COROS: CONTEXT-ROBUST TIME-SERIES FORECASTING WITH SCORE-BASED DIFFUSION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion models have emerged as an effective approach for probabilistic time-series forecasting, generating future trajectories from historical observations via iterative denoising. However, in many real-world applications the conditioning history may be contaminated by measurement noise or anomalies, which can degrade conditional generation. We propose COROS (COntext-RObust time-series forecasting with Score-based diffusion models), which extends score-based diffusion to the concatenated past–future sequence. By jointly denoising both the forecast horizon and the (possibly corrupted) history during sampling, COROS improves robust forecasting even under noisy conditions. Because noise is often introduced during observation, our method reduces sensitivity to corrupted historical data, thereby enhancing forecasting accuracy. Across five multivariate probabilistic forecasting benchmarks, COROS achieves competitive or best-in-class performance compared with 13 baselines. Furthermore, evaluation on standard LTSF benchmarks, where robustness to corrupted history is most transparently reflected in point-forecast degradation, demonstrates that COROS exhibits substantially smaller error growth compared to representative baselines as input noise increases.

1 INTRODUCTION

Time series forecasting predicts future observations from historical data by learning temporal dependencies between past and future, and is essential across domains such as physics, climate, healthcare, and finance (Lim & Zohren, 2021; Torres et al., 2021a; Masini et al., 2023). Among numerous deep learning methods proposed for this problem (Lim & Zohren, 2021; Torres et al., 2021b; Miller et al., 2024), diffusion-based conditional generative models have shown strong performance (Rasul et al., 2021; Tashiro et al., 2021; Yan et al., 2021). These methods train neural networks to approximate the conditional score $\nabla_{\mathbf{x}_t^{\text{pred}}} \log p(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}^{\text{hist}})$ at diffusion step t, and synthesize future observations via a denoising process (reverse SDE or ancestral sampling) conditioned on the historical data.

However, due to measurement errors or the occurrence of anomalies, the conditioning historical observations may contain noise, which can negatively impact forecasting performance (Rožanec et al., 2021). To address this, we propose COROS: **CO**ntext-**RO**bust time-series forecasting with **S**core-based diffusion models, which extends the denoising process of score-based generation to both future observations and conditional historical data, effectively reducing the inherent noise in the conditional inputs. Specifically, COROS learns the conditional score $\nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}} \mid \mathbf{x}^{\text{hist}})$ over the concatenation $\mathbf{x}^{\text{total}} = [\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}]$.

Modeling the total sequence within the diffusion process has two major advantages. First, high-frequency anomalies in the data are effectively mitigated after a few forward diffusion steps (Choi et al., 2022; Yang et al., 2023). Second, during the generation process, predictions and conditions are interdependently generated. This interdependence enables the reverse diffusion process to minimize the impact of anomalies in the historical data, as predictions are generated using predominantly non-anomalous conditions. To further leverage the noise-reduction capabilities of COROS and focus on prediction generation, we introduce an imbalanced weighting scheme in the loss function between the past and future components.

To further enhance the forecasting performance of our score-based conditional generation, we integrate classifier-free guidance (Ho & Salimans, 2022) into our framework. Our results show that

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

incorporating classifier-free methods into conditional generation reduces errors on most datasets, and this positive impact is further amplified when combined with COROS. Finally, in the experimental section, we demonstrate that our proposed method achieves better $CRPS_{sum}$ performance across 5 probabilistic time-series forecasting datasets, outperforming 13 baseline methods.

However, robustness to noisy history is clearly observed through *point-forecast* accuracy degradation. We therefore adopt mean squared error (MSE) for robustness experiments, following common practice in the LTSF literature, enabling intuitive comparison of error growth under increasing input corruption. We compare COROS with diffusion, Transformer, and linear baselines on 3 long-term time-series forecasting (LTSF) datasets with growing injected noise scale.

Contributions.

1. We propose COROS, a novel approach for score-based conditional generation in time-series forecasting. COROS reduces noise in conditional historical observations by synthesizing both future and historical observations through a denoising process.
2. We provide a theoretical analysis under standard assumptions that explains why total-sequence conditional diffusion can reduce sensitivity to corrupted histories, and we derive a practical training objective.
3. Across 5 probabilistic forecasting datasets and 3 LTSF datasets, our method outperforms 13 baselines and maintains robust predictions under input corruption.

2 Preliminaries and Problem Statement

2.1 Diffusion Models

Generative models aim to synthesize realistic data by learning the underlying probability distribution (Oussidi & Elhassouny, 2018; Harshvardhan et al., 2020; Cao et al., 2024). Diffusion models operate through two steps: (i) **noising** (forward) and (ii) **denoising** (reverse) (Yang et al., 2023).

Given $\mathbf{x} \sim p(\mathbf{x})$ and a discrete diffusion length T, DDPMs use the forward kernel

$$p(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\Big(\mathbf{x}_t; \sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t \mathbf{I}\Big), \quad t = 1, \dots, T,$$

with $\beta_t \in (0, 1)$. They train a learnable reverse kernel

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(t, \mathbf{x}_t),\, \boldsymbol{\Sigma}_\theta(t, \mathbf{x}_t)).$$

Score-based diffusion generalizes the process to continuous-time SDEs (Song et al., 2020). The forward and reverse SDEs can be written as

$$d\mathbf{x} = \mathbf{f}(t, \mathbf{x})\, dt + g(t)\, d\mathbf{w},$$
$$d\mathbf{x} = \Big(\mathbf{f}(t, \mathbf{x}) - g(t)^2 \nabla_{\mathbf{x}} \log p_t(\mathbf{x})\Big)\, dt + g(t)\, d\bar{\mathbf{w}},$$

where $t \in [0, 1]$, $\mathbf{w}$, $\bar{\mathbf{w}}$ are forward/backward Brownian motions.

Score-based models train a score network $s_\theta(t, \mathbf{x}_t)$ via score matching:

$$L_{\mathrm{SM}}(\theta) = \mathbb{E}_{t, \mathbf{x}_t}\Big[\lambda(t)\, \big\| s_\theta(t, \mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t) \big\|_2^2\Big].$$

Direct score matching is costly; using the Gaussian conditional of the forward SDE yields denoising score matching (DSM) (Vincent, 2011; Øksendal, 2014):

$$L_{\mathrm{DSM}}(\theta) = \mathbb{E}_{t, \mathbf{x}, \mathbf{x}_t}\Big[\lambda(t)\, \big\| s_\theta(t, \mathbf{x}_t) - \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t \mid \mathbf{x}) \big\|_2^2\Big].$$

For sampling, we use a predictor-corrector (PC) style predictor (Euler–Maruyama) without a corrector (as in our default experiments). A typical discretization of the reverse SDE is

$$\mathbf{x}_{t-\Delta t} = \mathbf{x}_t + \Big(\mathbf{f}(t, \mathbf{x}_t) - g(t)^2 s_\theta(t, \mathbf{x}_t)\Big)\Delta t + g(t)\sqrt{\Delta t}\, \mathbf{z}, \quad \mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}).$$

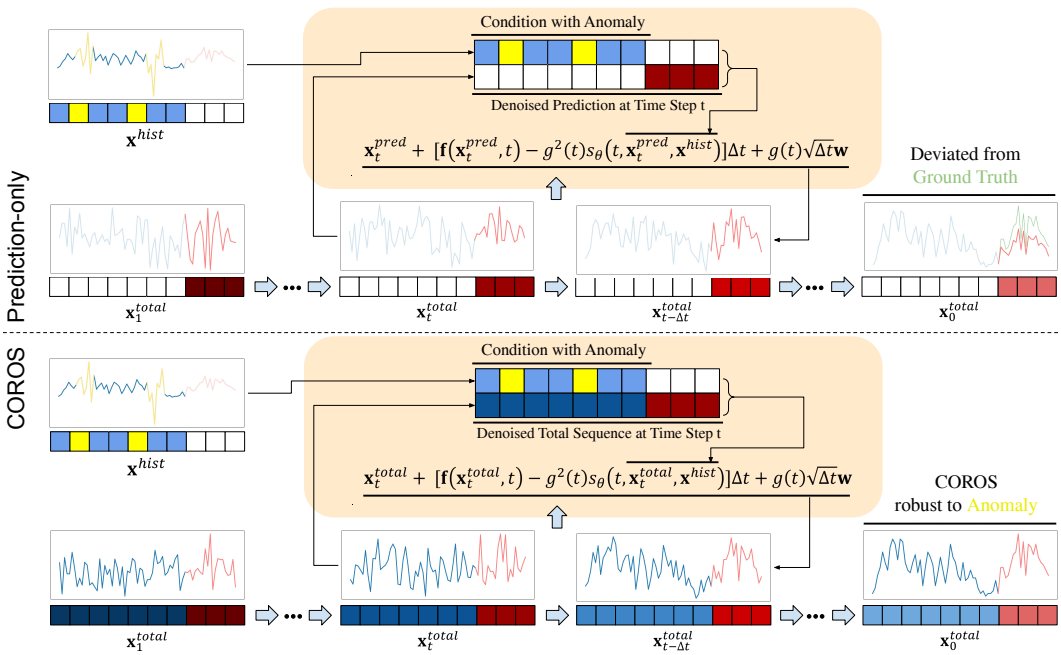


Figure 1: **Overall visualization of the sampling process of COROS.** Comparison between prediction-only generation (top) and COROS (bottom). Even when anomalies (highlighted in yellow) exist in the historical data, COROS iteratively applies reverse diffusion to produce denoised latent history during sampling, whereas prediction-only generation deviates from the ground truth.

2.2 Time-series Forecasting

Given a historical sequence $\mathbf{x}^{1:N}$, the task is to forecast the future sequence $\mathbf{x}^{N+1:N+T}$, where N represents the length of the historical data, and T represents the length of the prediction. Each data point $\mathbf{x}$ belongs to $\mathbb{R}^d$. For clarity, we denote the history sequence $\mathbf{x}^{1:N}$ by $\mathbf{x}^{\text{hist}}$, the future sequence $\mathbf{x}^{N+1:N+T}$ by $\mathbf{x}^{\text{pred}}$, and the ocncatenated sequence $\mathbf{x}^{1:N+T}$ by $\mathbf{x}^{\text{total}}$. Time-series forecasting has been widely studied to improve the accuracy of future predictions. However, the complex, intertwined characteristics of time-series data make it difficult to fully capture and understand its underlying patterns.

To address this challenge, researchers have increasingly turned to generative models, which aim to model the conditional likelihood of time-series data and provide a more comprehensive understanding of its structure. As a result, many time-series diffusion models have been proposed, which generally aim to learn the conditional distribution of future values given the history sequence, $p(\mathbf{x}^{\text{pred}}|\mathbf{x}^{\text{hist}})$ (Rasul et al., 2021; Tashiro et al., 2021). We provide a detailed explanation of their contributions and the rationale behind their target selection in Section H. Therefore, DDPM-based forecasting methods optimize the following equation:

$$L_{\text{SM}}^{\text{pred}}(\theta) = \mathbb{E}_{t,\mathbf{x}_t^{\text{pred}}}\left[\lambda(t)\left\|s_\theta(t,\mathbf{x}_t^{\text{pred}},\mathbf{x}^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{pred}}}\log p(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}^{\text{hist}})\right\|_2^2\right].$$

3 Proposed Method

In this section, we analyze existing prediction methods and propose COROS, which performs conditional diffusion over the concatenated past–future sequence to reduce sensitivity to corrupted histories and yield robust forecasts.

3.1 COROS: Context-Robust Time-Series Forecasting with Score-based Diffusion Models

Current methods train diffusion models by optimizing the conditional probability $p(\mathbf{x}^{\text{pred}} \mid \mathbf{c})$, where we denote the conditioning history $\mathbf{x}^{\text{hist}}$ by $\mathbf{c}$ to emphasize its role as model input. Here $\mathbf{c}$ denotes the

observed history fed to the model. During training, ($\mathbf{c} = \mathbf{x}^{\text{hist}}$). Under corrupted-history analysis, $\mathbf{c}$ may be noisy and remains fixed throughout sampling, whereas ($\mathbf{x}_t^{\text{hist}}$) denotes the history coordinates of the latent total state ($\mathbf{x}_t^{\text{total}} = [\mathbf{x}_t^{\text{hist}}, \mathbf{x}_t^{\text{pred}}]$) and is updated by the reverse SDE.

During training, they use a DDPM-style loss (Rasul et al., 2021; Tashiro et al., 2021; Kollovieh et al., 2023):

$$L_{\text{DDPM}}^{\text{pred}}(\theta) = \mathbb{E}_{t,\boldsymbol{\epsilon},\mathbf{x}^{\text{pred}}}\Big[\lambda(t)\,\big\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(t, \mathbf{x}_t^{\text{pred}}, \mathbf{c})\big\|_2^2\Big].$$

Song et al. (2021) proved that

$$-\mathbb{E}_{\mathbf{x}^{\text{total}}} \log p(\mathbf{x}^{\text{pred}} \mid \mathbf{c}) \le \frac{1}{2} L_{\text{SM}}^{\text{pred}}(\theta) + C_1,$$

where $L_{\text{SM}}^{\text{pred}}(\theta)$ is defined above. Under the SDE parameterization $\mathbf{x}_t = \boldsymbol{\mu}_t(\mathbf{x}) + \sigma_t \boldsymbol{\epsilon}$, $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, the noise-prediction network ϵ_θ is equivalent to a score network via $s_\theta(t, \mathbf{x}_t, \mathbf{c}) = -\boldsymbol{\epsilon}(t, \mathbf{x}_t, \mathbf{c})/\sigma_t$.

While intuitive, this approach is sensitive to noisy conditions. We propose **COROS**, which performs diffusion over the concatenated past–future sequence instead of only synthesizing the prediction sequence. COROS optimizes

$$-\mathbb{E}_{\mathbf{x}^{\text{total}}} \log p(\mathbf{x}^{\text{total}} \mid \mathbf{c}) \le \frac{1}{2} L_{\text{SM}}^{\text{total}}(\theta) + C_1,$$

with

$$L_{\text{SM}}^{\text{total}}(\theta) = \mathbb{E}_{t,\mathbf{c},\mathbf{x}_t^{\text{total}}}\Big[\lambda(t)\,\big\|s_\theta(t, \mathbf{x}_t^{\text{total}}, \mathbf{c}) - \nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}} \mid \mathbf{c})\big\|_2^2\Big].$$

Generating the total time-series offers two key advantages:

1. **Reduced condition mismatch at high noise levels:** After sufficient forward diffusion (i.e., at large t), the diffused distributions of clean and corrupted histories become closer, reducing conditioning mismatch in the early reverse steps.
2. **Joint past–future coupling during sampling:** The future trajectory is generated while the latent history segment is simultaneously updated, which reduces sensitivity to corruption in the observed history.

Forward diffusion reduces the mismatch between clean and corrupted conditions, which mainly affects the early reverse steps. Throughout the reverse process, COROS generates the forecast together with the latent history $\mathbf{x}_t^{\text{hist}}$, which is anchored by c and progressively denoised, whereas prediction-only diffusion conditions the forecast directly on c at every step. Appendices B–C examine the score sensitivities and local contraction of the trained network along this process.

The following lemma demonstrates how the KL divergence between noisy and clean conditions decreases as noise is added during the diffusion process, thereby mitigating the adverse effects of anomalies during the reverse diffusion process:

Lemma 1. *Let p be the true data distribution and $\hat{p}$ a distribution containing anomalies. Then, during the diffusion process:*

$$D_{KL}(\hat{p}_0 \| p_0) \ge D_{KL}(\hat{p}_t \| p_t) \ge D_{KL}(\hat{p}_1 \| p_1) \approx 0, \quad \forall t \in [0, 1].$$

The following theorem bridges the above theorem with COROS's robustness, showing that COROS preserves robustness between noisy and clean conditions:

Theorem 1. *Assume (A1)–(A3) in Appendix G. Let $H(t) := \int_t^1 g(s)^2\, ds$ and $G := H(0) = \int_0^1 g(s)^2\, ds > 0$. Then for the noised input condition $\mathbf{c}_s \equiv \mathbf{x}_s^{\text{hist}}$, the difference between forecasts under clean and corrupted conditions $\|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred'}\|$ satisfies:*

$$(i) \quad L \int_0^1 g(s)^2\, \|\mathbf{c}_s - \mathbf{c}'_s\|\, ds \;\le\; L\, G \sup_{s\in[0,1]} \|\mathbf{c}_s - \mathbf{c}'_s\|,$$

$$(ii) \quad L \int_0^1 g(s)^2\, e^{-m_x\, H(s)}\, \|\mathbf{c}_s - \mathbf{c}'_s\|\, ds \;\le\; \frac{L}{m_x}\big(1 - e^{-m_x G}\big) \sup_s \|\mathbf{c}_s - \mathbf{c}'_s\|.$$

Hence, the robustness constant of total-sequence sampling, $\frac{L}{m_x}(1 - e^{-m_x G})$, is strictly smaller than that of prediction-only sampling, LG.

Empirical diagnostics on the trained score network (Figure 6; Appendix C) show that these assumptions approximately hold along the sampling trajectories. In Theorem 1, we distinguish $\mathbf{x}^{\text{hist}}$ from condition and total sequence by denoting that of condition as $\mathbf{c}$. The above theorem shows that when noise is injected into the history data, under (A1)–(A3), COROS has a strictly smaller robustness constant than prediction-only; thus any Lipschitz evaluation functional inherits a no-worse (and, under nonzero perturbations, strictly better) worst-case sensitivity.

On the other hand, it is well known that directly computing $L_{SM}^{\text{pred}}(\theta)$ and $L_{SM}^{\text{total}}(\theta)$ is computationally prohibitive due to the need for statistical methods (Hyvärinen, 2005; Song et al., 2020). Therefore, we derive the denoising score-matching losses to train the score network s_θ and guarantee its convergence:

Proposition 1. *The DSM objectives corresponding to* $L_{\text{SM}}^{\text{pred}}(\theta)$ *and* $L_{\text{SM}}^{\text{total}}(\theta)$ *can be written as*

$$L_{\text{DSM}}^{\text{pred}}(\theta) = \mathbb{E}_{t,\mathbf{x}^{\text{total}},\mathbf{x}_t^{\text{total}}}\left[\lambda(t)\left\|s_\theta(t,\mathbf{x}_t^{\text{pred}},\mathbf{c}) - \nabla_{\mathbf{x}_t^{\text{pred}}}\log p(\mathbf{x}_t^{\text{total}} \mid \mathbf{x}^{\text{total}})\right\|_2^2\right],$$
$$L_{\text{DSM}}^{\text{total}}(\theta) = \mathbb{E}_{t,\mathbf{x}^{\text{total}},\mathbf{x}_t^{\text{total}}}\left[\lambda(t)\left\|s_\theta(t,\mathbf{x}_t^{\text{total}},\mathbf{c}) - \nabla_{\mathbf{x}_t^{\text{total}}}\log p(\mathbf{x}_t^{\text{total}} \mid \mathbf{x}^{\text{total}})\right\|_2^2\right].$$

Beyond using $L_{DSM}^{\text{total}}(\theta)$, we place additional emphasis on the prediction portion of the sequence. The weighted objective balances historical information and forecast generation. To achieve this balance, we introduce a hyperparameter γ to control the influence of the past history. The exact loss function is then defined as:

$$\Delta(\theta) = s_\theta(t,\mathbf{x}_t^{\text{total}},\mathbf{c}) - \nabla_{\mathbf{x}_t^{\text{total}}}\log p(\mathbf{x}_t^{\text{total}}|\mathbf{x}^{\text{total}}),$$
$$L(\theta) = \mathbb{E}_{t,\mathbf{x}^{\text{total}},\mathbf{x}_t^{\text{total}}}\lambda(t)[\gamma||\mathbf{m}\odot\Delta(\theta)||_2^2 + ||(1-\mathbf{m})\odot\Delta(\theta)||_2^2]$$

, where $\odot$ is a Hadamard product and $\mathbf{m} = \{0,1\}^{(N+T)\times d}$ is a mask vector that $\mathbf{m}_{ij} = 1$ if $i \leq N$ and 0 otherwise, dividing the past and future elements in our loss function.

3.2 Training and Inference

For training COROS, we employ the VP SDE (Song et al., 2020). γ controls how strongly we train the score on the history coordinates. $\gamma = 0.0$ reduces training to the prediction coordinates (prediction-only objective), while $\gamma = 1.0$ trains the score on both history and future coordinates equally. Neither extreme is optimal. As shown in Figure 2, COROS achieves the best results on the Solar and Electricity datasets when $\gamma = 0.1$ and $\gamma = 0.5$, respectively.

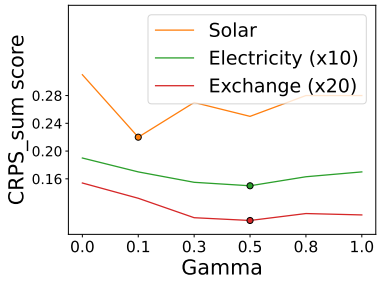


Figure 2: We report $CRPS_{sum}$ fixing other hyperparameters in Table 15, only varying γ from 0.0 to 1.0.

After training, we generate the total sequence from a history condition using a PC sampling procedure (Euler–Maruyama predictor) with default settings (Song et al., 2020). We also adapt classifier-free guidance (CFG) to our framework. For CFG, we adopt proportional training: with probability p_{cond} (default 0.2), train conditional $s_\theta(\mathbf{x}_t,\mathbf{c})$; with probability $1-p_{\text{cond}}$, train unconditional $s_\theta(\mathbf{x}_t,\mathbf{0})$. Detailed hyperparameters are provided in Appendix D.

4 Why Generate the Total Sequence Conditioned on Historical Data?

In this section, we conceptually and empirically justify the necessity of conditioning the generation of the total sequence on historical observations. To clearly demonstrate this, we conduct two additional comparisons: 1) total generation without conditioning (unconditional total generation) using replacement methods, and 2) unconditional generation guided by Observation Self-Guidance (OSG) (Kollovieh et al., 2023). Both comparisons are conducted using the same backbone for a fair comparison, highlighting the benefits and practical advantages of our proposed context-aware robustness approach.

Replacement Method. Previous approaches (Song et al., 2020; Ho et al., 2022) suggest continuously replacing the historical portion during diffusion with corresponding diffused historical values from the forward SDE. This approach solely utilizes denoised historical information without explicitly conditioning the score network on the observed history.

Specifically, at denoising step t, the history portion of $\mathbf{x}_t^{\text{total}}$ is replaced with forward-diffused historical values. Specifically, Replacement overwrites the history segment ($\mathbf{y}^{\text{hist}}$) with a forward-diffused surrogate ($\mathbf{y}_t^{\text{hist}}$), producing $\tilde{\mathbf{x}}_t = (\mathbf{y}_t^{\text{hist}}, \mathbf{x}_t^{\text{pred}})$ before applying the reverse SDE. Equivalently, the actual update obeys

$$\mathrm{d}\tilde{\mathbf{x}}_t = \Big[f - g^2 s_\theta\Big](\tilde{\mathbf{x}}_t)\,\mathrm{d}t + g\,\mathrm{d}\bar{\mathbf{w}}_t + \Pi_{\text{hist}}\Big(\underbrace{\Big[f - g^2 s_\theta\Big]\big(\mathbf{y}_t^{\text{hist}} - \tilde{\mathbf{x}}_t\big)}_{\text{replacement forcing}}\Big)\,\mathrm{d}t, \tag{1}$$

where Π_{hist} projects onto the history coordinates. The extra term in Equation 1 is an *exogenous, step-dependent* force that (i) violates the Markov consistency of the learned reverse dynamics (the drift now depends on an external process), (ii) makes the reverse trajectory follow a score field conditioned on *mismatched* history, and (iii) accumulates bias across steps; the stronger the diffusion noise (earlier t), the larger the mismatch and the propagated error. We point out that for Replacement and OSG baselines, the authors use an unconditional total-sequence diffusion model and enforce conditioning only at inference time.

Observation Self-Guidance (OSG). Another method, OSG, guides unconditional generation using score-SDE by computing gradients of the score network (Song et al., 2020). Although OSG improves performance notably (Solar dataset achieved 0.2490 ± 0.0094), it requires repeated gradient computations of the score network, resulting in significantly increased computational overhead and limiting practical applicability in high-dimensional settings (Electricity).

OSG adds an external force $\nabla_{\mathbf{x}} \log p(\mathbf{y}_{\text{obs}}|\mathbf{x}_t)$ at each reverse step. For a Gaussian observation model with predictor $\hat{y}(x_t)$ and noise covariance Σ, $\nabla_{\mathbf{x}_t} \log p(\mathbf{y}_{\text{obs}}|\mathbf{x}_t) = J_{\hat{\mathbf{y}}}(\mathbf{x}_t)^\top \Sigma^{-1}\big(\mathbf{y}_{\text{obs}} - \hat{\mathbf{y}}(\mathbf{x}_t)\big)$, so the magnitude of the external force scales with the residual $\mathbf{y}_{\text{obs}} - \hat{\mathbf{y}}(\mathbf{x}_t)$. This has two consequences: (i) the per-step, observation-dependent *direct forcing* splits trajectories and can introduce numerical instability (large residuals ⇒ stiff drifts, oscillation/divergence); (ii) computing the reverse update requires *additional backpropagations through the score/predictor at every step*, which yields substantial computational and memory overhead in high-dimensional multivariate settings.

Why COROS is Preferable. COROS conditions on the history *once* and denoises the *entire* sequence with the learned score without step-wise overwriting or external gradient forcing. Hence the reverse process remains Markov-consistent, avoids trajectory splits and bias accumulation from other methods, and matches the runtime of standard sampling (no extra per-step backward passes), yielding both stability and efficiency.

Table 1: Performance comparison of total sequence generation methods: conditioned COROS (ours), unconditional generation methods (Replacement and OSG), and prediction-only generation.

Method	Exchange	Solar	Electricity
Replacement	.0057±.0001	.2774±.0083	.0173±.0012
OSG	.0055±.0003	**.2490±.0094**	-
Prediction-only	.0063±.0002	.2871±.0202	.0210±.0013
COROS (ours)	**.0054±.0002**	.2501±.0080	**.0153±.0003**

Table 1 summarizes the experimental performance of the unconditional total-generation methods compared to COROS (conditioned total generation). Additionally, results of prediction-only generation are included for reference.

In short, the results highlight two key observations:

- **Conditioning Improves Forecasting Quality.** Conditioned COROS outperforms both unconditional total-generation (Replacement) and prediction-only generation methods across most datasets, while achieving comparable performance to OSG on Solar. This indicates that conditioning on historical data is beneficial, likely because it preserves residual information from the original conditions.
- **Practical Efficiency and Scalability.** Although OSG achieves slightly better performance on the Solar dataset, its requirement for repeated gradient computations makes it computationally expensive and restricts scalability. In contrast, COROS directly integrates denoising

into the reverse diffusion process, achieving comparable or superior performance across datasets with significantly lower computational overhead.

Notably, COROS uses $5.3\times$ less peak GPU memory than OSG at comparable forecasting quality on Solar (Appendix A.5). Overall, these empirical comparisons provide strong justification for conditioning on historical data when generating the total sequence, as COROS effectively balances performance and computational efficiency.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Our experiments consist of two components: (1) evaluating probabilistic forecasting performance on real-world multivariate datasets, and (2) assessing robustness to corrupted historical inputs under a standard Long-Term Time-Series Forecasting (LTSF) evaluation protocol. In both cases, we generate 50 sample trajectories at each test timestamp to compute the respective metrics. We select γ from $\{0.0, 0.1, 0.3, 0.5, 0.8, 1.0\}$ using the original validation data without injected corruption (Appendix D).

Main forecasting benchmarks. For the main experiments, we use five public time-series datasets: Exchange and Solar (Lai et al., 2017), Electricity[1], Taxi[2], and Wikipedia[3]. We compare COROS with 13 baselines covering representative transformer and diffusion-based models (see Appendix F).

We evaluate probabilistic forecasts using the normalized sum of Continuous Ranked Probability Score ($CRPS_{sum}$), a proper scoring rule that measures the discrepancy between the predictive cumulative distribution function (CDF) F and an observation x:

$$CRPS(F, x) \;=\; \int_{\mathbb{R}} \big(F(z) - \mathbb{I}(x \le z)\big)^2 \, dz.$$

We approximate F with an empirical CDF built from samples drawn from the predictive distribution and report

$$CRPS_{sum}(F, x) \;=\; \frac{CRPS(F, \sum_i x_{i,t})}{\sum_{i,t} |x_{i,t}|},$$

where the denominator normalizes by the total magnitude of the target.

While $CRPS_{sum}$ is the standard metric for probabilistic forecasting on our main benchmarks, robustness to corrupted conditioning is most transparently quantified by the degradation in *point-forecast* accuracy. We therefore adopt mean squared error (MSE) for robustness experiments, following common practice in the LTSF literature, which enables a direct and intuitive comparison of error growth under increasing input corruption.

Robustness under corrupted historical inputs (LTSF benchmarks). To directly test robustness against noisy conditions, we evaluate on three canonical LTSF benchmarks: Traffic (862 dimensions), Electricity (321 dimensions), and Weather (21 climate signals). These LTSF benchmarks span a wide range of dimensionalities and include high-dimensional settings comparable to our main benchmarks, providing a stress test for robustness. Note that the Electricity dataset used in the LTSF suite is different from the Electricity dataset used in our main experiments (e.g., in GluonTS they are distinguished as ELECTRICITY and ELECTRICITY-NIPS).

After training on the original benchmark data, we corrupt only half of the *historical input window* at test time by adding zero-mean Gaussian noise with four noise scales $\sigma \in \{0.0, 0.5, 1.0, 1.5\}$ (with σ expressed in units of the data standard deviation). We report MSE on the forecast window with prediction length 24 while following other hyperparameters from their official implementation code and average results over five random seeds. We compare COROS against PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024) (representative Transformer baselines) and DLinear (Zeng

[1] https://archive.ics.uci.edu/ml/datasets/ElectricityLoadDiagrams20112014

[2] https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page

[3] https://github.com/mbohlkeschneider/gluon-ts/tree/mv_release/datasets

Table 2: $CRPS_{sum}$ results on evaluation datasets. "–" indicates the method could not be run due to out-of-memory on our setting.

	Exchange	Solar	Electricity	Taxi	Wiki
VES	**.005±.000**	.900±.003	.880±.004	-	-
VAR	**.005±.000**	.830±.006	.039±.001	-	-
VAR-Lasso	.012±.000	.510±.006	.025±.000	-	3.10±.004
GARCH	.023±.000	.880±.002	.190±.001	-	-
KVAE	.014±.002	.340±.025	.051±.019	-	.095±.012
Vec-LSTM ind-scaling	.008±.001	.391±.017	.025±.001	.506±.005	.133±.002
Vec-LSTM low-copula	.007±.000	.319±.011	.064±.008	.326±.007	.241±.033
GP scaling	.009±.000	.368±.012	.022±.000	.183±.395	1.48±1.03
GP copula	.007±.000	.337±.024	.025±.002	.208±.183	.086±.004
Transformer MAF	.005±.003	.301±.014	.021±.000	.179±.002	.063±.003
MG-TSD	.007±.002	.308±.010	**.015±.002**	.116±.013	.053±.005
TimeGrad	.006±.001	.287±.020	.021±.001	.114±.020	.049±.002
CSDI	.007±.001	.298±.004	.017±.000	.123±.003	.047±.003
COROS	.006±.000	**.250±.007**	.018±.001	.122±.001	.052±.000
COROS-CFG	**.005±.000**	.277±.006	**.015±.000**	**.092±.001**	**.046±.001**

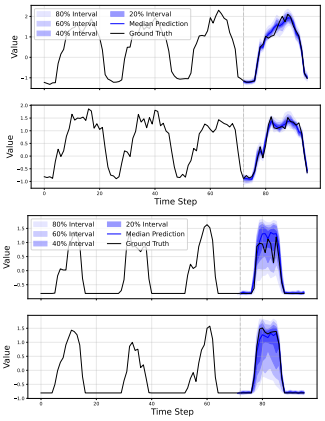


Figure 3: Fan plots on ELC(top) and SOLAR(bottom).

et al., 2022), RLinear (Li et al., 2023) (representative linear baselines). In this setting, we compute the point forecast as the sample mean following probabilistic forecasting protocol.

5.2 Experiments on Main Datasets

We present the $CRPS_{sum}$ performance of COROS and other baseline models in Table 2. We evaluate COROS over five random seeds, and we report both the mean and standard deviation. As shown in the table, COROS and COROS-CFG consistently outperform all competing models across every dataset, including other diffusion-based forecasting models. Notably, while diffusion-based forecasting models like TimeGrad and CSDI perform comparably on certain datasets, COROS consistently delivers superior results across a wide range of data complexities, from relatively low-dimensional datasets (e.g., Exchange) to high-dimensional ones (e.g., Wiki). Specifically, COROS attains the best score on Solar, while COROS-CFG yields additional gains on Exchange, Electricity, Taxi, and Wiki. To understand why CFG underperforms on Solar, we observed that Solar is periodic and low-noise; its predictive signal is already well captured by the base conditional score. We hypothesize that CFG may amplify residual noise in the history and deteriorate prediction quality. In contrast, CFG consistently improves performance on Exchange, Electricity, Taxi, and Wiki, suggesting benefits in noisier and higher-dimensional settings.

Table 3: Robustness to input corruption on LTSF benchmarks. We add Gaussian noise to the history window at test time and report MSE on the 24-step horizon.

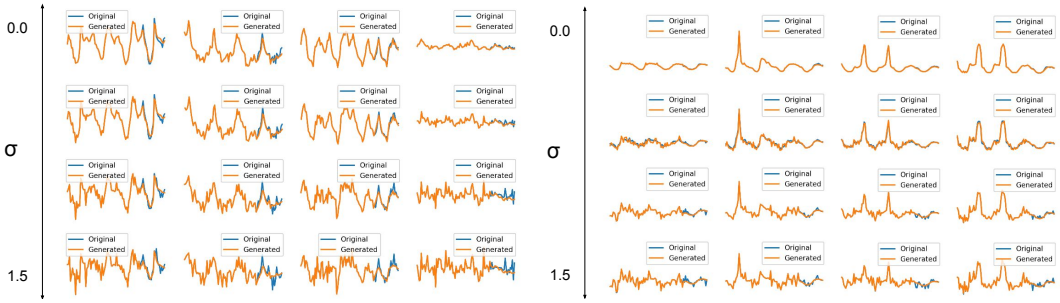
Dataset	Model	MSE on noise scale σ			
		0.0	**0.5**	**1.0**	**1.5**
Traffic	PatchTST	0.3556	0.6217	0.8776	1.1310
	PatchTST+NA	0.4366	0.6121	0.8112	1.0163
	iTransformer	0.3613	0.4081	0.6136	1.0166
	DLinear	0.6253	0.8055	1.0019	1.2023
	RLinear	0.6112	0.6971	0.9917	1.3385
	TimeGrad	0.5436	0.6120	0.7171	0.9028
	Prediction-Only	0.2774	0.5280	0.7194	0.8170
	COROS	**0.2510**	**0.3663**	**0.5790**	**0.7776**
Electricity	PatchTST	0.1546	0.5761	1.0182	1.4654
	PatchTST+NA	0.1807	0.4679	0.7865	1.1129
	iTransformer	0.1102	0.1563	0.3211	0.5156
	DLinear	0.1404	0.3710	0.6221	0.8783
	RLinear	0.1709	0.2577	0.5557	0.9064
	TimeGrad	0.1700	0.2280	0.3248	0.4399
	Prediction-Only	0.1293	0.1512	0.2266	0.4663
	COROS	**0.1024**	**0.1340**	**0.1881**	**0.2436**
Weather	PatchTST	0.4023	2.4870	4.7451	7.0465
	PatchTST+NA	0.4037	2.4210	4.6230	6.8711
	iTransformer	0.1056	0.1883	0.4388	0.7312
	DLinear	0.1240	0.2597	0.4073	0.5579
	RLinear	0.1302	0.2281	0.5620	0.9550
	TimeGrad	0.1352	0.1784	0.2653	0.4027
	Prediction-Only	0.1375	0.1692	0.2776	0.3774
	COROS	**0.1012**	**0.1601**	**0.2441**	**0.3101**

5.3 Experiments on LTSF Datasets for Robustness

Next, we evaluate robustness to corrupted historical inputs using standard LTSF benchmarks. Following the protocol described in Section 5.1, we add zero-mean Gaussian noise to the *history window only* at test time with increasing noise scale $\sigma \in \{0.0, 0.5, 1.0, 1.5\}$ (in units of the data standard deviation), and report MSE on the 24-step forecast horizon.

As shown in Table 3, COROS achieves the lowest MSE across all datasets and noise levels, and exhibits lowest MSE at every noise level. Figure 4 also provides representative qualitative examples,

showing that COROS's forecasts remain stable even under severe input corruption (e.g., $\sigma = 0.5$), where the observed history becomes close to pure noise. Notably, Transformer-based baselines (PatchTST and iTransformer) exhibit steep error growth as σ increases, indicating high sensitivity to corrupted histories. However, COROS (built on a DiffWave-style backbone) shows substantially smaller degradation across all noise levels.



(a) Traffic (b) Electricity

Figure 4: Representative samples of COROS's robustness across increasing injected noise from $\sigma \in \{0.0, 0.5, 1.0, 1.5\}$ on TRAFFIC (a) and ELECTRICITY (b).

Additionally, we apply a standard noise-augmentation technique for robust training on PatchTST, denoted by PatchTST+NA. We inject small Gaussian noise ($\sigma = 0.1$) only into the history sequence. Noise-augmentation method improves robustness of predictor, but reduces performance on almost-clean history since added noise blurs high-frequency information of time-series, consistent with previous survey paper (Wen et al., 2021). COROS achieves lower MSE than TimeGrad (Rasul et al., 2021) across all datasets and noise scales in Table 3. We additionally report $CRPS_{sum}$ results under corrupted histories in Appendix A.6.

5.4 ABLATIONS ON THE NUMBER OF SAMPLING STEPS

As indicated by the results, there are optimal "sweet spots" for the number of steps depending on the dataset. For example, COROS requires relatively fewer diffusion steps on datasets like Exchange and Electricity, whereas it benefits from higher steps on the Solar dataset to achieve the best performance. However, a large number of diffusion steps increases sampling time; we select a practical step budget (Table 15 in Appendix D) to balance accuracy and latency.

Table 4: $CRPS_{sum}$ by sampling steps.

Steps	Exchange	Electricity	Solar
50	.0057±.0003	.0168±.0003	.4540±.0125
100	**.0054±.0002**	**.0165±.0005**	.2829±.0090
200	.0057±.0002	.0168±.0007	.2501±.0070
250	.0059±.0004	.0166±.0005	.2313±.0059
500	.0057±.0002	.0166±.0002	**.2155±.0089**

We also point out that a notable distinction of COROS, compared to other diffusion-based forecasting models such as CSDI (Tashiro et al., 2021) and TimeGrad (Rasul et al., 2021), is its ability to adjust the number of sampling steps without the need for additional training at each specific step. This flexibility incurs no retraining cost, although sampling time scales with the number of steps.

6 CONCLUSION

We propose COROS: **CO**ntext-**RO**bust time-series forecasting with **S**core-based diffusion models, leveraging diffusion models to robustly forecast time-series data under anomalous or corrupted inputs. COROS effectively suppresses noisy conditions by generating the entire sequence, balancing historical and future components. Moreover, we introduce classifier-free guidance into diffusion-based forecasting, enhancing predictive accuracy. Extensive experiments demonstrate our model consistently outperforms 13 baselines across 3 LTSF and 5 probabilistic forecasting datasets.

Limitations. Our robustness evaluation injects controlled corruptions (Gaussian, spike, and quantization) into standard benchmark datasets at test time, since naturally corrupted series rarely provide clean references for measuring forecast degradation. Evaluating COROS on real-world data with naturally occurring anomalies (e.g. sensor faults) is left for future work.

AI USE STATEMENT

We used LLM to assist with manuscript drafting and editing, LaTeX formatting, and the interpretation and presentation of experimental results. The authors reviewed and revised the AI-assisted material for consistency with the underlying experiments and theoretical analysis and take responsibility for the final manuscript.

ETHICS STATEMENT

This study uses publicly available time-series benchmarks for their intended research purposes and credits the original datasets and baseline implementations. Our experiments evaluate forecasting robustness through controlled perturbations of benchmark inputs.

REPRODUCIBILITY STATEMENT

Experimental protocols are described in Section 5.1, with dataset configurations and hyperparameters in Appendix D. Appendix G provides the theoretical assumptions and proofs. Appendices B and C detail the sensitivity and Jacobian diagnostics, including trajectory sampling and evaluation settings.

REFERENCES

Hanqun Cao, Cheng Tan, Zhangyang Gao, Yilun Xu, Guangyong Chen, Pheng-Ann Heng, and Stan Z Li. A survey on generative diffusion models. *IEEE Transactions on Knowledge and Data Engineering*, 2024.

Jooyoung Choi, Jungbeom Lee, Chaehun Shin, Sungwon Kim, Hyunwoo Kim, and Sungroh Yoon. Perception prioritized training of diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11472–11481, 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *CoRR*, abs/2105.05233, 2021. URL https://arxiv.org/abs/2105.05233.

Xinyao Fan, Yueying Wu, Chang Xu, Yuhao Huang, Weiqing Liu, and Jiang Bian. Mg-tsd: Multi-granularity time series diffusion models with guided learning process, 2024. URL https://arxiv.org/abs/2403.05751.

Marco Fraccaro, Simon Kamronn, Ulrich Paquet, and Ole Winther. A disentangled recognition and nonlinear dynamics model for unsupervised learning, 2017. URL https://arxiv.org/abs/1710.05741.

GM Harshvardhan, Mahendra Kumar Gourisaria, Manjusha Pandey, and Siddharth Swarup Rautaray. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review*, 38:100285, 2020.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *CoRR*, abs/2006.11239, 2020. URL https://arxiv.org/abs/2006.11239.

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J. Fleet. Video diffusion models, 2022. URL https://arxiv.org/abs/2204.03458.

Robin John Hyndman, Anne B Koehler, J Keith Ord, and Ralph David Snyder. *Forecasting with Exponential Smoothing: The State Space Approach*. Springer, 2008. ISBN 9783540719168.

Aapo Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005. URL http://jmlr.org/papers/v6/hyvarinen05a.html.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting, 2023. URL https://arxiv.org/abs/2307.11494.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis, 2021. URL https://arxiv.org/abs/2009.09761.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. *CoRR*, abs/1703.07015, 2017. URL http://arxiv.org/abs/1703.07015.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping, 2023. URL https://arxiv.org/abs/2305.10721.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical Transactions of the Royal Society A*, 379(2194):20200209, 2021.

Haksoo Lim, Minjung Kim, Sewon Park, and Noseong Park. Regular time-series generation using sgm, 2023a. URL https://arxiv.org/abs/2301.08518.

Haksoo Lim, Sewon Park, Minjung Kim, Jaehoon Lee, Seonkyu Lim, and Noseong Park. Madsgm: Multivariate anomaly detection with score-based generative models. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, CIKM '23, pp. 1411–1420, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9798400701245. doi: 10.1145/3583780.3614956. URL https://doi.org/10.1145/3583780.3614956.

Haksoo Lim, Minjung Kim, Sewon Park, Jaehoon Lee, and Noseong Park. TSGM: Regular and irregular time-series generation using score-based generative models, 2024. URL https://openreview.net/forum?id=nFG1YmQTqi.

Yiming Liu, Kezhao Liu, Yao Xiao, Ziyi Dong, Xiaogang Xu, Pengxu Wei, and Liang Lin. Towards understanding the robustness of diffusion-based purification: A stochastic perspective, 2025. URL https://arxiv.org/abs/2404.14309.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting, 2024. URL https://arxiv.org/abs/2310.06625.

Helmut Lütkepohl. *New Introduction to Multiple Time Series Analysis*. Number 978-3-540-27752-1 in Springer Books. Springer, September 2005. ISBN ARRAY(0x505580b8). doi: 10.1007/978-3-540-27752-1. URL https://ideas.repec.org/b/spr/sprbok/978-3-540-27752-1.html.

Ricardo P Masini, Marcelo C Medeiros, and Eduardo F Mendes. Machine learning advances for time series forecasting. *Journal of economic surveys*, 37(1):76–111, 2023.

John A. Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I. Budak Arpinar, and Ninghao Liu. A survey of deep learning and foundation models for time series forecasting, 2024. URL https://arxiv.org/abs/2401.13912.

Byeonghu Na, Yeongmin Kim, HeeSun Bae, Jung Hyun Lee, Se Jung Kwon, Wanmo Kang, and Il-Chul Moon. Label-noise robust diffusion models, 2024. URL https://arxiv.org/abs/2402.17517.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers, 2023. URL https://arxiv.org/abs/2211.14730.

Bernt Øksendal. *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, 6th edition, January 2014. ISBN 3540047581. URL http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/3540047581.

Achraf Oussidi and Azeddine Elhassouny. Deep generative models: Survey. In *2018 International conference on intelligent systems and computer vision (ISCV)*, pp. 1–8. IEEE, 2018.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs Bergmann, and Roland Vollgraf. Multi-variate probabilistic time series forecasting via conditioned normalizing flows. *CoRR*, abs/2002.06103, 2020. URL https://arxiv.org/abs/2002.06103.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. *CoRR*, abs/2101.12072, 2021. URL https://arxiv.org/abs/2101.12072.

Jože Rožanec, Elena Trajkova, Klemen Kenda, Blaž Fortuna, and Dunja Mladenić. Explaining bad forecasts in global time series models. *Applied Sciences*, 11(19), 2021. ISSN 2076-3417. doi: 10.3390/app11199243. URL https://www.mdpi.com/2076-3417/11/19/9243.

David Salinas, Michael Bohlke-Schneider, Laurent Callot, Roberto Medico, and Jan Gasthaus. High-dimensional multivariate forecasting with low-rank gaussian copula processes. *CoRR*, abs/1910.03002, 2019. URL http://arxiv.org/abs/1910.03002.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *CoRR*, abs/2011.13456, 2020. URL https://arxiv.org/abs/2011.13456.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models, 2021. URL https://arxiv.org/abs/2101.09258.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. *CoRR*, abs/2107.03502, 2021. URL https://arxiv.org/abs/2107.03502.

José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: A survey. *Big Data*, 9(1):3–21, 2021a. doi: 10.1089/big.2020.0159. URL https://doi.org/10.1089/big.2020.0159. PMID: 33275484.

José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting: a survey. *Big Data*, 9(1):3–21, 2021b.

Roy van der Weide. Go-garch: A multivariate generalized orthogonal garch model. *Journal of Applied Econometrics*, 17(5):549–564, 2002. ISSN 08837252, 10991255. URL http://www.jstor.org/stable/4129271.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural Computation*, 23(7):1661–1674, 2011. doi: 10.1162/NECO_a_00142.

Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, IJCAI-2021, pp. 4653–4660. International Joint Conferences on Artificial Intelligence Organization, August 2021. doi: 10.24963/ijcai.2021/631. URL http://dx.doi.org/10.24963/ijcai.2021/631.

Tijin Yan, Hongwei Zhang, Tong Zhou, Yufeng Zhan, and Yuanqing Xia. Scoregrad: Multivariate probabilistic time series forecasting with continuous energy-based generative models. *CoRR*, abs/2106.10121, 2021. URL https://arxiv.org/abs/2106.10121.

Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models, 2021. URL https://arxiv.org/abs/2106.06041.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting?, 2022. URL https://arxiv.org/abs/2205.13504.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting, 2022. URL https://arxiv.org/abs/2201.12740.

A Additional Ablation Study

A.1 Comparison of LTSF Baselines within Our Environment

Throughout the main paper we used the standard predictor–corrector (PC) sampler (which we denote "Naive") with optional classifier-free guidance (CFG). Because LTSF methods such as DLinear (Zeng et al., 2022), FEDformer (Zhou et al., 2022) and PatchTST (Nie et al., 2023) are trained via deterministic regression losses, comparing them against our method gives a fairer picture of pure point forecasting quality.

To evaluate, we follow the "moderate-horizon" configuration used 72 historical steps → 24-step forecast for Solar, Electricity, Taxi, and Wiki. Metrics are RMSE and MAE (lower is better), which are standard metrics for deterministic forecasting methods. COROS results are averaged over five seeds; ± indicates one-standard-deviation.

Table 5: Moderate-horizon forecasting results.

Model	Dataset	Approach	RMSE	MAE
DLinear	Solar	–	30.12	18.60
	Electricity	–	409.82	49.88
	Taxi	–	5.07	3.34
	Wiki	–	6887.80	1471.84
FEDformer	Solar	–	30.08	19.36
	Electricity	–	602.59	80.57
PatchTST	Solar	–	31.00	18.86
	Electricity	–	348.36	45.28
COROS	Solar	CFG	30.34 ± 0.12	14.02 ± 0.04
		Naive	$\mathbf{28.53 \pm 0.43}$	$\mathbf{12.46 \pm 0.13}$
	Electricity	CFG	$\mathbf{321.85 \pm 36.44}$	40.39 ± 0.10
		Naive	326.00 ± 52.76	$\mathbf{40.35 \pm 0.12}$
	Taxi	CFG	$\mathbf{3.95 \pm 0.00}$	$\mathbf{2.61 \pm 0.00}$
		Naive	4.15 ± 0.00	2.74 ± 0.00
	Wiki	CFG	5910.28 ± 12.03	$\mathbf{698.78 \pm 0.64}$
		Naive	$\mathbf{5905.92 \pm 70.48}$	718.16 ± 17.88

Table 5 underscores that COROS remains strong even when evaluated with a fully deterministic sampler, while also highlighting its stability compared to popular regression-style LTSF architectures.

Point-forecast errors in the original data units provide a complementary view of these gains. In Table 5, the best COROS variant reduces MAE relative to the strongest reported regression baseline by 33% on Solar (18.60 to 12.46), 11% on Electricity (45.28 to 40.35), 22% on Taxi (3.34 to 2.61), and 53% on Wiki (1471.84 to 698.78).

Notably, Robustness in higher dimensions is worth noting. As dimensionality increases (Electricity → Wiki), LTSF models' errors grow quickly, whereas COROS's degradation is moderate—confirming our full-sequence denoising advantage. Moreover, resource constraints also became a concern. FEDformer required custom frequency hacks on Electricity and failed on Wiki; PatchTST ran out of GPU memory on several settings. COROS, while generative, scales gracefully once trained—it needs no architectural changes for different frequencies or horizons.

A.2 Ablations on LTSF baselines

We conduct an ablation study on robustness test across different history lengths. Table 6 shows that COROS consistently outperforms other LTSF baselines across different setting.

Table 6: Comparison of noise robustness (MSE) across different history lengths (72, 96, 120) and prediction lengths (24, 48, 72) on Traffic dataset. COROS consistently demonstrates superior stability against increasing noise scales.

(a) History Length 72

Hor.	Model	MSE on noise scale σ			
		0.0	0.5	1.0	1.5
24	PatchTST	0.3556	0.6217	0.8776	1.1310
	DLinear	0.6253	0.8055	1.0019	1.2023
	COROS	**0.2510**	**0.3663**	**0.5790**	**0.7776**
48	PatchTST	**0.2431**	0.5852	0.9105	1.1961
	DLinear	0.6185	0.7854	1.1564	1.5274
	COROS	0.2666	**0.3351**	**0.5204**	**0.7857**
72	PatchTST	0.3186	0.4803	0.7383	0.9737
	DLinear	0.4910	0.6467	0.9930	1.3393
	COROS	**0.3598**	**0.4687**	**0.6334**	**0.8023**

(b) History Length 96

Hor.	Model	MSE on noise scale σ			
		0.0	0.5	1.0	1.5
24	PatchTST	0.5393	0.8065	1.3778	1.9428
	DLinear	0.5905	0.8093	1.2956	1.7820
	COROS	**0.2296**	**0.3213**	**0.4768**	**0.6182**
48	PatchTST	0.3122	0.5029	0.8164	1.1058
	DLinear	0.4934	0.6682	1.0566	1.4448
	COROS	**0.2664**	**0.3616**	**0.5296**	**0.6811**
72	PatchTST	0.3064	0.4514	0.7057	0.9404
	DLinear	0.4637	0.6216	0.9722	1.3229
	COROS	**0.3031**	**0.4016**	**0.5825**	**0.7438**

(c) History Length 120

Hor.	Model	MSE on noise scale σ			
		0.0	0.5	1.0	1.5
24	PatchTST	0.3130	0.5995	1.0940	1.5595
	DLinear	0.5029	0.7251	1.2186	1.7120
	COROS	**0.2057**	**0.3365**	**0.4806**	**0.6162**
48	PatchTST	0.2965	0.4682	0.7809	1.0733
	DLinear	0.4527	0.6296	1.0229	1.4162
	COROS	**0.2442**	**0.3425**	**0.5098**	**0.6613**
72	PatchTST	0.3052	0.4544	0.7094	0.9433
	DLinear	0.4624	0.6248	0.9860	1.3472
	COROS	**0.2826**	**0.3484**	**0.5389**	**0.7061**

A.3 Ablations on different noise types

As another ablation, we apply different noise types other than Gaussian noise injection. Table 7 shows that COROS consistently maintains competitive performance on increasing noise scale. We describe detailed noise setup in Appendix D.

A.4 Training Cost Comparison

We compare training cost in high-dimensional settings with representative baselines on the Traffic dataset (862 variates). Figure 5 plots MSE against wall-clock training time per iteration, where bubble size indicates GPU memory footprint. As shown in the figure, COROS achieves the lowest MSE while keeping memory usage moderate. DLinear is the fastest and lightest baseline (0.91GB, 85ms/iter), but its error is also the largest. Transformer-based baselines such as iTransformer and PatchTST require substantially larger memory (7.50GB and 8.58GB, respectively), and PatchTST is also the slowest model. Notably, compared with the diffusion baseline CSDI, COROS uses nearly identical resources (2.94GB vs. 2.88GB, 413ms/iter vs. 384ms/iter) while achieving clearly lower MSE. These results support the accuracy benefit of full-sequence denoising at a compute and memory cost comparable to CSDI.

Table 7: Performance comparison on different injected noise type.

Dataset	Noise type	MSE on noise scale σ			
		0.0	0.5	1.0	1.5
Traffic	Gaussian noise	0.2510	0.3663	0.5790	0.7776
	Spike	0.2512	0.3154	0.3806	0.4504
	Quantization	0.2511	0.2505	0.2492	0.2498
Electricity	Gaussian noise	0.1024	0.1340	0.1881	0.2436
	Spike	0.1026	0.1001	0.1009	0.1101
	Quantization	0.1024	0.1024	0.1017	0.1024
Weather	Gaussian noise	0.1012	0.1601	0.2441	0.3101
	Spike	0.1010	0.1141	0.1333	0.1714
	Quantization	0.1014	0.1012	0.1032	0.1046

A.5 Memory Comparison with Observation Self-Guidance

We measure peak GPU memory for the Solar comparison in Table 1. OSG and COROS achieve closely matched $CRPS_{sum}$, with overlapping standard deviations. Their peak memory requirements are 18,043 MiB and 3,407 MiB, respectively (Table 8). Thus, COROS achieves comparable forecasting quality with $5.3\times$ lower peak memory usage. OSG computes an observation-guidance gradient through the score network at each reverse step, retaining activations for the backward pass. COROS evaluates the learned conditional score directly during denoising. On the 370-variate main Electricity benchmark, the OSG evaluation exceeds the available memory budget.

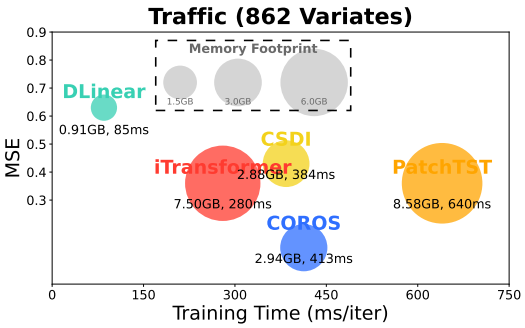


Figure 5: Training comparison with representative baselines.

The matched prediction-only comparison in Table 1 also isolates the forecasting benefit of total-sequence denoising: $CRPS_{sum}$ improves from 0.2871 to 0.2501 on Solar and from 0.0210 to 0.0153 on Electricity.

Table 8: Forecasting quality and peak GPU memory on Solar under the comparison in Table 1.

Method	$CRPS_{sum}$	Peak GPU memory (MiB)
OSG	0.2490 ± 0.0094	18,043
COROS	0.2501 ± 0.0080	3,407

A.6 Probabilistic Forecasting under Corrupted Histories

We evaluate $CRPS_{sum}$ on the LTSF Electricity (ECL) and Weather datasets using the corruption protocol of Table 3: Gaussian noise is applied to half of the historical time indices, $\sigma \in \{0, 0.5, 1.0, 1.5\}$,

and the prediction horizon is 24 steps. Results are averaged over five seeds, and the comparison uses the matched prediction-only diffusion control. Table 9 reports the mean and standard deviation.

Table 9: $CRPS_{sum}$ under corrupted historical inputs. Each entry reports mean $\pm$ standard deviation over five seeds.

Dataset	Method	$\sigma = 0.0$	0.5	1.0	1.5
Electricity	Prediction-Only	$.020 \pm .001$	$.027 \pm .001$	$.035 \pm .002$	$.064 \pm .002$
	COROS	$.018 \pm .001$	$.024 \pm .001$	$.034 \pm .002$	$.047 \pm .001$
Weather	Prediction-Only	$.058 \pm .005$	$.064 \pm .007$	$.087 \pm .008$	$.111 \pm .011$
	COROS	$.050 \pm .006$	$.063 \pm .006$	$.085 \pm .009$	$.101 \pm .009$

COROS achieves a lower mean $CRPS_{sum}$ at every noise scale on both datasets. From $\sigma = 0$ to $\sigma = 1.5$, $CRPS_{sum}$ increases by 0.029 for COROS and 0.044 for the prediction-only control on Electricity, and by 0.051 and 0.053 on Weather. The Electricity error-growth ratio is approximately $1.5\times$, and the Weather ratio is $1.04\times$. The same ordering appears in the MSE error growth summarized in Table 10. The probabilistic score aggregates variates and normalizes by target magnitude, which moderates the separation between methods relative to coordinate-level MSE.

Table 10: MSE error growth from $\sigma = 0$ to $\sigma = 1.5$, computed from Table 3.

Dataset	Prediction-Only	COROS
Traffic	0.5396	0.5266
Electricity	0.3370	0.1412
Weather	0.2399	0.2089

From $\sigma = 0$ to $\sigma = 1.5$, the MSE increases of the matched prediction-only control and COROS are 0.5396 and 0.5266 on Traffic, 0.3370 and 0.1412 on Electricity, and 0.2399 and 0.2089 on Weather. On Electricity, COROS reduces this error growth by 58%.

B Sensitivity to Latent History and Observed Conditions

The latent history serves as an auxiliary variable that is updated throughout sampling. The observed history anchors this state, and the evolving latent history supplies a progressively denoised context for forecasting. This interaction retains useful temporal information and reduces the propagation of corrupted observations into the forecast.

Table 11: Relative latent-history sensitivity U_{latent} under clean and corrupted conditions. Ratios use the corrupted value relative to the clean value, as reported in the diagnostic.

	Traffic			Electricity			Weather		
$1-t$	Clean	Corrupted	Ratio	Clean	Corrupted	Ratio	Clean	Corrupted	Ratio
0.000	.361	.408	1.13×	.292	.310	1.06×	.189	.211	1.11×
0.122	.268	.340	1.27×	.244	.260	1.07×	.111	.142	1.28×
0.245	.229	.269	1.17×	.204	.237	1.16×	.073	.106	1.45×
0.367	.211	.234	1.11×	.211	.213	1.01×	.067	.098	1.45×
0.489	.146	.193	1.32×	.183	.177	0.97×	.039	.088	2.26×
0.632	.101	.174	1.72×	.165	.201	1.22×	.039	.075	1.91×
0.754	.111	.133	1.21×	.142	.170	1.20×	.041	.104	2.55×
0.877	.092	.132	1.43×	.150	.171	1.14×	.034	.095	2.77×

We measure how the learned forecast score responds to the latent history and to the observed condition. The diagnostic uses trained COROS checkpoints on the Traffic, Electricity, and Weather test splits with history/prediction lengths of $72/24$ and the sampler of Section 5.1. We evaluate 14 trajectories per condition, formed from seven windows and two noise repeats, at nine diffusion times. The corrupted condition applies Gaussian noise with $\sigma = 1.0$ to half of the historical time indices. For each paired evaluation, the trajectory, diffusion time, and latent state are shared, and the observed condition is changed from clean to corrupted.

For the forecast-coordinate score s_p, we estimate the largest singular values of the two Jacobians using matrix-free operations:

$$L_{h\to p}(t) = \sigma_{\max}\left(\frac{\partial s_p}{\partial \mathbf{x}_t^{\text{hist}}}\right), \quad L_{c\to p}(t) = \sigma_{\max}\left(\frac{\partial s_p}{\partial \mathbf{c}}\right), \quad U_{\text{latent}}(t) = \frac{L_{h\to p}(t)}{L_{h\to p}(t) + L_{c\to p}(t)}.$$

The condition Jacobian is taken with respect to the historical entries selected for corruption. The latent history follows the VP diffusion scale, while the observed condition remains in data units; consequently, we interpret paired clean-to-corrupted changes at matched t and state. We retain eight diffusion times for numerical stability, excluding the terminal grid point where the score parameterization $s_\theta = -\epsilon_\theta/\sigma_t$ amplifies numerical error as σ_t approaches zero.

Table 11 reports U_{latent} for each retained time. It increases under corruption at all eight times on Traffic and Weather and at seven of eight times on Electricity. The paired median increase is $1.10\times$ on Electricity and $1.66\times$ on Weather, with increases at 69/112 and 98/112 paired states, respectively. On Traffic, the median observed-condition sensitivity is $0.76\times$ its clean value and the latent-history sensitivity is $1.02\times$. Weather exhibits the same pattern, at $0.66\times$ and $1.12\times$, respectively. On Electricity, both sensitivities increase, with the larger increase in the latent-history sensitivity ($1.15\times$ versus $1.07\times$). Across the three datasets, these paired changes show increased relative sensitivity to the evolving latent history under corruption.

C Jacobian Diagnostics along Reverse Trajectories

We evaluate the history and forecast score Jacobians along the reverse trajectories described in Appendix B to examine the local behavior associated with (A2) and (A3). With s_h and s_p denoting the history and forecast score coordinates, define

$$m_h^{\text{eff}}(t) = -\lambda_{\max}\left(\operatorname{Sym}\frac{\partial s_h}{\partial \mathbf{x}_t^{\text{hist}}}\right), \qquad m_p(t) = -\lambda_{\max}\left(\operatorname{Sym}\frac{\partial s_p}{\partial \mathbf{x}_t^{\text{pred}}}\right),$$

where $\operatorname{Sym}(J) = (J + J^\top)/2$. These quantities measure the least contractive local direction in each score block, with positive values corresponding to local contraction. We also evaluate the time-integrated history factor over the retained diffusion grid,

$$K_h(t) = \exp\left(-\int_t^1 g(s)^2 m_h^{\text{eff}}(s)\, ds\right).$$

The terminal grid point is excluded using the same numerical criterion as in Appendix B.

Table 12 reports the median local values. On Traffic, $m_h^{\text{eff}} > 0$ at 84.8% of sampled clean states and 87.5% of sampled corrupted states. Its median is positive at seven of the eight retained times, with at least 11 of 14 trajectories satisfying positivity at each of those times. Electricity and Weather have positive history-block medians at all eight retained times under both conditions. The integrated history factor is below one on all 14 trajectories in each dataset under both clean and corrupted conditions.

The forecast block exhibits a transition from positive to negative median curvature as the reverse process progresses. Traffic has positive medians through $1 - t = 0.367$, Electricity through $1 - t = 0.489$, and Weather through $1 - t = 0.632$ for clean conditions and $1 - t = 0.489$ for corrupted conditions. On Traffic, positive forecast-block curvature occurs at 50.0% of sampled clean states and 50.9% of sampled corrupted states; the positive region accounts for roughly 60% of $G = \int_0^1 g(s)^2\, ds$. Together, the measurements describe persistent integrated history contraction and a time-dependent curvature profile for the forecast coordinates.

Table 12: Median local history and forecast score curvature along reverse trajectories. Each dataset uses 14 trajectories per condition and the eight retained diffusion times.

	Traffic				**Electricity**				**Weather**			
	m_h^{eff}		m_p		m_h^{eff}		m_p		m_h^{eff}		m_p	
$1-t$	**Clean**	**Corrupted**	**Clean**	**Corrupted**	**Clean**	**Corrupted**	**Clean**	**Corrupted**	**Clean**	**Corrupted**	**Clean**	**Corrupted**
0.000	0.578	0.581	0.859	0.860	0.586	0.574	0.573	0.571	0.795	0.773	0.882	0.880
0.122	0.490	0.531	0.850	0.846	0.584	0.550	0.568	0.563	0.802	0.734	0.881	0.889
0.245	0.502	0.518	0.807	0.828	0.620	0.599	0.568	0.561	0.803	0.731	0.878	0.864
0.367	0.389	0.453	0.468	0.480	0.667	0.650	0.543	0.528	0.818	0.739	0.769	0.719
0.489	0.296	0.275	-0.940	-0.640	0.724	0.698	0.483	0.390	0.865	0.809	0.400	0.362
0.632	0.157	0.157	-2.180	-2.060	0.973	0.796	-0.332	-0.461	1.064	1.014	0.154	-0.398
0.754	0.136	0.243	-4.580	-4.370	1.405	1.139	-1.846	-2.167	1.708	1.247	-0.304	-1.196
0.877	-0.021	-0.057	-13.600	-13.800	3.145	2.626	-7.484	-6.464	4.658	2.762	-1.621	-3.640

D DESCRIPTIONS OF DATASETS, HYPERPARAMETERS AND MISCELLANEOUS ENVIRONMENTS

In this section, we describe model architecture, datasets and hyperparameters.

The GluonTS multivariate benchmark suite spans 8–2,000 variates and includes continuous-valued Exchange, Solar, and Electricity series and count-valued Taxi and Wiki series. The range of frequencies, horizons, and dimensionalities evaluates forecasting performance across distinct data regimes. Dataset configurations are given in Table 15.

We select γ from $\{0.0, 0.1, 0.3, 0.5, 0.8, 1.0\}$ using clean validation data and keep the selected value fixed throughout training and corrupted-input evaluation. Training and model selection use clean data; the corruption operators are applied to the historical input at test time.

Table 13 gives the standard LTSF splits used for Traffic, Electricity, and Weather, following the evaluation settings of PatchTST and iTransformer. All values denote the number of timesteps in each split. The Electricity series in this table has 321 variates; the main probabilistic Electricity benchmark in Table 15 has 370 variates.

Table 13: Dataset splits and sampling frequencies for the LTSF robustness experiments.

Dataset	Variates	Train	Validation	Test	Frequency
Traffic	862	12,185	1,757	3,509	Hourly
Electricity	321	18,317	2,633	5,261	Hourly
Weather	21	36,792	5,271	10,540	10 minutes

We first describe the diffusion architecture used in COROS. To effectively capture the conditional score function along the temporal axis, we adapt DiffWave (Kong et al., 2021) to our settings. Since COROS is based on DiffWave, we highlight the key differences. The input consists of the diffusion timestep, the diffused target data, and historical data, i.e. $t, \mathbf{x}^{\text{hist}}, \mathbf{x}_t^{\text{total}}$. Consistent with previous works (Ho et al., 2020; Kong et al., 2021), the timestep t is embedded into a continuous domain using sinusoidal embedding:

$$embedding(t) = [\sin(t/N^{0/d}, ..., \sin(t/N^{d-1/d}), \cos(t/N^{0/d}), ..., \cos(t/N^{d-1/d})]$$

, where d (embedding dimension) and N are hyperparameters set to 128 and 10,000, respectively. Furthermore, since our main diffusion-based forecasting baselines are DDPM methods, we use VP SDE and an Euler-Maruyama sampling predictor without corrector, which are a generalized formulation of DDPM and a default setting of VP SDE in Song et al. (2020), respectively. All experiments are conducted using the following software and hardware environments: UBUNTU 18.04 LTS, PYTHON 3.9.12, CUDA 9.1, NVIDIA Driver 470.141, i9 CPU, and GEFORCE RTX 2080 TI.

Table 14: Wall-clock inference latency of the Weather dataset (lower is better).

Method	Inference time (ms) ↓
TimeGrad	207.5
CSDI	23.6
COROS (ours)	24.1

To address concerns about diffusion sampling latency, we report wall-clock inference time of Weather dataset for LTSF tasks in Table 14. TimeGrad (Rasul et al., 2021) is substantially slower (207.5 ms) due to autoregressive denoising across the forecast horizon, whereas COROS achieves 24.1 ms, making it $\approx 8.6\times$ faster than TimeGrad and practically identical to the one-shot diffusion baseline CSDI (Tashiro et al., 2021) (23.6 ms; only 0.5 ms difference). Although COROS denoises the concatenated past–future sequence, its sampling loop is structurally the same as standard score-based one-shot diffusion (one forward pass per denoising step without extra per-step gradient/backward computations), and it reuses essentially the same DiffWave-style backbone parameters; thus the additional compute from including history coordinates has negligible impact on the wall-clock runtime in practice.

Next, we describe the noise-injection setup on LTSF experiments. For each train, validation, and test sample, we extract a window $\mathbf{x} = [\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}] \in \mathbb{R}^{(N+H)\times C}$, where $\mathbf{x}^{\text{hist}} \in \mathbb{R}^{N\times C}$ is the history window and $\mathbf{x}^{\text{pred}} \in \mathbb{R}^{H\times C}$ is the forecast horizon. To simulate corrupted conditioning, we apply a corruption operator $\mathcal{C}(\cdot)$ *only* to the history window:$\tilde{\mathbf{x}}^{\text{hist}} = \mathcal{C}(\mathbf{x}^{\text{hist}}), \tilde{\mathbf{x}}^{\text{pred}} = \mathbf{x}^{\text{pred}}$. All corruption magnitudes are expressed in the *normalized* space (i.e., in units of the data standard deviation). Unless stated otherwise, corruption is applied to half the randomly selected time indices. On selected timesteps t, we mainly use gaussian noise injection and apply other corruption methods:

- **Gaussian noise.** For a noise scale $\sigma > 0$, we corrupt the selected time indices by $\tilde{\mathbf{x}}_t^{\text{hist}} = \mathbf{x}_t^{\text{hist}} + \sigma\, \boldsymbol{\epsilon}_t, \;\; \boldsymbol{\epsilon}_t \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
- **Spike noise.** To model impulsive perturbations, we add Laplace noise: $\tilde{\mathbf{x}}_t^{\text{hist}} = \mathbf{x}_t^{\text{hist}} + \boldsymbol{\xi}_t, \;\; \boldsymbol{\xi}_t \sim \text{Laplace}(\mathbf{0}, \sigma\mathbf{I})$. where $\sigma > 0$ is the spike scale.
- **Quantization.** For a quantization step $\sigma > 0$, we quantize by rounding to the nearest multiple of δ: $\tilde{\mathbf{x}}_t^{\text{hist}} = \sigma \cdot \text{round}\big(\mathbf{x}_t^{\text{hist}}/\sigma\big)$.

Table 15: Description of datasets and hyperparameters.

	Dimension	Timesteps	Domain	γ	L_{hist}	L_{pred}	N_{step}	N_{iter}	w	N_{iter}^{CFG}	# test sample
Exchange	8	6071	$\mathbb{R}^+$	0.1	90	30	100	72	0.01	36	7
Solar	137	7009	$\mathbb{R}^+$	0.1	72	24	200	61	0.01	71	7
Electricity	370	5833	$\mathbb{R}^+$	0.5	72	24	50	44	0.01	66	7
Taxi	1214	1488	$\mathbb{N}$	0.1	48	24	50	14	0.1	24	56
Wiki	2000	792	$\mathbb{N}$	0.1	90	30	250	5	0.01	6	5

Algorithm 1 Training COROS via conditional DSM

Inputs: dataset $\mathcal{D} = \{(\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}})\}$; VP schedule $\beta(t)$; time sampler $t \sim \text{Unif}(0,1)$; loss weights γ.
Model: score network $s_\theta(t, \mathbf{x}_t^{\text{total}}, \mathbf{x}^{\text{hist}})$ predicting $\nabla_{\mathbf{x}_t^{\text{total}}} \log p_t(\mathbf{x}_t^{\text{total}} \mid \mathbf{x}^{\text{hist}})$, where $\mathbf{x}_t^{\text{total}} = [\mathbf{x}_t^{\text{hist}}, \mathbf{x}_t^{\text{pred}}]$.
Helpers: VP coefficients (α_t, σ_t) s.t. $\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \boldsymbol{\varepsilon}$; masks $M_{\text{hist}}, M_{\text{pred}}$ (same shape as $\mathbf{x}_t^{\text{total}}$) selecting history / future coordinates.
for each minibatch $(\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}) \sim \mathcal{D}$ **do**
  Sample $t \sim \text{Unif}(0,1)$; sample noises $\boldsymbol{\varepsilon}, \boldsymbol{\eta} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
  Form clean total sequence $\mathbf{x}^{\text{total}} = [\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}]$.
  Forward diffuse the total: $\mathbf{x}_t^{\text{total}} \leftarrow \alpha_t \mathbf{x}^{\text{total}} + \sigma_t \boldsymbol{\varepsilon}$.
  Score prediction: $\widehat{\mathbf{s}} \leftarrow s_\theta\big(t,\, \mathbf{x}_t^{\text{total}},\, \mathbf{x}^{\text{hist}}\big)$.
  DSM target: For VP, $\mathbf{s}^\star(t, \mathbf{x}_t^{\text{total}}, \mathbf{x}^{\text{hist}}) = -\dfrac{\boldsymbol{\varepsilon}}{\sigma_t}$ {(denoising score target)}
  Weighted loss (history vs. future):

$$\mathcal{L}(\theta) \;\leftarrow\; \gamma \left\| M_{\text{hist}} \odot (\widehat{\mathbf{s}} - \mathbf{s}^\star) \right\|_2^2 \;+\; \left\| M_{\text{pred}} \odot (\widehat{\mathbf{s}} - \mathbf{s}^\star) \right\|_2^2.$$

  Update $\theta \leftarrow \theta - \eta_\theta \nabla_\theta \mathcal{L}(\theta)$.
end for

E CLASSIFIER-FREE GUIDANCE (CFG)

While CFG is a well-known technique in discrete-time DDPMs, we adapt it to the continuous score-SDE framework. We thus provide additional details on how our continuous-time interpretation modifies the standard CFG approach for time-series.

To incorporate an auxiliary classifier in naïve conditional generation, Dhariwal & Nichol (2021) introduced classifier guidance, modifying the standard denoising process by adjusting the estimated noise. Originally, $\epsilon(\mathbf{x}_t|\mathbf{c}) \sim -\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c})$ is replaced with $\tilde{\epsilon}(\mathbf{x}_t|\mathbf{c}) = \epsilon(\mathbf{x}_t|\mathbf{c}) - w\sigma_t \nabla_{\mathbf{x}_t} \log p(\mathbf{c}|\mathbf{x}_t)$, where w is a weighting term, and an additional classifier is trained to calculate $p(\mathbf{c}|\mathbf{x}t)$. From the perspective of score-based SDEs, this approach can be interpreted as altering the score function $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c})$ to $\nabla_{\mathbf{x}_t} \log \tilde{p}(\mathbf{x}_t|\mathbf{c}) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c}) + w \nabla_{\mathbf{x}_t} \log p(\mathbf{c}|\mathbf{x}_t) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t|\mathbf{c}) p(\mathbf{c}|\mathbf{x}_t)^w$, which means $\tilde{p}(\mathbf{x}_t|\mathbf{c}) \sim p(\mathbf{x}_t|\mathbf{c}) p(\mathbf{c}|\mathbf{x}_t)^w$ and effectively incorporating the classifier into the generative process.

To address the dependency on an additional classifier, Ho & Salimans (2022) proposed classifier-free guidance (**CFG**), allowing the generation process to be guided without the need for a separately trained classifier. In CFG, the model learns the modified noise estimate $\tilde{\epsilon}(\mathbf{x}_t|\mathbf{c}) = (1+w)\epsilon(\mathbf{x}_t|\mathbf{c}) - w\epsilon(\mathbf{x}_t)$ by

training a single model that handles both conditional and unconditional generations. This is achieved by training with zero-padding for the unconditional case, resulting in $\tilde{\epsilon}_\theta(\mathbf{x}_t, \mathbf{c}) = (1 + w)\epsilon_\theta(\mathbf{x}_t, \mathbf{c}) - w\epsilon_\theta(\mathbf{x}_t, \mathbf{0})$.

Generally, applying Classifier-Free Guidance (CFG) to historical data can be seen as a subset of conditional generation, which is a natural concept. However, in many existing conditional diffusion approaches (e.g., text-to-image or time-series forecasting), noisy conditioning inputs can degrade performance, as highlighted by (Na et al., 2024) (see Appendix C of their paper). This is why CFG has rarely been adopted in time-series forecasting methods so far—if the historical data is noisy, naive CFG can amplify that noise and harm predictions.

In our framework, though, COROS resists historical noise during the diffusion process, alleviating overdependency on noisy historical sequence. Joint denoising reduces sensitivity to historical noise, and CFG strengthens the useful conditional signal to guide the prediction. This synergy between COROS and CFG is described in Section 3, where we show that while naive CFG can degrade performance under noisy conditions, our approach remains robust precisely because of the joint denoising mechanism.

F Explanations about Baselines

Classical Methods.

- **VAR/VAR-Lasso** (Lütkepohl, 2005): Vector AutoRegression (VAR) estimates linear dependencies across multiple time-series. VAR-Lasso adds an ℓ_1 penalty to mitigate overfitting in high-dimensional data.
- **GARCH** (van der Weide, 2002): A model that captures time-varying volatility (conditional variance), particularly popular in financial contexts.
- **VES** (Hyndman et al., 2008): A vectorized exponential smoothing method using state-space formulations to handle multivariate trends.

VAE/State-Space Method.

- **KVAE** (Fraccaro et al., 2017): Combines a Kalman Filter with a Variational Autoencoder to learn complex latent dynamics for forecasting.

Deep Learning-Based Methods.

- **Vec-LSTM (ind-scaling and low-copula)** (Salinas et al., 2019): LSTM-based multivariate forecasting; the "ind-scaling" version assumes independent output distributions, while the "low-copula" version models their joint distribution via copulas.
- **GP scaling/copula** (Salinas et al., 2019): Similar to Vec-LSTM but leverages Gaussian Processes for uncertainty estimation, offering more flexible distributional modeling at higher computational cost.
- **Transformer MAF** (Rasul et al., 2020): A Transformer architecture combined with a Masked Autoregressive Flow for modeling complex, long-range dependencies in multivariate time series.

Diffusion-Based Methods.

- **MG-TSD** (Fan et al., 2024): A diffusion-based probabilistic time-series model that uses multi-granularity (fine→coarse) views of the series to guide the denoising process.
- **TimeGrad** (Rasul et al., 2021): A DDPM-based approach that generates forecasts autoregressively, one step at a time.
- **CSDI** (Tashiro et al., 2021): Primarily designed for time-series imputation but can also perform one-shot forecasting of the future horizon by treating it as a masked region.

LTSF Deterministic Baselines.

- **DLinear** (Zeng et al., 2022): A decomposition-linear model that applies simple linear projections for long-term forecasting.
- **RLinear** (Li et al., 2023): A linear regression baseline that directly maps the input window to the prediction horizon.
- **PatchTST** (Nie et al., 2023): A patch-based Transformer that models long contexts efficiently by grouping time steps into patches.
- **iTransformer** (Liu et al., 2024): An inverted Transformer that treats variables as tokens to better capture multivariate dependencies for LTSF.

Unlike these baselines, our **COROS** method denoises the entire time-series (both past and future) during generation. By jointly modeling historical and future observations, COROS can more effectively handle noisy inputs, leading to improved probabilistic forecasts.

G Detailed Proof

In this section, we give detailed proof of Theorems.

G.1 Noisy-cleaned alignment at initial steps

Theorem 2. *Suppose that p is a given data distribution and $\hat{p}$ is a distribution with anomaly data. Then the following inequality holds along the diffusion process:*

$$D_{KL}(\hat{p}_0||p_0) \geq D_{KL}(\hat{p}_t||p_t) \geq D_{KL}(\hat{p}_1||p_1) \approx 0,$$

where $t \in [0, 1]$.

Remark. Prior adversarial purification works Yoon et al. (2021); Liu et al. (2025) show adding noise removes high-frequency anomalies but also destroys the original data. Thus, backward sampling from Gaussian noise yields a denoised sample unrelated to the original unless guided by conditions. We adopt this principle for time-series forecasting: by generating the entire sequence (both past and future), COROS detaches from potentially noisy historical observations while still leveraging the condition for directional guidance.

Proof. We begin by expressing the KL-divergence of the left-hand term as:

$$\begin{aligned} D_{\mathrm{KL}}(\hat{p}_0(\mathbf{x}) \parallel p_0(\mathbf{x})) &= D_{\mathrm{KL}}(\hat{p}_0(\mathbf{x}) \parallel p_0(\mathbf{x})) - D_{\mathrm{KL}}(\hat{p}_s(\mathbf{x}) \parallel p_s(\mathbf{x})) + D_{\mathrm{KL}}(\hat{p}_s(\mathbf{x}) \parallel p_s(\mathbf{x})) \\ &= -\int_0^s \frac{\partial D_{\mathrm{KL}}(\hat{p}_t(\mathbf{x}) \parallel p_t(\mathbf{x}))}{\partial t} dt + D_{\mathrm{KL}}(\hat{p}_s(\mathbf{x}) \parallel p_s(\mathbf{x})). \end{aligned}$$

Next, consider the Fokker-Planck equation, which governs the evolution of the probability distribution over time:

$$\frac{\partial p_t(\mathbf{x})}{\partial t} = \nabla_{\mathbf{x}} \cdot \left(\frac{1}{2} g^2(t) p_t(\mathbf{x}) \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) - f(\mathbf{x}, t) p_t(\mathbf{x}) \right) = \nabla_{\mathbf{x}} \cdot \left(h_p(\mathbf{x}, t) p_t(\mathbf{x}) \right).$$

Now, the derivative of the KL-divergence with respect to time can be computed as follows:

$$
\begin{aligned}
\frac{\partial D_{\mathrm{KL}}(\hat{p}_t(\mathbf{x}) \,\|\, p_t(\mathbf{x}))}{\partial t} &= \frac{\partial}{\partial t} \int \hat{p}_t(\mathbf{x}) \log \frac{\hat{p}_t(\mathbf{x})}{p_t(\mathbf{x})} \, d\mathbf{x} \\
&= \int \frac{\partial \hat{p}_t(\mathbf{x})}{\partial t} \log \frac{\hat{p}_t(\mathbf{x})}{p_t(\mathbf{x})} \, d\mathbf{x} + \int \underbrace{\frac{\partial \hat{p}_t(\mathbf{x})}{\partial t} \, d\mathbf{x}}_{=0} - \int \frac{\hat{p}_t(\mathbf{x})}{p_t(\mathbf{x})} \frac{\partial p_t(\mathbf{x})}{\partial t} \, d\mathbf{x} \\
&= \int \nabla_{\mathbf{x}} \cdot (h_{\hat{p}}(\mathbf{x}, t) \hat{p}_t(\mathbf{x})) \log \frac{\hat{p}_t(\mathbf{x})}{p_t(\mathbf{x})} \, d\mathbf{x} - \int \frac{\hat{p}_t(\mathbf{x})}{p_t(\mathbf{x})} \nabla_{\mathbf{x}} \cdot (h_p(\mathbf{x}, t) p_t(\mathbf{x})) \, d\mathbf{x} \\
&= - \int \hat{p}_t(\mathbf{x}) \left[h_{\hat{p}}^{\top}(\mathbf{x}, t) - h_p^{\top}(\mathbf{x}, t) \right] \left[\nabla_{\mathbf{x}} \log \hat{p}_t(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right] d\mathbf{x} \\
&= -\frac{1}{2} \int \hat{p}_t(\mathbf{x}) g(t)^2 \left\| \nabla_{\mathbf{x}} \log \hat{p}_t(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_t(\mathbf{x}) \right\|_2^2 d\mathbf{x}. \leq 0
\end{aligned}
$$

Thus, we have shown that:

$$
D_{\mathrm{KL}}(\hat{p}_0(\mathbf{x}) \,\|\, p_0(\mathbf{x})) \geq D_{\mathrm{KL}}(\hat{p}_s(\mathbf{x}) \,\|\, p_s(\mathbf{x})) \geq D_{\mathrm{KL}}(\hat{p}_1(\mathbf{x}) \,\|\, p_1(\mathbf{x})).
$$

Since both $\hat{p}_1$ and p_1 follow a standard normal distribution, i.e., $\hat{p}_1, p_1 \sim \mathcal{N}(0, \mathbf{I})$, it follows that:

$$
D_{\mathrm{KL}}(\hat{p}_1(\mathbf{x}) \,\|\, p_1(\mathbf{x})) \approx 0
$$

Hence, the inequality in the theorem is proved.

□

G.2 Robustness of COROS

We assume the generative structure

$$
p(\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}, \tilde{\mathbf{x}}^{\text{hist}}) = p(\mathbf{x}^{\text{hist}}) \, p(\mathbf{x}^{\text{pred}} \mid \mathbf{x}^{\text{hist}}) \, p(\tilde{\mathbf{x}}^{\text{hist}} \mid \mathbf{x}^{\text{hist}})
$$

and apply independent VP forward kernels to $(\mathbf{x}^{\text{hist}}, \mathbf{x}^{\text{pred}}, \tilde{\mathbf{x}}^{\text{hist}})$ to obtain $(\mathbf{x}_t^{\text{hist}}, \mathbf{x}_t^{\text{pred}}, \mathbf{c}_t)$ with $\mathbf{c}_t \equiv \tilde{\mathbf{x}}_t^{\text{hist}}$. Throughout we abbreviate $p_t(\cdot) = p(\cdot \text{ at time } t)$.

Assumption 1. *(A1)* (Lipschitz-in-history score for the future) *Define* $\phi_{\mathbf{h}}(\mathbf{p}) := \nabla_{\mathbf{p}} \log p_t(\mathbf{p} \mid \mathbf{h})$. *There exists* $L > 0$ *such that* $\|\phi_{\mathbf{h}}(\mathbf{p}) - \phi_{\mathbf{h}'}(\mathbf{p})\| \leq L \, \|\mathbf{h} - \mathbf{h}'\|$ *uniformly in* $(\mathbf{h}, \mathbf{h}', \mathbf{p}, t)$.

(A2) (Strong log-concavity in $\mathbf{x}^{\text{hist}}$) $-\log p_t(\mathbf{x}^{hist} \mid \mathbf{c})$ *is* m_x*-strongly convex in* $\mathbf{x}^{hist}$*, uniformly over* $(\mathbf{c}, t)$*; i.e.,*

$$
\left(\nabla_{\mathbf{x}^{hist}} \log p_t(\mathbf{x}^{hist} \mid \mathbf{c}) - \nabla_{\mathbf{x}^{hist}} \log p_t(\mathbf{x}^{hist\,\prime} \mid \mathbf{c}) \right)^{\top} (\mathbf{x}^{hist} - \mathbf{x}^{hist\,\prime}) \leq -m_x \, \|\mathbf{x}^{hist} - \mathbf{x}^{hist\,\prime}\|^2.
$$

(A3) (Coercivity in $\mathbf{x}^{\text{pred}}$) $-\log p_t(\mathbf{x}^{pred} \mid \mathbf{x}^{hist})$ *is* m_y*-strongly convex in* $\mathbf{x}^{pred}$ *uniformly in* $(\mathbf{x}^{hist}, t)$.

Remarks. Assumption (A1) covers conditional exponentials (e.g., conditionally Gaussian models), and (A2)–(A3) are standard to ensure contractive reverse flows.[4] These assumptions are satisfied by typical conditionally log-concave families (e.g., linear/affine–Gaussian and more general exponential families with Lipschitz natural parameters) and by neural score models whose Jacobians are Lipschitz/sector-bounded via spectral or weight normalization.

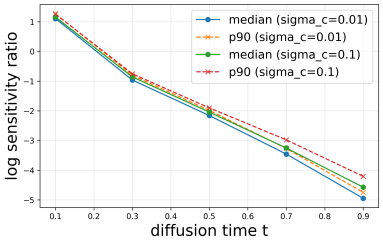


Figure 6: We report Lipschitz proxy on Traffic dataset. The plot shows trained network preserves stability.

To sanity-check the *Lipschitz-in-history* requirement in assumptions, we empirically estimate a Lipschitz proxy by

[4] COROS trains the total-sequence *conditional* score via DSM and samples with a PC sampler; the DSM equivalence for conditionals is given in Theorem 1.

fixing the same forward-noised input $\mathbf{x}_t$ and perturbing only the history/condition $\mathbf{h} \mapsto \mathbf{h}' = \mathbf{h} + \sigma_c \boldsymbol{\xi}$ (masked to observed history coordinates). We then compute, for each diffusion time t,

$$\widetilde{R}_{\mathrm{pred}}(t) := \frac{\mathrm{MSE}\big(\big(s_\theta(t, \mathbf{x}_t, \mathbf{h}) - s_\theta(t, \mathbf{x}_t, \mathbf{h}')\big)\big)}{\mathrm{MSE}((\mathbf{h} - \mathbf{h}')) + \varepsilon},$$

where $\mathbf{m}_{\mathrm{hist}}$ and $\mathbf{m}_{\mathrm{pred}}$ select history and forecast coordinates, respectively. Across the tested settings (e.g., `traffic` with history/prediction lengths $72/24$ and $\sigma_c \in \{0.01, 0.1\}$), the median and 90th percentile of $\log \widetilde{R}_{\mathrm{pred}}(t)$ remain bounded and decrease monotonically with t, with similar trends across perturbation scales. This indicates that the score restricted to future coordinates does not exhibit exploding sensitivity to history in practice, which is consistent as a mild smoothness assumption.

Lemma 2. *For any $t \in (0, 1]$,*

$$p_t(\mathbf{x}_t^{hist}, \mathbf{x}_t^{pred} \mid \mathbf{c}_t) = p_t(\mathbf{x}_t^{hist} \mid \mathbf{c}_t) \cdot p_t(\mathbf{x}_t^{pred} \mid \mathbf{x}_t^{hist}),$$

hence $\mathbf{x}_t^{pred} \perp \mathbf{c}_t \mid \mathbf{x}_t^{hist}$.

Proof. By construction $p(\mathbf{x}^{\mathrm{hist}}, \mathbf{x}^{\mathrm{pred}}, \tilde{\mathbf{x}}^{\mathrm{hist}}) = p(\mathbf{x}^{\mathrm{hist}})\, p(\mathbf{x}^{\mathrm{pred}} \mid \mathbf{x}^{\mathrm{hist}})\, p(\tilde{\mathbf{x}}^{\mathrm{hist}} \mid \mathbf{x}^{\mathrm{hist}})$, and the VP forward kernels act independently on $(X, Y, \widetilde{X})$. The Markov property yields $C_t \perp Y_t \mid X_t$ and thus the factorization $p_t(\mathbf{x}_t^{\mathrm{hist}}, \mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{c}_t) = p_t(\mathbf{x}_t^{\mathrm{hist}} \mid \mathbf{c}_t) \cdot p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x}_t^{\mathrm{hist}})$. □

From Lemma 2,

$$\nabla_{\mathbf{x}_t^{\mathrm{pred}}} \log p_t(\mathbf{x}_t^{\mathrm{hist}}, \mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{c}_t) = \nabla_{\mathbf{x}_t^{\mathrm{pred}}} \log p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x}_t^{\mathrm{hist}}),$$

$$\nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{hist}}, \mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{c}_t) = \nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{hist}} \mid \mathbf{c}_t) + \nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x}_t^{\mathrm{hist}}).$$

Thus the $\mathbf{x}^{\mathrm{pred}}$-component of the *total* score depends on the (being-denoised) $\mathbf{x}_t^{\mathrm{hist}}$, whereas the prediction-only score uses the marginal $p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{c}_t)$.

Lemma 3. *For any $(\mathbf{x}_t^{pred}, \mathbf{c}_t)$,*

$$\nabla_{\mathbf{x}_t^{pred}} \log p_t(\mathbf{x}_t^{pred} \mid \mathbf{c}_t) = \mathbb{E}\Big[\nabla_{\mathbf{x}_t^{pred}} \log p_t(\mathbf{x}_t^{pred} \mid \mathbf{x}_t^{hist}) \;\Big|\; \mathbf{x}_t^{pred}, \mathbf{c}_t\Big]. \tag{2}$$

Proof. Differentiating $\log p_t(\mathbf{x}_t^{\mathrm{pred}}, \mathbf{c}_t) = \log \int p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x})\, p_t(\mathbf{x} \mid \mathbf{c}_t)\, d\mathbf{x}$ under the integral and applying Bayes' rule yields (2). □

We now compare the reverse SDE drifts (omitting the common diffusion term $g(t)\, d\overline{\mathbf{w}}_t$).

First, the **prediction-only** reverse process in $\mathbf{x}^{\mathrm{pred}}$ takes drift in

$$\dot{\mathbf{x}}_t^{\mathrm{pred}} = \mathbf{f}_{\mathrm{pred}}(t, \mathbf{x}_t^{\mathrm{pred}}) - g(t)^2\, \nabla_{\mathbf{x}_t^{\mathrm{pred}}} \log p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{c}_t), \tag{3}$$

with $\mathbf{x}_t^{\mathrm{hist}} \equiv \mathbf{c}_t$ fixed as a (noisy) condition.

On the other hand, COROS takes drift in

$$\dot{\mathbf{x}}_t^{\mathrm{hist}} = \mathbf{f}_{\mathrm{hist}}(t, \mathbf{x}_t^{\mathrm{hist}}) - g(t)^2 \Big(\nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{hist}} \mid \mathbf{c}_t) + \nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x}_t^{\mathrm{hist}})\Big), \tag{4}$$

$$\dot{\mathbf{x}}_t^{\mathrm{pred}} = \mathbf{f}_{\mathrm{pred}}(t, \mathbf{x}_t^{\mathrm{pred}}) - g(t)^2\, \nabla_{\mathbf{x}_t^{\mathrm{pred}}} \log p_t(\mathbf{x}_t^{\mathrm{pred}} \mid \mathbf{x}_t^{\mathrm{hist}}). \tag{5}$$

Hence, $\mathbf{x}_t^{\mathrm{hist}}$ is aligned with clean data by $\nabla_{\mathbf{x}_t^{\mathrm{hist}}} \log p_t(\mathbf{x}_t^{\mathrm{hist}} \mid \mathbf{c}_t)$, and $\mathbf{x}_t^{\mathrm{pred}}$ uses the increasingly denoised $\mathbf{x}_t^{\mathrm{hist}}$.

Proposition 2. *Under (A2) and the VP drift $\mathbf{f}_{hist}(t, \mathbf{x}) = -\frac{1}{2}\beta(t)\, \mathbf{x}$, consider two runs of (4) with the same $\{\mathbf{x}_t^{pred}\}$ and $\mathbf{c}_t$, but different $\mathbf{x}_t^{hist}$ and $\mathbf{x}_t^{hist\,\prime}$. Then*

$$\frac{d}{dt} \|\mathbf{x}_t^{\mathrm{hist}} - \mathbf{x}_t^{\mathrm{hist}\,\prime}\|^2 \le -2m_x g(t)^2 \|\mathbf{x}_t^{\mathrm{hist}} - \mathbf{x}_t^{\mathrm{hist}\,\prime}\|^2,$$

$$\Rightarrow \quad \|\mathbf{x}_t^{\mathrm{hist}} - \mathbf{x}_t^{\mathrm{hist}\,\prime}\| \le \exp\Big(-m_x \int_t^1 g(s)^2\, \mathrm{d}s\Big) \|\mathbf{x}_1^{\mathrm{hist}} - \mathbf{x}_1^{\mathrm{hist}\,\prime}\|.$$

Thus $\mathbf{x}_t^{hist}$ contracts exponentially toward the mode/mean of $p_t(\mathbf{x}^{hist} \mid \mathbf{c}_t)$ along the reverse flow.

Proof. Monotonicity of $\nabla_{\mathbf{x}} \log p_t(\mathbf{x} \mid \mathbf{c})$ with parameter m_x controls the symmetric part of the Jacobian of the drift in (4); the contribution of f_{hist} further aids dissipation. The cross-term $\nabla \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}_t^{\text{hist}})$ is 1-Lipschitz in $\mathbf{x}_t^{\text{hist}}$ under (A1) at $\mathbf{x}_t^{\text{pred}}$ and is dominated by m_x; the standard Grönwall argument for deriving the solution of ODE gives the stated inequality. □

We measure robustness by a Lipschitz-type sensitivity of the *output* $\mathbf{x}_0^{\text{pred}}$ with respect to the *condition path* $\mathbf{c}_\cdot$. Fix two noisy conditions $\mathbf{c}_\cdot$ and $\mathbf{c}'_\cdot$, and couple the reverse noise so differences stem only from the drifts.

Theorem 3. *Assume (A1)–(A3) in Appendix G. Let* $H(t) := \int_t^1 g(s)^2\, ds$ *and* $G := H(0) = \int_0^1 g(s)^2\, ds > 0$. *Then for noised input condition* $\boldsymbol{c}_s \equiv \boldsymbol{x}_s^{hist}$, *the difference between the forecasts under* $\mathbf{c}$ *and* $\mathbf{c}'$, $\|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\|$, *is bounded by*

$$\textbf{\textit{(prediction-only)}} \quad L \int_0^1 g(s)^2 \, \|\mathbf{c}_s - \mathbf{c}'_s\| \, ds \;\le\; L\,G \sup_{s\in[0,1]} \|\mathbf{c}_s - \mathbf{c}'_s\|,$$

$$\textbf{\textit{(total-sequence)}} \quad L \int_0^1 g(s)^2 \, e^{-m_x\, H(s)} \, \|\mathbf{c}_s - \mathbf{c}'_s\| \, ds \;\le\; \frac{L}{m_x}\big(1 - e^{-m_x G}\big) \sup_s \|\mathbf{c}_s - \mathbf{c}'_s\|.$$

Consequently,

$$\frac{L}{m_x}\big(1 - e^{-m_x G}\big) \;<\; L\,G \quad \Rightarrow \quad \sup \|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\|_{\textit{total-sequence}} \;<\; \sup \|\mathbf{x}_0^{pred} - \mathbf{x}_0^{pred\,\prime}\|_{\textit{pred-only}}.$$

Thus, under the stated assumptions, the total-sequence *conditional score has a strictly tighter worst-case sensitivity bound with respect to condition perturbations than the* prediction-only *score.*

Proof. Let $\Delta(t) := \mathbf{x}_t^{\text{pred}} - \mathbf{x}_t^{\text{pred}\,\prime}$. We couple the two reverse processes (prediction-only or total-sequence) with the *same* reverse Gaussian noise so that pathwise differences arise only from the drift terms

Write the two reverse drifts (prediction-only vs. total) as

$$\dot{\mathbf{x}}_t^{\text{pred}} = \mathbf{f}_{\text{pred}}(t, \mathbf{x}_t^{\text{pred}}) - g(t)^2\, S_t, \qquad \dot{\mathbf{x}}_t^{\text{pred}\,\prime} = \mathbf{f}_{\text{pred}}(t, \mathbf{x}_t^{\text{pred}\,\prime}) - g(t)^2\, S'_t,$$

where S_t and S'_t are, respectively,

$$S_t = \begin{cases} \nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{c}_t) & \text{(prediction)}, \\ \nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}_t^{\text{hist}}) & \text{(total)}, \end{cases} \quad S'_t = \begin{cases} \nabla_{\mathbf{x}_t^{\text{pred}\,\prime}} \log p_t(\mathbf{x}_t^{\text{pred}\,\prime} \mid \mathbf{c}'_t) & \text{(prediction)}, \\ \nabla_{\mathbf{x}_t^{\text{pred}\,\prime}} \log p_t(\mathbf{x}_t^{\text{pred}\,\prime} \mid \mathbf{x}_t^{\text{hist}\,\prime}) & \text{(total)}. \end{cases}$$

Subtracting and taking the inner product with the unit vector $\mathbf{u}(t) := \Delta(t)/\|\Delta(t)\|$ (for $\Delta \neq 0$) yields

$$\begin{aligned} \frac{d}{dt}\|\Delta(t)\| &= \mathbf{u}(t)^\top \big(\dot{\mathbf{x}}_t^{\text{pred}} - \dot{\mathbf{x}}_t^{\text{pred}\,\prime}\big) \\ &= \mathbf{u}^\top \big(\mathbf{f}_{\text{pred}}(t, \mathbf{x}_t^{\text{pred}}) - \mathbf{f}_{\text{pred}}(t, \mathbf{x}_t^{\text{pred}\,\prime})\big) - g(t)^2\, \mathbf{u}^\top \big(S_t - S'_t\big). \end{aligned}$$

Under VP, $\mathbf{f}_{\text{pred}}(t, \mathbf{y}) = -\frac{1}{2}\beta(t)\mathbf{y}$ contributes a *contractive* term $-\frac{1}{2}\beta(t)\|\Delta(t)\|$. Moreover, by Assumption (A3) (m_y-strong convexity in $\mathbf{x}^{\text{pred}}$),

$$\big(\nabla_{\mathbf{x}^{\text{pred}}} \log p_t(\mathbf{x}^{\text{pred}} \mid \cdot) - \nabla_{\mathbf{x}^{\text{pred}}} \log p_t(\mathbf{x}^{\text{pred}\,\prime} \mid \cdot)\big)^\top (\mathbf{x}^{\text{pred}} - \mathbf{x}^{\text{pred}\,\prime}) \;\le\; -\,m_y\, \|\Delta(t)\|^2,$$

which implies

$$-\,\mathbf{u}^\top \big(S_t - S'_t\big) \;\le\; -\,m_y\, \|\Delta(t)\| \;+\; \Gamma_t,$$

with the *forcing* term

$$\Gamma_t := \begin{cases} \big\|\nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{c}_t) - \nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{c}'_t)\big\|, & \text{(prediction-only)}, \\ \big\|\nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}_t^{\text{hist}}) - \nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}_t^{\text{hist}\,\prime})\big\|, & \text{(total)}. \end{cases}$$

Absorbing the extra $-\frac{1}{2}\beta(t)\|\Delta\|$ (beneficial contraction) into m_y and combining the above,

$$\frac{d}{dt}\|\Delta(t)\| \;\le\; -\,m_y\, g(t)^2\, \|\Delta(t)\| \;+\; g(t)^2\, \Gamma_t. \tag{6}$$

Case 1: prediction-only. By the Fisher/mixture identity (Lemma 3),

$$\nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{c}_t) = \mathbb{E}\big[\nabla_{\mathbf{x}_t^{\text{pred}}} \log p_t(\mathbf{x}_t^{\text{pred}} \mid \mathbf{x}_t^{\text{hist}}) \,\big|\, \mathbf{x}_t^{\text{pred}}, \mathbf{c}_t\big].$$

Hence, using Jensen and (A1) (Lipschitz in the history argument),

$$\begin{aligned}\Gamma_t &= \Big\| \mathbb{E}\big[\phi_{\mathbf{x}_t^{\text{hist}}}(\mathbf{x}_t^{\text{pred}}) \mid \mathbf{x}_t^{\text{pred}}, \mathbf{c}_t\big] - \mathbb{E}\big[\phi_{\mathbf{x}_t^{\text{hist}}}(\mathbf{x}_t^{\text{pred}}) \mid \mathbf{x}_t^{\text{pred}}, \mathbf{c}_t'\big] \Big\| \\ &\le L \cdot \mathsf{W}_1\big(p_t(\mathbf{x}_t^{\text{hist}} \mid \mathbf{c}_t),\, p_t(\mathbf{x}_t^{\text{hist}} \mid \mathbf{c}_t')\big),\end{aligned}$$

where W_1 is the 1-Wasserstein distance. Under the VP channel and strong log-concavity (A2), the posterior map $\mathbf{c}_t \mapsto p_t(\mathbf{x}_t^{\text{hist}} \mid \mathbf{c}_t)$ is 1-Lipschitz in W_1 (e.g., by contraction of Gaussian channels / Brascamp–Lieb), yielding

$$\Gamma_t \;\le\; L\,\|\mathbf{c}_t - \mathbf{c}_t'\|.$$

Plugging this into (6) and using $e^{-m_y(H(0)-H(s))} \le 1$,

$$\|\Delta(0)\| \le \int_0^1 g(s)^2\, L\, \|\mathbf{c}_s - \mathbf{c}_s'\|\, ds \le L\, G \sup_{s\in[0,1]} \|\mathbf{c}_s - \mathbf{c}_s'\|,$$

.

Case 2: total-sequence. Here

$$\Gamma_t = \big\|\phi_{\mathbf{x}_t^{\text{hist}}}(\mathbf{x}_t^{\text{pred}}) - \phi_{\mathbf{x}_t^{\text{hist}\,\prime}}(\mathbf{x}_t^{\text{pred}})\big\| \le L\,\|\mathbf{x}_t^{\text{hist}} - \mathbf{x}_t^{\text{hist}\,\prime}\| \quad \text{by (A1).}$$

By Proposition 2 (contractivity of the $\mathbf{x}^{\text{hist}}$-flow under (A2)),

$$\|\mathbf{x}_t^{\text{hist}} - \mathbf{x}_t^{\text{hist}\,\prime}\| \;\le\; e^{-m_x\, H(t)}\, \|\mathbf{x}_1^{\text{hist}} - \mathbf{x}_1^{\text{hist}\,\prime}\|.$$

With the standard synchronous terminal coupling of the reverse SDE, $\mathbf{x}_1^{\text{hist}}$ and $\mathbf{x}_1^{\text{hist}\,\prime}$ share the same Gaussian noise, so their difference is controlled by the difference of the (forward) conditions; in particular,

$$\|\mathbf{x}_1^{\text{hist}} - \mathbf{x}_1^{\text{hist}\,\prime}\| \;\le\; \sup_{u\in[0,1]} \|\mathbf{c}_u - \mathbf{c}_u'\|.$$

Therefore

$$\Gamma_t \;\le\; L\, e^{-m_x\, H(t)} \sup_u \|\mathbf{c}_u - \mathbf{c}_u'\|.$$

Plugging this bound into Step 2, we obtain

$$\|\Delta(0)\| \le L \sup_u \|\mathbf{c}_u - \mathbf{c}_u'\| \int_0^1 e^{-m_y(H(0)-H(s))}\, g(s)^2\, e^{-m_x H(s)}\, ds.$$

Since $e^{-m_y(H(0)-H(s))} \le 1$, it suffices to compute

$$\int_0^1 g(s)^2\, e^{-m_x H(s)}\, ds.$$

Using $H'(s) = -g(s)^2$, the change of variables $u = H(s)$ gives

$$\int_0^1 g(s)^2 e^{-m_x H(s)} ds = \int_{u=G}^0 \big(-e^{-m_x u}\big)\, du = \int_0^G e^{-m_x u}\, du = \frac{1 - e^{-m_x G}}{m_x}.$$

Hence

$$\|\Delta(0)\| \;\le\; \frac{L}{m_x}\big(1 - e^{-m_x G}\big) \sup_s \|\mathbf{c}_s - \mathbf{c}_s'\|.$$

Finally, since $1 - e^{-a} < a$ for all $a > 0$, we have

$$\frac{L}{m_x}\big(1 - e^{-m_x G}\big) \;<\; L\,G,$$

which combined with the two bounds proves the strict inequality and completes the proof. □

Remark. Positivity of m_y in (A3) is used only to bound the common factor $e^{-m_y(H(0)-H(s))}$ by one. Keeping this factor in (6), the two bounds share the same positive weight, and the total-sequence integrand is additionally multiplied by $e^{-m_x H(s)} < 1$; hence the total-sequence bound remains strictly smaller for any sign of m_y.

G.3 Conditional DSM Equivalence on the Future Coordinate

Theorem 4. *For each $L_{SM}^{pred}(\theta)$ and $L_{SM}^{total}(\theta)$, its denoising score matching are represented as follows:*

$$L_{DSM}^{pred}(\theta) = \mathbb{E}_{t,\mathbf{x}^{total},\mathbf{x}_t^{total}}[\lambda(t)||s_\theta(t,\mathbf{x}_t^{pred},\mathbf{c}) - \nabla_{\mathbf{x}_t^{pred}} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total})||_2^2]$$
$$L_{DSM}^{total}(\theta) = \mathbb{E}_{t,\mathbf{x}^{total},\mathbf{x}_t^{total}}[\lambda(t)||s_\theta(t,\mathbf{x}_t^{total},\mathbf{c}) - \nabla_{\mathbf{x}_t^{total}} logp(\mathbf{x}_t^{total}|\mathbf{x}^{total})||_2^2]$$

Remark 1. Therefore, the prediction-only DSM target coincides with the x_t^{pred}-coordinate of the joint DSM target. Under coordinate-wise score SDE framework, $\nabla_{x_t^{\text{pred}}} \log p(x_t^{\text{total}} \mid x^{\text{total}}) \approx \nabla_{x_t^{\text{pred}}} \log p(x_t^{\text{pred}} \mid x^{\text{pred}})$. Hence prediction-only diffusion and COROS share the same future-coordinate Gaussian denoising label during training. COROS differs by additionally learning the history-coordinate score and by coupling past and future through the reverse-time dynamics.

Remark 2. Our total-generation loss *recovers*—rather than merely matches—the same conditional score function. Concretely: (i) although COROS is trained on the full-sequence score, it still preserves the conditional distribution $p(\mathbf{x}^{\text{pred}} \mid \mathbf{x}^{\text{hist}})$; (ii) by generating the entire sequence, the model simultaneously denoises the historical part, improving robustness when the input history contains anomalies.

Proof. We prove denoising score matching loss of prediction, $L_{DSM}^{total}(\theta)$. The result of $L_{DSM}^{pred}(\theta)$ can be derived similarly. For simplicity, we do not distinguish given condition with historic part of total sequence. We start from decomposing it:

$$\begin{aligned} L_{SM}^{total}(\theta) = &-2 \cdot \mathbb{E}_t \mathbb{E}_{\mathbf{x}^{hist}} \mathbb{E}_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total}|\mathbf{x}^{hist})\rangle \\ &+ \mathbb{E}_t \mathbb{E}_{\mathbf{x}^{hist}} \mathbb{E}_{\mathbf{x}_t^{total}} \left[\left\| s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}) \right\|_2^2 \right] + C_1 \end{aligned}$$

Here, C_1 is a constant that does not depend on the parameter θ, and $\langle \cdot, \cdot \rangle$ means the inner product. Then, the first part's expectation of the right-hand side can be expressed as follows:

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total}|\mathbf{x}^{hist})\rangle \\ &= \int_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total}|\mathbf{x}^{hist})\rangle \mathrm{p}(\mathbf{x}_t^{total}|\mathbf{x}^{hist}) d\mathbf{x}_t^{total} \\ &= \int_{\mathbf{x}^{total}} \int_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \frac{1}{\mathrm{p}(\mathbf{x}^{hist})} \frac{\partial \mathrm{p}(\mathbf{x}_t^{total},\mathbf{x}^{hist},\mathbf{x}^{total})}{\partial \mathbf{x}_t^{total}} \rangle d\mathbf{x}_t^{total} d\mathbf{x}^{total} \\ &= \int_{\mathbf{x}^{total}} \int_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \frac{\partial \mathrm{p}(\mathbf{x}_t^{total}|\mathbf{x}^{total}))}{\partial \mathbf{x}_t^{total}} \rangle \mathrm{p}(\mathbf{x}^{total}|\mathbf{x}^{hist}) d\mathbf{x}_t^{total} d\mathbf{x}^{total} \\ &= \mathbb{E}_{\mathbf{x}^{total}} \left[\int_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \frac{\partial \mathrm{p}(\mathbf{x}_t^{total}|\mathbf{x}^{total}))}{\partial \mathbf{x}_t^{total}} \rangle d\mathbf{x}_t^{total} \right] \\ &= \mathbb{E}_{\mathbf{x}^{total}} \left[\int_{\mathbf{x}_t^{total}} \langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total}|\mathbf{x}^{total})\rangle \mathrm{p}(\mathbf{x}_t^{total}|\mathbf{x}^{total}) d\mathbf{x}_t^{total} \right] \\ &= \mathbb{E}_{\mathbf{x}^{total}} \mathbb{E}_{\mathbf{x}_t^{total}} [\langle s_\theta(t,\mathbf{x}_t^{total},\mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total}|\mathbf{x}^{total})\rangle] \end{aligned}$$

The second part's expectation of the right-hand side can be rewritten similarly, therefore we can derive following result:

$$\begin{aligned} L_{SM}^{total}(\theta) &= -2 \cdot \mathbb{E}_t \mathbb{E}_{\mathbf{x}^{total}} \mathbb{E}_{\mathbf{x}_t^{total}} \langle s_\theta(t, \mathbf{x}_t^{total}, \mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total} | \mathbf{x}^{hist}) \rangle \\ &\quad + \mathbb{E}_t \mathbb{E}_{\mathbf{x}^{hist}} \mathbb{E}_{\mathbf{x}^{total}} \mathbb{E}_{\mathbf{x}_t^{total}} \left[\left\| s_\theta(t, \mathbf{x}_t^{total}, \mathbf{x}^{hist}) \right\|_2^2 \right] + C_1 \\ &= L_{DSM}^{total}(\theta) + C_1 \end{aligned}$$

C is a constant that does not depend on the parameter θ.

Similarly, we compute $L_{DSM}^{pred}(\theta)$. We give proof on inner product part and the remaining terms follow directly from the case of $L_{DSM}^{total}(\theta)$.

$$\begin{aligned} &\mathbb{E}_{\mathbf{x}_t^{pred}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{pred}} \log p(\mathbf{x}_t^{pred} | \mathbf{x}^{hist}) \rangle \\ &= \int_{\mathbf{x}_t^{pred}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{pred}} \log p(\mathbf{x}_t^{pred} | \mathbf{x}^{hist}) \rangle \mathrm{p}(\mathbf{x}_t^{pred} | \mathbf{x}^{hist}) d\mathbf{x}_t^{pred} \\ &= \int_{\mathbf{x}^{pred}} \int_{\mathbf{x}_t^{total}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \frac{1}{\mathrm{p}(\mathbf{x}^{hist})} \frac{\partial \mathrm{p}(\mathbf{x}_t^{total}, \mathbf{x}^{hist}, \mathbf{x}^{pred})}{\partial \mathbf{x}_t^{pred}} \rangle d\mathbf{x}_t^{total} d\mathbf{x}^{pred} \\ &= \int_{\mathbf{x}^{pred}} \int_{\mathbf{x}_t^{total}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \frac{\partial \mathrm{p}(\mathbf{x}_t^{total} | \mathbf{x}^{total}))}{\partial \mathbf{x}_t^{total}} \rangle \frac{\mathrm{p}(\mathbf{x}^{hist}, \mathbf{x}^{pred})}{\mathbf{x}^{hist}} d\mathbf{x}_t^{total} d\mathbf{x}^{pred} \\ &= \mathbb{E}_{\mathbf{x}^{pred}} \left[\int_{\mathbf{x}_t^{total}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \frac{\partial \mathrm{p}(\mathbf{x}_t^{total} | \mathbf{x}^{total}))}{\partial \mathbf{x}_t^{total}} \rangle d\mathbf{x}_t^{total} \right] \\ &= \mathbb{E}_{\mathbf{x}^{pred}} \left[\int_{\mathbf{x}_t^{total}} \langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total} | \mathbf{x}^{total}) \rangle \mathrm{p}(\mathbf{x}_t^{total} | \mathbf{x}^{total}) d\mathbf{x}_t^{total} \right] \\ &= \mathbb{E}_{\mathbf{x}^{pred}} \mathbb{E}_{\mathbf{x}_t^{total}} [\langle s_\theta(t, \mathbf{x}_t^{pred}, \mathbf{x}^{hist}), \nabla_{\mathbf{x}_t^{total}} \log p(\mathbf{x}_t^{total} | \mathbf{x}^{total}) \rangle] \end{aligned}$$

□

H Related work

Existing diffusion-based forecasting models are broadly divided into two categories: models targeting the prediction sequence score function, $\nabla_{\mathbf{x}_t^{\text{pred}}} \log p(\mathbf{x}_t^{\text{pred}} | \mathbf{x}^{\text{hist}})$, and those modeling the entire sequence score, $\nabla_{\mathbf{x}_t^{\text{total}}} \log p(\mathbf{x}_t^{\text{total}} | \mathbf{x}^{\text{hist}})$.

TimeGrad (Rasul et al., 2021) and CSDI (Tashiro et al., 2021) belong to the first category. TimeGrad generates predictions autoregressively, predicting one step ahead iteratively, whereas CSDI generates the entire prediction sequence in a single step. Although one-shot generation can be efficient, it may introduce higher variance in samples, prompting CSDI to stabilize performance by averaging multiple samples. Based on the continuous SDE format, MadSGM (Lim et al., 2023b) extended diffusion usage to time-series anomaly detection, leveraging samples, probability and score function simultaneously.

In the second category, models like TSDiff (Kollovieh et al., 2023) generate the complete sequence, leveraging history-guided sampling to enhance conditional generation. Additionally, Lim et al. (2023a) and Lim et al. (2024) propose autoregressive generation in a latent space to handle irregularly sampled data effectively, offering improved modeling of complex time dependencies.

Our proposed method, COROS, combines advantages from both categories, integrating predictive accuracy and guidance mechanisms within a unified framework for robust and flexible time-series forecasting.