FAIP-DIFF: FLUCTUATION-AWARE INFORMATION-PRESERVING DIFFUSION FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

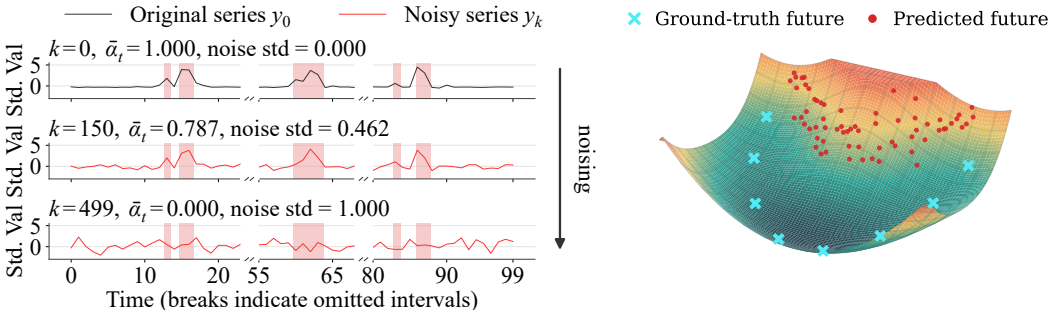
Conventional probabilistic forecasting methods often rely on predefined distributional assumptions to characterize multiple possible future outcomes. Diffusion models have become a promising alternative, as they can flexibly learn data distributions without restrictive parametric assumptions. However, existing diffusion-based time-series forecasting methods have not fully solved the problem of temporal pattern distortion that occurs during the diffusion process of non-stationary time series. Moreover, we found that temporal pattern distortion manifests in temporal fluctuation suppression during forward noising and denoising trajectory drift during reverse sampling. To address these questions, we propose FAIP-Diff, a fluctuation-aware information-preserving diffusion framework for probabilistic time series forecasting. FAIP-Diff estimates the conditional mean and volatility as forecasting priors. To alleviate temporal fluctuation suppression during forward noising, FAIP-Diff introduces an adaptive noise remodulation strategy that adjusts the corruption process based on position-dependent forecasting difficulty, enhancing learning in fluctuation-sensitive regions. Second, to mitigate denoising trajectory drift during reverse sampling, FAIP-Diff develops a constrained-space guided sampling mechanism that softly rectifies the predicted noise at each denoising step, steering generated trajectories toward realistic temporal patterns while preserving predictive diversity. Additionally, we introduce a volatility-aware weighted training objective that further alleviates temporal pattern distortion through adaptive loss weighting. Experiments across diverse real-world benchmarks demonstrate that FAIP-Diff delivers more accurate and reliable forecasts, particularly under complex non-stationary dynamics.

1 INTRODUCTION

Time-series forecasting (TSF) is a fundamental task in a wide range of real-world applications, including energy scheduling (Liu et al., 2025a), traffic management (Zhu et al., 2026), meteorological monitoring (Akın et al., 2026), and disease dynamics (Frishberg et al., 2023). Existing TSF methods typically exploit historical observations to characterize temporal dependencies and recurring patterns for future prediction. However, real-world time series are inherently non-stationary, so patterns learned from historical data may not remain reliable as dynamics evolve (Fan et al., 2023; Kim et al., 2022). As a result, deterministic forecasts may fail to characterize the multiple plausible outcomes of the future. Probabilistic forecasting therefore becomes essential, as it explicitly models predictive uncertainty and produces distributions or multiple plausible future trajectories (Cao et al., 2026; Shen et al., 2024).

Conventional probabilistic forecasting methods often rely on predefined distributional assumptions to characterize multiple possible future outcomes (Cini et al., 2025). However, real-world time series typically exhibit complex and non-stationary distributions that are difficult to specify a priori. Diffusion models therefore provide a promising alternative, as they can flexibly learn complex data distributions without restrictive parametric assumptions (Rishi et al., 2025). In TSF, standard diffusion models progressively perturb clean data with Gaussian noise and learn a reverse denoising process to recover plausible future trajectories (Ye et al., 2025). Despite their strong generative forecasting capabilities, standard diffusion mechanisms are largely agnostic to the intrinsic temporal

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.



(a) Temporal fluctuation suppression.

(b) Denoising trajectory drift.

Figure 1: Illustration of temporal pattern distortion in diffusion-based TSF. (a) Temporal fluctuation suppression: high-fluctuation patterns in the shaded regions are progressively suppressed as Gaussian noise increases. (b) Denoising trajectory drift: the predicted trajectory gradually deviates from the ground-truth trajectory during reverse sampling.

patterns of non-stationary time series. As a result, meaningful temporal patterns may be disrupted during forward noising and inadequately recovered during reverse sampling, leading to what we term **temporal pattern distortion**.

Specifically, temporal pattern distortion manifests in two complementary aspects: (a) **temporal fluctuation suppression** during forward noising and (b) **denoising trajectory drift** during reverse sampling. First, during forward noising, Gaussian perturbations are progressively injected without explicitly accounting for the heterogeneous fluctuation characteristics of time-series data. As illustrated in Figure 1a, progressive Gaussian perturbation can suppress informative high-fluctuation patterns, such as local variations and abrupt changes. Since such patterns occur less frequently than smooth observations, their contribution to model learning is relatively limited under a uniform noise schedule, making them more difficult to recover and motivating adaptive emphasis on fluctuation-intensive regions (Chen & Gupta, 2024; Seo & Gupta, 2026).

Second, during multi-step reverse sampling, small denoising errors can accumulate across diffusion steps, causing the generated trajectory to progressively drift away from the ground-truth trajectory, as illustrated in Figure 1b. Without explicit structural guidance, such deviations may propagate throughout the reverse process and drive generated samples toward less plausible regions of the data space. Therefore, an effective constraint is required to correct the denoising trajectory and guide it toward realistic temporal patterns.

To address these questions, we propose FAIP-Diff, a Fluctuation-Aware Information-Preserving Diffusion framework for probabilistic time-series forecasting. FAIP-Diff first estimates the conditional mean and volatility as forecasting priors. Then, to alleviate temporal fluctuation suppression during forward noising, FAIP-Diff introduces **an adaptive noise remodulation strategy** that adjusts the corruption process based on position-dependent forecasting difficulty, thereby improving learning in fluctuation-sensitive regions. Second, to mitigate denoising trajectory drift during reverse sampling, FAIP-Diff develops **a constrained-space guided sampling mechanism** that softly rectifies the predicted noise at each denoising step, steering generated trajectories toward realistic temporal patterns while preserving predictive diversity. Additionally, we introduce a volatility-aware weighted training objective that further alleviates temporal pattern distortion through adaptive loss weighting. We evaluate FAIP-Diff on multiple widely used TSF benchmarks and primarily evaluate its distributional accuracy and calibration using the Continuous Ranked Probability Score (CRPS) and Quantile Interval Coverage Error (QICE). Experimental results demonstrate that FAIP-Diff improves probabilistic forecasting performance over competitive baselines.

In summary, our main contributions are as follows:

• We propose FAIP-Diff, a Fluctuation-Aware Information-Preserving Diffusion framework for probabilistic time-series forecasting, which explicitly addresses temporal pattern distortion throughout both forward noising and reverse sampling.

• To alleviate temporal fluctuation suppression during forward noising, we develop an adaptive noise remodulation strategy that adjusts the corruption strength based on position-dependent forecasting difficulty, enhancing learning in fluctuation-sensitive regions.

• To mitigate denoising trajectory drift during reverse sampling, we introduce a constrained-space guided sampling mechanism that leverages structural priors learned from training data to progressively steer generated trajectories toward plausible temporal patterns.

• Extensive experiments on multiple benchmark datasets demonstrate the effectiveness of FAIP-Diff in terms of distributional accuracy and calibration.

2 Related Work

Time Series Forecasting. Transformer-based models introduce sparse attention and decomposition mechanisms to capture long-horizon patterns (Zhou et al., 2021; Wu et al., 2021). Frequency-domain modeling and multi-resolution temporal structures have also been explored to improve representation quality and robustness (Zhou et al., 2022b; Liu et al., 2022a). Subsequent works further address non-stationarity and frequency-enhanced historical memory for complex real-world series (Liu et al., 2022b; Zhou et al., 2022a). More recent architectures revisit the representation of time series through patch-level modeling, temporal variation modeling, and inverted Transformer structures (Nie et al., 2023; Wu et al., 2023; Liu et al., 2024). Long-context forecasting further shows the importance of scaling temporal context for unified time-series tasks (Liu et al., 2025b). Other competitive forecasting backbones, such as Crossformer, DLinear, and N-HiTS, show that multiscale structures, simple decomposition, and hierarchical interpolation can also be effective for long-horizon prediction (Zhang & Yan, 2023; Zeng et al., 2023; Challu et al., 2023). However, these deterministic predictions make it difficult to fully learn the dynamic changes of non-stationary data. Probabilistic prediction methods have therefore gained attention. Representative approaches include likelihood-based autoregressive models and attention-based probabilistic forecasting (Salinas et al., 2020; Lim et al., 2021). Normalizing-flow-based forecasting and recent dynamical-prior generative models further improve distribution modeling for multivariate or long-horizon probabilistic forecasting (Rasul et al., 2021b; Wu et al., 2025).

Diffusion Models for Time Series. Research has found that the iterative sampling mechanism of diffusion models is highly suitable for TSF. Diffusion models generate data by gradually corrupting samples with Gaussian noise and learning a reverse denoising process (Ho et al., 2020; Song et al., 2021; Nichol & Dhariwal, 2021). TimeGrad introduces autoregressive denoising diffusion for multivariate probabilistic prediction, while CSDI applies conditional score-based diffusion to time series estimation and prediction (Rasul et al., 2021a; Tashiro et al., 2021). Subsequent studies combine diffusion models with structured state-space modeling and non-autoregressive conditioning for time-series prediction (Alcaraz & Strodthoff, 2023; Shen & Kwok, 2023). Interpretable temporal representation learning has also been incorporated into diffusion-based time-series generation (Yuan & Qiao, 2024). Recent works further extend diffusion models to conditional time-series generation, irregular time series, and data-scarce generation (Narasimhan et al., 2024; Fadlon et al., 2025; Gonen et al., 2025). Other recent studies investigate latent-to-data generation, elucidate diffusion forecasting, and multimodal forecasting losses (Shen et al., 2026; Carrillo et al., 2026; Zhang et al., 2026). Prototype- or structure-guided generation has also been explored, such as K-ProtoDiff, which injects prototype-aware information into diffusion generation (Duan et al., 2026). The aforementioned time-series forecasting methods still rely on predefined output distributions or implicitly learned volatility. Most diffusion-based time-series prediction methods use a fixed forward noise schedule. Moreover, existing diffusion models often overlook temporal structures, making it difficult to preserve informative fluctuations and prevent trajectory drift.

3 Preliminaries

In this section, we present the basic diffusion-based forecasting framework adopted in FAIP-Diff. To overcome the limitations of single-point future prediction and generate multiple plausible future trajectories, this work leverages the distinctive training mechanism of diffusion models. In time series forecasting, the sliding window is usually used to slice the given time series samples. $\mathbf{y}$ are the future data and $\mathbf{x}$ are the historical data.

During the training of diffusion models for TSF, the historical data $\mathbf{x} \in \mathbb{R}^{T\times N}$ is used as a condition to add noise to the real future data $\mathbf{y}_0 \in \mathbb{R}^{O\times N}$, where T and O respectively represent the length of historical data and future data, and N is the number of variables. During inference, the historical data $\mathbf{x}$ is used to construct the noise-added result, and the real future data $\mathbf{y}_0$ is not involved in the calculation.

A standard diffusion model gradually corrupts the clean future sequence $\mathbf{y}_0$ through a Markovian forward noising process q, with the result $\mathbf{y}_k$ of diffusion step being

$$\mathbf{y}_k \sim q(\mathbf{y}_k \mid \mathbf{y}_0), \quad \mathbf{y}_k = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + \sqrt{1-\bar{\alpha}_k}\,\boldsymbol{\epsilon} \quad (\bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i, k \in K), \tag{1}$$

where $\boldsymbol{\epsilon}$ is sampled from a standard Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{I})$, k is the current diffusion step, K is the total number of diffusion steps, α_k denotes the signal preservation ratio at step k, and $\bar{\alpha}_k$ denotes the cumulative signal preservation ratio after k diffusion steps. Let $\{\beta_k\}_{k=1}^{K}$ be a predefined noise variance schedule, where $\beta_k \in (0, 1)$ controls the amount of Gaussian noise injected at diffusion step k. We define $\alpha_k = 1 - \beta_k$.

The reverse process p_θ recovers clean future sequences under the diffusion model with parameter θ. Given historical observations x, the denoising model learns the conditional reverse transition

$$p_\theta(\mathbf{y}_{k-1} \mid \mathbf{y}_k, x) = \mathcal{N}(\boldsymbol{\mu}_{\theta,k}, \boldsymbol{\sigma}^2_{\theta,k}), \tag{2}$$

where $\hat{\boldsymbol{\epsilon}}_{\theta,k}$ and $\boldsymbol{\sigma}^2_{\theta,k}$ denote the outputs predicted by the denoising neural network, and the detailed derivations of $\boldsymbol{\mu}^2_{\theta,k}$ are provided in Appendix A.2 (see Eq.53). Each reverse-diffusion step estimates the noise variance and the future data based on the current state, and the future data are given by

$$\tilde{\mathbf{y}}_0^{(k)} = \frac{\mathbf{y}_k - \sqrt{1-\bar{\alpha}_k}\,\hat{\boldsymbol{\epsilon}}_{\theta,k}}{\sqrt{\bar{\alpha}_k}}. \tag{3}$$

Starting from a last noisy state $\boldsymbol{y}_K$, the model repeatedly applies the reverse transitions and finally obtains the generated future trajectory $\hat{\mathbf{y}}_0$. The Kullback–Leibler (KL) divergence measures the discrepancy between the two probability distributions q and p_θ. It serves as the training loss for progressively optimizing the model and is defined as

$$\mathcal{L}_{\mathrm{KL}}^{(k)} = D_{\mathrm{KL}}\left(q(\mathbf{y}_{k-1} \mid \mathbf{y}_k, \mathbf{y}_0) \,\|\, p_\theta(\mathbf{y}_{k-1} \mid \mathbf{y}_k, \mathbf{x})\right), \qquad k \geq 2. \tag{4}$$

However, standard diffusion models struggle to address temporal pattern distortion in the data. Section 4 will elaborate on how FAIP-Diff further alleviates this issue.

4 METHOD

In this section, we provide a detailed description of the proposed FAIP-Diff model, which consists of basic forecasting priors and three main components, as illustrated in Figure 2, including prior conditional networks (Section 4.1), an adaptive noise remodulation strategy (Section 4.2), a constrained-space guided sampling strategy (Section 4.3), and a volatility-aware weighted training objective for explicitly learning local future uncertainty (Section 4.4).

4.1 PRIOR CONDITIONAL NETWORKS

The conditional mean network is built on a Transformer block and uses the historical sequence to estimate the predictive center of the future sequence (Ye et al., 2025). The latter half of the historical sequence $\mathbf{x}$ is extracted and concatenated with $\mathbf{0} \in \mathbb{R}^{O\times N}$, and the historical data are standardized. A Transformer block then captures temporal dependencies, and inverse standardization yields the predicted future trajectory. $\boldsymbol{\mu}_x$ is the prediction at the zero-masked positions.

$$\boldsymbol{\mu}_x = \text{Transformer}(\mathbf{x} \,\|\, \mathbf{0}) \in \mathbb{R}^{O\times N} \tag{5}$$

The conditional variance network is based on a three-layer MLP architecture and computes the variance $\boldsymbol{\sigma}^2$ of $\mathbf{x}$ using a sliding window (Ye et al., 2025). Along the temporal dimension, the resulting variance $\boldsymbol{\sigma}^2$ is mapped to the future prediction horizon, yielding the following output

$$\boldsymbol{\sigma}_x^2 = \text{MLP}(\boldsymbol{\sigma}^2) \in \mathbb{R}^{O\times N} \tag{6}$$

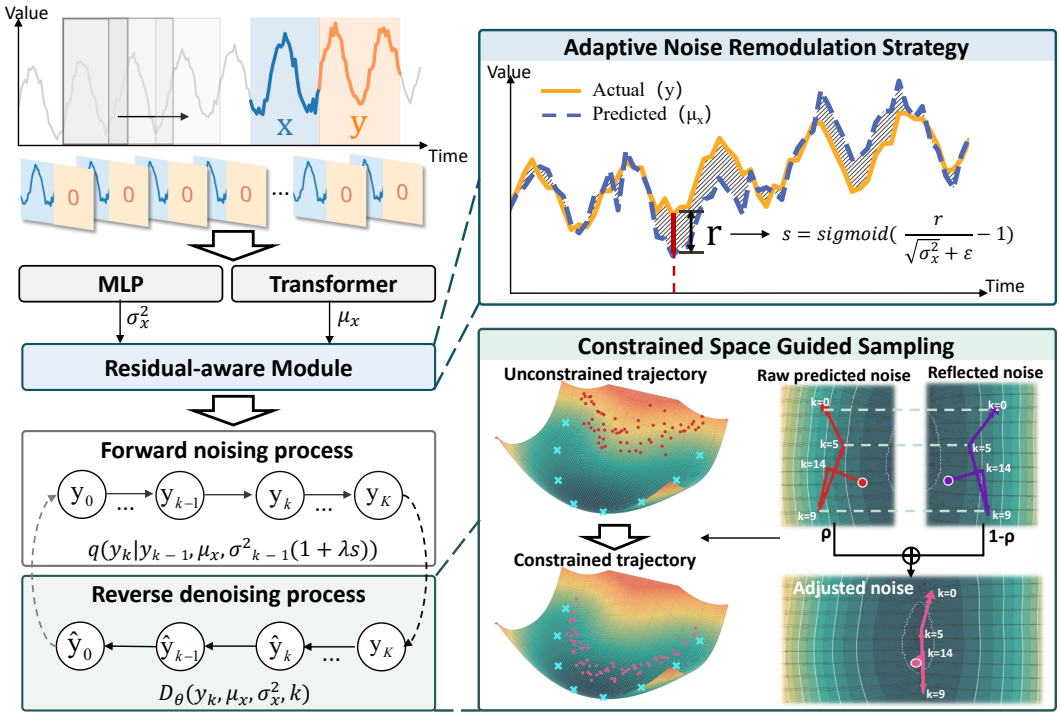


Figure 2: Overall framework of FAIP-Diff. The time series is first segmented into sliding windows and fed into prior conditional networks. During training, the adaptive noise remodulation strategy computes an anomaly-aware score from the residual and adaptively adjusts the forward diffusion noise variance accordingly. During inference, the constrained-space guided sampling strategy corrects trajectory drift and guides the reverse denoising process.

4.2 Adaptive Noise Remodulation Strategy

To alleviate temporal fluctuation suppression during forward noising, FAIP-Diff introduces an **adaptive noise remodulation strategy**. Since the conditional mean $\boldsymbol{\mu}_x$ is predicted from historical conditions, its prediction accuracy reflects, to some extent, the forecasting difficulty at each time step. During training, the model first computes the absolute residual $\mathbf{r}$ between the ground-truth future sequence $\mathbf{y}_0$ and the conditional mean $\boldsymbol{\mu}_x$. Furthermore, since the raw residuals may be affected by the magnitude of local fluctuations, FAIP-Diff further normalizes the residual using the predicted conditional volatility $\boldsymbol{\sigma}_x^2$. A sigmoid function maps the normalized residual to obtain the anomaly-aware score $\mathbf{s}$ as

$$\mathbf{s} = \text{sigmoid}\left(\frac{\mathbf{r}}{\sqrt{\boldsymbol{\sigma}_x^2 + \varepsilon}} - 1\right), \quad \mathbf{r} = |\mathbf{y}_0 - \boldsymbol{\mu}_x|, \tag{7}$$

where ε is a small positive constant for numerical stability. A higher $\mathbf{s}$ indicates that the corresponding temporal position is more difficult to predict relative to its predicted volatility level.

Based on the conditional volatility predicted by historical sequences and the true local volatility $\boldsymbol{\sigma}_y^2$, the baseline forward noise variance used in FAIP-Diff is given in Eq. 8. The detailed derivation process of the noise variance can be found in Appendix A.1 (see Eqs. 22-25).

$$\boldsymbol{\sigma}_{\text{base},k}^2 = (\bar{\beta}_k - \tilde{\beta}_k)\boldsymbol{\sigma}_x^2 + \tilde{\beta}_k \boldsymbol{\sigma}_y^2. \tag{8}$$

The anomaly-aware score is used to modulate the base noise variance, so that regions with large residuals receive more targeted perturbations during the forward diffusion process:

$$\boldsymbol{\sigma}_{adapt,k}^2 = \boldsymbol{\sigma}_{base,k}^2 \odot (1 + \lambda \mathbf{s}), \tag{9}$$

where λ controls the modulation strength and $\odot$ denotes vector multiplication. The final result is

$$\boldsymbol{\epsilon}_k = \boldsymbol{\epsilon} \odot \sqrt{\boldsymbol{\sigma}_{adapt,k}^2}, \tag{10}$$

where $\boldsymbol{\epsilon}$ is sampled from a standard Gaussian distribution. Based on the standard forward diffusion process shown in Eq. 1 and the forecasting priors introduced in Section 4.1 . The detailed derivation process of noise variance can be found in Appendix A.1(see Eqs. 22-25). During the training phase, the adjusted noise addition result is

$$\mathbf{y}_k = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + (1 - \sqrt{\bar{\alpha}_k})\,\boldsymbol{\mu}_x + \boldsymbol{\epsilon}_k. \tag{11}$$

Additionally, for the validation and test sets, the ground-truth future values are not used for noise injection. Instead, the state at the diffusion step is initialized directly based on the conditional priors. Therefore, the highest noise state $\mathbf{y}_K$ obtained by direct initialization in the inference stage is

$$\mathbf{y}_K = \boldsymbol{\mu}_x + \sqrt{\boldsymbol{\sigma}_x^2} \odot \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{12}$$

4.3 Constrained Space Guided Sampling

To mitigate denoising trajectory drift during reverse sampling, FAIP-Diff introduces **Constrained Space Guided Sampling**. The ground-truth future windows in the training set are first flattened into vectors, and principal component analysis (PCA) is then applied to extract the top-R orthogonal basis vectors as

$$\mathbf{B} = \mathrm{PCA}(\{\mathrm{Flatten}(\mathbf{y})\}, R). \tag{13}$$

These basis vectors span a low-dimensional subspace that represents the dominant variation patterns of real future sequences in the training data. The subspace is constructed only from training future windows and remains fixed during validation and testing, avoiding future information leakage. It serves as an external structural prior rather than an additional input to the denoising network.

Training Phase. Based on the standard reverse diffusion step in Eq. 2 and the forecasting priors introduced in Section 4.1. A diffusion step is randomly selected, and the noise variance and the future data at that step are computed as

$$\hat{\boldsymbol{\sigma}}^2_{\epsilon,k} = (\bar{\beta}_k - \tilde{\beta}_k)\boldsymbol{\sigma}_x^2 + \tilde{\beta}_k \boldsymbol{\sigma}^2_{y,k}. \tag{14}$$

$$\tilde{\mathbf{y}}_0^{(k)} = \frac{\mathbf{y}_k - (1 - \sqrt{\bar{\alpha}_k})\,\boldsymbol{\mu}_x - \hat{\boldsymbol{\epsilon}}_k \odot \sqrt{\hat{\boldsymbol{\sigma}}^2_{\epsilon,k}}}{\sqrt{\bar{\alpha}_k}}. \tag{15}$$

Inference Phase. During inference, the model parameters remain fixed, and the noisy data are reconstructed through a progressive denoising process. In addition, FAIP-Diff projects the predicted noise onto the future-pattern subspace learned during training to construct the reflected noise direction with respect to this subspace. The reflected noise is softly fused with the original predicted noise to obtain the corrected denoising noise $\boldsymbol{\epsilon}'_k$ as

$$\begin{aligned} \boldsymbol{\epsilon}_{ref} &= 2\hat{\boldsymbol{\epsilon}}_k \mathbf{B}^\top \mathbf{B} - \hat{\boldsymbol{\epsilon}}_k, \\ \boldsymbol{\epsilon}'_k &= (1 - \rho)\hat{\boldsymbol{\epsilon}}_{\boldsymbol{k}} + \rho \boldsymbol{\epsilon}_{ref}, \end{aligned} \tag{16}$$

where $\rho \in [0, 1]$ controls the regularization strength. The corrected noise is reshaped to the original tensor form and used to compute the reverse mean for sampling. That is, $\hat{\boldsymbol{\epsilon}}_k$ is replaced with $\boldsymbol{\epsilon}'_k$. When $\rho = 0$, the method reduces to standard reverse diffusion sampling.

4.4 Volatility-Aware Weighted Training Objective

Uniformly aggregating training errors may underemphasize infrequent but informative high-fluctuation regions. To address this issue, we introduce a **Volatility-Aware Weighted Training Objective**, which adaptively reweights the volatility prediction loss according to residual intensity. The residual-aware weight is defined as

$$\mathbf{w} = \mathrm{Clip}\left(\frac{\mathbf{r}}{\mathrm{Mean}(\mathbf{r}) + \varepsilon}, 0, c\right), \tag{17}$$

where c is the clipping threshold and the role of ε is consistent with that of ε in Eq. 7. Volatility-guided weighted loss and the overall training objective as

$$\mathcal{L}_{var} = \mathrm{Mean}\left[\mathbf{w} \odot \left(\sqrt{\boldsymbol{\sigma}_x^2} - \sqrt{\boldsymbol{\sigma}_y^2}\right)^2\right]. \tag{18}$$

$$\mathcal{L} = \mathcal{L}_{KL} + \mathcal{L}_{mean} + \mathcal{L}_{var}. \tag{19}$$

By optimizing joint loss, FAIP-Diff has further enhanced its sample generation capability, central trajectory prediction capability, and volatility calibration capability.

5 EXPERIMENTS

In this section, we evaluate the effectiveness of FAIP-Diff on multiple widely used multivariate TSF datasets. The experiments are designed to investigate whether FAIP-Diff outperforms existing methods in probabilistic TSF (Section 5.2), whether the adaptive noise remodulation strategy, constrained space guided sampling, and volatility-aware weighted training objective are individually effective (Section 5.3), whether the key components of FAIP-Diff can preserve the information contained in the data (Section I), how key hyperparameters affect model performance (Section 5.4), and how computationally efficient FAIP-Diff is (Section 5.5).

5.1 EXPERIMENTAL SETUP

Datasets We conduct experiments on eight widely used multivariate TSF datasets, including ETT (Zhou et al., 2021), Electricity, Weather, Exchange Rate, and ILI (Wu et al., 2021). These datasets cover different application scenarios and exhibit varying degrees of non-stationarity. Appendix B presents the dataset introduction and analysis, along with the corresponding training settings.

Baselines To comprehensively evaluate the performance of FAIP-Diff, we compare it with both conventional TSF methods and diffusion-based forecasting methods, such as Autoformer (Wu et al., 2021), Informer (Zhou et al., 2021), TimesNet (Wu et al., 2023), TimeGrad (Rasul et al., 2021a), CSDI (Tashiro et al., 2021), TimeDiff (Shen & Kwok, 2023), Diffusion-TS (Yuan & Qiao, 2024), TMDM (Li et al., 2024), and NsDiff (Ye et al., 2025), K-ProtoDiff (Duan et al., 2026).

Evaluation Metrics Since FAIP-Diff is designed for probabilistic forecasting, we mainly adopt CRPS and QICE as the core evaluation metrics. The calculation process for these indicators is provided in Appendix C. In addition, we report MAE and MSE as point forecasting metrics.

Implementation Details The number of diffusion steps is set to 20. The adaptive noise remodulation strength λ is set to 1.0, and the reflection strength ρ in structural regularized sampling is set to 0.6. Train the exported future pattern subspace to retain 20 principal components.

5.2 MAIN RESULTS

As shown in Table 1, FAIP-Diff achieves better overall performance on most datasets, with particularly clear advantages on Weather, Exchange Rate, and ILI, which contain stronger local fluctuations. These results suggest that adaptive noise remodulation and volatility-guided weighted learning improve the modeling of local uncertainty in non-stationary sequences.

Compared with the strong diffusion-based baseline NsDiff, FAIP-Diff consistently improves across all four metrics on Exchange Rate, ILI, and Weather. These gains are especially pronounced on highly variable, non-stationary datasets, where irregular fluctuations make future trajectories difficult to recover. By placing greater learning emphasis on challenging temporal regions and constraining trajectory drift during reverse sampling, FAIP-Diff captures local uncertainty more effectively while maintaining realistic temporal structures. Although TimeDiff obtains a better CRPS on Weather, its considerably higher QICE, MAE, and MSE indicate weaker calibration and deterministic accuracy. In contrast, FAIP-Diff provides more balanced performance across distributional quality, probabilistic calibration, and point-forecasting accuracy, demonstrating greater robustness under complex temporal dynamics.

5.3 ABLATION STUDY

We conduct ablation studies to assess the contributions of the adaptive noise remodulation strategy (ANR), constrained space guided sampling (CSG), and the volatility-aware weighted objective (VAW). As shown in Table 2, each component reduces the average CRPS, MAE, and MSE compared with the base model, although ANR alone slightly increases QICE. Pairwise combinations further improve most cases. The full model obtains the lowest average values across all four metrics, suggesting that the three components provide complementary benefits for forecasting accuracy and calibration. Appendix G contains a complete ablation experiment.

Table 1: Forecasting performance comparison on multivariate time-series datasets. Each dataset reports CRPS, QICE, MAE, and MSE. Lower values are better. The best results are highlighted in bold, and the second-best results are underlined. "–" denotes unavailable results.

Model	ETTh1				ETTh2				ETTm1				ETTm2			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
TimeGrad	0.606	6.731	0.813	1.062	1.212	9.488	1.496	3.462	0.647	6.693	0.831	1.218	0.775	6.962	0.967	1.690
CSDI	0.492	3.107	0.708	0.949	0.647	5.331	0.900	1.226	0.524	2.828	0.752	1.002	0.817	8.106	1.069	1.723
TimeDiff	0.465	14.931	**0.479**	0.517	0.471	14.813	0.485	0.456	0.464	14.795	0.477	0.537	0.316	13.385	0.333	0.268
DiffusionTS	0.603	6.423	0.774	1.089	1.168	9.577	1.411	3.273	0.574	5.605	0.744	1.030	1.035	9.959	1.232	2.372
TMDM	0.452	2.821	0.607	0.696	0.383	4.471	0.490	0.512	0.375	2.567	0.455	0.494	0.289	2.610	0.395	0.315
NsDiff	0.392	1.470	0.523	0.594	0.358	2.074	0.490	0.514	0.346	2.041	0.455	0.488	0.256	**2.030**	0.352	0.281
K-ProtoDiff	0.532	11.119	0.597	0.758	0.403	13.052	0.445	0.382	0.513	10.261	0.589	0.693	0.378	12.908	0.406	0.289
Autoformer	–	–	0.641	0.726	–	–	0.802	1.086	–	–	0.522	0.564	–	–	0.463	0.431
Informer	–	–	0.930	1.445	–	–	1.892	5.486	–	–	0.818	1.138	–	–	1.473	3.594
TimesNet	–	–	0.499	0.505	–	–	0.455	0.433	–	–	**0.425**	**0.408**	–	–	0.354	0.300
FAIP-Diff	**0.367**	**1.317**	0.486	**0.497**	**0.320**	**1.388**	**0.439**	**0.378**	**0.330**	**1.794**	0.429	0.436	**0.246**	2.151	**0.323**	**0.235**

Model	Electricity				Exchange Rate				ILI				Weather			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
TimeGrad	0.397	7.118	0.504	0.505	0.826	9.464	1.058	1.567	1.140	6.519	1.414	4.197	–	–	–	–
CSDI	0.577	7.506	0.822	1.007	0.855	7.864	1.081	1.701	1.244	7.693	1.481	4.515	0.322	14.032	0.354	0.258
TimeDiff	0.750	15.466	0.764	0.879	0.433	14.556	0.446	0.402	1.153	14.942	1.169	3.958	**0.138**	10.670	0.305	0.228
DiffusionTS	0.633	8.205	0.856	1.072	1.251	10.411	1.564	3.628	1.612	10.090	1.788	6.053	0.298	5.383	0.609	0.625
TMDM	0.461	10.562	0.359	0.257	0.336	6.393	0.430	0.334	0.967	6.217	1.175	3.636	–	–	0.241	0.180
NsDiff	0.290	**6.685**	**0.306**	0.209	0.324	5.930	0.412	0.300	0.806	5.598	0.985	2.846	0.224	5.208	0.276	0.223
K-ProtoDiff	–	–	–	–	0.625	13.798	0.657	0.634	1.797	15.951	1.799	6.777	0.314	11.384	0.352	0.320
Autoformer	–	–	0.343	0.236	–	–	0.589	0.447	–	–	–	–	–	–	0.457	0.446
Informer	–	–	0.464	0.408	–	–	0.581	0.432	–	–	–	–	–	–	0.513	0.567
TimesNet	–	–	0.334	0.238	–	–	0.425	0.326	–	–	–	–	–	–	0.293	0.254
FAIP-Diff	**0.285**	7.361	**0.306**	**0.202**	**0.264**	**4.842**	**0.338**	**0.230**	**0.663**	**4.841**	**0.812**	**1.907**	0.197	**3.765**	**0.238**	**0.158**

5.4 Parameter Sensitivity Analysis

We evaluate the sensitivity of FAIP-Diff to different prediction horizons and the results are shown in Figure 3. Please refer to Appendix F for details. As the prediction horizon increases, FAIP-Diff relatively stable on the ETT datasets but exhibits a larger performance degradation on Exchange Rate. This suggests that long-term forecasting scenarios remain more challenging.

We analyze the sensitivity of FAIP-Diff to the adaptive noise remodulation strength λ, reflection strength ρ, and number of prototype components R. As shown in Figure 4, excessively large values of λ and ρ may lead to denoising difficulty or overly constrain sample diversity. Although $R = 40$ achieves better CRPS, MAE, and MSE, $R = 20$ yields the best QICE while also achieving superior performance on the other metrics. A compact subspace also retains the dominant temporal structures while reducing the inclusion of low-variance or redundant directions.

Table 2: Average ablation results across datasets.

Model	CRPS	QICE(%)	MAE	MSE
Base	0.375	3.900	0.475	0.682
+ANR	0.366	4.096	0.453	0.650
+CSG	0.366	3.697	0.455	0.646
+VAW	0.359	3.581	0.447	0.609
+ANR+CSG	0.353	3.803	0.441	0.586
+ANR+VAW	0.346	3.597	0.431	0.572
+VAW+CSG	0.350	3.726	0.435	0.564
FAIP-Diff	**0.334**	**3.474**	**0.420**	**0.505**

5.5 Efficiency Analysis

We further evaluate the computational efficiency of FAIP-Diff and representative diffusion-based forecasting baselines on ETTh1. Table 3 summarizes the results. FAIP-Diff has the same number of parameters as NsDiff. Its inference time increases slightly from 7.84s to 8.30s on 12 test inputs. In terms of memory usage, FAIP-Diff requires 103.13MB peak GPU memory.

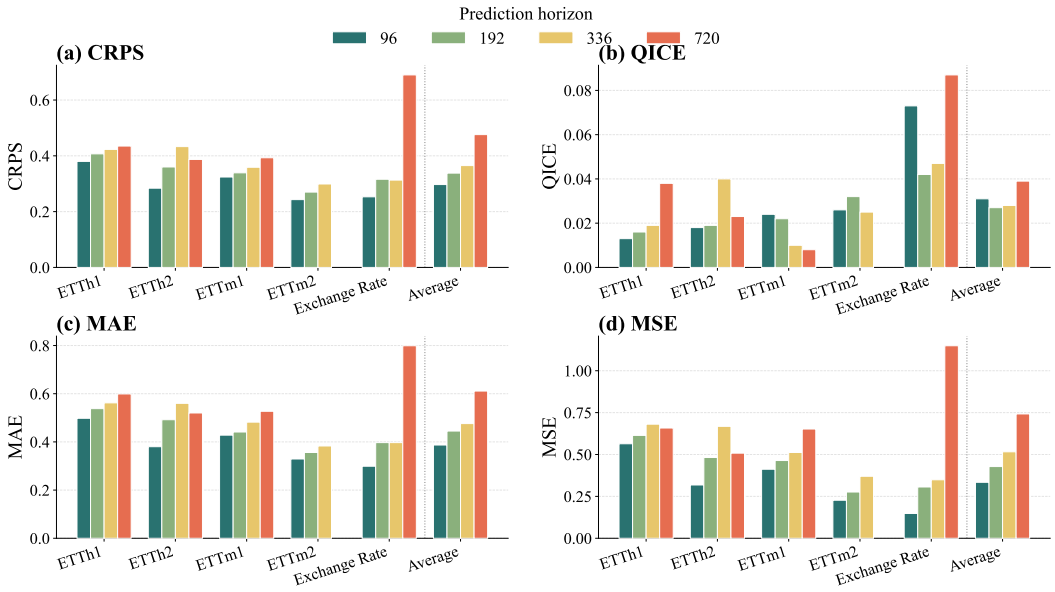


Figure 3: Prediction-horizon sensitivity of FAIP-Diff on the ETT and Exchange Rate datasets.

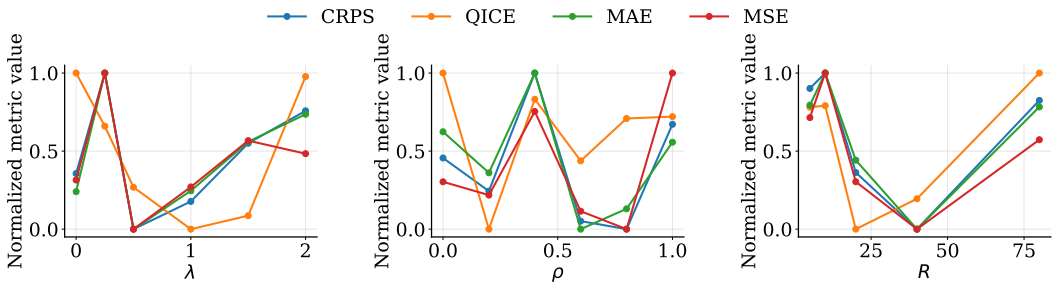


Figure 4: Parameter sensitivity of FAIP-Diff with respect to λ, ρ, and R.

This suggests that the proposed constrained-space guidance introduces limited memory overhead. FAIP-Diff keeps the same parameter scale as NsDiff, while providing additional fluctuation-aware and structure-guided forecasting capabilities.

Table 3: Efficiency comparison on ETTh1.

Model	Params (M)	Init. (s)	Infer. (s)	ms/Input	Inputs/s	GPU Mem. (MB)
CSDI	0.61	0.17	107.44	8952.97	0.11	74.93
TimeDiff	3.24	0.16	40.48	3372.99	0.30	30.49
TMDM	9.07	0.12	2.32	193.50	5.17	141.96
NsDiff	9.44	0.13	7.84	653.29	1.53	103.03
FAIP-Diff	9.44	0.14	8.30	691.41	1.45	103.13

6 Conclusion and Future Work

FAIP-Diff introduces an adaptive noise remodulation strategy to alleviate temporal fluctuation suppression during forward noising and develops a constrained-space guided sampling mechanism to mitigate denoising trajectory drift during reverse sampling. Experiments on multiple benchmarks show that these components can improve distributional accuracy and calibration.

Future work will explore dynamic mechanisms and multimodal extensions. First, real-world time series often exhibit time-varying dynamics. Learning step-dependent or context-dependent guidance mechanisms may further improve adaptability under changing temporal patterns. Second, many forecasting scenarios involve multimodal information. Incorporating such multimodal data into diffusion forecasting may help the model improve uncertainty estimation.

AI USE STATEMENT

In this work, we have not used generative AI tools for any task requiring disclosure, and the remaining required disclosure tasks are not applicable to this work. We have not used generative AI tools to produce experimental results, fabricate data, construct baselines, or generate unsupported scientific claims. We used generative AI tools only to edit the manuscript for grammar and readability. All AI-assisted edits were reviewed by the authors to ensure that they did not alter the technical content, scientific claims, or intended meaning. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

We confirm that our work has been conducted in accordance with the ICLR Code of Ethics `https://iclr.cc/public/CodeOfEthics`. The study does not involve human subjects, sensitive personal data, or experiments that may cause harm to individuals or groups. The datasets used are publicly available and no personally identifiable information is included. Our methodology and findings are intended for academic purposes and do not pose foreseeable risks of misuse. We have carefully considered issues of fairness, bias, and privacy, and to the best of our knowledge, our research maintains integrity and complies with all applicable ethical standards.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide the complete implementation of FAIP-Diff in the anonymous supplementary repository at `https://anonymous.4open.science/r/FAIP_Diff-7884`. The released materials include data-loading and preprocessing procedures, model training and probabilistic sampling pipelines, evaluation metrics, and dependency specifications. We will make an anonymized code repository available during review, and release the maintained public repository upon acceptance.

REFERENCES

Pelin Akın, Tuba Koç, and Haydar Koç. Hybrid LSTM Model with Efficient Hyperparameter Tuning for Earthquake Magnitude Prediction in Turkey. *Soil Dynamics and Earthquake Engineering*, 200:109753, 2026.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-Based Conditional ECG Generation with Structured State Space Models. *Computers in Biology and Medicine*, 163:107115, 2023.

Defu Cao, Wen Ye, Yizhou Zhang, Sam Griesemer, and Yan Liu. PINFDiT: Energy-Based Physics-Informed Diffusion Transformers for General-purpose Time Series Tasks. In *International Conference on Learning Representations*, pp. 156753–156799, 2026.

Edgardo Solano Carrillo, Sreerag V Naveenachandran, and Julia Niebling. TEDM: Time Series Forecasting with Elucidated Diffusion Models. In *International Conference on Learning Representations*, pp. 1–31, 2026.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramírez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural Hierarchical Interpolation for Time Series Forecasting. In *Association for the Advancement of Artificial Intelligence*, pp. 6989–6997, 2023.

Xiaoqian Chen and Lalit Gupta. Training LSTMS with Circular-Shift Epochs for Accurate Event Forecasting in Imbalanced Time Series. *Expert Systems with Applications*, 238:121701, 2024.

Andrea Cini, Alexander Jenkins, Danilo Mandic, Cesare Alippi, and Filippo Maria Bianchi. Relational Conformal Prediction for Correlated Time Series. In *International Conference on Machine Learning*, pp. 1–17, 2025.

Yuhang Duan, Lin Lin, and Xiaoshuai Wu. K-ProtoDiff: Key Prototypes-Guided Diffusion for Time Series Generation. In *Association for the Advancement of Artificial Intelligence*, pp. 20959–20967, 2026.

Gal Fadlon, Idan Arbiv, Nimrod Berman, and Omri Azencot. A Diffusion Model for Regular Time Series Generation from Irregular Data with Completion and Masking. In *Advances in Neural Information Processing Systems*, pp. 1–36, 2025.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-TS: A General Paradigm for Alleviating Distribution Shift in Time Series Forecasting. In *Association for the Advancement of Artificial Intelligence*, volume 37, pp. 7522–7529, 2023.

Amit Frishberg, Neta Milman, Ayelet Alpert, Hannah Spitzer, Ben Asani, Johannes B. Schiefelbein, Evgeny Bakin, Karen Regev-Berman, Siegfried G. Priglinger, Joachim L. Schultze, Fabian J. Theis, and Shai S. Shen-Orr. Reconstructing Disease Dynamics for Mechanistic Insights and Clinical Benefit. *Nature Communications*, 14(1):6840, 2023.

Tal Gonen, Itai Pemper, Ilan Naiman, Nimrod Berman, and Omri Azencot. Time Series Generation Under Data Scarcity: A Unified Generative Modeling Approach. In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 172751–172790, 2025.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, pp. 6840–6851, 2020.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible Instance Normalization for Accurate Time-Series Forecasting against Distribution Shift. In *International Conference on Learning Representations*, pp. 1–25, 2022.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-Modulated Diffusion Models for Probabilistic Multivariate Time Series Forecasting. In *International Conference on Learning Representations*, pp. 18604–18622, 2024.

Bryan Lim, Sercan Ö. Arık, Nicolas Loeff, and Tomas Pfister. Temporal Fusion Transformers for Interpretable Multi-Horizon Time Series Forecasting. *International Journal of Forecasting*, 37 (4):1748–1764, 2021.

Peiyuan Liu, Beiliang Wu, Yifan Hu, Naiqi Li, Tao Dai, Jigang Bao, and Shu-Tao Xia. TimeBridge: Non-Stationarity Matters for Long-term Time Series Forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840, 2025a.

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X. Liu, and Schahram Dustdar. Pyraformer: Low-Complexity Pyramidal Attention for Long-Range Time Series Modeling and Forecasting. In *International Conference on Learning Representations*, pp. 1–20, 2022a.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary Transformers: Exploring the Stationarity in Time Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted Transformers Are Effective for Time Series Forecasting. In *International Conference on Learning Representations*, pp. 11116–11140, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-Context Transformers for Unified Time Series Forecasting. In *International Conference on Learning Representations*, pp. 83982–84006, 2025b.

Sai Shankar Narasimhan, Shubhankar Agarwal, Oguzhan Akcin, Sujay Sanghavi, and Sandeep P. Chinchali. Time Weaver: A Conditional Time Series Generation Model. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 37293–37320, 2024.

Alex Nichol and Prafulla Dhariwal. Improved Denoising Diffusion Probabilistic Models. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. In *International Conference on Learning Representations*, pp. 1–24, 2023.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive Denoising Diffusion Models for Multivariate Probabilistic Time Series Forecasting. In *International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868, 2021a.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. Multivariate Probabilistic Time Series Forecasting via Conditioned Normalizing Flows. In *International Conference on Learning Representations*, pp. 1–19, 2021b.

J Rishi, Gvs Mothish, and Deepak Subramani. Conditional Diffusion Model with Nonlinear Data Transformation for Time Series Forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51703–51723, 2025.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks. *International Journal of Forecasting*, 36 (3):1181–1191, 2020.

Junhyeon Seo and Pawan Gupta. Adaptive Expert-Guided Deep Imbalanced Regression for Global PM2.5 Forecasting with Temporal Convolutional Networks and GEOS-FP Inputs. *npj Climate and Atmospheric Science*, 9(1):171, 2026.

Lifeng Shen and James T. Kwok. Non-autoregressive Conditional Diffusion Models for Time Series Prediction. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31016–31029, 2023.

Lifeng Shen, Weiyu Chen, and James T. Kwok. Multi-Resolution Diffusion Models for Time Series Forecasting. In *International Conference on Learning Representations*, pp. 49042–49060, 2024.

Lifeng Shen, Kai Syun Hou, Weiyu Chen, and James T. Kwok. Latent-to-Data Cascaded Diffusion Models for Unconditional Time Series Generation. In *International Conference on Learning Representations*, pp. 35058–35079, 2026.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-Based Generative Modeling through Stochastic Differential Equations. In *International Conference on Learning Representations*, pp. 1–36, 2021.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional Score-based Diffusion Models for Probabilistic Time Series Imputation. In *Advances in Neural Information Processing Systems*, volume 34, pp. 24804–24816, 2021.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting. In *Advances in Neural Information Processing Systems*, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-Variation Modeling for General Time Series Analysis. In *International Conference on Learning Representations*, pp. 1–23, 2023.

Xingjian Wu, Xiangfei Qiu, Hongfan Gao, Jilin Hu, Bin Yang, and Chenjuan Guo. K^2VAE: A Koopman-Kalman Enhanced Variational AutoEncoder for Probabilistic Time Series Forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67562–67583, 2025.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary Diffusion For Probabilistic Time Series Forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72112–72130, 2025.

Xinyu Yuan and Yan Qiao. Diffusion-TS: Interpretable Diffusion for General Time Series Generation. In *International Conference on Learning Representations*, pp. 41582–41610, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are Transformers Effective for Time Series Forecasting? In *Association for the Advancement of Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer Utilizing Cross-Dimension Dependency for Multivariate Time Series Forecasting. In *International Conference on Learning Representations*, pp. 1–21, 2023.

Yunhao Zhang, Wenyao Hu, Jiale Zheng, Lujia Pan, and Junchi Yan. MMPD: Diverse Time Series Forecasting via Multi-Mode Patch Diffusion Loss. In *International Conference on Learning Representations*, pp. 1–29, 2026.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. In *Association for the Advancement of Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing MA, Xue Wang, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, and Rong Jin. FiLM: Frequency improved Legendre Memory Model for Long-term Time Series Forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 12677–12690, 2022a.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency Enhanced Decomposed Transformer for Long-term Series Forecasting. In *International Conference on Machine Learning*, volume 162, pp. 27268–27286, 2022b.

Kechao Zhu, Linlong Chen, Danfeng Wang, and Kun Fan. Research on Traffic Flow Prediction Based on Dual-Channel Hybrid Graph Learning with Trend-Aware Attention. *Scientific Reports*, 16:19867, 2026.

A Theoretical Analysis of the Proposed Components

In this appendix, we provide theoretical interpretations of the three key components in FAIP-Diff, include adaptive noise remodulation strategy A.1, constrained space guided sampling A.2, and the volatility-aware weighted training objective A.3. The analysis does not claim unconditional performance improvement for arbitrary data distributions, but shows that the proposed designs are consistent with heteroscedastic uncertainty modeling, sampling-time structural regularization, and importance-weighted risk minimization.

A.1 Adaptive Noise Remodulation Strategy

For a ground-truth future sequence $\mathbf{y}_0$, the conditional mean prediction is denoted by $\boldsymbol{\mu}_x$, and the predicted conditional volatility is denoted by $\boldsymbol{\sigma}_x^2$. The absolute residual is defined as

$$\mathbf{r} = |\mathbf{y}_0 - \boldsymbol{\mu}_x|. \tag{20}$$

FAIP-Diff defines an anomaly-aware score by normalizing the residual with the predicted conditional volatility:

$$\mathbf{s} = \text{sigmoid}\left(\frac{\mathbf{r}}{\sqrt{\boldsymbol{\sigma}_x^2 + \varepsilon}} - 1\right), \qquad \mathbf{s} \in (0, 1), \tag{21}$$

where ε is a small positive constant for numerical stability.

Following the notation in Section 4.2, let $\{\beta_i\}_{i=1}^K$ denote the predefined diffusion noise schedule and $\alpha_i = 1 - \beta_i$. At diffusion step k, the cumulative noise coefficient is

$$\bar{\beta}_k = 1 - \bar{\alpha}_k = 1 - \prod_{i=1}^{k} \alpha_i = \sum_{i=1}^{k} \left(\beta_i \prod_{j=i+1}^{k} \alpha_j\right). \tag{22}$$

In our implementation, another schedule-dependent coefficient$\tilde{\beta}_t$ is constructed from the diffusion schedule:

$$\tilde{\beta}_k = \sum_{i=1}^{k} \left(\prod_{j=i}^{k} \alpha_j\right) - \sum_{i=1}^{k} \left(\alpha_i^2 \prod_{j=i+1}^{k} \alpha_j\right). \tag{23}$$

Since $\prod_{j=i}^{k} \alpha_j = \alpha_i \prod_{j=i+1}^{k} \alpha_j$,we can further expand $\tilde{\beta}_k$ and ultimately obtain

$$\tilde{\beta}_k = \sum_{i=1}^{k} \left(\alpha_i \prod_{j=i+1}^{k} \alpha_j \right) - \sum_{i=1}^{k} \left(\alpha_i^2 \prod_{j=i+1}^{k} \alpha_j \right) = \sum_{i=1}^{k} \left(\alpha_i \beta_i \prod_{j=i+1}^{k} \alpha_j \right). \tag{24}$$

$$\begin{aligned} \bar{\beta}_k - \tilde{\beta}_k &= \sum_{i=1}^{k} \left(\beta_i \prod_{j=i+1}^{k} \alpha_j \right) - \sum_{i=1}^{k} \left(\alpha_i \beta_i \prod_{j=i+1}^{k} \alpha_j \right) \\ &= \sum_{i=1}^{k} \left(\beta_i^2 \prod_{j=i+1}^{k} \alpha_j \right). \end{aligned} \tag{25}$$

The base forward noise variance used in FAIP-Diff is computed as

$$\boldsymbol{\sigma}_{\text{base},k}^2 = (\bar{\beta}_k - \tilde{\beta}_k)\boldsymbol{\sigma}_x^2 + \tilde{\beta}_k \boldsymbol{\sigma}_y^2, \tag{26}$$

where $\boldsymbol{\sigma}_y^2$ denotes the target local volatility estimated from the real sequence. Equivalently, substituting the expanded forms gives

$$\boldsymbol{\sigma}_{\text{base},k}^2 = \left[\sum_{i=1}^{k} \left(\beta_i^2 \prod_{j=i+1}^{k} \alpha_j \right) \right] \boldsymbol{\sigma}_x^2 + \left[\sum_{i=1}^{k} \left(\alpha_i \beta_i \prod_{j=i+1}^{k} \alpha_j \right) \right] \boldsymbol{\sigma}_y^2. \tag{27}$$

FAIP-Diff further modulates the base variance using the anomaly-aware score:

$$\boldsymbol{\sigma}_{\text{adapt},k}^2 = \boldsymbol{\sigma}_{\text{base},k}^2 \odot (1 + \lambda \mathbf{s}), \qquad \lambda \geq 0, \tag{28}$$

where λ controls the modulation strength.

Property 1: monotonicity with respect to residual intensity. Let

$$\mathbf{u} = \frac{\mathbf{r}}{\sqrt{\boldsymbol{\sigma}_x^2 + \epsilon}} - 1. \tag{29}$$

Since $\mathbf{s} = \text{sigmoid}(\mathbf{u})$, we have

$$\frac{\partial \mathbf{s}}{\partial \mathbf{r}} = \text{sigmoid}(\mathbf{u}) \left(1 - \text{sigmoid}(\mathbf{u})\right) \frac{1}{\sqrt{\boldsymbol{\sigma}_x^2 + \epsilon}} > 0. \tag{30}$$

Therefore,

$$\frac{\partial \boldsymbol{\sigma}_{\text{adapt},k}^2}{\partial \mathbf{r}} = \lambda \boldsymbol{\sigma}_{\text{base},k}^2 \odot \frac{\partial \mathbf{s}}{\partial \mathbf{r}} > 0, \tag{31}$$

when $\lambda > 0$ and $\boldsymbol{\sigma}_{\text{base},k}^2 > 0$. This indicates that temporal positions with larger normalized residuals receive stronger forward perturbations.

Property 2: bounded noise amplification. Because $\mathbf{s} \in (0, 1)$, the adaptive variance satisfies

$$\boldsymbol{\sigma}_{\text{base},k}^2 < \boldsymbol{\sigma}_{\text{adapt},k}^2 < (1 + \lambda)\boldsymbol{\sigma}_{\text{base},k}^2. \tag{32}$$

Thus, adaptive noise remodulation increases the perturbation strength in difficult regions while avoiding unbounded variance amplification.

Interpretation. With the adaptive variance, the noisy future state is generated as

$$\mathbf{y}_k = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + (1 - \sqrt{\bar{\alpha}_k})\boldsymbol{\mu}_x + \left(\boldsymbol{\epsilon} \odot \sqrt{\boldsymbol{\sigma}_{\text{adapt},k}^2} \right), \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \tag{33}$$

Therefore, the corresponding forward distribution is

$$q(\mathbf{y}_k \mid \mathbf{y}_0, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2, \boldsymbol{\sigma}_y^2, \mathbf{s}) = \mathcal{N}\left(\mathbf{y}_k; \boldsymbol{\mu}_{q,k}, \boldsymbol{\sigma}_{q,k}^2\right), \tag{34}$$

where

$$\boldsymbol{\mu}_{q,k} = \sqrt{\bar{\alpha}_k}\mathbf{y}_0 + (1 - \sqrt{\bar{\alpha}_k})\boldsymbol{\mu}_x, \tag{35}$$

and

$$\boldsymbol{\sigma}_{q,k}^2 = \boldsymbol{\sigma}_{\text{adapt},k}^2. \tag{36}$$

Compared with a uniform forward noise schedule, this formulation introduces a heteroscedastic perturbation process that explicitly depends on local forecasting difficulty and conditional volatility.

A.2 CONSTRAINED-SPACE GUIDED SAMPLING

FAIP-Diff constructs a future-pattern subspace from ground-truth future windows in the training set. Let each future window be flattened as

$$\mathbf{v}^{(i)} = \mathrm{Flatten}(\mathbf{y}^{(i)}) \in \mathbb{R}^{ON}, \qquad i = 1, \dots, U, \tag{37}$$

where O is the prediction horizon, N is the number of variables, and U is the number of training future windows used for subspace construction. PCA is applied to extract the top-R orthogonal basis vectors:

$$\mathbf{B} = [b_1, b_2, \dots, b_R], \qquad \mathbf{B}^\top \mathbf{B} = \mathbf{I}. \tag{38}$$

The resulting constrained space is

$$\mathcal{S} = \mathrm{span}(\mathbf{B}). \tag{39}$$

During reverse diffusion, the denoising network predicts the denoising noise and reverse variance:

$$\hat{\boldsymbol{\epsilon}}_k, \hat{\boldsymbol{\sigma}}_k^2 = D_\theta(\mathbf{y}_k, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2, k). \tag{40}$$

The predicted noise is flattened and projected onto the constrained future-pattern subspace. Following the implementation convention in Section 4.3, we write the projection as

$$\boldsymbol{\epsilon}_{proj} = \mathrm{Proj}_{\mathcal{S}}(\hat{\boldsymbol{\epsilon}}_k) = \hat{\boldsymbol{\epsilon}}_k \mathbf{B}^\top \mathbf{B}. \tag{41}$$

The reflected noise with respect to the subspace is

$$\boldsymbol{\epsilon}_{ref} = 2\boldsymbol{\epsilon}_{proj} - \hat{\boldsymbol{\epsilon}}_k. \tag{42}$$

The corrected denoising noise is obtained by soft fusion:

$$\boldsymbol{\epsilon}'_k = (1-\rho)\hat{\boldsymbol{\epsilon}}_k + \rho \boldsymbol{\epsilon}_{ref}, \qquad \rho \in [0, 1]. \tag{43}$$

Equivalently,

$$\boldsymbol{\epsilon}'_k = (1-2\rho)\hat{\boldsymbol{\epsilon}}_k + 2\rho\, \mathrm{Proj}_{\mathcal{S}}(\hat{\boldsymbol{\epsilon}}_k). \tag{44}$$

Property 1: decomposition of the corrected noise. Decompose the predicted noise into its subspace component and orthogonal component:

$$\hat{\boldsymbol{\epsilon}}_k = \boldsymbol{\epsilon}_\parallel + \boldsymbol{\epsilon}_\perp, \tag{45}$$

where

$$\boldsymbol{\epsilon}_\parallel = \mathrm{Proj}_{\mathcal{S}}(\hat{\boldsymbol{\epsilon}}_k), \qquad \boldsymbol{\epsilon}_\perp = \hat{\boldsymbol{\epsilon}}_k - \mathrm{Proj}_{\mathcal{S}}(\hat{\boldsymbol{\epsilon}}_k). \tag{46}$$

Then,

$$\boldsymbol{\epsilon}_{ref} = \boldsymbol{\epsilon}_\parallel - \boldsymbol{\epsilon}_\perp. \tag{47}$$

Therefore, the corrected noise becomes

$$\begin{aligned} \boldsymbol{\epsilon}'_k &= (1-\rho)(\boldsymbol{\epsilon}_\parallel + \boldsymbol{\epsilon}_\perp) + \rho(\boldsymbol{\epsilon}_\parallel - \boldsymbol{\epsilon}_\perp) \\ &= \boldsymbol{\epsilon}_\parallel + (1-2\rho)\boldsymbol{\epsilon}_\perp. \end{aligned} \tag{48}$$

This shows that the subspace component is preserved, while the off-subspace component is controlled by the reflection strength ρ.

Property 2: control of off-subspace deviation. The distance between the original predicted noise and the constrained space is

$$d(\hat{\boldsymbol{\epsilon}}_k, \mathcal{S}) = \|\boldsymbol{\epsilon}_\perp\|_2. \tag{49}$$

For the corrected noise, the distance becomes

$$d(\boldsymbol{\epsilon}'_k, \mathcal{S}) = \|(1-2\rho)\boldsymbol{\epsilon}_\perp\|_2 = |1-2\rho|\, \|\boldsymbol{\epsilon}_\perp\|_2. \tag{50}$$

When $\rho = 0$, FAIP-Diff reduces to standard reverse diffusion sampling:

$$\boldsymbol{\epsilon}'_k = \hat{\boldsymbol{\epsilon}}_k. \tag{51}$$

When $\rho = 0.5$, the corrected noise is exactly projected onto the constrained space:

$$\boldsymbol{\epsilon}'_k = \boldsymbol{\epsilon}_\parallel. \tag{52}$$

A moderate value of ρ therefore controls the off-subspace denoising direction while preserving stochasticity in the reverse process.

Effect on reverse diffusion. Let the reverse mean computed with the original predicted noise be

$$\boldsymbol{\mu}_{k-1} = \mu_\theta(\mathbf{y}_k, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2, \hat{\boldsymbol{\epsilon}}_k). \tag{53}$$

After constrained-space correction, the reverse mean becomes

$$\boldsymbol{\mu}'_{k-1} = \mu_\theta(\mathbf{y}_k, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2, \boldsymbol{\epsilon}'_k). \tag{54}$$

The reverse transition used for sampling can then be written as

$$p_\theta(\mathbf{y}_{k-1} \mid \mathbf{y}_k, \mathbf{x}) = \mathcal{N}\left(\mathbf{y}_{k-1}; \boldsymbol{\mu}'_{k-1}, \hat{\boldsymbol{\sigma}}_k^2\right). \tag{55}$$

Interpretation. Since the constrained space $\mathcal{S}$ is constructed only from training future windows, it captures dominant future variation patterns observed in the training data without using validation or test future information. The sampling strategy does not change the conditional inputs of the denoising network. Instead, it acts as a sampling-time structural regularizer by correcting the predicted denoising direction at each reverse step. This encourages generated trajectories to remain closer to realistic temporal structures while preserving sample diversity through the stochastic reverse variance.

A.3 Volatility-Aware Weighted Training Objective

The conditional volatility network predicts $\boldsymbol{\sigma}_x^2$, while the target local volatility computed from the real sequence is denoted by $\boldsymbol{\sigma}_y^2$. A basic volatility regression objective can be written as

$$\mathcal{L}_{\text{var}}^{\text{base}} = \text{Mean}\left[\left(\sqrt{\boldsymbol{\sigma}_x^2} - \sqrt{\boldsymbol{\sigma}_y^2}\right)^2\right]. \tag{56}$$

However, non-stationary time series are heteroscedastic, and the forecasting difficulty varies across temporal positions. FAIP-Diff therefore introduces a residual-based weight:

$$\mathbf{w} = \text{Clip}\left(\frac{\mathbf{r}}{\text{Mean}(\mathbf{r}) + \varepsilon}, 0, c\right), \tag{57}$$

where c is the clipping threshold. The volatility-aware weighted objective is

$$\mathcal{L}_{\text{var}} = \text{Mean}\left[\mathbf{w} \odot \left(\sqrt{\boldsymbol{\sigma}_x^2} - \sqrt{\boldsymbol{\sigma}_y^2}\right)^2\right]. \tag{58}$$

Importance-weighted risk interpretation. Let $z = (\mathbf{x}, \mathbf{y})$ denote a training example, and define the local volatility prediction error as

$$\ell_{\text{var}}(z) = \left(\sqrt{\boldsymbol{\sigma}_x^2} - \sqrt{\boldsymbol{\sigma}_y^2}\right)^2. \tag{59}$$

The unweighted volatility risk is

$$\mathcal{R}_{\text{var}} = \mathbb{E}_{z\sim p_{\text{data}}}\left[\ell_{\text{var}}(z)\right]. \tag{60}$$

The residual-weighted objective minimizes

$$\mathcal{R}_{\text{var}} = \mathbb{E}_{z\sim p_{\text{data}}}\left[\mathbf{w}(z)\ell_{\text{var}}(z)\right]. \tag{61}$$

This is equivalent to minimizing an unweighted risk under a reweighted data distribution

$$p_w(z) = \frac{\mathbf{w}(z)p_{\text{data}}(z)}{\mathbb{E}_{z\sim p_{\text{data}}}[\mathbf{w}(z)]}. \tag{62}$$

Thus,

$$\mathcal{R}_{\text{var}} = \mathbb{E}_{z\sim p_{\text{data}}}[\mathbf{w}(z)]\mathbb{E}_{z\sim p_w}\left[\ell_{\text{var}}(z)\right]. \tag{63}$$

Since $\mathbf{w}$ increases with residual intensity, high-residual and high-volatility regions receive larger effective importance in the volatility learning objective.

Gradient effect. For one temporal position, define

$$\mathbf{e}_{\mathrm{var}} = \sqrt{\boldsymbol{\sigma}_x^2} - \sqrt{\boldsymbol{\sigma}_y^2}. \tag{64}$$

The local weighted volatility loss is

$$\ell_{\mathrm{var}} = \mathbf{w} \odot \mathbf{e}_{\mathrm{var}}^2. \tag{65}$$

Its gradient with respect to $\sqrt{\boldsymbol{\sigma}_x^2}$ is

$$\frac{\partial \ell_{\mathrm{var}}}{\partial \sqrt{\boldsymbol{\sigma}_x^2}} = 2\mathbf{w} \odot \mathbf{e}_{\mathrm{var}}. \tag{66}$$

Compared with the unweighted objective, the gradient magnitude is scaled by $\mathbf{w}$. Therefore, volatility prediction errors in high-residual regions produce larger parameter updates, encouraging the volatility estimator to better capture local uncertainty.

Calibration interpretation. A probabilistic forecasting model should assign wider prediction intervals to high-uncertainty regions and narrower intervals to stable regions. If volatility is underestimated in high-residual regions, the predictive distribution becomes overconfident and poorly calibrated. By emphasizing volatility errors in these regions, the proposed weighted objective helps improve conditional variance estimation and benefits distributional calibration.

Reverse-distribution training objective. In addition to the volatility-aware objective, FAIP-Diff trains the denoising network using a noise-prediction term together with a variance-matching term. Under the base conditional forward process, the target reverse posterior is written as

$$q_{\mathrm{base}}\left(\mathbf{y}_{k-1} \mid \mathbf{y}_k, \mathbf{y}_0, \mathbf{x}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{q,k}, \tilde{\boldsymbol{\sigma}}_k^2\right), \tag{67}$$

where $\boldsymbol{\mu}_{q,k}$ and $\tilde{\boldsymbol{\sigma}}_k^2$ are the posterior mean and variance derived from the base conditional diffusion process.

The learned reverse transition is

$$p_\theta\left(\mathbf{y}_{k-1} \mid \mathbf{y}_k, \mathbf{x}\right) = \mathcal{N}\left(\boldsymbol{\mu}_{\theta,k}, \hat{\boldsymbol{\sigma}}_k^2\right), \tag{68}$$

where

$$\left(\hat{\boldsymbol{\epsilon}}_k, \hat{\boldsymbol{\sigma}}_k^2\right) = D_\theta\left(\mathbf{y}_k, \mathbf{x}, \boldsymbol{\mu}_x, \boldsymbol{\sigma}_x^2, k\right). \tag{69}$$

Motivated by the Gaussian reverse KL divergence, our implementation employs the following simplified surrogate objective:

$$\mathcal{L}_{\mathrm{KL}} = \mathrm{Mean}\left[\left(\boldsymbol{\epsilon} - \hat{\boldsymbol{\epsilon}}_k\right)^2 + \frac{\tilde{\boldsymbol{\sigma}}_k^2}{\hat{\boldsymbol{\sigma}}_k^2} - \log\left(\frac{\tilde{\boldsymbol{\sigma}}_k^2}{\hat{\boldsymbol{\sigma}}_k^2}\right)\right], \tag{70}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ is the standard Gaussian noise used to construct the adaptively perturbed state. The first term is a simplified noise-prediction surrogate for reverse-mean learning, while the remaining terms match the predicted reverse variance to the target posterior variance. Constant terms and the common scaling factor are omitted.

Conditional mean objective. The conditional mean network provides the initial future estimate $\boldsymbol{\mu}_x$ from the historical observations. It is trained directly against the ground-truth future sequence using

$$\mathcal{L}_{\mathrm{mean}} = \mathrm{Mean}\left[\left(\boldsymbol{\mu}_x - \mathbf{y}_0\right)^2\right]. \tag{71}$$

This objective encourages $\boldsymbol{\mu}_x$ to provide an informative conditional center for the forward perturbation process and the reverse generation process.

Combining the diffusion loss, conditional mean loss, and volatility-aware weighted loss, the complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{mean}} + \mathcal{L}_{\mathrm{var}}. \tag{72}$$

Through this joint objective, FAIP-Diff simultaneously learns the denoising direction, reverse uncertainty, conditional forecasting center, and position-dependent future volatility.

Table 4: Experimental settings for different time-series datasets.

Dataset	Input Length (T)	Prediction Length (O)	Batch Size
ETTh1	168	192	32
ETTh2	168	192	32
ETTm1	168	192	32
ETTm2	168	192	32
Exchange Rate	168	24	32
ILI	168	36	16
Weather	168	24	8
Electricity	168	192	4

B Dataset description and corresponding experimental setup

We evaluate FAIP-Diff on eight widely used multivariate TSF benchmarks, including four ETT subsets, Electricity, Exchange Rate, Weather, and ILI. These datasets cover diverse real-world scenarios such as energy systems, electricity consumption, financial markets, meteorological monitoring, and epidemic dynamics. They also exhibit different degrees of seasonality, non-stationarity, local fluctuation, and abrupt changes, making them suitable for evaluating probabilistic forecasting models under heterogeneous temporal patterns.

The ETT datasets contain electricity transformer temperature records collected from power transformer stations. ETTh1 and ETTh2 are sampled hourly, while ETTm1 and ETTm2 are sampled every 15 minutes. These datasets contain clear periodic patterns and local variations, and are commonly used in long-horizon forecasting studies. The Electricity dataset records the electricity consumption of multiple clients, showing strong daily and weekly periodicity as well as multivariate correlations. The Exchange Rate dataset contains daily exchange rates of several countries, which are highly non-stationary and exhibit strong local fluctuations. The Weather dataset consists of multiple meteorological variables, such as temperature, humidity, pressure, and wind speed, and contains both smooth trends and short-term fluctuations. The ILI dataset records the ratio of patients with influenza-like illness, characterized by seasonal epidemic patterns and abrupt peaks.

Figure 5 presents the boxplots of the eight benchmark datasets after variable-wise standardization. The visualization shows clear differences in distributional spread, skewness, and outlier patterns across datasets. In particular, Exchange Rate and ILI contain more pronounced extreme values and heavy-tailed behaviors, while ETT and Electricity exhibit relatively regular but still heterogeneous distributions. These distributional differences motivate the need for fluctuation-aware uncertainty modeling and structurally regularized diffusion sampling.

As summarized in Table 4, the input window length is fixed at $T = 168$ for all datasets, while the prediction horizon and batch size are configured according to the forecasting characteristics, dimensionality, and memory requirements of each dataset.

C Evaluation Metrics for Probabilistic Forecasting

Since probabilistic forecasting models output predictive distributions or multiple future samples rather than a single deterministic forecast, evaluation should measure both accuracy and uncertainty calibration. In this work, we mainly use the Continuous Ranked Probability Score (CRPS) and Quantile Interval Coverage Error (QICE).

C.1 Continuous Ranked Probability Score

The Continuous Ranked Probability Score (CRPS) is a widely used metric for evaluating probabilistic forecasts. Given a predictive cumulative distribution function F and an observation y, CRPS is defined as

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{+\infty} \left(F(z) - \mathbb{I}\{y \le z\}\right)^2 dz, \tag{73}$$

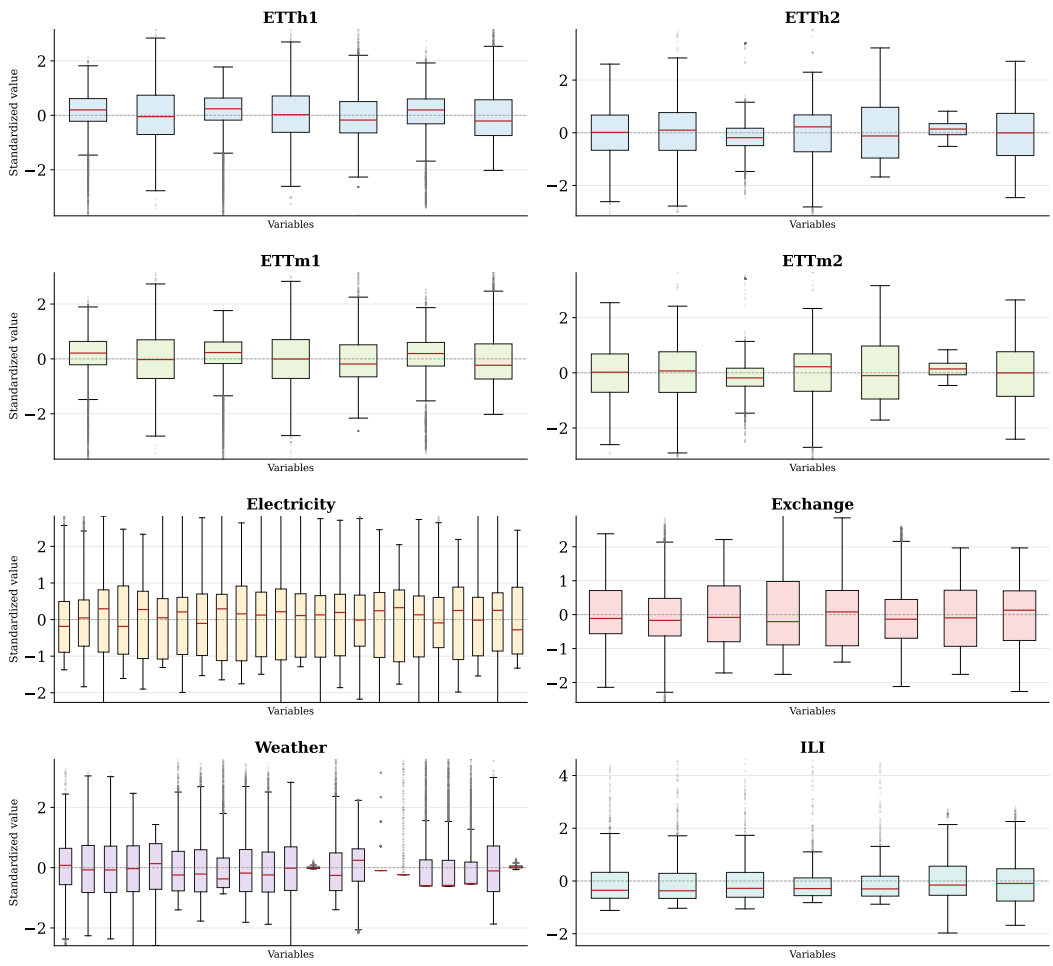


Figure 5: Variable-wise boxplots of the eight multivariate time-series benchmark datasets after standardization. Each subplot corresponds to one dataset, and each box represents the distribution of one variable. The datasets exhibit heterogeneous distributional ranges, outlier patterns, and variable-level volatility, supporting the need for fluctuation-aware probabilistic forecasting.

where $\mathbb{I}\{\cdot\}$ is the indicator function.

CRPS can be interpreted as a generalization of the absolute error to probabilistic forecasts. A better CRPS indicates that the predictive distribution is more concentrated around the true observation while remaining well calibrated. If the predictive distribution degenerates to a deterministic point forecast, CRPS reduces to the mean absolute error.

In sample-based probabilistic forecasting, the predictive distribution is represented by M generated samples:

$$\left\{\hat{\mathbf{y}}^{(1)}, \hat{\mathbf{y}}^{(2)}, \ldots, \hat{\mathbf{y}}^{(M)}\right\}. \tag{74}$$

The empirical CRPS can be computed as

$$\mathrm{CRPS} = \frac{1}{M}\sum_{m=1}^{M}\left|\hat{y}^{(m)} - y\right| - \frac{1}{2M^2}\sum_{m=1}^{M}\sum_{j=1}^{M}\left|\hat{y}^{(m)} - \hat{y}^{(j)}\right|. \tag{75}$$

The first term measures the discrepancy between generated samples and the ground truth, while the second term measures the dispersion among generated samples. Therefore, CRPS jointly evaluates predictive accuracy and distributional sharpness.

For multivariate time series, CRPS is averaged over all prediction steps and variables:

$$\mathrm{CRPS} = \frac{1}{ON} \sum_{o=1}^{O} \sum_{n=1}^{N} \mathrm{CRPS}\left(F_{o,n}, y_{o,n}\right), \tag{76}$$

where O denotes the prediction horizon and N denotes the number of variables.

C.2 Quantile Interval Coverage Error

Quantile Interval Coverage Error (QICE) evaluates the calibration of predictive distributions by comparing the empirical coverage of quantile intervals with the expected coverage.

Given D generated samples for a target value y, we first compute a set of quantile boundaries:

$$q_0, q_1, \ldots, q_D, \tag{77}$$

where D is the number of quantile intervals. These boundaries divide the predictive distribution into M intervals:

$$[q_{d-1}, q_d], \qquad d = 1, \ldots, D. \tag{78}$$

For a well-calibrated predictive distribution, the true observations should fall into each quantile interval with approximately equal probability:

$$\frac{1}{D}. \tag{79}$$

Let c_d denote the empirical fraction of observations that fall into the d-th quantile interval:

$$c_d = \frac{1}{L} \sum_{\ell=1}^{L} \mathbb{I}\left\{ q_{d-1}^{(\ell)} < y^{(\ell)} \leq q_d^{(\ell)} \right\}, \tag{80}$$

where L is the number of evaluated target points. QICE is defined as

$$\mathrm{QICE} = \sum_{d=1}^{D} \left| c_d - \frac{1}{D} \right|. \tag{81}$$

A lower QICE indicates better calibration. If the predictive distribution is perfectly calibrated, each quantile interval contains approximately the same proportion of observations, and QICE approaches zero.

C.3 Interpretation of CRPS and QICE

CRPS and QICE evaluate different aspects of probabilistic forecasts. CRPS jointly measures accuracy and sharpness: it rewards predictive distributions that assign high probability mass near the true observation while penalizing overly dispersed samples. QICE mainly evaluates calibration: it measures whether the empirical coverage of predictive quantile intervals matches the expected coverage.

Therefore, a good probabilistic forecasting model should achieve both low CRPS and low QICE. Low CRPS indicates that generated samples are close to the ground truth, while low QICE indicates that the predictive distribution is well calibrated. In diffusion-based TSF, these metrics are particularly suitable because the model naturally generates multiple future samples through repeated reverse diffusion sampling.

D Algorithms

This section summarizes the three core components of FAIP-Diff in algorithmic form, include adaptive noise remodulation strategy, constrained-space guided sampling, and volatility-aware weighted learning. We use $\mathbf{x}$ to denote the historical input time-series window, $\mathbf{y}_0$ to denote the ground-truth future sequence, and $\mathbf{y}_k$ to denote the noisy future state at diffusion step k. The conditional mean and variance predictions are denoted by $\boldsymbol{\mu}_x$ and $\boldsymbol{\sigma}_x^2$, respectively.

D.1 Adaptive Noise Remodulation Strategy

Algorithm 1 summarizes the proposed adaptive forward noise remodulation strategy. The conditional mean and variance are first estimated from the historical input. The normalized forecasting residual is then used to characterize the prediction difficulty at each future position and adaptively enhance the corresponding forward noise variance.

Algorithm 1 Adaptive Noise Remodulation Strategy

Require: Historical input $\mathbf{x}$, ground-truth future $\mathbf{y}_0$, temporal features, diffusion step k, modulation strength λ, numerical constant ϵ
Ensure: Noisy future state $\mathbf{y}_k$
1: Predict the conditional mean $\boldsymbol{\mu}_x$ from the historical input and temporal features
2: Predict the conditional variance $\boldsymbol{\sigma}_x^2$ from the historical input
3: Compute the absolute residual between $\mathbf{y}_0$ and $\boldsymbol{\mu}_x$
4: Normalize the residual using the conditional standard deviation and obtain the residual-aware anomaly score $\mathbf{s}$
5: Estimate the target local volatility $\boldsymbol{\sigma}_y^2$ from the real historical and future sequence
6: Compute the base forward noise variance $\boldsymbol{\sigma}_{\mathrm{base},k}^2$ using the diffusion schedule, $\boldsymbol{\sigma}_x^2$, and $\boldsymbol{\sigma}_y^2$
7: Remodulate $\boldsymbol{\sigma}_{\mathrm{base},k}^2$ using $\mathbf{s}$ and λ to obtain the adaptive forward noise variance $\boldsymbol{\sigma}_{\mathrm{adapt},k}^2$
8: Sample Gaussian noise $\boldsymbol{\epsilon}_k \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
9: Scale $\boldsymbol{\epsilon}_k$ according to $\boldsymbol{\sigma}_{\mathrm{adapt},k}^2$
10: Construct $\mathbf{y}_k$ using the conditional forward noising process defined in Appendix A.1
11: **return** $\mathbf{y}_k$

D.2 Constrained-Space Guided Sampling

FAIP-Diff constructs a constrained future-pattern space from real future windows in the training set. As summarized in Algorithm 2, the resulting space captures the dominant temporal and cross-variable patterns of the training futures. It is constructed before reverse sampling and remains fixed during validation and testing.

Algorithm 2 Construction of the Training-Derived Constrained Space

Require: Training future windows $\{\mathbf{y}_0^{(i)}\}_{i=1}^{M}$, number of retained principal components R
Ensure: Constrained-space basis $\mathbf{B}$ and future-pattern space $\mathcal{S}$
1: Flatten each training future window along the prediction and variable dimensions
2: Stack the S flattened future windows into a sample matrix
3: Center the sample matrix along the training-sample dimension
4: Apply PCA and retain the leading R principal components
5: Form the orthogonal constrained-space basis $\mathbf{B}$
6: Construct the future-pattern space $\mathcal{S} = \mathrm{span}(\mathbf{B})$
7: **return** $\mathbf{B}, \mathcal{S}$

Algorithm 3 presents the reverse sampling procedure. Unlike prototype-conditioned generation methods that directly feed prototype representations into the denoising network, FAIP-Diff uses $\mathcal{S}$ as an external structural constraint. At each reverse step, the predicted noise is projected and reflected with respect to this space before it is used to calculate the next diffusion state.

Algorithm 3 Constrained-Space Guided Reverse Sampling

Require: Historical input $\mathbf{x}$, temporal features, trained conditional mean and variance networks, denoising network, constrained-space basis $\mathbf{B}$, reflection strength ρ, total number of diffusion steps K
Ensure: Generated future sample $\hat{\mathbf{y}}_0$
Predict the conditional mean $\boldsymbol{\mu}_x$ from the historical input and temporal features
Predict the conditional variance $\boldsymbol{\sigma}_x^2$ from the historical input
Sample terminal Gaussian noise $\mathbf{z}_K \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
Initialize the terminal state $\mathbf{y}_K$ using $\boldsymbol{\mu}_x$, $\boldsymbol{\sigma}_x^2$, and $\mathbf{z}_K$
for $k = K-1, K-2, \ldots, 1$ **do**
    Feed $\mathbf{y}_k$, $\boldsymbol{\mu}_x$, $\boldsymbol{\sigma}_x^2$, and k into the denoising network
    Obtain the predicted noise $\hat{\boldsymbol{\epsilon}}_k$ and reverse variance $\hat{\boldsymbol{\sigma}}_k^2$
    Flatten $\hat{\boldsymbol{\epsilon}}_k$ along the prediction and variable dimensions
    Project the flattened noise onto the constrained space $\mathcal{S}$
    Reflect the predicted noise with respect to $\mathcal{S}$
    Fuse the original and reflected noise using reflection strength ρ
    Reshape the fused result to obtain the corrected noise $\boldsymbol{\epsilon}'_k$
    Calculate the reverse mean using $\mathbf{y}_k$, $\boldsymbol{\mu}_x$, $\boldsymbol{\sigma}_x^2$, and $\boldsymbol{\epsilon}'_k$
    Sample the next state $\mathbf{y}_{k-1}$ using the reverse mean and $\hat{\boldsymbol{\sigma}}_k^2$
end for
Perform the final deterministic denoising step
Set the resulting clean future sequence as $\hat{\mathbf{y}}_0$
return $\hat{\mathbf{y}}_0$

For probabilistic forecasting, Algorithm 3 is repeated M times using independent Gaussian noise realizations. Given the same historical input $\mathbf{x}$, this produces $\{\hat{\mathbf{y}}_0^{(m)}\}_{m=1}^{M}$, which forms an empirical approximation to the conditional predictive distribution.

D.3 Volatility-Aware Weighted Learning

Algorithm 4 summarizes the volatility-aware weighted learning strategy. The residual-based weight $\mathbf{w}$ increases the contribution of difficult temporal positions to conditional variance learning, thereby reducing the dominance of smooth and easy-to-predict observations.

Algorithm 4 Volatility-Aware Weighted Learning

Require: Ground-truth future $\mathbf{y}_0$, conditional mean $\boldsymbol{\mu}_x$, conditional variance $\boldsymbol{\sigma}_x^2$, target local volatility $\boldsymbol{\sigma}_y^2$, clipping threshold c, numerical constant ϵ
Ensure: Volatility-aware loss $\mathcal{L}_{\text{var}}$
Compute the absolute residual between $\mathbf{y}_0$ and $\boldsymbol{\mu}_x$
Normalize the residual and apply clipping to obtain the residual-based weight $\mathbf{w}$
Compute the difference between the predicted conditional standard deviation and the target local standard deviation
Weight the position-wise volatility errors using $\mathbf{w}$
Average the weighted errors over the batch, prediction, and variable dimensions to obtain $\mathcal{L}_{\text{var}}$
return $\mathcal{L}_{\text{var}}$

D.4 Joint Training Procedure

Algorithm 5 summarizes the joint training process of FAIP-Diff. For each batch, one diffusion step is randomly sampled for every training example. Therefore, the complete reverse chain is not executed during training. Instead, the model directly constructs the noisy state associated with the sampled step and learns to predict the corresponding Gaussian noise and reverse variance.

Algorithm 5 Joint Training of FAIP-Diff

Require: Training pairs $(\mathbf{x}, \mathbf{y}_0)$, temporal features, conditional mean network, conditional variance network, denoising network, total number of diffusion steps K
Ensure: Trained FAIP-Diff parameters
for each training batch **do**
  Predict $\boldsymbol{\mu}_x$ from the historical input and temporal features
  Predict $\boldsymbol{\sigma}_x^2$ from the historical input
  Randomly sample diffusion steps k and their symmetric counterparts within the batch
  Apply Algorithm 1 to construct the corresponding noisy future states $\mathbf{y}_k$
  Feed $\mathbf{y}_k$, $\boldsymbol{\mu}_x$, $\boldsymbol{\sigma}_x^2$, and k into the denoising network
  Obtain $\hat{\boldsymbol{\epsilon}}_k$ and $\hat{\boldsymbol{\sigma}}_k^2$
  Calculate the target reverse posterior variance from the forward diffusion process
  Calculate the diffusion loss $\mathcal{L}_{\mathrm{KL}}$
  Calculate the conditional mean loss $\mathcal{L}_{\mathrm{mean}}$
  Apply Algorithm 4 to calculate $\mathcal{L}_{\mathrm{var}}$
  Combine the three objectives and update all trainable parameters
end for
return Trained FAIP-Diff parameters

The complete training objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{KL}} + \mathcal{L}_{\mathrm{mean}} + \mathcal{L}_{\mathrm{var}}. \tag{82}$$

Here, $\mathcal{L}_{\mathrm{KL}}$ jointly trains the noise prediction and reverse-variance estimation of the denoising network, $\mathcal{L}_{\mathrm{mean}}$ trains the conditional mean predictor $\boldsymbol{\mu}_x$, and $\mathcal{L}_{\mathrm{var}}$ trains the conditional variance predictor $\boldsymbol{\sigma}_x^2$ with greater emphasis on high-residual positions. Their detailed definitions and theoretical interpretations are provided in Appendix A.3.

E Hyperparameter and Symbol Description

Table 5: Main notation used in FAIP-Diff.

Symbol	Description
$\mathbf{x}$	Historical input time-series window
$\mathbf{y}_0$	Ground-truth future sequence
$\mathbf{y}_k$	Noisy future state at diffusion step k
$\hat{\mathbf{y}}_0$	Generated future sample
T	Historical window length
O	Prediction horizon
N	Number of variables
k	Current diffusion step
K	Total number of diffusion steps
$\boldsymbol{\mu}_x$	Conditional mean prediction
$\boldsymbol{\sigma}_x^2$	Conditional volatility or variance prediction
$\boldsymbol{\sigma}_y^2$	Target local volatility estimated from real future windows
$\hat{\boldsymbol{\epsilon}}_k$	Noise predicted by the denoising network
$\boldsymbol{\epsilon}'_k$	Corrected noise after constrained-space reflection
$\hat{\boldsymbol{\sigma}}_k^2$	Reverse variance predicted by the denoising network
$\mathbf{s}$	Residual-aware anomaly score
$\mathbf{w}$	Residual-based weight for volatility learning
$\boldsymbol{\sigma}_{\mathrm{base},k}^2$	Base forward noise variance
$\boldsymbol{\sigma}_{\mathrm{adapt},k}^2$	Residual-aware adaptive forward noise variance
$\mathbf{B}$	Training-derived constrained-space basis
$\mathcal{S}$	Constrained future-pattern space, $\mathcal{S} = \mathrm{span}(\mathbf{B})$
M	Number of training future windows used for subspace construction
R	Number of retained principal components
ρ	Reflection strength for constrained-space guided sampling
λ	Residual-aware noise modulation strength
c	Clipping threshold for residual-based volatility weighting

Table 6: Main implementation hyperparameters.

Hyperparameter	Value	Description
Input window length T	168	Length of historical observations
Prediction horizon O	192	Length of future sequence to forecast
Diffusion steps K	20	Number of diffusion denoising steps
β_{start}	0.0001	Initial value of the linear noise schedule
β_{end}	0.01	Final value of the linear noise schedule
Noise schedule	Linear	Diffusion noise schedule type
Batch size	32	Training batch size
Number of samples S	100	Number of generated forecasting samples
Transformer hidden size	512	Hidden dimension of the mean network
Number of attention heads	8	Attention heads in the mean network
Encoder layers	2	Number of encoder layers
Decoder layers	1	Number of decoder layers
Feed-forward dimension	1024	Dimension of Transformer feed-forward layers
Dropout	0.05	Dropout rate
Activation	GELU	Activation function
Rolling length	96	Window length for local volatility estimation
Retained components R	20	Number of PCA components for constrained-space construction
Reflection strength ρ	0.6	Weight of reflected noise in reverse sampling
Noise modulation strength λ	1.0	Strength of residual-aware noise modulation
Residual weight clipping c	10	Upper bound for volatility loss weighting

F Horizon Sensitivity Results

This appendix provides additional results on prediction horizon sensitivity. The goal of this experiment is to examine how FAIP-Diff behaves under different forecast lengths. The input window length is fixed to 168, the number of diffusion steps is set to 20, and 100 samples are generated for probabilistic forecasting. For the ETT and Exchange Rate datasets, we evaluate prediction horizons of 96, 192, 336, and 720. For ILI and Weather, shorter horizons are used due to their different sampling frequencies and data characteristics.

The horizon sensitivity results show that prediction difficulty generally increases as the forecast horizon becomes longer. On ETTh1, ETTm1, ETTm2, and Exchange Rate, CRPS, MAE, and MSE tend to increase from short to longer prediction horizons, indicating that long-range forecasting introduces larger distributional and point-estimation errors. This is consistent with the common observation that longer horizons require the model to maintain temporal structure over a wider future range.

However, the results are not strictly monotonic across all datasets and metrics. As shown in Table 7, ETTh2 obtains worse results at horizon 336 than at horizon 720 on several metrics. This does not contradict the motivation of FAIP-Diff. The reported metrics are averaged over the entire prediction window, and different horizons may cover different mixtures of smooth segments, periodic patterns, local fluctuations, and abrupt changes. Therefore, a longer horizon does not necessarily lead to a larger averaged error on every dataset. Instead, the non-monotonic behavior reflects that forecasting difficulty is unevenly distributed across temporal positions, which is also one of the motivations for the proposed fluctuation-aware modeling strategy.

As shown in Table 8, the results on ILI and Weather further show that prediction difficulty depends strongly on dataset characteristics. ILI exhibits irregular behavior across horizons, likely because its epidemic peaks and abrupt changes appear only in specific temporal regions. Weather shows relatively better performance at short horizons but also varies with the selected forecast length. These observations suggest that horizon sensitivity is affected not only by the prediction length itself, but also by the local temporal structures contained in each forecast window.

Table 7: Prediction horizon sensitivity results on ETT and Exchange Rate datasets.

Horizon	ETTh1				ETTh2				ETTm1			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
96	0.380	0.013	0.498	0.564	0.284	0.018	0.380	0.318	0.324	0.024	0.428	0.412
192	0.407	0.016	0.538	0.614	0.360	0.019	0.492	0.482	0.339	0.022	0.441	0.464
336	0.423	0.019	0.562	0.681	0.433	0.040	0.560	0.668	0.359	0.010	0.482	0.512
720	0.435	0.038	0.599	0.658	0.387	0.023	0.520	0.508	0.393	0.008	0.527	0.652
Horizon	ETTm2				Exchange Rate				Average			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
96	0.243	0.026	0.329	0.227	0.253	0.073	0.299	0.148	0.297	0.031	0.387	0.334
192	0.270	0.032	0.356	0.276	0.316	0.042	0.397	0.306	0.338	0.027	0.445	0.428
336	0.299	0.025	0.383	0.370	0.313	0.047	0.397	0.349	0.365	0.028	0.476	0.516
720	–	–	–	–	0.690	0.087	0.799	1.149	0.476	0.039	0.611	0.742

Table 8: Prediction horizon sensitivity results on ILI and Weather.

ILI

Horizon	CRPS	QICE	MAE	MSE
9	0.496	0.076	0.590	0.669
18	0.314	0.063	0.370	0.237
36	0.878	0.046	1.060	3.307
72	0.623	0.042	0.758	1.923

Weather

Horizon	CRPS	QICE	MAE	MSE
4	0.228	0.047	0.242	0.147
8	0.174	0.044	0.213	0.110
16	0.206	0.057	0.216	0.156
32	–	–	–	–

G Complete Ablation Study

This appendix presents the complete ablation results of FAIP-Diff on all eight benchmark datasets. Overall, the complete model achieves the best metric combinations on 19 out of the 32 dataset, including a tied best CRPS result on ETTm1. Moreover, it improves upon the base model on 30 out of the 32 comparisons. These results demonstrate that the three components are largely complementary and that their joint use leads to more accurate and robust probabilistic forecasts.

As shown in Table 9, the complete model delivers consistent improvements across different benchmarks. For example, it achieves the best results for all four metrics on ETTh2 and obtains the lowest CRPS, MAE, and MSE on both Exchange Rate and ILI, demonstrating the complementary effects of the three components on distributional and point forecasting accuracy. Meanwhile, some individual or pairwise variants perform better on specific metrics, such as QICE on ETTh1 and MAE and MSE on Weather. These results suggest that each component has its own advantages, while their combination provides more robust overall performance across datasets and evaluation criteria.

Taken together, the complete model ranks among the top two methods for CRPS on all eight datasets and achieves particularly strong results for MAE and MSE. The single-component and pairwise variants occasionally perform better on a specific metric, but none of them provides the same level of consistent performance across datasets. This confirms that ANR, VAW, and CSG have complementary effects and supports incorporating all three components into the final FAIP-Diff framework.

H Visualize noise changes

Figure 6 illustrates how the noise adjustment modifies the temporal structure of the predicted noise. The raw prediction $\hat{\epsilon}_k$ contains considerable high-frequency variation, although different variables already exhibit distinct positive or negative patterns. After PCA projection, the dominant variable-dependent and slowly varying temporal structures become clearer, while irregular local fluctuations are suppressed. The reflected component emphasizes the deviations associated with the projection

Table 9: Complete ablation results on eight forecasting benchmarks. Lower values indicate better performance. The best and second-best results are highlighted in bold and underlined, respectively.

Method	ETTh1				ETTh2				ETTm1				ETTm2			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
Base	0.392	1.470	0.523	0.594	0.358	2.074	0.490	0.514	0.346	2.041	0.455	0.488	0.256	2.030	0.352	0.281
+ANR	0.392	**1.112**	0.525	0.560	0.341	2.688	0.460	0.439	0.334	1.603	0.442	0.462	0.254	2.391	0.327	0.254
+CSG	0.392	1.332	0.519	0.579	0.332	1.971	0.451	0.413	0.341	1.883	0.450	0.450	0.255	1.888	0.332	0.246
+VAW	0.381	1.227	0.503	0.568	0.354	2.074	0.475	0.463	0.340	1.808	0.450	0.458	0.255	2.239	0.333	0.256
+ANR+CSG	0.388	1.371	0.520	0.579	0.347	1.784	0.477	0.495	**0.330**	2.243	**0.426**	**0.428**	0.254	**1.667**	0.335	0.247
+ANR+VAW	0.368	1.134	0.493	0.523	0.337	1.481	0.456	0.439	0.334	**1.295**	0.445	0.444	0.251	2.523	0.325	0.243
+VAW+CSG	0.378	1.148	0.495	0.526	0.358	2.311	0.477	0.484	0.339	2.161	0.441	0.450	0.255	2.368	0.335	0.258
FAIP-Diff	**0.367**	1.317	**0.486**	**0.497**	**0.320**	**1.388**	**0.439**	**0.378**	**0.330**	1.794	0.429	0.436	**0.246**	2.151	**0.323**	**0.235**

Method	Exchange Rate				ILI				Weather				ECL			
	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE	CRPS	QICE	MAE	MSE
Base	0.324	5.930	0.412	0.300	0.806	5.598	0.985	2.846	0.227	5.374	0.276	0.223	0.290	**6.685**	0.306	0.209
+ANR	0.303	6.254	0.381	0.263	0.797	5.937	0.961	2.875	0.227	5.372	0.230	0.146	**0.280**	7.412	0.300	0.201
+CSG	0.309	5.512	0.394	0.283	0.792	**4.825**	0.958	2.850	0.215	4.456	0.229	0.140	0.290	7.709	0.303	0.204
+VAW	0.292	4.854	0.372	0.262	0.746	5.413	0.879	2.488	0.214	**3.104**	0.262	0.175	0.290	7.931	0.298	0.200
+ANR+CSG	0.283	4.532	0.369	0.250	0.726	6.016	0.863	2.335	0.212	5.380	0.238	0.151	**0.280**	7.427	0.300	0.201
+ANR+VAW	0.288	4.786	0.360	0.247	0.713	5.556	0.844	2.340	**0.194**	4.659	0.221	0.139	0.286	7.345	0.306	0.203
+VAW+CSG	0.278	**4.206**	0.355	0.235	0.694	5.175	0.855	2.223	0.207	5.077	**0.210**	**0.120**	0.287	7.361	0.309	0.215
FAIP-Diff	**0.264**	4.842	**0.338**	**0.230**	**0.663**	4.841	**0.812**	**1.907**	0.197	3.765	0.238	0.158	0.282	7.694	**0.295**	**0.198**

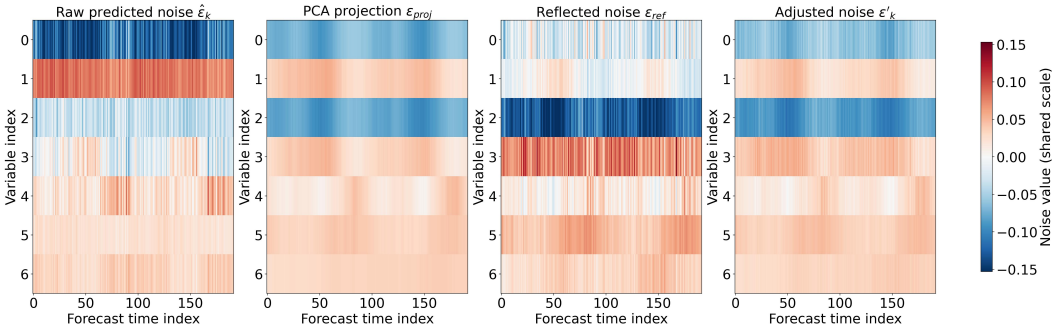
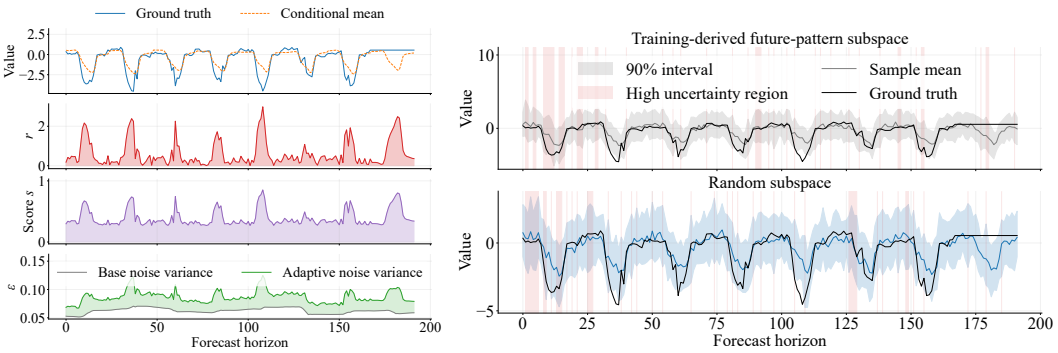


Figure 6: Visualization of the temporal noise adjustment process. From left to right are the raw predicted noise $\hat{\epsilon}_k$, its PCA projection ϵ_{proj}, the reflected component ϵ_{ref}, and the adjusted noise ϵ'_k. Each row corresponds to one variable, and each column represents a forecast time index. All panels use the same color scale.

and therefore displays stronger local variations. By combining these components, the adjusted noise ϵ'_k preserves the principal temporal and cross-variable structure while reducing isolated, rapidly changing perturbations. For example, the consistently negative pattern of variable 2 and the positive patterns of variables 3–6 are retained after adjustment, but their temporal evolution becomes substantially smoother. This visualization suggests that the projection-based adjustment acts as a structured correction rather than a uniform rescaling, allowing the denoising process to maintain meaningful variable-specific dynamics while limiting unstable noise fluctuations.

I Qualitative Visualization

To further illustrate how the proposed components work during forecasting, we provide qualitative visualizations on a representative ETTh1 forecast window. Figure 7a visualizes the relationship among the conditional mean prediction, residual magnitude, anomaly-aware score, and adaptive forward noise variance. First group comparison demonstration several local drops and turning points are not fully captured by the conditional mean, leading to visible residual peaks. The Groups 2 and 3 show that these residual peaks are consistently reflected in the anomaly-aware score. The last group further shows how the anomaly-aware score affects the forward diffusion process. Figure 7b com-



(a) Adaptive noise remodulation.

(b) Training-derived subspace vs. random subspace.

Figure 7: Qualitative visualization on ETTh1. The left figure shows how residual peaks lead to higher anomaly-aware scores and adaptive noise variance. The right figure compares constrained-space guided sampling using the training-derived future-pattern subspace and a random subspace.

Table 10: Window-level analysis of the Adaptive Noise Remodulation Strategy on the ILI dataset. The high-volatility subset contains the 20% of evaluated windows with the largest future-sequence volatility. CRPS wins count the windows in which Base+ANR obtains a lower CRPS than Base. Coverage changes are reported in percentage points.

Subset	Windows	Base CRPS	ANR CRPS	CRPS Impr. (%)	CRPS Wins	Coverage Change
All windows	50	0.6329	0.5851	5.40	37/50	+3.70
High-volatility windows	10	0.8177	0.6829	16.41	10/10	+4.44
Remaining windows	40	0.5867	0.5606	2.65	27/40	+3.51

pares constrained-space guided sampling using the training-derived future-pattern subspace with a random subspace, where the random subspace does not contain meaningful temporal structure. The comparison shows that the training-derived subspace produces a sample mean that is more consistent with the recurrent temporal pattern of the ground truth. In contrast, the random subspace leads to more evident shape or phase deviations in several forecast regions.

J Case Study

J.1 Case Study of Adaptive Noise Remodulation

To examine the behavior of the Adaptive Noise Remodulation Strategy (ANR), we compare basic model with Base+ANR on 50 matched forecasting windows from the ILI dataset. Table 10 summarizes the window-level results for all evaluated windows and for subsets with different volatility levels. Figure 8 further presents a representative high-volatility case at the single-variable level.

Aggregate window-level results. As reported in Table 10, ANR reduces the average CRPS from 0.6329 to 0.5851 across all 50 windows, corresponding to an average relative improvement of 5.40%. CRPS is improved in 37 of the 50 windows. ANR also increases the average coverage of the 90% prediction interval by 3.70 percentage points. This coverage improvement is accompanied by a 13.60% increase in interval width.

The effect is more pronounced for the high-volatility subset. As shown in the second row of Table 10, ANR improves CRPS in all 10 windows in the highest volatility quintile and reduces their average CRPS from 0.8177 to 0.6829, corresponding to an average improvement of 16.41%. By comparison, the remaining 40 windows obtain an average improvement of 2.65%. This comparison suggests that ANR provides larger benefits in windows containing stronger local fluctuations.

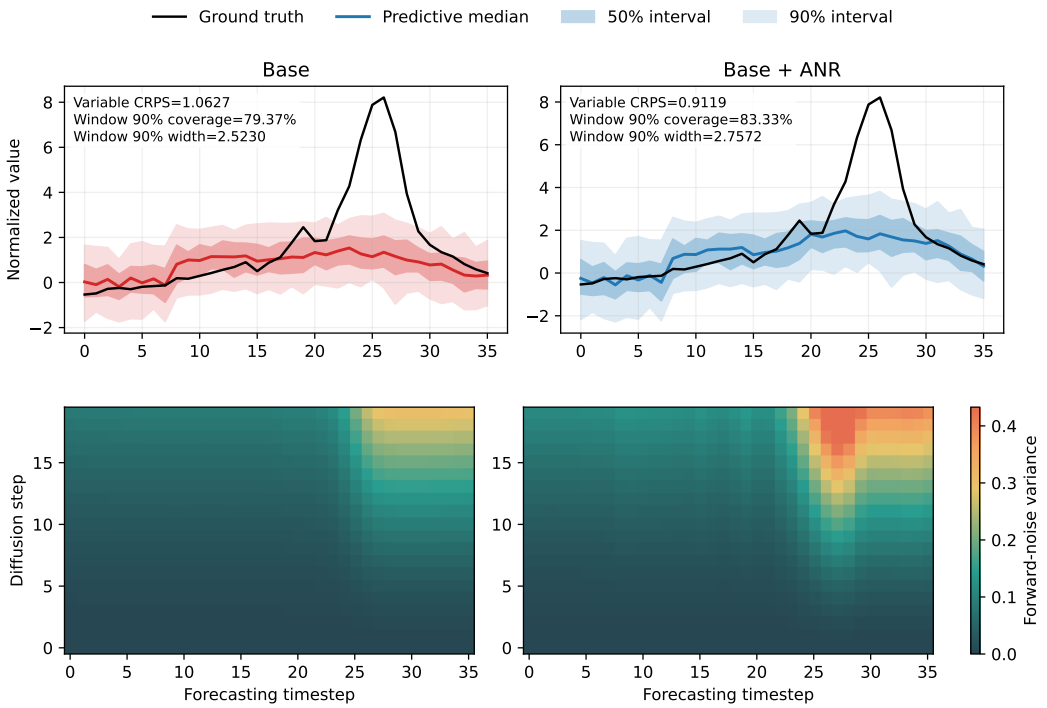


Figure 8: Representative ILI forecasting case for the Adaptive Noise Remodulation Strategy. The selected case is Window 2 and variable 3. The upper-left and upper-right panels show the probabilistic forecasts produced by Base and Base+ANR, respectively. The displayed CRPS values are calculated only for the visualized variable. The lower panels show the reconstructed forward-noise variance over diffusion steps and forecasting positions.

Representative single-variable case. Figure 8 presents Window 2, whose volatility is at the 88th percentile among the evaluated windows. For the displayed variable, CRPS decreases from 1.0627 to 0.9119, corresponding to a 14.19% improvement.

As shown in the upper panels, the basic model predictive median responds only weakly to the rapid increase around forecasting steps 23–29. ANR raises the predictive level around this region and follows the rising segment more closely. The 90% interval coverage increases from 79.37% to 83.33%, while its average width increases from 2.5230 to 2.7572. Thus, the improvement is associated with both an adjustment of the predictive median and a moderate expansion of predictive uncertainty. Nevertheless, neither model fully captures the largest observed peak, indicating that the case remains challenging after noise remodulation.

The lower panels visualize the reconstructed forward-noise variance at different diffusion steps and forecasting positions. Compared with the basic model, ANR assigns larger noise variance to the later diffusion steps, particularly around forecasting steps 23–35. The strongest adjustment occurs near steps 25–29, which coincides with the rapid rise and peak in the target sequence. This pattern is consistent with the intended role of ANR. It increases the noise allocated to locally difficult positions instead of uniformly modifying the complete forecasting horizon.

The aggregate and case-level results suggest that ANR is particularly useful for windows containing strong local fluctuations. It reduces the scale mismatch and forecasting errors in the high-volatility subset while improving prediction-interval coverage. In addition, the moderate widening of the prediction intervals indicates that ANR assigns greater predictive uncertainty to difficult-to-forecast regions. This adaptive allocation provides broader coverage where forecast errors are more likely, thereby producing better-calibrated and more reliable probabilistic forecasts.

Table 11: Multivariate case analysis on five consecutive forecasting windows from the Weather dataset. The scale-shift percentile is calculated relative to the 30 evaluated test windows. CRPS is aggregated across forecast variables within each window. A positive improvement indicates that constrained-space guided sampling obtains a better CRPS than unconstrained sampling.

Window	Scale Shift	Percentile (%)	Base CRPS	Guided CRPS	Improvement (%)
17	3.186	60.0	0.2482	0.2478	0.17
18	3.210	63.3	0.2483	0.2433	2.03
19	3.238	66.7	0.2495	0.2423	2.89
20	3.265	70.0	0.2445	0.2454	−0.38
21	3.294	73.3	0.2434	0.2424	0.39
Average	—	—	0.2468	0.2442	1.03

J.2 Case Study under Local Scale Distribution Shift

To examine the results of constrained-space guided sampling under local distribution shift, we analyze a sequence of test windows from the Weather dataset. We first report the multivariate forecasting results and then visualize the behavior of an individual variable. The two analyses use different levels of aggregation: Table 11 reports CRPS aggregated across variables, whereas Figure 9 presents the predictions for channel 13.

Multivariate analysis. Table 11 reports the results for five consecutive forecasting windows with increasing local scale discrepancy. Constrained-space guidance reduces the average CRPS from 0.2468 to 0.2442, corresponding to an average improvement of 1.03%. Improvements are observed in four of the five windows. In particular, Window 19 shows the largest reduction, with CRPS decreasing from 0.2495 to 0.2423. However, Window 20 exhibits a slight degradation of 0.38%, indicating that the effect of guidance is not uniform across all shifted windows. Overall, the multivariate results suggest that the guidance mechanism can moderately improve predictive distribution alignment in several local scale-shift cases.

Single-variable analysis. Figure 9 visualizes channel 13 in Windows 18, 19, and 21. The scale-shift score increases from 3.210 to 3.294 across these examples. For each row, the left panel shows the unconstrained prediction, while the right panel shows the prediction after constrained-space guidance. The corresponding channel-level CRPS improvements are 0.52%, 0.74%, and 0.37%, respectively.

The constrained predictions retain prediction intervals of a similar scale while producing moderate adjustments to the predictive median and sample distribution. Positive CRPS changes are consistently observed in all three displayed cases, although the improvements remain relatively small. This indicates that the guidance mechanism provides locally stable corrections for the selected variable rather than substantially changing the entire predictive distribution. The channel-level improvements in Figure 9 are not expected to match the multivariate improvements in Table 11, since the figure evaluates only channel 13, whereas the table aggregates errors across all variables. These results provide complementary evidence at the multivariate and individual-variable levels.

K Matched-Seed Significance Analysis

We assess the consistency of FAIP-Diff's improvements over NsDiff using matched random seeds. For each dataset, the two models are trained and evaluated under the same dataset-specific configuration and random seeds. Depending on the dataset, 7–12 paired runs are available. For each dataset and metric, we compare FAIP-Diff with NsDiff using a two-sided Wilcoxon signed-rank test on results paired by random seed.

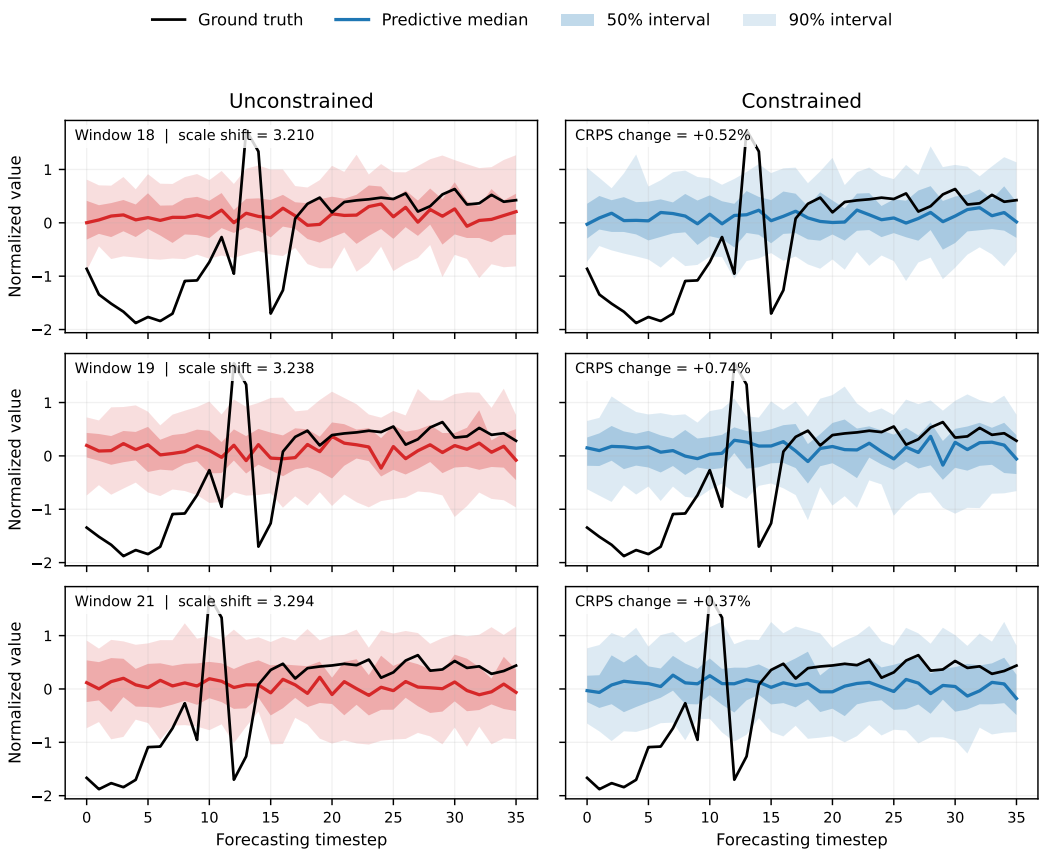


Figure 9: Single-variable forecasting cases under increasing local scale discrepancy on the Weather dataset. The rows correspond to Windows 18, 19, and 21 from top to bottom. The left column shows unconstrained predictions, while the right column shows predictions obtained using constrained-space guidance. The black curve denotes the ground-truth future sequence, and the colored curve denotes the predictive median. The dark and light shaded regions represent the 50% and 90% prediction intervals, respectively. The reported CRPS changes are calculated for channel 13.

For visualization, the relative improvement for dataset d, seed s, and metric m is defined as

$$\Delta_{d,s}^{(m)} = \frac{M_{d,s}^{(m)}(\text{NsDiff}) - M_{d,s}^{(m)}(\text{FAIP} - \text{Diff})}{M_{d,s}^{(m)}(\text{NsDiff})} \times 100\%. \tag{83}$$

All four metrics are minimized; hence, a positive value indicates better performance by FAIP-Diff. The mean and 95% confidence interval shown in Figure 10 are calculated from the seed-level relative improvements, with the interval estimated by paired bootstrap resampling. These confidence intervals visualize variation across seeds and are distinct from the Wilcoxon tests.

Figure 10 shows positive mean improvements for CRPS, QICE, MAE, and MSE on all eight datasets. Only the QICE of Electricity has not improved. The seed-level points also reveal variation in the magnitude of the improvements, particularly on ILI and Weather.

Because we test eight datasets for each metric, we apply the Holm procedure separately within each metric to adjust its eight dataset-wise Wilcoxon p-values. Figure 11 compares the raw and adjusted values. Its horizontal axis is $-\log_{10}(p)$, so points farther to the right correspond to smaller p-values. The dashed line marks $p = 0.05$.

As shown in Figure 11, 31 of the 32 dataset–metric comparisons remain significant after Holm correction. The only exception is QICE on Electricity, for which the raw and adjusted p-values are both 0.054688. This comparison also has a negative mean relative improvement in Figure 10.

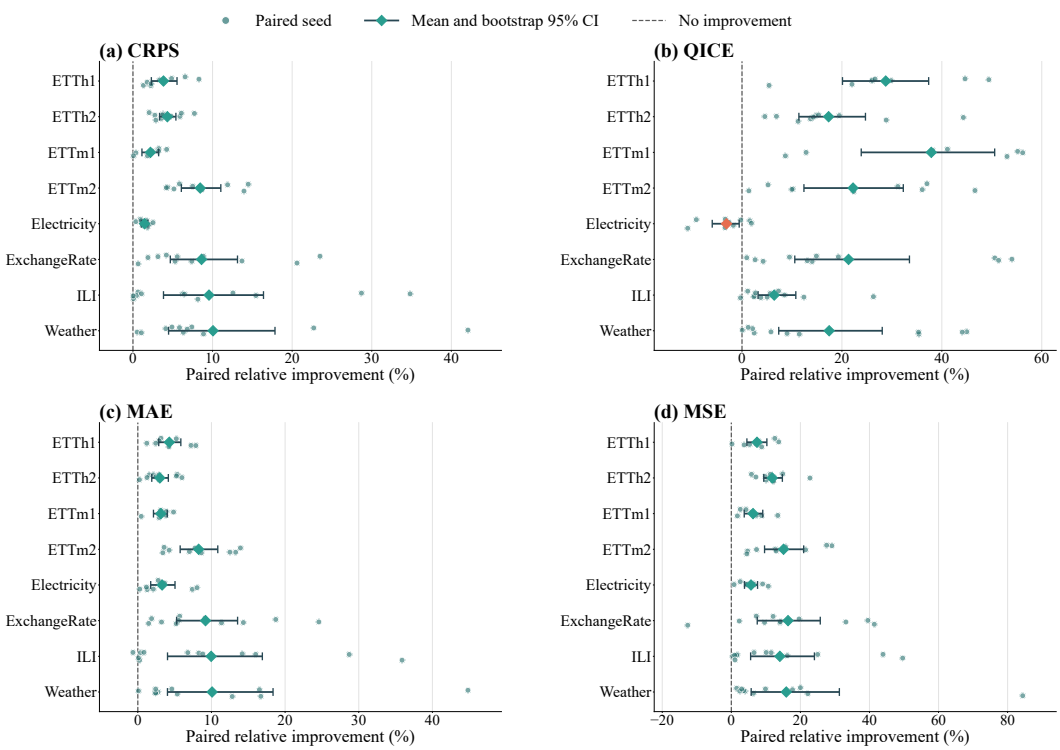


Figure 10: Paired relative improvements of FAIP-Diff over NsDiff across matched random seeds.

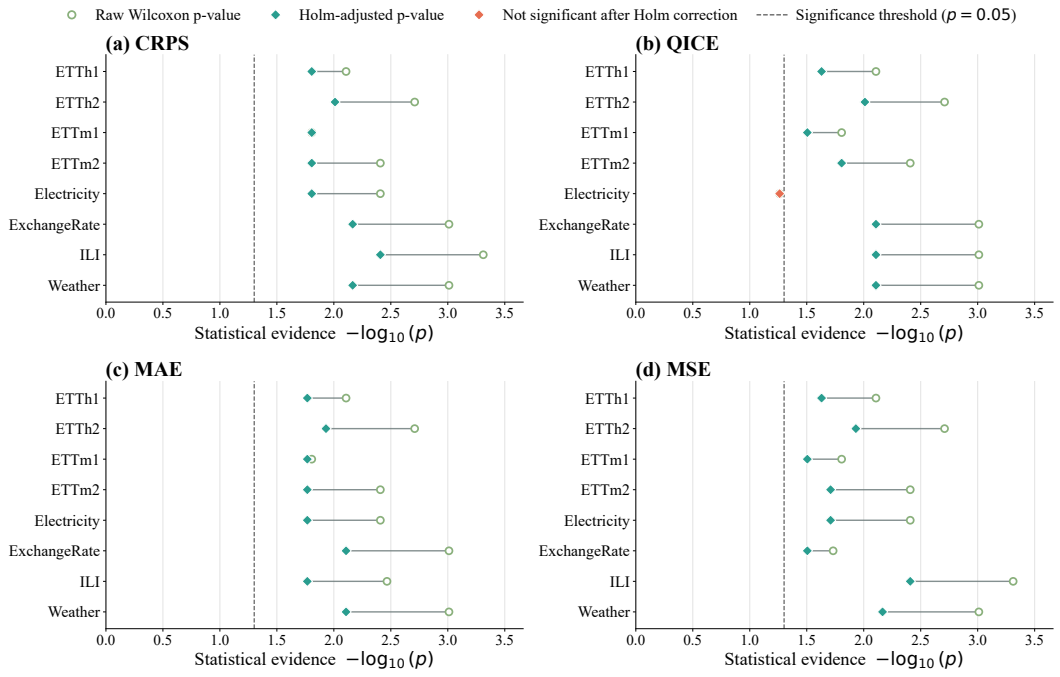


Figure 11: Raw paired Wilcoxon p-values and Holm-adjusted p-values for FAIP-Diff versus NsDiff.

Overall, the matched-seed analysis shows improvements in nearly all metrics across the evaluated datasets, with QICE on the Electricity dataset as the sole exception.