(\boldsymbol{\theta}_0)p(\boldsymbol{\theta}_0|\mathbf{x})$ and then propagate this information to time t via the transition kernel $p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0)$. In Eq. (6) we write the score, and then re-express the joint probability $p(\boldsymbol{\theta}_0|\mathbf{x})p(\boldsymbol{\theta}_t|\boldsymbol{\theta}_0) = p(\boldsymbol{\theta}_0, \boldsymbol{\theta}_t|\mathbf{x})$ as $p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t, \mathbf{x})p(\boldsymbol{\theta}_t|\mathbf{x})$, which allows us to separate the contribution of the new prior guidance from the original score model $s(\boldsymbol{\theta}_t, t, \mathbf{x})$. In multiple steps we exploit the fact that multiplicative constants inside the integral disappear under the score.

We can draw samples from $q(\boldsymbol{\theta}_t|\mathbf{x})$ via the reverse diffusion process using the modified score:

$$\nabla_{\boldsymbol{\theta}_t} \log q(\boldsymbol{\theta}_t|\mathbf{x}) \approx \nabla_{\boldsymbol{\theta}_t} \log \mathbb{E}_{p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t, \mathbf{x})} \left[\rho(\boldsymbol{\theta}_0)\right] + s(\boldsymbol{\theta}_t, t, \mathbf{x}). \tag{9}$$

where first term estimates how the new prior's influence propagates to time t (guidance term) and the second term is our trained score model. This is a common way to implement a guidance function (Chung et al., 2023; Song et al., 2023a,b; Rissanen et al., 2024), where now the guidance function is the prior ratio. In the rest of this section, we apply several approximation techniques to estimate the guidance term.

3.1. Approximating the Guidance Function

To approximate the guidance term in Eq. (9) efficiently while maintaining flexible inference-time priors, we introduce two approximations. Following recent work (Song et al., 2023a; Peng et al., 2024; Rissanen et al., 2024), we first model the reverse transition kernel as a Gaussian distribution. We then introduce a novel approach that represents $\rho(\boldsymbol{\theta})$ as a Gaussian mixture model. This representation enables both an analytical solution and preserves flexibility in the model. While previous research on inverse problems has explored guidance with linear-Gaussian observation models (Song et al., 2023a), these can be viewed as special cases of our method when using a single mixture component.

Reverse Transition Kernel Approximation. We first approximate the reverse transition kernel $p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t)$ as a Gaussian distribution centered at $\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t)$, obtained from the score function via Tweedie's formula, Eq. (4). This approximation is common in the guidance literature (Chung et al., 2023; Song et al., 2023a; Peng et al., 2024; Rissanen et al., 2024; Finzi et al., 2023; Bao et al., 2022). For the covariance matrix $\boldsymbol{\Sigma}_{0|t}$, we adopt a simple yet effective approximation inspired by Song et al. (2023a); Ho et al. (2022):

$$\boldsymbol{\Sigma}_{0|t} = \frac{\sigma(t)^2}{1+\sigma(t)^2}\mathbf{I}. \tag{10}$$

This approximation acts as a time-dependent scaling factor that naturally aligns with the diffusion process – starting at the identity matrix when $t = 1$ and approaching zero as $t \to 0$, effectively increasing the precision of our prior guidance at smaller timesteps.

Prior Ratio Approximation. We then approximate the prior ratio function $\rho(\boldsymbol{\theta}) = \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta})}$ as a generalized mixture of Gaussians:

$$\rho(\boldsymbol{\theta}) \approx \sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{\theta}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \qquad \rho(\boldsymbol{\theta}) \geq 0, \tag{11}$$

where $\{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\}_{i=1}^K$ represent the weights, means and covariance matrices of the mixture. Since this represents a ratio rather than a distribution, the mixture weights need not be positive nor sum to one, as long as the ratio remains non-negative, potentially enabling more expressive approximations such as subtractive mixtures (Loconte et al., 2024). Notably, when $p(\boldsymbol{\theta})$ is uniform (as in our experiments), $\rho(\boldsymbol{\theta})$ reduce to $q(\boldsymbol{\theta})$, and we directly specify it as a Gaussian mixture. For non-uniform training distributions, the ratio function can be fit with a generalized Gaussian mixture approximation, which can theoretically approximate any continuous function (Sorenson and Alspach, 1971).

Guidance Term. With these Gaussian approximations, the guidance term becomes:

$$\nabla_{\boldsymbol{\theta}_t} \log \mathbb{E}_{p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t,\mathbf{x})}\left[\rho(\boldsymbol{\theta}_0)\right] \approx \nabla_{\boldsymbol{\theta}_t} \log \int \sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \boldsymbol{\Sigma}_{0|t})\mathrm{d}\boldsymbol{\theta}_0. \tag{12}$$

This integral can be solved analytically (full derivation in Appendix A.2), yielding:

$$\nabla_{\boldsymbol{\theta}_t} \log \mathbb{E}_{p(\boldsymbol{\theta}_0|\boldsymbol{\theta}_t,\mathbf{x})}[\rho(\boldsymbol{\theta}_0)] \approx \frac{\sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)(\boldsymbol{\mu}_i - \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t))^{\mathbf{T}} \widetilde{\boldsymbol{\Sigma}}_i^{-1} \nabla_{\boldsymbol{\theta}_t} \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t)}{\sum_{i=1}^{K} w_i \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)} \tag{13}$$

$$= \sum_{i}^{K} \tilde{w}_i (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t))^{\mathbf{T}} \widetilde{\boldsymbol{\Sigma}}_i^{-1} \nabla_{\boldsymbol{\theta}_t} \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \tag{14}$$

where $\widetilde{\boldsymbol{\Sigma}}_i = \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_{0|t}$ and $\tilde{w}_i = w_i \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i) / \sum_{j=1}^{K} w_j \mathcal{N}(\boldsymbol{\mu}_j|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_j)$. For typical inverse problems where parameter dimensionality is below 100, these calculations remain computationally tractable. However, higher-dimensional problems would require additional approximations, particularly for the log determinant and matrix inversion.

Finally, the PriorGuide update to the mean of the reverse kernel can be expressed concisely using Tweedie's formula, Eq. (4), and our derived guidance term, Eq. (14):

$$\mu_{0|t}^{\text{new}}(\boldsymbol{\theta}_t) = \mu_{0|t}(\boldsymbol{\theta}_t) + \sigma(t)^2 \sum_{i}^{K} \tilde{w}_i (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t))^{\mathbf{T}} \widetilde{\boldsymbol{\Sigma}}_i^{-1} \nabla_{\boldsymbol{\theta}_t} \boldsymbol{\mu}_{0|t}. \tag{15}$$

This update intuitively combines the original prediction $\mu_{0|t}(\boldsymbol{\theta}_t)$ with a weighted sum of correction terms from our new prior. The correction magnitude is controlled by both the noise schedule $\sigma(t)^2$ and the distance between the mixture components and current prediction.

4. Experiments

We evaluate PriorGuide using the base model from Simformer (Gloeckler et al., 2024), trained with the variance exploding SDE (Song et al., 2021). Notably, our method requires no modifications to the original diffusion model's training procedure and works by adjusting the guidance term at inference time as described in Section 3.

Two Moons with Correlated Prior. The two moons example is a common benchmark for simulation-based inference. Here, we add a strong correlated prior $q(\boldsymbol{\theta})$ to test how our method handles a multi-modal scenario (Fig. A.2). PriorGuide correctly captures the multimodality of the problem through its posterior distribution. For validation, we compare PriorGuide's results with a ground truth baseline obtained by retraining the base model with $q(\boldsymbol{\theta})$. For quantitative validation, we compared samples from PriorGuide and the retrained model across 10 different observations $\mathbf{x}$ using the Classifier 2-Sample Tests (C2ST) score (Lopez-Paz and Oquab, 2017). The C2ST score measures how well a classifier can distinguish between two sets of samples, with 0.5 indicating indistinguishable samples. Between the retrained model and PriorGuide samples, we obtain a score of 0.623 ± 0.044. For context, the score between the base diffusion model and standard MCMC samples is 0.523 ± 0.016, demonstrating that PriorGuide generates comparable samples without requiring retraining. See Appendix A.3 for model details.

Benchmark SBI Tasks. Finally, we evaluate PriorGuide on two simulation-based inference tasks of increasing complexity: the Ornstein-Uhlenbeck Process (OUP), a time-series model

		Uniform $\boldsymbol{\theta}$ sampling				Mixture $\boldsymbol{\theta}$ sampling	
		Simformer	ACE	ACEP	PriorGuide	Simformer	PriorGuide
OUP	RMSE (↓)	0.61(0.03)	0.59(0.00)	0.21(0.02)	**0.17**(0.01)	0.51(0.04)	**0.40**(0.02)
	MMD (↓)	0.19(0.01)	0.15(0.00)	0.04(0.00)	**0.03**(0.00)	0.16(0.02)	**0.09**(0.01)
Turin	RMSE (↓)	0.25(0.00)	0.25(0.00)	0.10(0.01)	**0.07**(0.00)	0.26(0.00)	**0.18**(0.00)
	MMD (↓)	0.11(0.00)	0.11(0.00)	0.02(0.00)	**0.01**(0.00)	0.08(0.00)	**0.04**(0.00)

Table 1: Comparison of SBI task metrics for $\boldsymbol{\theta}$ prediction; mean (standard deviation) over 5 runs. Best results are bolded. *Left*: Uniform sampling distribution for $\boldsymbol{\theta}$, with an informative Gaussian prior given to ACEP and PriorGuide. *Right*: Correlated mixture sampling distribution, with the same distribution given as prior to PriorGuide.

with two latent variables (Uhlenbeck and Ornstein, 1930), and the Turin model (Turin et al., 1972), a radio propagation simulator with four parameters that generates 101-dimensional signal data.

For both tasks, we set the sampling distribution of $\boldsymbol{\theta}$ in two ways: (i) as a uniform distribution and (ii) as a correlated Gaussian mixture distribution. We can then test the ability of a model of incorporating prior information by passing useful information about the sampled $\boldsymbol{\theta}$. In the uniform case, we provide information by sampling the prior location from a Gaussian around the true $\boldsymbol{\theta}$, and giving that Gaussian prior to models that support runtime priors, following Chang et al., 2024. In the correlated Gaussian mixture case, we pass a prior that exactly matches the true inference-time sampling distribution. Further experimental details are provided in Appendix A.3.

As a baseline, we compare our method, PriorGuide, with the same base SimFormer model *without* prior guidance (Gloeckler et al., 2024). We also consider another amortized inference method, the Amortized Conditioning Engine (ACE; Chang et al., 2024), whose ACEP variant affords runtime incorporation of *factorized* priors seen during training. Table 1 presents the benchmark results. In the uniform $\boldsymbol{\theta}$ case, we compare PriorGuide with an informative Gaussian prior against Simformer and ACE (both without priors), and ACE with the same simple prior (ACEP). In the mixture sampling case, we compare base SimFormer with PriorGuide guided by the sampling distribution as prior.[1] PriorGuide outperforms all baselines in both settings, demonstrating its capabilities of incorporating prior information at test time without retraining. Example visualizations of results on the SBI experiments are presented in Appendix A.4.

5. Discussion

In this work, we introduced PriorGuide, a technique that enables the use of flexible, user-defined priors at inference time for diffusion-based amortized inference methods. Our experiments demonstrate that PriorGuide can effectively recover posterior distributions under new priors. This capability is particularly valuable in scientific applications where prior knowledge is often refined post-training, for prior sensitivity analysis or with large inference models, where retraining is undesirable.

1. ACEP does not afford complex correlated priors, so it is not included.

Acknowledgments

PC, DH, and LA were supported by the Research Council of Finland (Flagship programme: Finnish Center for Artificial Intelligence FCAI). NL was funded by Business Finland (project 3576/31/2023). LA was also supported by Research Council of Finland grants 358980 and 356498. SR was supported by Research Council of Finland grant 334600. The authors wish to thank the Finnish Computing Competence Infrastructure (FCCI), Aalto Science-IT project, and CSC–IT Center for Science, Finland, for the computational and data storage resources provided, including access to the LUMI supercomputer, owned by the EuroHPC Joint Undertaking, hosted by CSC (Finland) and the LUMI consortium (LUMI project 462000551).

References

Fan Bao, Chongxuan Li, Jun Zhu, and Bo Zhang. Analytic-DPM: an analytic estimate of the optimal reverse variance in diffusion probabilistic models. In *International Conference on Learning Representations*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Paul E. Chang, Nasrulloh Loka, Daolang Huang, Ulpu Remes, Samuel Kaski, and Luigi Acerbi. Amortized probabilistic conditioning for optimization, simulation and inference, 2024.

Hyungjin Chung, Jeongsol Kim, Michael T Mccann, Marc L Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations, ICLR 2023*. The International Conference on Learning Representations, 2023.

Kyle Cranmer, Johann Brehmer, and Gilles Louppe. The frontier of simulation-based inference. *Proceedings of the National Academy of Sciences*, 117(48):30055–30062, 2020.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Lasse Elsemüller, Hans Olischläger, Marvin Schmitt, Paul-Christian Bürkner, Ullrich Köthe, and Stefan T. Radev. Sensitivity-aware amortized Bayesian inference. *Transactions on Machine Learning Research*, 2024.

Marc Anton Finzi, Anudhyan Boral, Andrew Gordon Wilson, Fei Sha, and Leonardo Zepeda-Núñez. User-defined event sampling and uncertainty quantification in diffusion models for physical dynamical systems. In *International Conference on Machine Learning*, pages 10136–10152. PMLR, 2023.

Marta Garnelo, Dan Rosenbaum, Chris J Maddison, Tiago Ramalho, David Saxton, Murray Shanahan, Yee Whye Teh, Danilo J Rezende, and SM Ali Eslami. Conditional neural processes. In *International Conference on Machine Learning*, pages 1704–1713, 2018.

Andrew Gelman, John B Carlin, Hal S Stern, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*, volume 3nd edition. Chapman and Hall/CRC, 2014.

Manuel Gloeckler, Michael Deistler, Christian Weilbach, Frank Wood, and Jakob H Macke. All-in-one simulation-based inference. In *International Conference on Machine Learning*. PMLR, 2024.

David Greenberg, Marcel Nonnenmacher, and Jakob Macke. Automatic posterior transformation for likelihood-free inference. In *International Conference on Machine Learning*, pages 2404–2414. PMLR, 2019.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications.*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851, 2020.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, volume 35, pages 18954–18967. Curran Associates, Inc., 2022.

Noah Hollmann, Samuel Müller, Lennart Purucker, Arjun Krishnakumar, Max Körfer, Shi Bin Hoo, Robin Tibor Schirrmeister, and Frank Hutter. Accurate predictions on small data with a tabular foundation model. *Nature*, 637(8045):319–326, 2025.

Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, pages 26565–26577, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *Proceedings of the Eleventh International Conference on Learning Representations*. ICLR, May 2023.

Lorenzo Loconte, Aleksanteri M. Sladek, Stefan Mengel, Martin Trapp, Arno Solin, Nicolas Gillis, and Antonio Vergari. Subtractive mixture models via squaring: Representation and learning. In *International Conference on Learning Representations (ICLR)*, 2024.

David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. In *International Conference on Learning Representations*, 2017.

Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Luc Van Gool, and Radu Timofte. Repaint: Inpainting using denoising diffusion probabilistic models. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Troels Pedersen. Stochastic multipath model for the in-room radio channel based on room electromagnetics. *IEEE Transactions on Antennas and Propagation*, 67(4):2591–2603, 2019.

Xinyu Peng, Ziyang Zheng, Wenrui Dai, Nuoqian Xiao, Chenglin Li, Junni Zou, and Hongkai Xiong. Improving diffusion models for inverse problems using optimal posterior covariance. In *International Conference on Learning Represntations*, 2024.

Stefan T Radev, Ulf K Mertens, Andreas Voss, Lynton Ardizzone, and Ullrich Köthe. Bayesflow: Learning complex stochastic models with invertible neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 33(4):1452–1466, 2020.

Severi Rissanen, Markus Heinonen, and Arno Solin. Free hunch: Denoiser covariance estimation for diffusion models without extra costs, 2024.

Christian P Robert. *The Bayesian choice: from decision-theoretic foundations to computational implementation*, volume 2nd edition. Springer, 2007.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015.

Jiaming Song, Arash Vahdat, Morteza Mardani, and Jan Kautz. Pseudoinverse-guided diffusion models for inverse problems. In *International Conference on Learning Representations*, 2023a.

Jiaming Song, Qinsheng Zhang, Hongxu Yin, Morteza Mardani, Ming-Yu Liu, Jan Kautz, Yongxin Chen, and Arash Vahdat. Loss-guided diffusion models for plug-and-play controllable generation. In *International Conference on Machine Learning*, pages 32483–32498. PMLR, 2023b.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*. ICLR, May 2021.

H.W. Sorenson and D.L. Alspach. Recursive Bayesian estimation using gaussian sums. *Automatica*, 7(4):465–479, 1971. ISSN 0005-1098.

George L Turin, Fred D Clapp, Tom L Johnston, Stephen B Fine, and Dan Lavry. A statistical model of urban multipath propagation. *IEEE Transactions on Vehicular Technology*, 21 (1):1–9, 1972.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

George Whittle, Juliusz Ziomek, Jacob Rawling, and Michael A Osborne. Distribution transformers: Fast approximate Bayesian inference with on-the-fly prior adaptation. *arXiv preprint arXiv:2502.02463*, 2025.

Jonas Wildberger, Maximilian Dax, Simon Buchholz, Stephen Green, Jakob H Macke, and Bernhard Schölkopf. Flow matching for scalable simulation-based inference. *Advances in Neural Information Processing Systems*, 36, 2024.

Appendix A. Appendix

A.1. Related Work

PriorGuide builds on advances in three key areas: diffusion models for inverse problems, simulation-based inference (SBI), and guidance techniques for controllable generation. Recent work has adapted diffusion models to scientific applications with intractable forward models, treating inverse problems as conditional generation (Chung et al., 2023). Methods like those in Gloeckler et al. (2024) train diffusion models to directly approximate the posterior. However, these approaches fix the prior during training, limiting their flexibility. Recent work in Elsemüller et al. (2024); Chang et al. (2024); Whittle et al. (2025) showed the effectiveness of inference time priors, but the approach is limited. In inverse problems, reconstruction guidance (Chung et al., 2023) incorporates likelihood gradients during sampling. Related approaches from Rissanen et al. (2024); Finzi et al. (2023); Bao et al. (2022); Peng et al. (2024) use Tweedie's formula to guide sampling, but focus on refining the likelihood term rather than modifying the prior. PriorGuide uniquely repurposes guidance mechanisms to inject new prior information, combining the flexibility of score-based methods with the expressiveness of Gaussian mixture priors.

Limitations. While PriorGuide offers significant flexibility, it has several important limitations: First, the computational cost scales with parameter dimensionality due to the weighted averaging over Gaussian components. Very high-dimensional problems may require additional approximations to maintain efficiency. Furthermore, PriorGuide assumes the new prior ratio can be well-approximated by a Gaussian mixture. While highly expressive, this may not capture all possible prior distributions, particularly those with heavy tails or discrete components. Future work could develop automatic conversion of arbitrary priors into approximate Gaussian mixtures. Additionally, integrating PriorGuide with in-painting style guidance techniques could enhance its applicability to a wider range of inverse problems by removing the need to specify conditioning variables upfront, offering further flexibility.

A.2. Gaussian Integration

Here is the detailed derivation for Eq. (14) from the main text:

$$\nabla_{\boldsymbol{\theta}_t} \log \mathbb{E}\left[\boldsymbol{\rho}(\boldsymbol{\theta}_0)\right] \approx \nabla_{\boldsymbol{\theta}_t} \log \int \sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)\mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \boldsymbol{\Sigma}_{0|t})d\boldsymbol{\theta}_0, \tag{A.1}$$

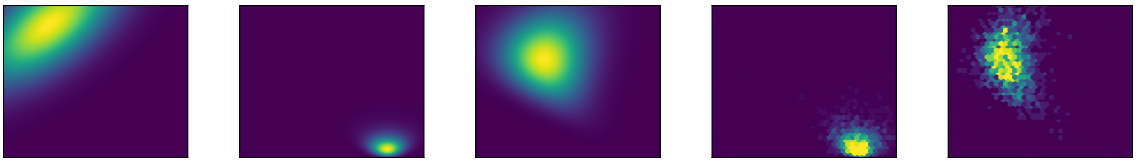
$$= \nabla_{\boldsymbol{\theta}_t} \log \sum_{i=1}^{K} \int \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{\theta}_0, \boldsymbol{\Sigma}_i)\mathcal{N}(\boldsymbol{\theta}_0|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \boldsymbol{\Sigma}_{0|t})d\boldsymbol{\theta}_0. \tag{A.2}$$

The step above uses the symmetry property of Gaussian distributions: if $\mathbf{a} \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ then $\boldsymbol{\mu} \sim \mathcal{N}(\mathbf{a}, \boldsymbol{\Sigma})$. This allows us to swap $\boldsymbol{\theta}_0$ and $\boldsymbol{\mu}_i$ in the first Gaussian. Furthermore,

$$= \nabla_{\boldsymbol{\theta}_t} \log \sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i|\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_{0|t}), \tag{A.3}$$

using the standard result for the convolution of two Gaussian distributions:

$$\int \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)\mathcal{N}(\boldsymbol{\mu}_1|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)d\boldsymbol{\mu}_1 = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2) \tag{A.4}$$



(a) Prior **(b)** Likelihood **(c)** Posterior **(d)** Unguided **(e)** PriorGuide

Figure A.1: Posterior inference with and without PriorGuide. The plots show the mean μ and standard deviation σ parameters of a Gaussian toy model. Prior (a) and likelihood (b) from some observations $\mathbf{x}$ (not shown) yield Bayesian posterior (c). A standard diffusion model trained on a uniform distribution over μ, σ (no prior) matches the likelihood (d). PriorGuide can implement the specified prior at runtime, matching the Bayesian posterior (e).

For notational convenience, we define $\widetilde{\boldsymbol{\Sigma}}_i = \boldsymbol{\Sigma}_i + \boldsymbol{\Sigma}_{0|t}$ continuing with the derivation:

$$= \nabla_{\boldsymbol{\theta}_t} \log \sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i), \tag{A.5}$$

$$= \frac{\nabla_{\boldsymbol{\theta}_t} \sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)}{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)} \quad \text{(chain rule)}, \tag{A.6}$$

$$= \frac{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i) \nabla_{\boldsymbol{\theta}_t} \log \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)}{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)} \quad \text{(since } \nabla f = f \nabla \log f\text{)}, \tag{A.7}$$

$$= \frac{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i) \nabla_{\boldsymbol{\theta}_t} \left(-\frac{1}{2} (\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t) - \boldsymbol{\mu}_i)^\top \widetilde{\boldsymbol{\Sigma}}_i^{-1} (\boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t) - \boldsymbol{\mu}_i) \right)}{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i),} \tag{A.8}$$

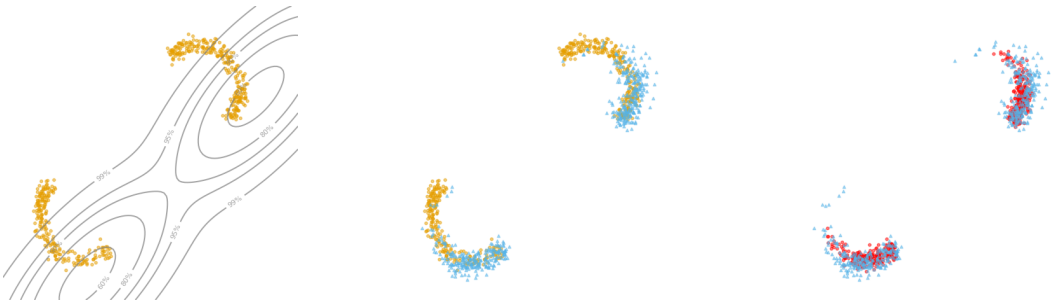
$$= \frac{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i) (\boldsymbol{\mu}_i - \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t))^{\mathbf{T}} \widetilde{\boldsymbol{\Sigma}}_i^{-1} \nabla_{\boldsymbol{\theta}_t} \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t)}{\sum_{i=1}^{K} \mathcal{N}(\boldsymbol{\mu}_i | \boldsymbol{\mu}_{0|t}(\boldsymbol{\theta}_t), \widetilde{\boldsymbol{\Sigma}}_i)}. \tag{A.9}$$

A.3. Experimental Details

Toy Gaussian Example. A Gaussian likelihood is chosen for tractability, where $x \mid \boldsymbol{\theta} \sim \mathcal{N}(x; \theta_1, \theta_2^2)$ so $\boldsymbol{\theta} \in \mathbb{R}^2$. The original prior $p(\boldsymbol{\theta})$ is uniform over $[0, 1]^2$, while the new prior $q(\boldsymbol{\theta})$ is a multivariate Gaussian distribution:

$$q(\boldsymbol{\theta}) = \mathcal{N}\left(\boldsymbol{\theta}; \begin{bmatrix} 0.3 \\ 0.8 \end{bmatrix}, \begin{bmatrix} 0.039 & 0.025 \\ 0.025 & 0.04 \end{bmatrix}\right) \tag{A.10}$$

where θ_1 represents the mean and θ_2 the standard deviation of the likelihood. This choice of prior introduces correlation between the mean and standard deviation parameters while concentrating probability mass in a specific region of the parameter space. The $\mathbf{x}$ for likelihood calculations for training are 10 samples from a given $\boldsymbol{\theta}^{(i)}$ therefore $\mathbf{x}^{(i)} \in \mathbb{R}^{10}$. The base model was trained with $10,000$ simulations. The network architecture and training scheme was taken from the base configuration in Gloeckler et al. (2024). In Fig. A.1 a histogram



(a) Prior v samples **(b)** PriorGuide v samples **(c)** PriorGuide v retrained

Figure A.2: Two moons with correlated prior. The ● points are samples from the diffusion model trained with uniform prior $p(\boldsymbol{\theta})$. Contours of the new prior $q(\boldsymbol{\theta})$. The ▲ points are PriorGuide samples using this new prior. **(c)** compares these against samples ● from a model retrained with the new prior.

plot shows the sample frequency as a comparison for the posterior density which can be computed exactly.

Two Moons with Correlated Prior. We use the standard two moons example in the SBI package detailed in Greenberg et al. (2019), where $\boldsymbol{\theta} \in \mathbb{R}^2$ and $\mathbf{x} \in \mathbb{R}^2$. The original prior $p(\boldsymbol{\theta})$ is uniform over $[-1, 1]^2$, while the new prior $q(\boldsymbol{\theta})$ is a multivariate mixture Gaussian distribution:

$$q(\boldsymbol{\theta}) = \frac{1}{2}\mathcal{N}\left(\boldsymbol{\theta}; \begin{bmatrix}0.2\\0.2\end{bmatrix}, \begin{bmatrix}0.01 & 0.007\\0.007 & 0.01\end{bmatrix}\right) + \frac{1}{2}\mathcal{N}\left(\boldsymbol{\theta}; \begin{bmatrix}-0.2\\-0.2\end{bmatrix}, \begin{bmatrix}0.01 & 0.007\\0.007 & 0.01\end{bmatrix}\right) \tag{A.11}$$

where the mixture weights are equal so 0.5, and each component shares the same covariance matrix with correlation coefficient. The base model was trained with $10,000$ simulations and same network architecture as in the previous example.

Ornstein-Uhlenbeck Process (OUP). OUP is a well-established stochastic process frequently applied in financial mathematics and evolutionary biology for modeling mean-reverting dynamics (Uhlenbeck and Ornstein, 1930). The model is defined as:

$$y_{t+1} = y_t + \Delta y_t, \quad \Delta y_t = \theta_1 \left[\exp(\theta_2) - y_t\right] \Delta t + 0.5w, \quad \text{for } t = 1, \ldots, T,$$

where we set $T = 25$, $\Delta t = 0.2$, and initialize $x_0 = 10$. The noise term follows a Gaussian distribution, $w \sim \mathcal{N}(0, \Delta t)$. We define $p(\boldsymbol{\theta})$ as a uniform prior, $U([0, 2] \times [-2, 2])$, over the latent parameters $\boldsymbol{\theta} = (\theta_1, \theta_2)$.

For this OUP task, the base model is trained on 10,000 simulations. We evaluate the performance using Maximum Mean Discrepancy (MMD) with an exponentiated quadratic kernel with a lengthscale of 1, and Root Mean Squared Error (RMSE). Each experiment is evaluated using 100 randomly sampled $\boldsymbol{\theta}$. For each $\boldsymbol{\theta}$, we generate 1,000 posterior samples, repeating this process over five runs.

We define two new prior distributions $q(\boldsymbol{\theta})$ for the OUP experiments: (i) The *simple prior* consists of Gaussian distributions with a standard deviation set to 5% of the parameter range.

Each prior's mean is sampled from a Gaussian centered on the true parameter value, using the same standard deviation (similar to Chang et al., 2024). (ii) The *complex prior*, a mixture of two slightly correlated bivariate Gaussians with equal component weights ($\pi_1 = \pi_2 = 0.5$):

$$q(\boldsymbol{\theta}) = \pi_1 \mathcal{N}\left(\begin{pmatrix} 0.5 \\ -1.0 \end{pmatrix}, \begin{pmatrix} 0.06 & 0.01 \\ 0.01 & 0.06 \end{pmatrix}\right) + \pi_2 \mathcal{N}\left(\begin{pmatrix} 1.3 \\ 0.5 \end{pmatrix}, \begin{pmatrix} 0.06 & 0.01 \\ 0.01 & 0.06 \end{pmatrix}\right). \tag{A.12}$$

Turin Model. Turin is a widely used time-series model for simulating radio wave propagation (Turin et al., 1972; Pedersen, 2019). This model generates high-dimensional, complex-valued time-series data and is governed by four key parameters: G_0 determines the reverberation gain, T controls the reverberation time, λ_0 defines the arrival rate of the point process, and σ_N^2 represents the noise variance.

The model assumes a frequency bandwidth of $B = 0.5$ GHz and simulates the transfer function H_k at $N_s = 101$ evenly spaced frequency points. The observed transfer function at the k-th frequency point, Y_k, is defined as:

$$Y_k = H_k + W_k, \quad k = 0, 1, \ldots, N_s - 1,$$

where W_k represents additive zero-mean complex Gaussian noise with circular symmetry and variance σ_W^2. The transfer function H_k is expressed as:

$$H_k = \sum_{l=1}^{N_{\text{points}}} \alpha_l \exp(-j2\pi\Delta f k \tau_l),$$

where the time delays τ_l are sampled from a homogeneous Poisson point process with rate λ_0, and the complex gains α_l are modeled as independent zero-mean complex Gaussian random variables. The conditional variance of the gains is given by:

$$\mathbb{E}[|\alpha_l|^2 | \tau_l] = \frac{G_0 \exp(-\tau_l / T)}{\lambda_0}.$$

To obtain the time-domain signal $\tilde{y}(t)$, an inverse Fourier transform is applied:

$$\tilde{y}(t) = \frac{1}{N_s} \sum_{k=0}^{N_s - 1} Y_k \exp(j2\pi k \Delta f t),$$

where $\Delta f = B/(N_s - 1)$ represents the frequency spacing. Finally, the real-valued output is computed by taking the absolute square of the complex signal and applying a logarithmic transformation:

$$y(t) = 10 \log_{10}(|\tilde{y}(t)|^2).$$

We follow the same training and experimental setup as in OUP. In this Turin case, all parameters are normalized to $[0, 1]$ using the transformation: $\tilde{x} = \frac{x - x_{\min}}{x_{\max} - x_{\min}}$, where $\tilde{x}$ is the normalized value. The true parameter bounds are: $G_0 \in [10^{-9}, 10^{-8}]$, $T \in [10^{-9}, 10^{-8}]$, $\lambda_0 \in [10^7, 5 \times 10^9]$, $\sigma_N^2 \in [10^{-10}, 10^{-9}]$.

For this Turin problem, the *simple prior* follows the same specification as in OUP, while the *complex prior* is also a multivariate Gaussian mixture with equal component weights

but with different component parameters, adjusted to match the Turin model's parameter dimension and normalized range, defined as:

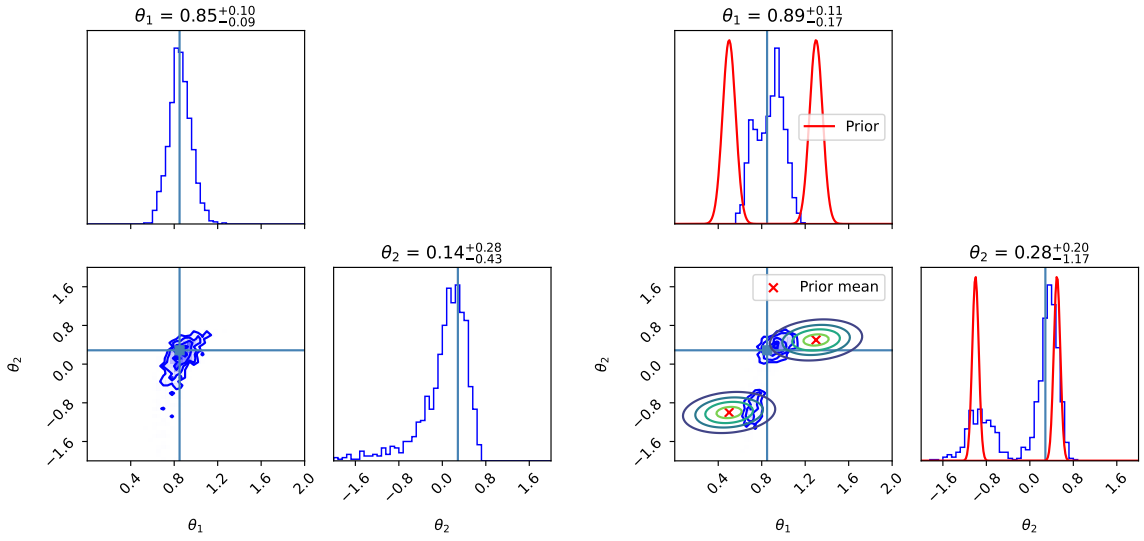
$$q(\boldsymbol{\theta}) = \pi_1 \mathcal{N}\left(\begin{pmatrix}0.30\\0.30\\0.70\\0.70\end{pmatrix}, \begin{pmatrix}0.01 & 0.005 & 0.005 & 0.005\\0.005 & 0.01 & 0.005 & 0.005\\0.005 & 0.005 & 0.01 & 0.005\\0.005 & 0.005 & 0.005 & 0.01\end{pmatrix}\right) \tag{A.13}$$

$$+ \pi_2 \mathcal{N}\left(\begin{pmatrix}0.70\\0.70\\0.30\\0.30\end{pmatrix}, \begin{pmatrix}0.01 & 0.005 & 0.005 & 0.005\\0.005 & 0.01 & 0.005 & 0.005\\0.005 & 0.005 & 0.01 & 0.005\\0.005 & 0.005 & 0.005 & 0.01\end{pmatrix}\right). \tag{A.14}$$

A.4. SBI Mixture Prior Corner plots

As a representative visualization of the SBI experiments, we present example corner plots of posterior samples for the case where the sampling distribution of $\boldsymbol{\theta}$ follows a mixture distribution in both the OUP and Turin SBI tasks. These plots illustrate marginal pairwise relationships between sampled latent parameters and demonstrate that PriorGuide can handle complex priors, producing posterior results that are reasonable given the prior structure.

Fig. A.3 presents the corner plots for the OUP case, comparing Simformer and PriorGuide. The higher-dimensional Turin task is shown in Fig. A.4 and Fig. A.5 for Simformer and PriorGuide, respectively.



(a) **(b)**

Figure A.3: **OUP model.** Comparison of posterior samples between Simformer and PriorGuide. The light blue line is the true parameter value. The bottom left corner of **(b)** shows the sampling mixture distribution (and prior); see Eq. (A.12) for detail. **(a)** Simformer results (without prior guidance), where the model fails to capture the true mixture distribution of $\boldsymbol{\theta}$. **(b)** PriorGuide helps the base model generate posterior results that align well with the structure of the complex prior.

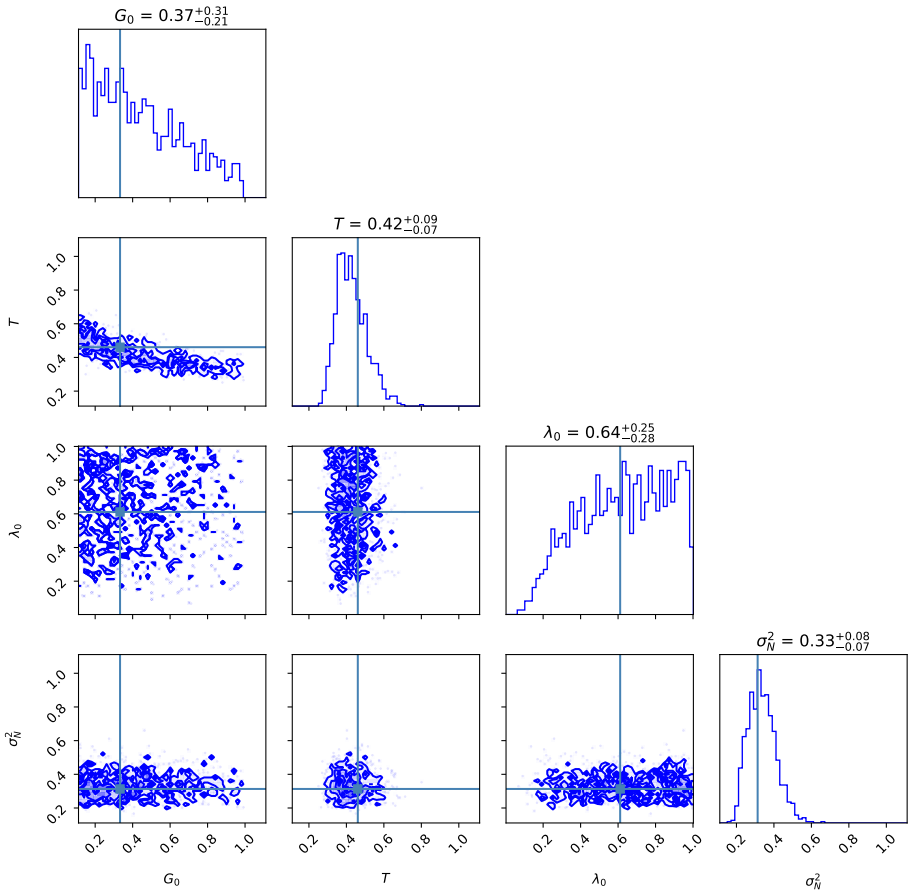


Figure A.4: **Turin model (SimFormer).** Posterior samples using Simformer, without prior guidance. The light blue line is the true parameter value. The sampling distribution is the mixture described in Eq. (A.13) (see bottom left corner of Fig. A.5 for visualization). Since the model is trained on a uniform prior, it yields a wide posterior that fails to capture the multimodality of the true $\boldsymbol{\theta}$ distribution.

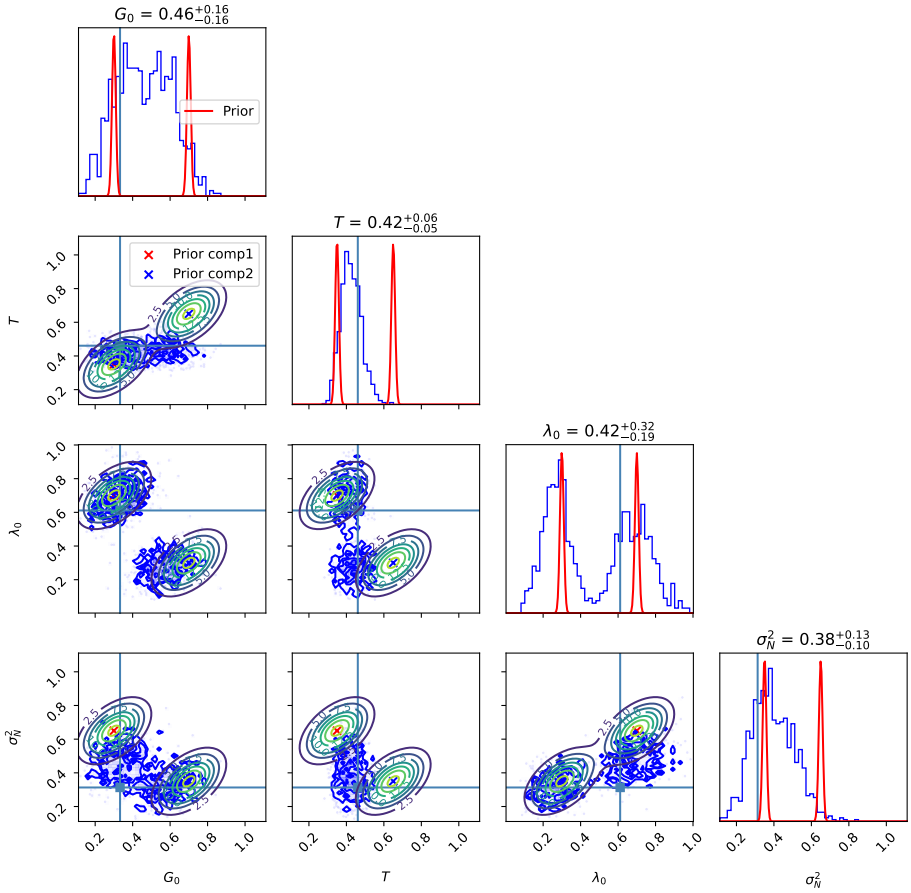


Figure A.5: **Turin model (PriorGuide).** Posterior samples from PriorGuide. Compared to the Simformer without prior guidance (Fig. A.4), PriorGuide significantly improves posterior estimation, aligning it more closely with the complex prior structure while using the same model as the Simformer, without retraining. Note that the contour plots represent the sampling distribution (prior).