MIXTURE REFERENCE ASSISTED DIFFUSION FOR PROBABILISTIC TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic time series forecasting is essential for simulating future states and supporting decision making. However, real world time series often exhibit temporal heterogeneity, where local statistical patterns shift across time. While diffusion models provide a flexible framework for capturing predictive uncertainty, conventional formulations rely on neural networks to learn these varying local patterns implicitly, complicating the reverse denoising process. To address this limitation, we introduce Mixture Reference Assisted Diffusion (MiraDiff), a framework that decouples conditional denoising into an analytical local statistical reference and a learnable neural correction. Specifically, MiraDiff first fits a Gaussian mixture to training segments to capture recurring local patterns. This frozen reference supplies closed form clean signal estimates across diffusion timesteps, while a neural network conditioned on observed history learns residual corrections to capture long range contexts and cross variable dependencies. Extensive experiments across eight real world datasets demonstrate that MiraDiff delivers consistently competitive probabilistic forecasting performance on both distribution calibration and temporal dependency fidelity. Further ablation studies confirm that introducing closed form local statistics provides an effective statistical inductive bias, substantially alleviating the learning burden on the neural network.

1 INTRODUCTION

Probabilistic time series forecasting aims to characterize possible future trajectories given an observed history. It supports critical operational decisions in power systems and supply chains, including generation scheduling (Pinson, 2013), reserve allocation (Matos & Bessa, 2011), and inventory planning (Salinas et al., 2017). Although deterministic forecasting models have achieved substantial advances in predictive accuracy (Zhou et al., 2021; Nie et al., 2023; Liu et al., 2024), their point predictions alone do not quantify uncertainty, limiting their usefulness when decisions must account for alternative outcomes. In contrast, conditional generative models characterize distributions over future sequences, enabling uncertainty to be represented through an ensemble of plausible trajectories (Rasul et al., 2021; Tashiro et al., 2021; Ye et al., 2025). However, real-world time series often exhibit varying local levels, fluctuation magnitudes, and temporal correlations across different intervals, which is a phenomenon known as temporal heterogeneity (Kim et al., 2022; Song et al., 2021). Under such heterogeneity, the statistical behavior of the future distribution shifts dynamically over time, making it challenging for generative models to capture both diverse local dynamics and global temporal dependencies solely through end-to-end learning.

In recent years, diffusion models have emerged as a flexible framework for probabilistic forecasting (Ho et al., 2020; Song et al., 2021). TimeGrad and CSDI established early foundations by formulating conditional diffusion for temporal data, while TimeDiff advanced this direction via nonautoregressive forecasting (Rasul et al., 2021; Tashiro et al., 2021; Shen & Kwok, 2023). Subsequent works introduced domain specific inductive biases, such as trend and seasonal decomposition in Diffusion-TS (Yuan & Qiao, 2024), conditional mean and variance modeling in TMDM and NsDiff (Li et al., 2024; Ye et al., 2025), and Gaussian process priors within flow matching formulations in TSFlow (Kollovieh et al., 2025). Despite these advances, existing methods mainly rely on parameterized neural networks to infer clean signals during reverse denoising. However, such end-to-end parameterization struggles when temporal heterogeneity is pronounced, where requiring the network to jointly memorize diverse

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.

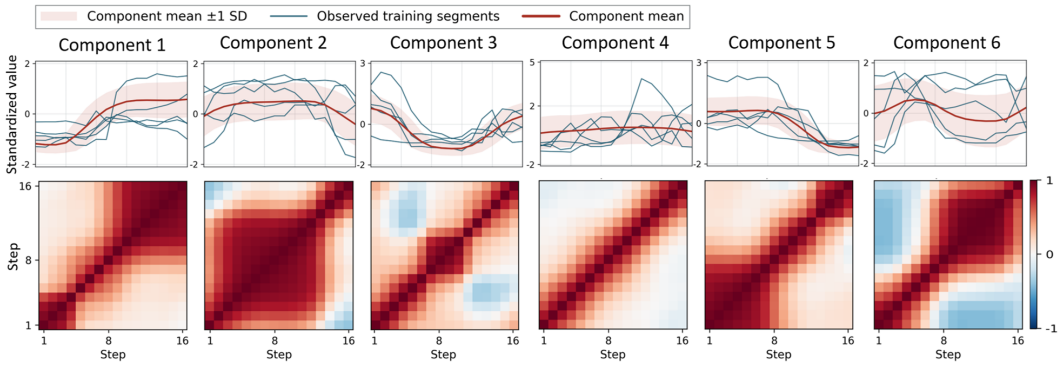


Figure 1: Visualization of learned mixture components on the Electricity dataset. Top row illustrates prototype profiles μ_k, marginal standard deviation bands, and training segments sampled at fixed Mahalanobis distance percentiles (10%, 30%, 50%, 70%, 90%) from each component mean. Bottom row displays the corresponding correlation matrices normalized from Σ_k, demonstrating structured intra segment dependencies across time steps.

local configurations and model complex conditional transitions imposes an excessive learning burden, frequently compromising local temporal fidelity.

Crucially, observed training sequences naturally exhibit recurring local patterns that can be summarized into an explicit statistical prior (Mueen et al., 2009). As illustrated in Fig.1, empirical segments cluster into interpretable prototype shapes with strong intra segment correlations rather than isotropic noise. Owing to the conjugate properties of Gaussian noise, such an empirical reference permits analytic posterior inference to yield a clean local signal estimate during reverse diffusion. This insight motivates a decoupled denoising paradigm where an explicit statistical reference provides a reliable, data derived local baseline, while the neural network is tasked solely with learning residual corrections governed by historical context and broader spatiotemporal dependencies.

To this end, we propose Mixture Reference Assisted Diffusion (**MiraDiff**), a two stage framework that decouples conditional denoising into explicit local statistical inference and conditional neural correction. In the first stage, a Gaussian mixture model is fitted to short training segments to represent recurring local patterns. In the second stage, with the mixture parameters frozen, closed form posterior inference under Gaussian corruption provides an analytic clean signal estimate for each noisy segment. This local estimate is converted into a reference velocity that dynamically adapts to the input without updating reference parameters. A neural network conditioned on historical observations then predicts a residual correction to capture cross patch and cross channel dependencies, effectively reconciling the empirical statistical reference with the target conditional forecasting distribution. The main contributions of this work are summarized as follows:

- We propose a reference assisted diffusion framework that combines an analytic estimate derived from a frozen empirical segment distribution with a history conditioned neural correction, preserving standard Gaussian diffusion mechanics.
- We formulate a Gaussian mixture reference and its corresponding velocity representation, providing a principled decomposition of the reverse denoising function into an explicit statistical reference and a learnable conditional correction.
- Extensive experiments on real-world datasets demonstrate that MiraDiff consistently achieves superior probabilistic forecasting performance, while comprehensive ablation studies confirm the effectiveness of incorporating explicit local statistical inductive biases.

2 Preliminaries

2.1 Problem Statement

Probabilistic Time Series Forecasting. Let $H \in \mathbb{R}^{L \times C}$ denote the observed history and $Y \in \mathbb{R}^{T \times C}$ the future sequence, where $L, T, C \in \mathbb{N}_+$ specify the history length, forecast horizon, and number

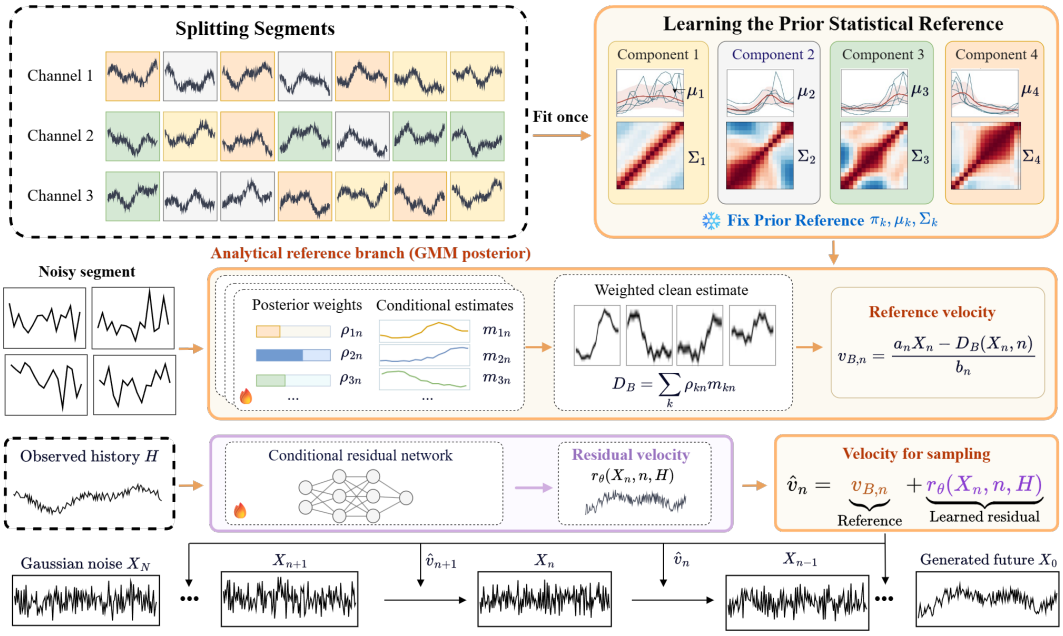


Figure 2: Overview of MiraDiff. A Gaussian mixture model fitted to training segments provides an analytic reference during reverse diffusion, while a neural correction conditioned on the observed history supplies the remaining contextual and cross channel dependencies.

of channels, respectively. Given training pairs of historical and future observations, the objective is to learn a conditional distribution $p_\theta(Y \mid H)$ that approximates the underlying data distribution $p_{\text{data}}(Y \mid H)$. At inference time, the model generates an ensemble of complete future trajectories $Y^{(s)} \sim p_\theta(\cdot \mid H)$ for $s = 1, \ldots, S$, where θ denotes the learnable model parameters and $S \in \mathbb{N}_+$ is the number of generated samples. Each $Y^{(s)} \in \mathbb{R}^{T \times C}$ represents a possible future evolution, and the ensemble characterizes predictive uncertainty over the forecast horizon.

2.2 Gaussian diffusion and velocity prediction

Let $N \in \mathbb{N}_+$ be the number of diffusion steps and let $X_n \in \mathbb{R}^{T \times C}$ denote the noisy future at step $n \in \{1, \ldots, N\}$ under Gaussian diffusion (Ho et al., 2020; Rasul et al., 2021). Under Gaussian corruption, $X_n = a_n Y + b_n \epsilon$, where vector $\epsilon \sim \mathcal{N}(0, I_{TC})$, $a_n, b_n > 0$, and $a_n^2 + b_n^2 = 1$. The coefficients are defined in Appendix A.1, and the experimental schedule is specified in Appendix B. The velocity target $v_n^* = a_n \epsilon - b_n Y$ gives the reconstruction identity (Salimans & Ho, 2022)

$$Y = a_n X_n - b_n v_n^*. \tag{1}$$

This parameterization allows an analytic clean-signal estimate and a neural correction to be combined through a common velocity representation. Section 3 elaborates on this decomposition.

3 Model

In this section, we introduce MiraDiff, a probabilistic forecasting framework that combines local statistical inference with conditional neural denoising. We first present the statistical mixture reference and derive its closed form clean signal estimate under Gaussian corruption. We then describe how this analytic estimate is incorporated into velocity prediction and reverse diffusion sampling. Finally, we outline the two stage training procedure for empirical reference fitting and neural network optimization.

3.1 LOCAL MIXTURE REFERENCE AND POSTERIOR INFERENCE

To characterize recurring local temporal dynamics, we represent short subsequences from each channel as ordered vectors. Let $z \in \mathbb{R}^P$ denote such a clean segment, where $P \in \{1, \ldots, T\}$ is the segment length and each coordinate corresponds to a consecutive temporal position. This formulation allows the reference distribution to capture both the characteristic profile of a local trajectory and the temporal dependence among adjacent observations. Clean segments are extracted and pooled across temporal intervals and channels from the training sequences to learn a shared representation of recurring patterns. Specifically, we fit a Gaussian mixture model with $K \in \mathbb{N}_+$ components parameterized by $\varphi = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^K$, where $\pi_k > 0$ represents the mixture weight with $\sum_{k=1}^K \pi_k = 1$, $\mu_k \in \mathbb{R}^P$ is the component mean, and $\Sigma_k \in \mathbb{R}^{P \times P}$ is a symmetric positive definite covariance matrix. The resulting reference density is given by

$$q_0(z; \varphi) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(z; \mu_k, \Sigma_k), \tag{2}$$

where each mean vector μ_k describes an empirical prototype profile, while the covariance matrix Σ_k accounts for intra segment temporal correlations and local variations. As illustrated in Fig. 1, the mixture components fitted on the Electricity dataset capture distinct local regimes, such as rising trends, stable plateaus, and turning points. Furthermore, the correlation heatmaps exhibit pronounced off diagonal structures rather than diagonal white noise, validating that adjacent temporal steps within a segment are intrinsically coupled. Additional fitted reference visualizations across datasets are provided in Appendix E. Although the mixture parameters φ remain frozen after the initial fitting stage, the posterior assignment across components dynamically adapts to each noisy input and its corresponding noise level, allowing the reference to produce input dependent local estimates throughout diffusion.

To compute this adaptive posterior at arbitrary diffusion step n, we project the fitted prior onto the noisy observation space. Under forward diffusion, the conditional transition distribution of a noisy segment x given clean segment z is $\mathcal{N}(x; a_n z, b_n^2 I_P)$. Conditioned on component k with prior density $\mathcal{N}(z; \mu_k, \Sigma_k)$, marginalizing over the clean segment yields the noisy component distribution in closed form via Gaussian convolution,

$$p_n(x \mid k) = \int \mathcal{N}(x; a_n z, b_n^2 I_P) \, \mathcal{N}(z; \mu_k, \Sigma_k) \, dz = \mathcal{N}(x; a_n \mu_k, V_{k,n}), \tag{3}$$

where the noisy covariance matrix is given by

$$V_{k,n} = a_n^2 \Sigma_k + b_n^2 I_P. \tag{4}$$

Summing over all components with their prior weights π_k yields the noisy marginal density $p_n(x) = \sum_{k=1}^K \pi_k \, \mathcal{N}(x; a_n \mu_k, V_{k,n})$, under which reconstructing the clean segment reduces to evaluating the conditional expectation $\mathbb{E}[z \mid x]$ at diffusion step n. Specifically, we obtain the posterior responsibility $\rho_{k,n}(x)$ by updating the empirical prior weight π_k with the noisy component likelihood $\mathcal{N}(x; a_n \mu_k, V_{k,n})$ via Bayes' rule. Conditioned on component k, the clean segment z and noisy observation x follow a joint Gaussian distribution, yielding the analytical conditional mean $m_{k,n}(x)$ through standard Gaussian conditioning. These two closed form quantities are given by

$$\rho_{k,n}(x) = \frac{\pi_k \mathcal{N}(x; a_n \mu_k, V_{k,n})}{\sum_{\ell=1}^K \pi_\ell \mathcal{N}(x; a_n \mu_\ell, V_{\ell,n})}, \qquad m_{k,n}(x) = \mu_k + a_n \Sigma_k V_{k,n}^{-1}(x - a_n \mu_k). \tag{5}$$

The posterior weights satisfy $\sum_{k=1}^K \rho_{k,n}(x) = 1$, dynamically reweighting the components based on how well their diffused profiles match the observed segment. Within each component, the projection term $a_n \Sigma_k V_{k,n}^{-1}$ acts as an adaptive filter gain that updates the prototype profile μ_k via the observed residual $x - a_n \mu_k$. Crucially, the off diagonal entries of Σ_k capture intra segment dependencies, propagating information across adjacent time steps and explicitly injecting structured continuity into the local reconstruction. Finally, aggregating these component predictions over the posterior responsibilities yields the analytical reference denoiser $D_B(x, n) \in \mathbb{R}^P$, expressed as

$$D_B(x, n) = \sum_{k=1}^{K} \rho_{k,n}(x) \, m_{k,n}(x). \tag{6}$$

Eq. (6) translates the empirical segment prior into an analytic clean signal estimate for any arbitrary diffusion step. By employing soft mixture weighting, this estimator handles ambiguity among multiple recurring motifs without prematurely committing to a single prototype. Under the reference distribution, $D_B(x, n)$ represents the exact minimum mean squared error estimator of the clean segment. Applying this operator across all temporal segments and channels provides an analytical baseline that preserves local temporal dynamics, effectively reducing the optimization burden on the subsequent neural denoising network.

3.2 Reference assisted conditional diffusion

In Section 3.1, the posterior operator D_B is formulated for an individual univariate segment of length P. To extend this estimate across the full multivariate forecasting horizon $X_n \in \mathbb{R}^{T\times C}$, we evaluate D_B independently on each nonoverlapping temporal segment within every channel and assemble the outputs in their corresponding coordinates, denoted by $\mathcal{D}_B(X_n, n) \in \mathbb{R}^{T\times C}$. Substituting the local reconstruction $\mathcal{D}_B(X_n, n)$ in place of the unknown clean future Y in the velocity definition of Eq. 1 yields the analytical reference velocity $v_B(X_n, n) \in \mathbb{R}^{T\times C}$, given by

$$v_B(X_n, n) = \frac{a_n X_n - \mathcal{D}_B(X_n, n)}{b_n}. \tag{7}$$

Intuitively, $v_B(X_n, n)$ provides an empirical estimate of the ideal velocity target v^* by relying solely on local intra segment statistics. This reference velocity $v_B(X_n, n)$ is evaluated deterministically during both training and reverse sampling. Because $\mathcal{D}_B$ operates independently on individual temporal segments and channels, it inherently omits historical context, cross segment continuity, and cross channel correlations. To compensate for these missing global dependencies, we introduce a parameterized neural network $r_\theta(X_n, n, H) \in \mathbb{R}^{T\times C}$ taking the historical context H, the full noisy sequence X_n, and the diffusion step n as inputs. Operating in the velocity space provides a fundamental advantage, since velocity vector fields admit natural linear superposition. This property allows the complex denoising dynamics to be decoupled additively into an analytical local drift and a neural global correction. Therefore the total predicted velocity can be formulated as

$$\widehat{v}_\theta(X_n, n, H) = v_B(X_n, n) + r_\theta(X_n, n, H). \tag{8}$$

Equivalently, substituting Eq. 8 into the reconstruction identity in Eq. 1 reveals that the implied clean sequence decomposes into

$$\widehat{Y}_\theta(X_n, n, H) = \mathcal{D}_B(X_n, n) - b_n r_\theta(X_n, n, H). \tag{9}$$

Eq. 9 reveals that the predicted trajectory decouples into an explicit prior $\mathcal{D}_B$ and a neural residual r_θ. Formulating this decomposition in the velocity space provides a distinct optimization advantage over directly predicting clean signals. Directly estimating clean targets across different diffusion steps suffers from severe variance in signal magnitude. In contrast, the additive velocity formulation naturally introduces the noise coefficient b_n to scale the neural residual, dynamically enabling substantial global corrections under high noise levels while suppressing network fluctuations as the noise level diminishes toward zero.

Theoretical interpretation and reverse sampling. Under a squared velocity loss, the population optimal neural correction satisfies $r^*(I) = \mathbb{E}[v_n^* \mid I] - v_B(X_n, n)$, where $I = (X_n, n, H)$. Incorporating v_B strictly preserves the Bayes optimal denoiser while strategically reallocating the learning budget of a capacity constrained network. Instead of fitting recurring local temporal patterns, the network focuses its representational bandwidth on cross segment transitions, cross channel correlations, and historical conditioning. Through the lens of score matching (Vincent, 2011), this optimal correction satisfies

$$r^*(X_n, n, H) = -\frac{b_n}{a_n}\nabla_{X_n} \log \frac{p_n(X_n \mid H)}{Q_n(X_n)}, \tag{10}$$

where $p_n(X_n \mid H)$ is the conditional noisy density and $Q_n(X_n)$ is the marginal density induced by the local reference. Rather than fitting the entire conditional score from scratch, the network learns the log density ratio gradient relative to the analytical reference anchor Q_n.

Algorithm 1 Training

Input: Training pairs $\mathcal{D}_{\mathrm{tr}}$, segments $\mathcal{Z}_{\mathrm{tr}}$, schedule, budget U

Fit φ on $\mathcal{Z}_{\mathrm{tr}}$ via regularized EM and freeze it.
Initialize network θ with a zero final layer.
for $u = 1, \ldots, U$ **do**
  Sample batch (H, Y), step n, and noise ϵ.
  Form $X_n = a_n Y + b_n \epsilon$ and $v_n^* = a_n \epsilon - b_n Y$.
  Partition X_n into length-P segments.
  Compute $\widehat{Y}_B \leftarrow \mathcal{D}_B(X_n, n; \varphi)$.
  Set $v_B \leftarrow (a_n X_n - \widehat{Y}_B)/b_n$.
  Evaluate $r \leftarrow r_\theta(X_n, n, H)$.
  Set $\widehat{v}_\theta \leftarrow v_B + r$.
  Update θ using $\|\widehat{v}_\theta - v_n^*\|_F^2$.
end for
return (φ, θ).

Algorithm 2 Inference

Input: Context H, reference φ, network θ, schedule $\{a_n, b_n, c_n, d_n, \widetilde{\beta}_n\}_{n=1}^N$

Initialize $X_N \sim \mathcal{N}(0, I_{TC})$.
for $n = N, \ldots, 1$ **do**
  Compute $\widehat{Y}_B \leftarrow \mathcal{D}_B(X_n, n; \varphi)$.
  Evaluate $r \leftarrow r_\theta(X_n, n, H)$.
  Form $\widehat{Y}_\theta \leftarrow \widehat{Y}_B - b_n r$.
  if $n > 1$ **then**
    Sample $\xi_n \sim \mathcal{N}(0, I_{TC})$.
    Update $X_{n-1} \leftarrow c_n \widehat{Y}_\theta + d_n X_n + \sqrt{\widetilde{\beta}_n}\, \xi_n$.
  else
    Set $X_0 \leftarrow \widehat{Y}_\theta$.
  end if
end for
return X_0.

This additive formulation naturally subsumes standard diffusion as an uninformative reference special case. During reverse sampling, the total predicted velocity $\widehat{v}_\theta$ is converted into the reverse trajectory mean using standard Gaussian diffusion schedules, leaving the sampling pipeline seamlessly intact with explicit updates provided in Appendix A.

3.3 Training procedure

The model is optimized through a two stage training procedure. In the first stage, the local statistical reference is fitted to a collection of univariate training segments $\mathcal{Z}_{\mathrm{tr}} = \{z_i\}_{i=1}^M$, where M denotes the total number of extracted segments, by maximizing the average log likelihood,

$$\ell_{\mathrm{ref}}(\varphi) = \frac{1}{M} \sum_{i=1}^{M} \log q_0(z_i; \varphi), \tag{11}$$

where the reference parameters φ are estimated via expectation maximization with covariance regularization and subsequently frozen throughout downstream learning (Dempster et al., 1977). Detailed configurations regarding segment extraction and covariance shrinkage are provided in Appendix B. In the second stage, with the analytical reference fixed, the neural network parameters θ are trained by minimizing the expected mean squared velocity error,

$$\mathcal{L}(\theta) = \mathbb{E}_{H,Y,n,\epsilon}\left[\frac{\|\widehat{v}_\theta(X_n, n, H) - v_n^*\|_F^2}{TC}\right], \tag{12}$$

where the expectation is taken over historical contexts and future ground truths (H, Y) from the training distribution, uniformly sampled diffusion steps $n \sim \mathcal{U}\{1, \ldots, N\}$, and standard Gaussian noise $\epsilon \sim \mathcal{N}(0, I_{TC})$, with $\|\cdot\|_F$ denoting the Frobenius norm. Because $\widehat{v}_\theta = v_B + r_\theta$, minimizing Eq. 12 is equivalent to training the neural correction network r_θ to fit the residual target $v_n^* - v_B(X_n, n)$. This decoupled design isolates the lightweight statistical modeling from neural optimization, introducing inductive guidance without changing the neural backbone architecture.

4 Experiments

In this section, we empirically evaluate MiraDiff on real world multivariate time series datasets. Comprehensive experiments regarding probabilistic forecasting performance, temporal dependency fidelity, and the impact of prior configurations are conducted in the following subsections.

Table 1: Performance comparisons on eight real world datasets in terms of QICE and CRPS. Lower values are better. The best and second best results are boldfaced and underlined, respectively.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Weather		ECL		Traffic		Solar	
Method	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS	QICE	CRPS
TimeGrad	3.462	0.542	6.522	0.972	8.045	0.658	2.657	0.475	8.058	0.838	4.324	0.331	3.341	0.304	8.667	0.401
CSDI	3.924	0.480	6.167	0.822	4.162	0.552	9.972	0.799	7.237	0.364	6.875	0.636	6.569	0.808	8.116	0.424
TimeDiff	14.936	0.470	14.324	0.411	14.823	0.473	13.413	0.316	14.584	0.353	15.534	0.751	15.460	0.779	14.947	0.697
DiffusionTS	3.828	0.617	8.398	1.130	6.293	0.540	11.105	0.986	7.918	0.400	5.581	0.670	2.898	0.757	7.378	0.503
TMDM	2.350	0.436	5.602	0.478	2.139	0.332	1.968	0.279	4.952	0.289	10.580	0.455	9.435	0.585	9.111	0.354
NsDiff	1.820	0.413	**3.045**	0.395	2.406	0.362	**1.682**	0.290	6.800	0.326	8.166	0.342	4.016	0.375	5.580	0.302
D^3VAE	3.411	0.591	6.738	1.106	6.804	0.463	5.226	0.719	9.192	0.439	7.530	0.401	7.836	0.456	8.043	0.415
SSSD	5.904	0.522	10.472	1.038	5.972	0.466	11.046	0.954	8.323	0.396	9.932	0.929	9.075	1.059	10.050	0.688
TSFlow	**1.208**	0.418	4.092	0.455	2.120	0.382	2.363	0.270	3.428	0.288	4.017	0.493	**1.525**	0.556	2.622	0.250
MG-TSD	6.030	0.658	6.863	0.636	3.065	0.523	6.808	0.975	8.167	0.523	4.202	0.278	3.663	0.280	7.302	0.265
MiraDiff (ours)	1.517	**0.360**	3.401	**0.351**	**1.367**	**0.321**	1.812	**0.267**	**2.680**	**0.220**	**3.532**	**0.251**	2.838	**0.264**	**1.065**	**0.214**

Datasets. We evaluate all models on eight widely adopted multivariate time series datasets spanning energy, weather, and transportation domains. These include four electricity transformer temperature benchmarks, ETTh1, ETTh2, ETTm1, and ETTm2 (Zhou et al., 2021), as well as Weather (Wu et al., 2021), Solar (Lai et al., 2018), Electricity, and Traffic. Comprehensive dataset statistics, chronological train validation test splits, preprocessing protocols, and evaluation horizons are detailed in Appendix B.

Baselines. We compare MiraDiff against ten representative probabilistic forecasting and time series generative frameworks. These encompass the autoregressive diffusion model TimeGrad (Rasul et al., 2021), the conditional score based model CSDI (Tashiro et al., 2021), the non autoregressive diffusion model TimeDiff (Shen & Kwok, 2023), the interpretable diffusion framework Diffusion-TS (Yuan & Qiao, 2024), the transformer modulated diffusion model TMDM (Li et al., 2024), the nonstationary diffusion model NsDiff (Ye et al., 2025), the variational generative model D3VAE (Li et al., 2022), the structured state space diffusion model SSSD (Lopez Alcaraz & Strodthoff, 2023), the continuous flow matching model TSFlow (Kollovieh et al., 2025), and the multi granularity diffusion framework MG-TSD (Fan et al., 2024). Implementation and evaluation details for MiraDiff are provided in Appendix B.

4.1 Forecasting Performance and Trajectory Fidelity

Evaluation metrics. Probabilistic forecasting quality is assessed via two complementary metrics, namely the continuous ranked probability score (CRPS) and the quantile interval coverage error (QICE). Specifically, CRPS evaluates predictive accuracy and distributional sharpness across the entire forecast horizon, whereas QICE measures marginal calibration by quantifying deviations from nominal coverage levels (Gneiting et al., 2007). Lower values reflect superior forecasting precision and sharper calibration across both metrics. Formal mathematical formulations for both criteria are provided in Appendix B.

Forecasting performance. Table 1 reports the MiraDiff consistently achieves the lowest CRPS across all evaluated datasets, demonstrating superior predictive accuracy and tight predictive intervals. Compared with TimeDiff and TSFlow, its CRPS gains are particularly clear on Traffic and Solar, where the forecasts must capture pronounced local fluctuations. Together, these results show that MiraDiff maintains strong predictive performance across datasets with different temporal characteristics, while also achieving competitive calibration.

Trajectory fidelity. Fig. 3 compares the forecast trajectories of MiraDiff with TSFlow, TMDM, and SSSD across ETTh1, Electricity, and Traffic. On ETTh1, TSFlow and TMDM attenuate the periodic fluctuations or miss deep troughs, whereas MiraDiff more closely follows their amplitude. On the bursty Electricity and Traffic datasets, TMDM and SSSD produce overly wide intervals, and TSFlow fails to follow transient spikes. MiraDiff keeps compact intervals that stay close to the ground truth over the whole horizon. These results show that MiraDiff preserves fine temporal structure without inflating uncertainty. Additional trajectory comparisons across eight datasets and more baseline methods are provided in Appendix C.

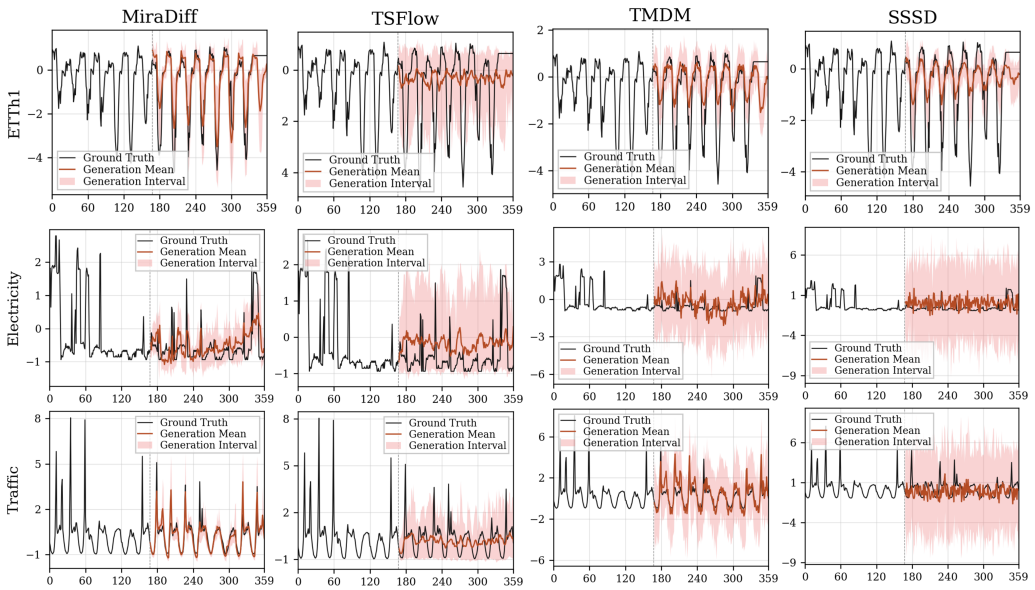


Figure 3: Probabilistic forecast trajectories on ETTh1, Electricity, and Traffic. The black curve shows the observed series, the orange curve shows the mean of generated forecasts, and the shaded region shows their interval. The vertical dashed line separates the observed history from the forecast horizon.

4.2 Distributional Alignment and Ablation Studies

Distributional alignment and segment fidelity. We first evaluate whether generative models reproduce empirical data distributions within localized time series subsequences. Specifically, each 16 step temporal segment is treated as an ordered continuous vector. Observed segments and their model generated counterparts are projected onto a shared two dimensional space using t-SNE (van der Maaten & Hinton, 2008) to inspect sample coverage and cluster structure. Meanwhile, to verify whether generated values preserve the true distribution shape, especially around mode peaks and extreme tails, we aggregate normalized observations across segments and fit continuous probability curves via kernel density estimation (KDE). To quantify distributional discrepancies without reliance on spatial projections or bandwidth choices, we further compute the empirical Wasserstein 1 distance,

$$W_1(F_{\text{real}}, F_{\text{gen}}) = \int_{\mathbb{R}} |F_{\text{real}}(z) - F_{\text{gen}}(z)| \, \mathrm{d}z, \tag{13}$$

where F_{real} and F_{gen} denote the empirical cumulative distribution functions of ground truth and synthesized trajectories, respectively. As shown in Fig. 4, MiraDiff aligns with ground truth segments with markedly higher accuracy than baseline models. In the low dimensional projection, segments synthesized by MiraDiff adhere closely to empirical cluster boundaries, whereas competing baselines exhibit severe sample dispersion. This behavior is matched by density estimations, where MiraDiff accurately traces both peak concentrations and tail decay patterns. Additional dataset and baseline comparisons are provided in Appendix D.

Table 2: Ablation of hierarchical reference prior configurations evaluated by CRPS and Temporal Variogram Score (VS) across eight datasets.

Reference	ETTh1		ETTh2		ETTm1		ETTm2		Weather		ECL		Traffic		Solar	
	CRPS	VS	CRPS	VS	CRPS	VS	CRPS	VS	CRPS	VS	CRPS	VS	CRPS	VS	CRPS	VS
Standard	0.501	0.106	0.480	0.082	0.362	0.073	0.304	0.057	0.314	0.064	0.282	0.052	0.279	0.087	0.288	0.054
Diagonal	0.518	0.112	0.547	0.090	0.379	0.077	0.298	0.055	0.326	0.061	0.295	0.059	0.271	0.085	0.362	0.058
Single	0.404	0.088	0.435	0.075	0.335	0.065	**0.267**	0.050	0.268	0.058	0.261	0.048	0.312	0.091	0.246	0.047
Mixture	**0.360**	**0.079**	**0.351**	**0.068**	**0.321**	**0.058**	0.286	**0.047**	**0.220**	**0.051**	**0.251**	**0.044**	**0.264**	**0.082**	**0.214**	**0.042**

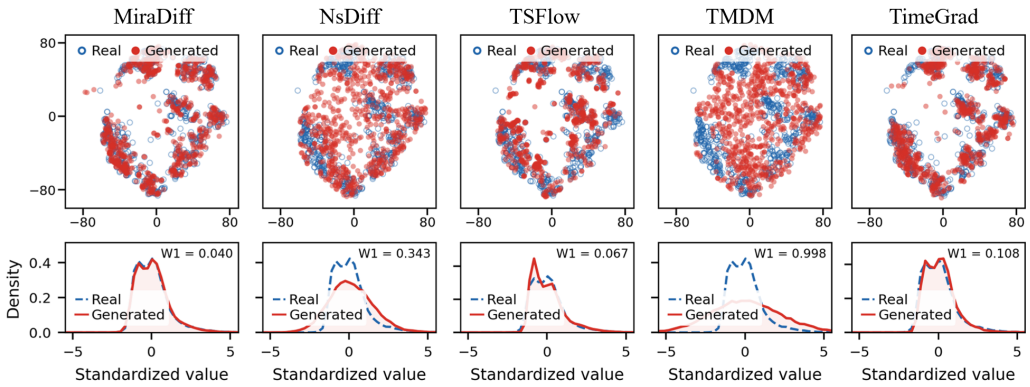


Figure 4: Distributional comparison of observed and generated 16 step temporal segments on Traffic dataset. The upper panel projects ordered segment vectors onto a shared two dimensional latent plane via t-SNE, and the lower panel displays pooled marginal densities alongside empirical W_1 distances.

Hierarchical ablation studies. To systematically dissect the source of these gains, we compare four hierarchical reference configurations across all datasets under both CRPS and Temporal Variogram Score (VS) (Scheuerer & Hamill, 2015), where lower values indicate better performance and VS measures the temporal dependency structure with formal definitions detailed in Appendix.B. **Standard** employs no analytical reference prior, requiring the neural network to predict the entire velocity field from scratch. **Diagonal** incorporates empirical means and position wise marginal variances with all off diagonal covariance elements set to zero, isolating the effect of static marginal statistics. **Single** restricts the reference prior to a single unimodal Gaussian distribution with $K = 1$, retaining intra segment covariance without any mixture modeling. **Mixture** represents our proposed formulation, which utilizes a Gaussian mixture model with multiple components ($K > 1$) to capture multimodal segment patterns. All configurations maintain identical diffusion schedules and network architectures, isolating the empirical benefits of the analytical reference prior.

As reported in Table 2, simply matching marginal statistics via Diagonal setting yields negligible gains or even degrades performance across both metrics, demonstrating that static marginal calibration without temporal covariance disrupts sequential continuity. In contrast, incorporating intra segment covariance in the Single setting leads to substantial improvements across most datasets. This confirms that modeling correlation structures across positions is essential for capturing temporal dependencies. Scaling the analytical prior to our proposed Mixture reference achieves the lowest VS across all eight datasets and the best CRPS in most cases, notably reducing CRPS to 0.220 on Weather and 0.351 on ETTh2.

5 Conclusion

In this paper, we present MiraDiff, a probabilistic time series forecasting framework that combines an empirical local reference prior with conditional diffusion. A Gaussian mixture model fitted on clean training segments provides an analytical signal estimate at each reverse step, while a neural residual network incorporates the observed history and temporal dependencies beyond individual segments. This formulation introduces structured local statistics without altering the standard forward process. Extensive experiments across eight real world datasets demonstrate competitive probabilistic forecasting performance on both distribution calibration and temporal dependency fidelity. Hierarchical ablation studies confirm that incorporating intra segment covariance and multimodal patterns provides an effective statistical inductive bias that aligns with real temporal dynamics. Overall, MiraDiff bridges interpretable local statistics and generative diffusion processes, confirming that appropriate statistical inductive biases are crucial for neural networks to maintain temporal consistency and capture underlying sequential dynamics.

AI USE STATEMENT

Generative AI tools, including ChatGPT and Codex, were used to help develop and refine the methodology and experiments, assist with mathematical exposition and code implementation, interpret results, prepare figures, and draft and edit parts of the manuscript. The authors reviewed the AI-assisted material against the cited literature, implementation, and saved experimental records, and take responsibility for the final claims, results, code, figures, and text.

REPRODUCIBILITY STATEMENT

Section 3 and Algorithms 1 and 2 describe the model, reference fitting, and sampling procedures. Appendix A provides the mathematical derivations. Appendix B documents the architecture, data preprocessing, experimental configurations, model selection, sampling protocol, and evaluation metrics. Additional analyses and visualizations are provided in the remaining appendices. An anonymized implementation with training and evaluation scripts is available at `https://anonymous.4open.science/r/MiraDiff-CEAB/`

REFERENCES

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–38, 1977.

Xinyao Fan, Yueying Wu, Chang Xu, Yu-Hao Huang, Weiqing Liu, and Jiang Bian. MG-TSD: Multi-granularity time series diffusion models with guided learning process. In *International Conference on Learning Representations*, 2024.

Tilmann Gneiting, Fadoua Balabdaoui, and Adrian E. Raftery. Probabilistic forecasts, calibration and sharpness. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(2): 243–268, 2007.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, 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*, 2022.

Marcel Kollovieh, Marten Lienen, David Lüdke, Leo Schwinn, and Stephan Günnemann. Flow matching with gaussian process priors for probabilistic time series forecasting. In *International Conference on Learning Representations*, 2025.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long- and short-term temporal patterns with deep neural networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23009–23022, 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*, 2024.

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*, 2024.

Juan Miguel Lopez Alcaraz and Nils Strodthoff. Diffusion-based time series imputation and forecasting with structured state space models. *Transactions on Machine Learning Research*, 2023.

Manuel A. Matos and Ricardo J. Bessa. Setting the operating reserve using probabilistic wind power forecasts. *IEEE Transactions on Power Systems*, 26(2):594–603, 2011.

Abdullah Mueen, Eamonn Keogh, Qiang Zhu, Sydney Cash, and Brandon Westover. Exact discovery of time series motifs. In *Proceedings of the 2009 SIAM International Conference on Data Mining*, pp. 473–484. SIAM, 2009.

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*, 2023.

Pierre Pinson. Wind energy: Forecasting challenges for its operational management. *Statistical Science*, 28(4):564–585, 2013.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 2021.

Tim Salimans and Jonathan Ho. Progressive distillation for fast sampling of diffusion models. In *International Conference on Learning Representations*, 2022.

David Salinas, Valentin Flunkert, and Jan Gasthaus. DeepAR: Probabilistic forecasting with autoregressive recurrent networks, 2017.

Michael Scheuerer and Thomas M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31016–31029. PMLR, 2023.

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*, 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.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

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

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*, volume 34, 2021.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 72112–72130. PMLR, 2025.

Xinyu Yuan and Yan Qiao. Diffusion-ts: Interpretable diffusion for general time series generation. In *International Conference on Learning Representations*, 2024.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11106–11115, 2021.

A MATHEMATICAL DETAILS

A.1 REFERENCE POSTERIOR AND REVERSE DIFFUSION

CLEAN REFERENCE AND GAUSSIAN POSTERIOR

Let the forecasting horizon contain T time steps and C channels, with nonoverlapping segments of length P and $J = T/P$. The operator $\mathcal{P}_{j,c} : \mathbb{R}^{T\times C} \to \mathbb{R}^P$ extracts segment j from channel c. Fitting a shared Gaussian mixture $q_0(\cdot;\varphi)$ to training segments gives the factorized clean reference

$$Q_0(Y) = \prod_{j=1}^{J}\prod_{c=1}^{C} q_0(\mathcal{P}_{j,c}Y;\varphi). \tag{A.1}$$

The reference is independent across segments and channels. The history H and dependencies beyond each segment enter through the neural correction in Section 3.

For a clean segment z in component k, let $z \mid k \sim \mathcal{N}(\mu_k, \Sigma_k)$ and $x = a_n z + b_n \epsilon$, where $\epsilon \sim \mathcal{N}(0, I_P)$. The noisy covariance is $V_{k,n} = a_n^2\Sigma_k + b_n^2 I_P$, and $\mathrm{Cov}(z, x \mid k) = a_n\Sigma_k$. Gaussian conditioning therefore yields the component mean $m_{k,n}(x) = \mu_k + a_n\Sigma_k V_{k,n}^{-1}(x - a_n\mu_k)$, as used in the main text, and the component posterior covariance

$$C_{k,n} = \Sigma_k - a_n^2\Sigma_k V_{k,n}^{-1}\Sigma_k. \tag{A.2}$$

Bayes' rule gives the responsibilities $\rho_{k,n}(x)$ defined in Section 3; the complete local posterior is $q(z \mid x, n) = \sum_{k=1}^{K}\rho_{k,n}(x)\mathcal{N}(z; m_{k,n}(x), C_{k,n})$. Its mean is the analytic clean estimate $\mathcal{D}_B(x,n) = \sum_k \rho_{k,n}(x)m_{k,n}(x)$. Applying this estimate independently to all JC segments gives $\mathcal{D}_B(X_n, n)$.

STABLE EVALUATION OF THE REFERENCE VELOCITY

Directly evaluating $\big(a_n x - m_{k,n}(x)\big)/b_n$ can lose precision when b_n is small. Using $a_n^2 + b_n^2 = 1$ and $V_{k,n} - \Sigma_k = b_n^2(I_P - \Sigma_k)$ gives the equivalent component velocity

$$v_{B,k}(x,n) = b_n V_{k,n}^{-1}[a_n(I_P - \Sigma_k)x - \mu_k]\,. \tag{A.3}$$

The reference velocity for a segment is $\sum_k \rho_{k,n}(x)v_{B,k}(x,n)$, which equals $\big(a_n x - \mathcal{D}_B(x,n)\big)/b_n$. In implementation, an eigendecomposition $\Sigma_k = U_k\Lambda_k U_k^\top$ reduces solves and Gaussian log densities to the eigenvalues $a_n^2(\Lambda_k)_{p,p} + b_n^2$. Responsibilities are normalized with a log sum exp operation. Reference calculations use double precision before conversion to the neural tensor dtype.

DIFFUSION COEFFICIENTS

For completeness, let $\beta_n \in (0,1)$, $\alpha_n = 1-\beta_n$, $\bar\alpha_n = \prod_{i=1}^{n}\alpha_i$, and $\bar\alpha_0 = 1$. The forward coefficients used in Section 2.2 are $a_n = \sqrt{\bar\alpha_n}$ and $b_n = \sqrt{1-\bar\alpha_n}$. Define the standard DDPM coefficients $c_n = \beta_n\sqrt{\bar\alpha_{n-1}}/(1-\bar\alpha_n)$, $d_n = \sqrt{\alpha_n}(1-\bar\alpha_{n-1})/(1-\bar\alpha_n)$, and $\widetilde\beta_n = \beta_n(1-\bar\alpha_{n-1})/(1-\bar\alpha_n)$.

REVERSE SAMPLING UPDATE

The predicted clean future is $\widehat{Y}_\theta = \mathcal{D}_B(X_n,n) - b_n r_\theta(X_n,n,H)$, so the reverse conditional mean becomes

$$\mu_\theta(X_n,n,H) = c_n\mathcal{D}_B(X_n,n) - c_n b_n r_\theta(X_n,n,H) + d_n X_n. \tag{A.4}$$

For $n \geq 2$ and $\xi_n \sim \mathcal{N}(0, I_{TC})$, the reverse update is

$$X_{n-1} = \mu_\theta(X_n,n,H) + \sqrt{\widetilde\beta_n}\,\xi_n. \tag{A.5}$$

At $n = 1$, $c_1 = 1$, $d_1 = 0$, and $\widetilde\beta_1 = 0$, hence $X_0 = \widehat{Y}_\theta(X_1,1,H)$. Initializing $X_N \sim \mathcal{N}(0, I_{TC})$ and applying these transitions defines the conditional generator

$$p_\theta(Y \mid H) = \int p_N(X_N)\prod_{n=1}^{N} p_\theta(X_{n-1} \mid X_n, H)\,dX_{1:N}, \qquad X_0 = Y, \tag{A.6}$$

where the $n = 1$ transition is the deterministic distribution concentrated at $\widehat{Y}_\theta(X_1,1,H)$.

A.2 Optimal correction and score interpretation

Population optimal residual

Write $\mathcal{I} = (X_n, n, H)$ and let $v_n^* = a_n\epsilon - b_n Y$ be the velocity target. For the squared velocity loss in Eq. 12, the population optimal correction over an unrestricted function class is

$$r^*(\mathcal{I}) = \mathbb{E}[v_n^* \mid \mathcal{I}] - v_B(X_n, n). \tag{A.7}$$

Because $v_B(X_n, n)$ is determined by X_n and n, an unrestricted residual can represent the same conditional optimal total velocity as a model without a reference. The decomposition assigns the locally computable part to v_B and the remaining conditional part to the learned residual.

Reference choices and the standard diffusion case

Several reference choices fit this formulation. With $K = 1$, $\mu_1 = 0$, and $\Sigma_1 = I_P$, the reference clean estimate is $\mathcal{D}_B(x, n) = a_n x$, hence $v_B = 0$ and the model reduces to the Standard velocity predictor. A diagonal empirical covariance retains segment position means and variances without correlations between positions. An empirical full covariance with $K = 1$ additionally retains within segment covariance. With $K > 1$, the responsibilities vary with the noisy segment and diffusion step, producing a mixture weighted posterior estimate. These are choices of reference within the same framework; MiraDiff can also use the empirical $K = 1$ reference.

Derivation of the score ratio

Finally, let $p_n(X_n \mid H)$ be the noisy marginal of the target conditional distribution, distinct from the reference noisy marginal $Q_n(X_n) = \prod_{j,c} q_n(\mathcal{P}_{j,c}X_n; \varphi)$. For $a_n, b_n > 0$, differentiating the Gaussian forward convolution gives $\nabla_{X_n} \log p_n(X_n \mid H) = -\big(X_n - a_n\mathbb{E}[Y \mid X_n, H]\big)/b_n^2$. Combining this identity with the definition of velocity yields

$$\mathbb{E}[v_n^* \mid \mathcal{I}] = \frac{a_n X_n - \mathbb{E}[Y \mid X_n, H]}{b_n} = -\frac{b_n}{a_n}\left[X_n + \nabla_{X_n} \log p_n(X_n \mid H)\right]. \tag{A.8}$$

Applying the same calculation to the factorized reference, for which $\mathcal{D}_B(X_n, n)$ is the posterior mean, gives

$$v_B(X_n, n) = \frac{a_n X_n - \mathcal{D}_B(X_n, n)}{b_n} = -\frac{b_n}{a_n}\left[X_n + \nabla_{X_n} \log Q_n(X_n)\right]. \tag{A.9}$$

The X_n terms cancel on subtracting these identities. Consequently,

$$r^*(\mathcal{I}) = -\frac{b_n}{a_n}\nabla_{X_n} \log \frac{p_n(X_n \mid H)}{Q_n(X_n)}. \tag{A.10}$$

This identity characterizes the population optimal correction as the score difference between the history conditioned target and the fitted local reference. It describes the role of the residual under the specified squared loss, without requiring the empirical reference to match the conditional target distribution.

B Implementation and evaluation protocol

B.1 Datasets and preprocessing

Table B.1 summarizes the eight multivariate datasets and the number of Gaussian components selected for the main comparison. The number of time steps refers to the observations used in this study. All datasets use a history length of $L = 168$, a forecasting horizon of $T = 192$, and nonoverlapping reference segments of length $P = 16$.

The data are split chronologically into training, validation, and test sets. The ETT datasets use an approximately $60/20/20$ split, and the other datasets use an approximately $70/10/20$ split. Per-channel normalization statistics and the reference mixture are fitted using training data only. Missing observations in the history are filled causally, while unobserved future values are excluded from evaluation. For comparison with archived baselines, ETT, Electricity, and Solar are scored in their historical training-scaled coordinates; Weather and Traffic use their own training scalers.

Table B.1: Dataset statistics and reference configurations selected for Table 1.

Dataset	Time steps	Channels	Sampling interval	K
ETTh1	14,400	7	1 hour	2
ETTh2	14,400	7	1 hour	6
ETTm1	57,600	7	15 minutes	2
ETTm2	57,600	7	15 minutes	1
Weather	52,696	21	10 minutes	8
Electricity	26,304	321	1 hour	6
Traffic	17,544	862	1 hour	6
Solar	52,560	137	10 minutes	3

B.2 Model and reference configuration

The history encoder uses channelwise temporal convolutions and pooled channel features. Its representation width is 32, and the correction network uses width 64 with gated dilated residual blocks. The output layer is initialized to zero so that the initial total velocity equals the analytic reference velocity.

The reference is fitted to training segments without additional per-segment normalization, using a full-covariance Gaussian mixture and expectation maximization. An additive $10^{-3}I_P$ term regularizes each component covariance. The fitted mixture is then frozen during neural training and sampling. The selected number of components K varies by dataset as shown in Table B.1, while $P = 16$ throughout. For the controlled comparison in Table 2, Standard uses no reference, Diagonal and Single use $K = 1$ with diagonal and full covariance respectively, and Mixture evaluates a separate $K > 1$ reference. In particular, the selected Table 1 configurations for ETTm2 use $K = 1$ and correspond to the Single reference setting. Their Mixture ablation results are separate from the selected main-table checkpoints.

B.3 Training, selection, and sampling

Neural models are trained with AdamW, a learning rate of 2×10^{-4}, weight decay of 10^{-4}, an effective batch size of 8, and gradient norm clipping at 5. The diffusion process uses $N = 100$ steps, a cosine schedule with offset 0.008, and $\beta_n \leq 0.999$.

The main test panels use 32 origins on each dataset. Predictions use 100 samples per origin. The controlled comparisons use the same dataset splits, neural architecture, diffusion schedule, and evaluation conventions; the reference form is varied as described above. External baselines retain their original training and sampling settings.

B.4 Evaluation metrics

For S scalar predictive samples $u_1, \ldots, u_S$ and observed value u, empirical CRPS is computed as

$$\widehat{\mathrm{CRPS}}_S = \frac{1}{S}\sum_{s=1}^{S}|u_s - u| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{r=1}^{S}|u_s - u_r|. \tag{B.1}$$

The denominator S^2 follows the empirical-ensemble convention used for the reported results. CRPS is averaged over observed forecast positions and channels, then equally over test origins.

Temporal Variogram Score (VS) measures discrepancies in within-channel temporal differences. For lag $\ell \in \{1, 4, 8, 24\}$, let $\mathcal{O}_\ell$ contain forecast pairs (t, c) for which both observations are available, and let $\mathcal{A}_+$ contain the nonempty lags. With predictive trajectories $x^{(1)}, \ldots, x^{(S)}$ and realized future y, the score is

$$\mathrm{VS} = \frac{1}{|\mathcal{A}_+|}\sum_{\ell\in\mathcal{A}_+}\frac{1}{|\mathcal{O}_\ell|}\sum_{(t,c)\in\mathcal{O}_\ell}\left[|y_{t,c} - y_{t-\ell,c}|^{1/2} - \frac{1}{S}\sum_{s=1}^{S}|x^{(s)}_{t,c} - x^{(s)}_{t-\ell,c}|^{1/2}\right]^2. \tag{B.2}$$

VS is averaged equally over origins with at least one valid temporal pair.

For QICE, the empirical predictive deciles define ten bins. If f_j is the fraction of observed targets falling in bin j, pooled across origins, forecast positions, and channels, then

$$\mathrm{QICE} = \frac{1}{10} \sum_{j=1}^{10} |f_j - 0.1|. \tag{B.3}$$

Empirical deciles are linearly interpolated, and out of range targets are assigned to the end bins. The values in Table 1 report $100 \times \mathrm{QICE}$.

C Additional Trajectory Visualizations

Figures C.1 and C.2 extend the trajectory comparison in Fig. 3 to eight datasets and additional baselines. The black curve shows the observed series, while the orange curve and shaded region show the mean and interval of generated forecasts. The vertical dashed line separates the observed history from the forecast horizon.

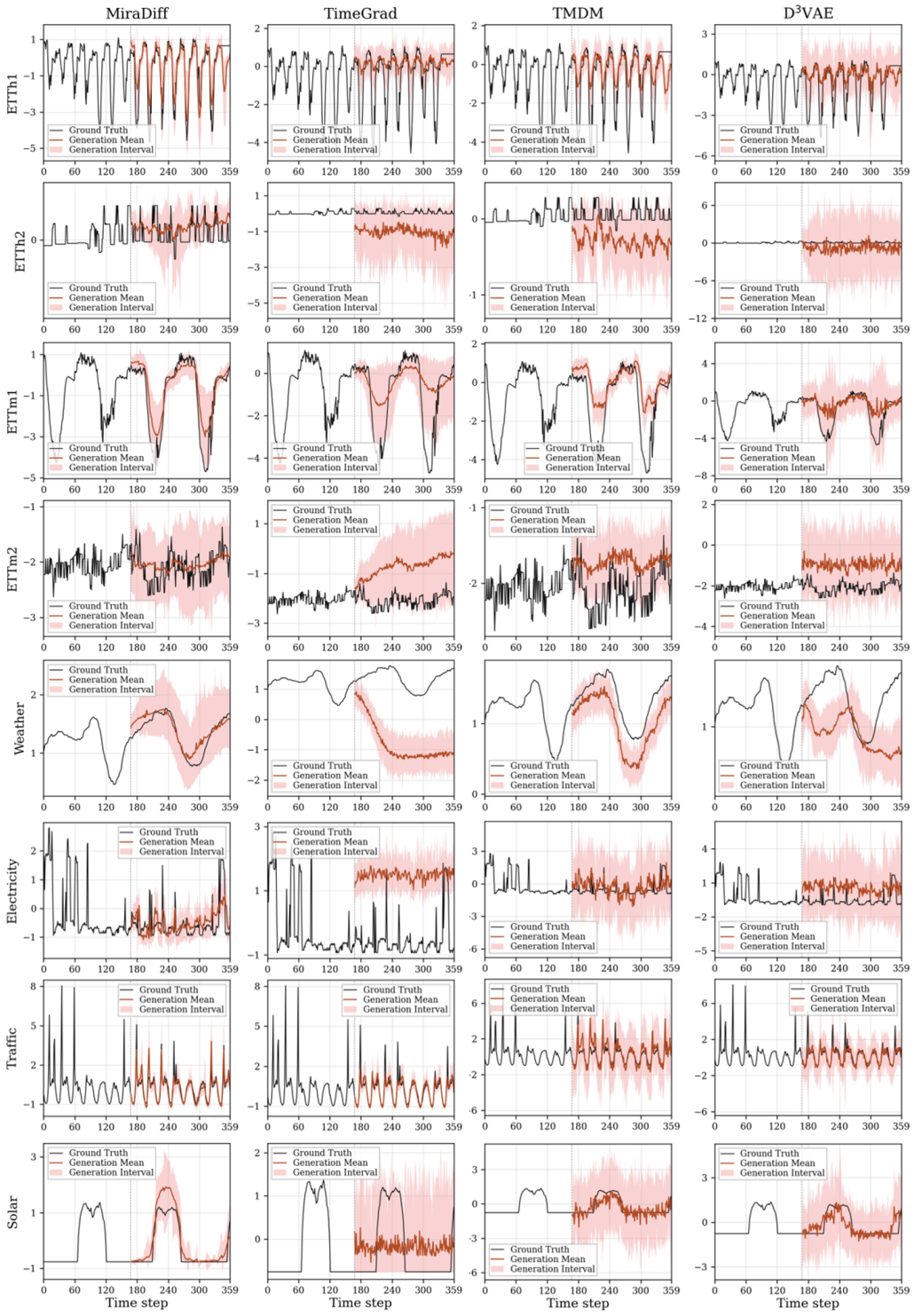


Figure C.1: Forecast trajectory comparisons across eight datasets for MiraDiff, TimeGrad, TMDM, and D^{3}VAE.

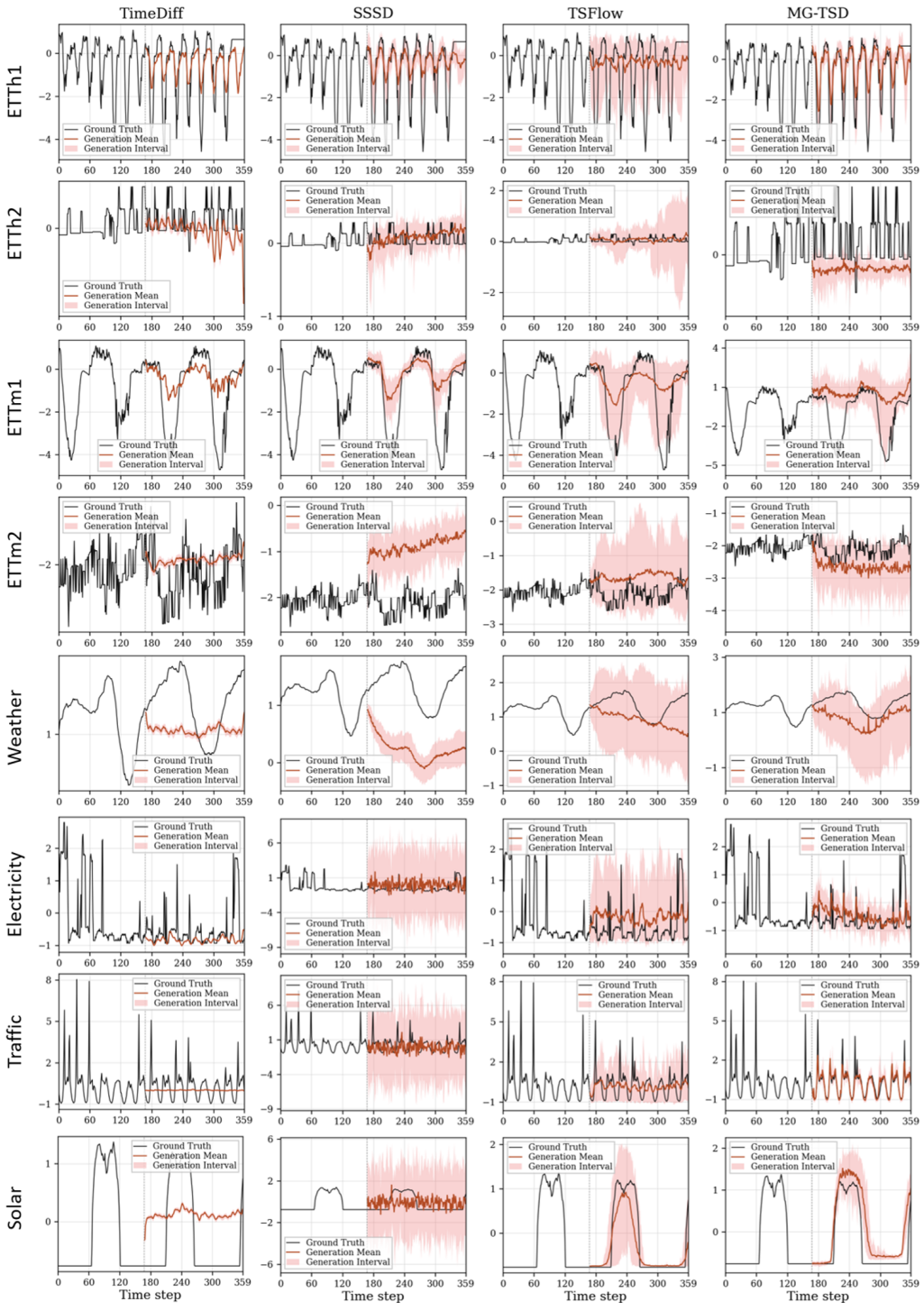


Figure C.2: Forecast trajectory comparisons across eight datasets for TimeDiff, SSSD, TSFlow, and MG-TSD.

D Additional Distributional Comparisons

Fig. D.1–Fig. D.6 extend the t-SNE and KDE comparisons to six datasets and additional baselines. MiraDiff attains the lowest W_1 on ETTh1, Electricity, Weather, and Traffic, and the second lowest on ETTm2 and Solar.

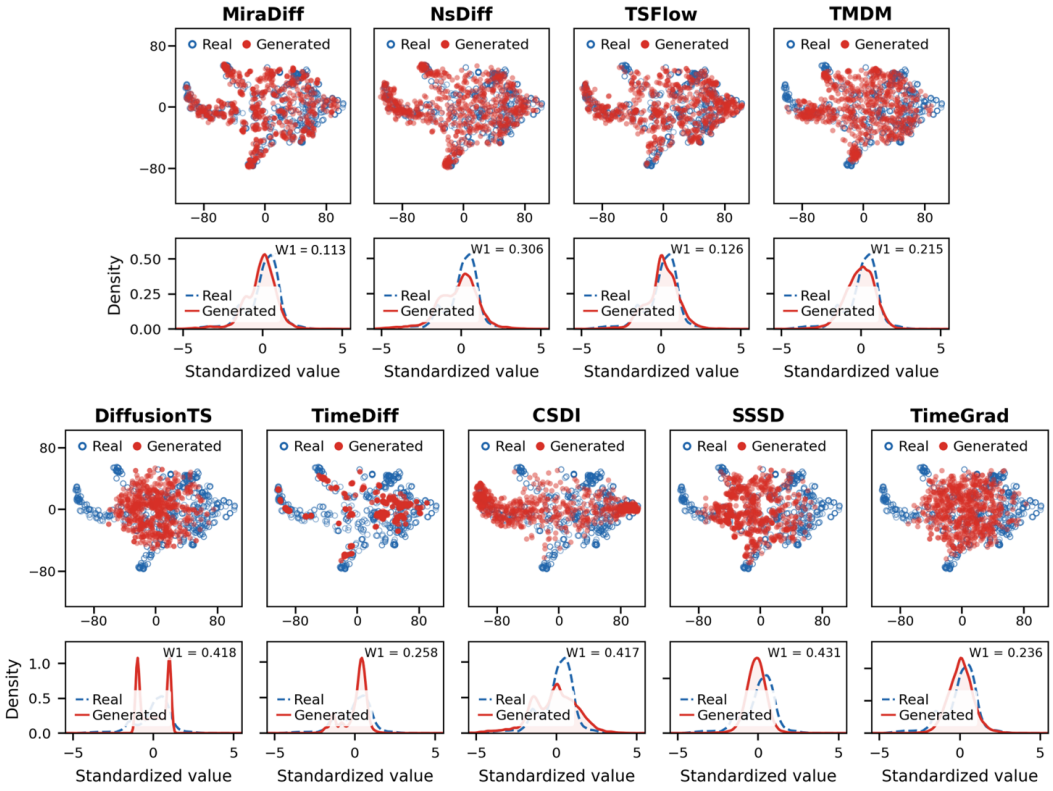


Figure D.1: t-SNE projections and KDE comparisons of observed and generated samples on **ETTh1**. The KDE panels report Wasserstein-1 distances (W_1).

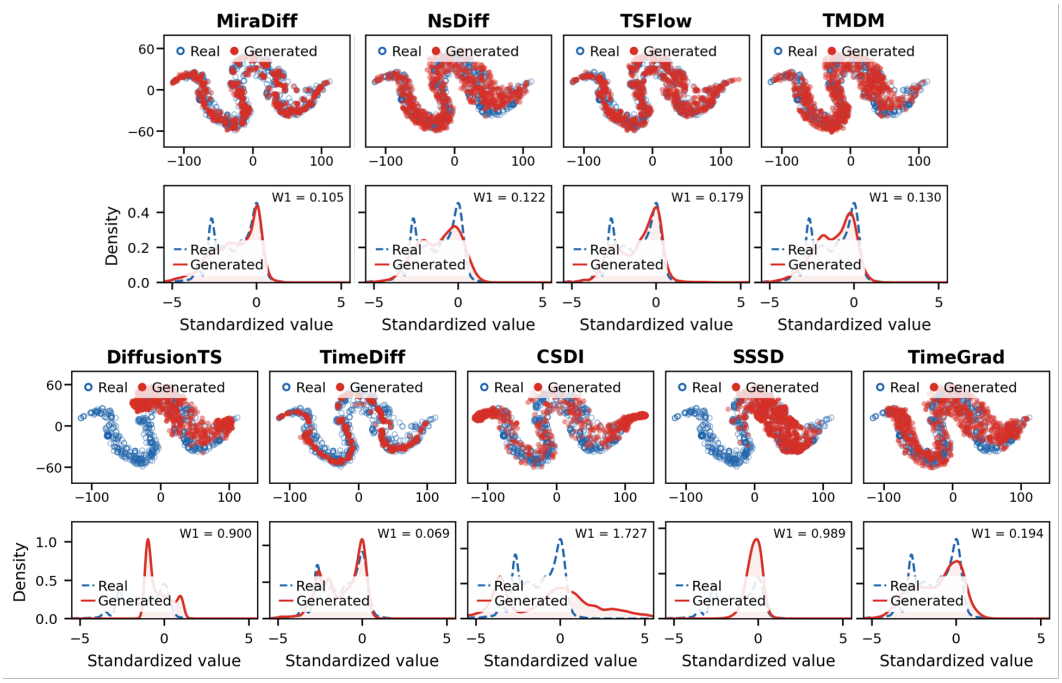


Figure D.2: t-SNE projections and KDE comparisons of observed and generated samples on **ETTm2**. The KDE panels report W_1 distances.

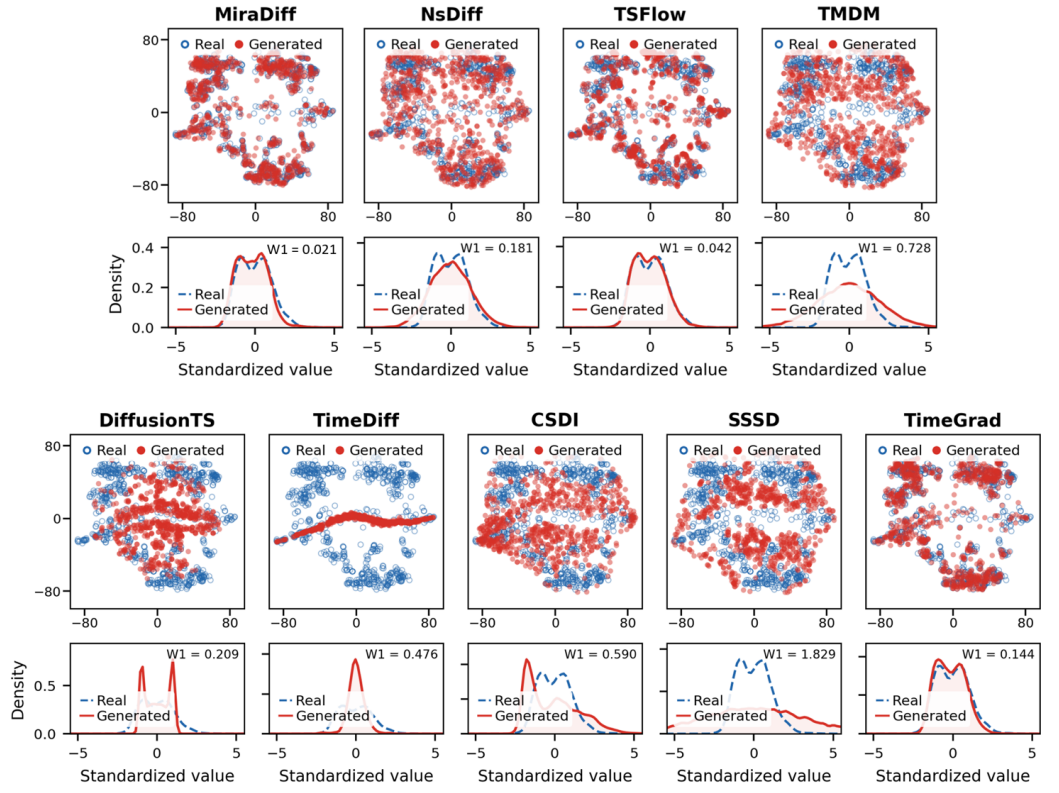


Figure D.3: t-SNE projections and KDE comparisons of observed and generated samples on **Electricity**. The KDE panels report W_1 distances.

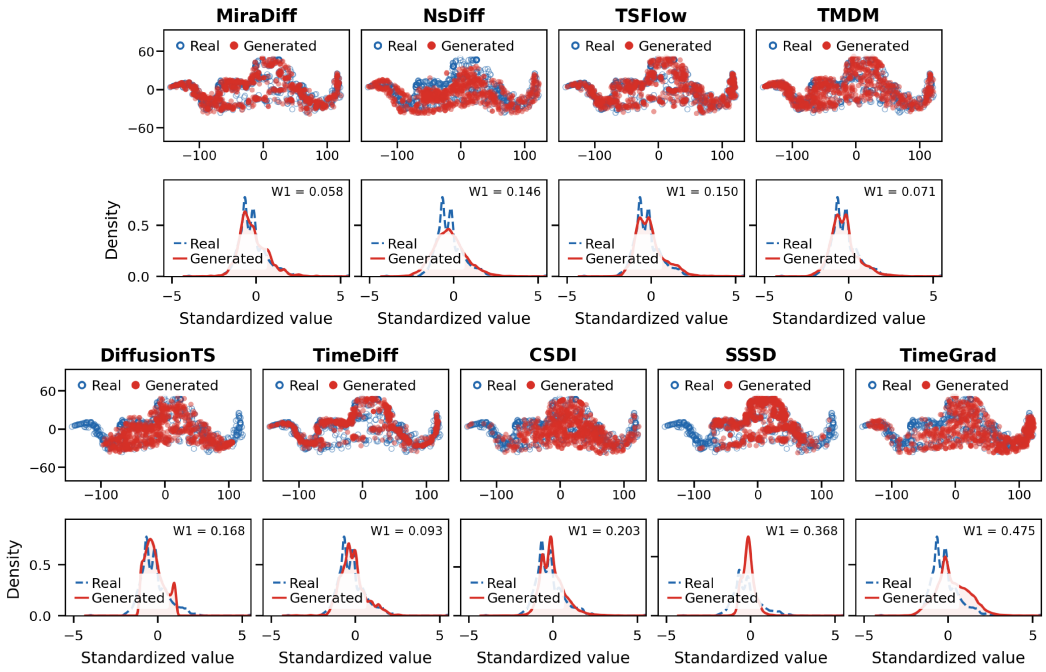


Figure D.4: t-SNE projections and KDE comparisons of observed and generated samples on **Weather**. The KDE panels report W_1 distances.

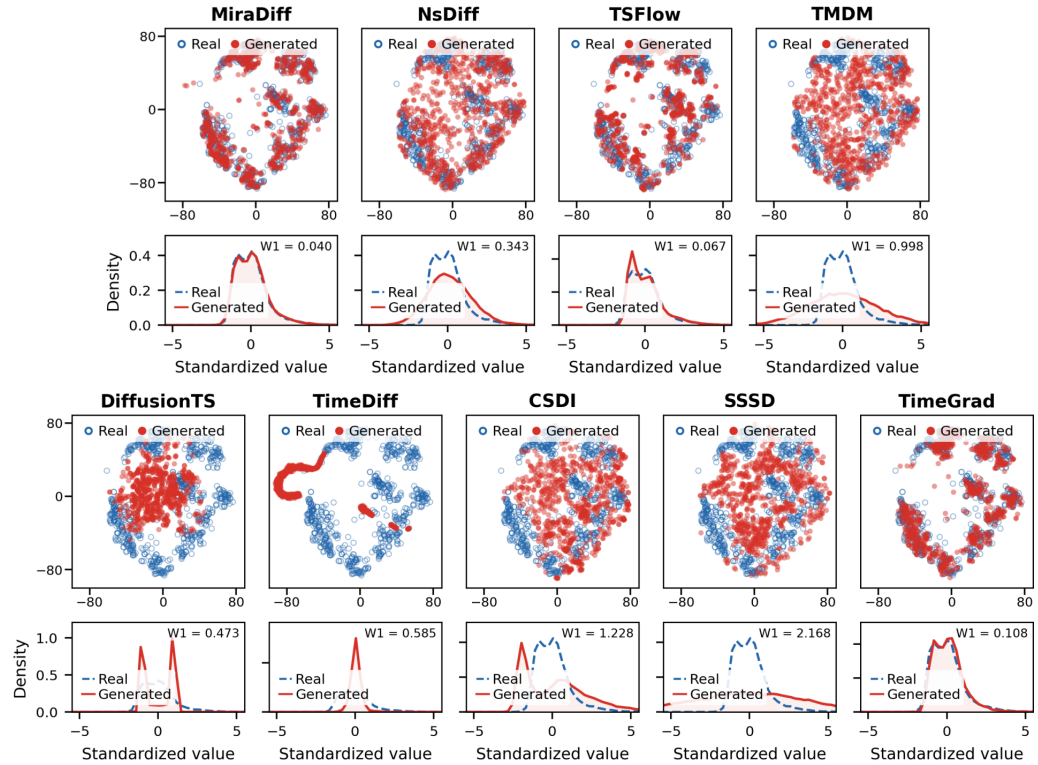


Figure D.5: t-SNE projections and KDE comparisons of observed and generated samples on **Traffic**. The KDE panels report W_1 distances.

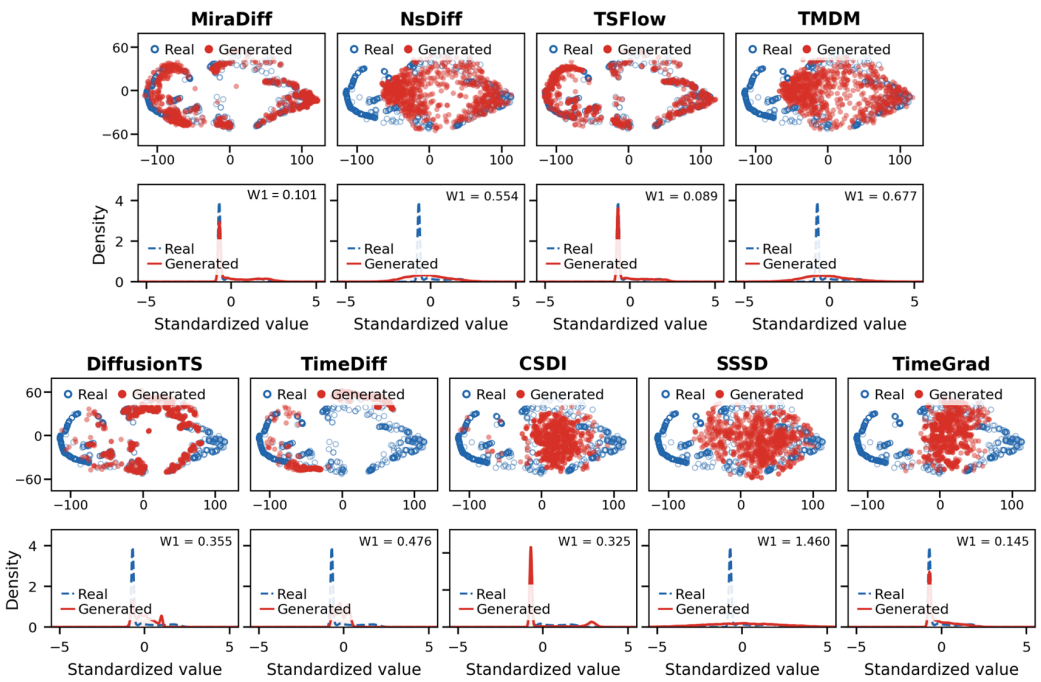


Figure D.6: t-SNE projections and KDE comparisons of observed and generated samples on **Solar**. The KDE panels report W_1 distances.

E SUPPLEMENTARY REFERENCE VISUALIZATION

Fig. E.1 to Fig. E.10 show illustrative eight component Gaussian mixture fits to training segments from ETTh1, Electricity, Weather, Traffic, and Solar. Each dataset is shown with observed segments, component means and pointwise standard deviation bands, and within segment correlation matrices. These visualization fits are separate from the configurations used for forecasting.

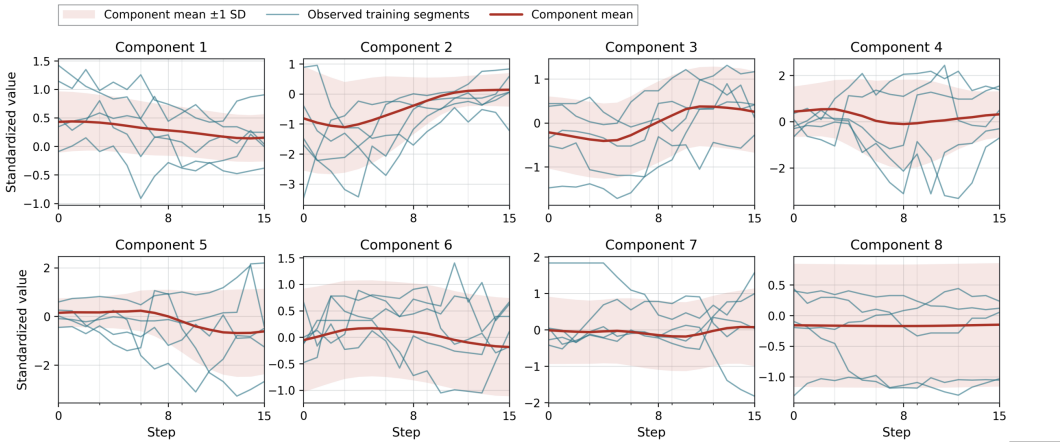


Figure E.1: Real observed **ETTh1** training segments and the fitted reference with eight components.

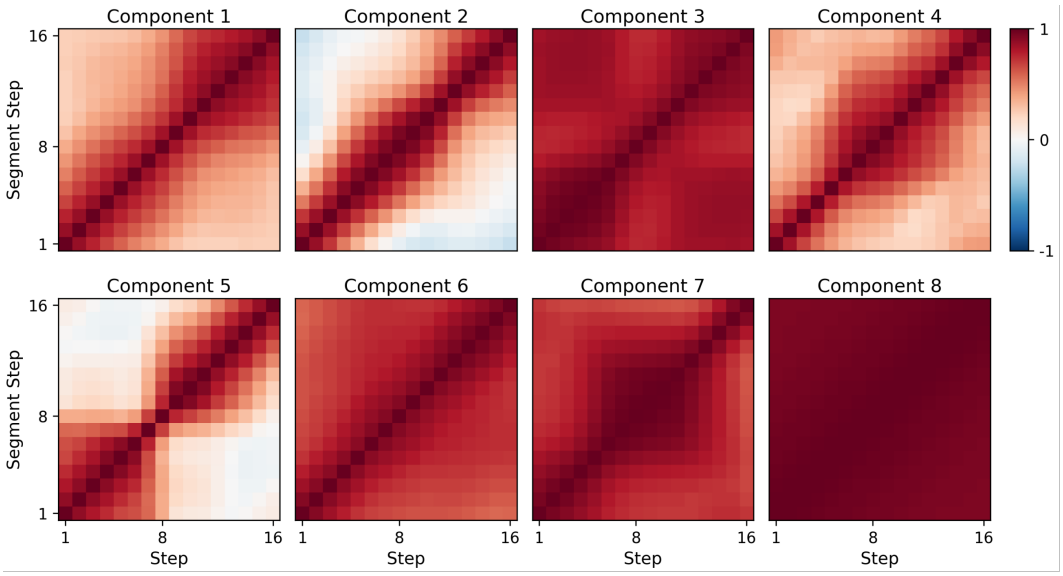


Figure E.2: Correlations among the 16 segment positions for each of the eight fitted **ETTh1** components.

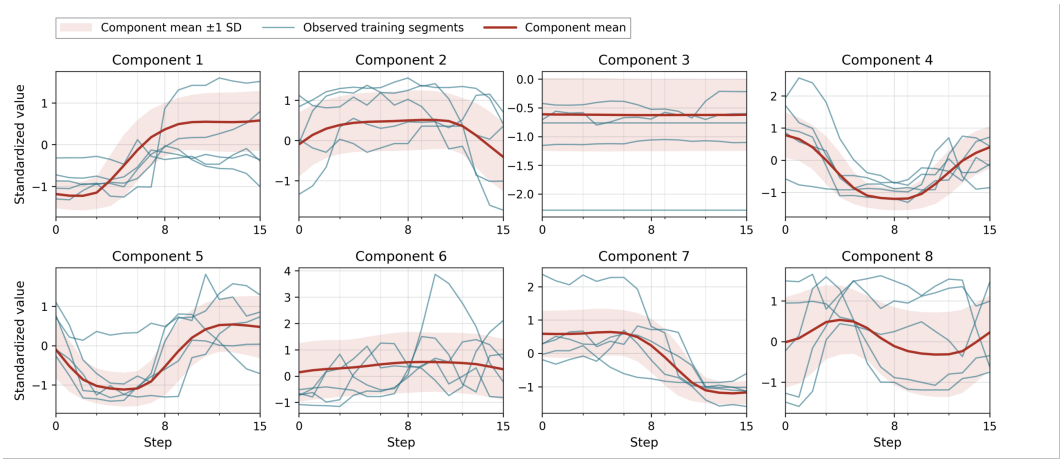


Figure E.3: Real observed **Electricity** training segments and the fitted reference with eight components.

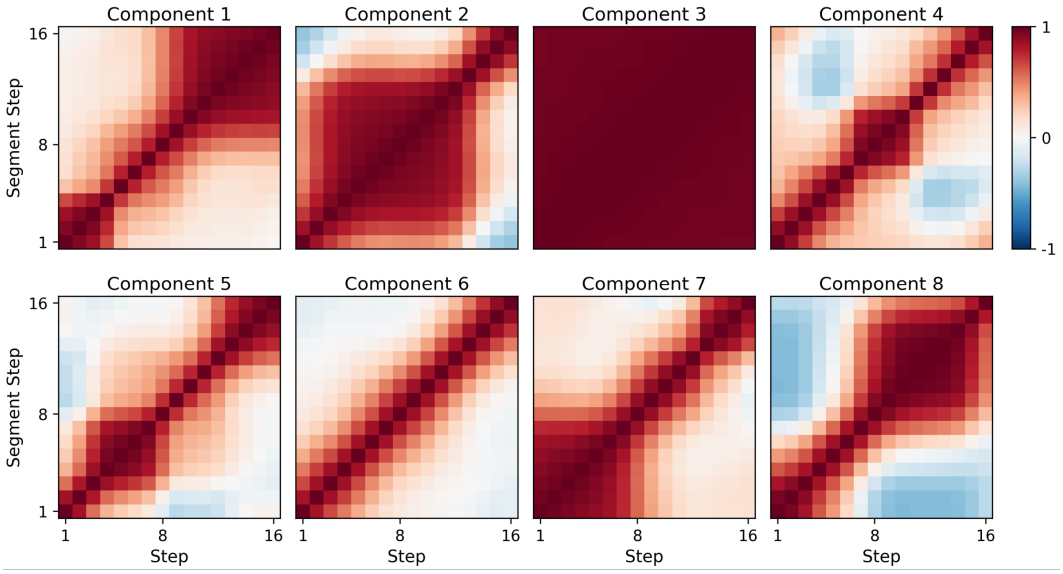


Figure E.4: Correlations among the 16 segment positions for each of the eight fitted **Electricity** components.

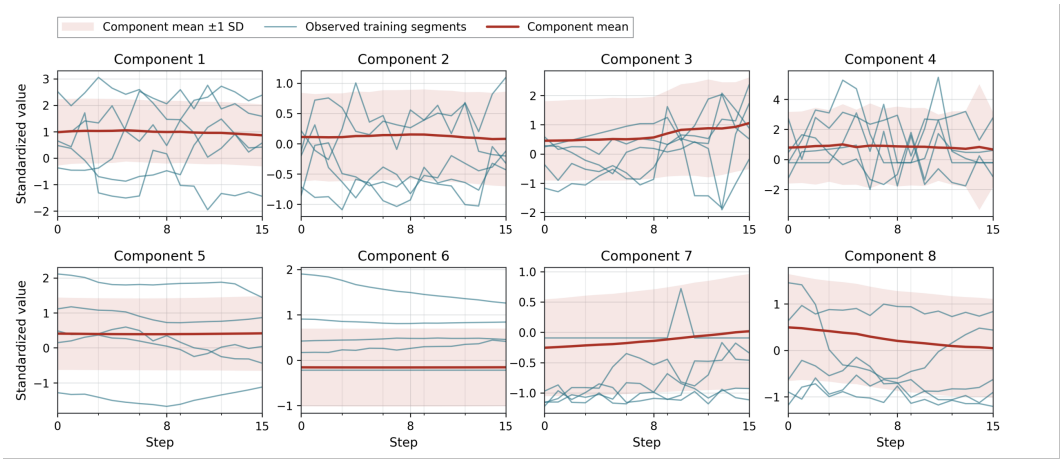


Figure E.5: Real observed **Weather** training segments and the fitted reference with eight components.

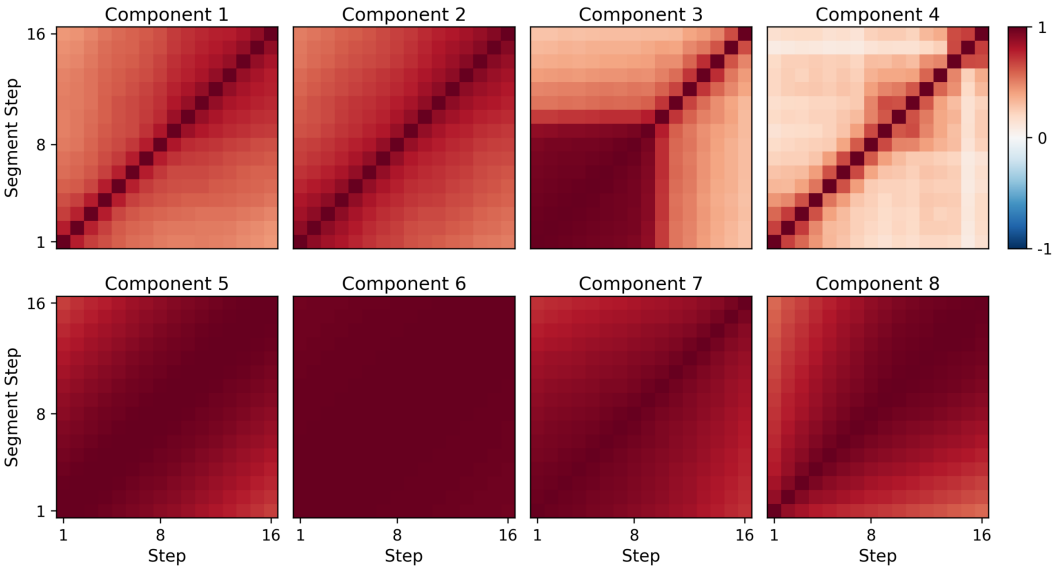


Figure E.6: Correlations among the 16 segment positions for each of the eight fitted **Weather** components.

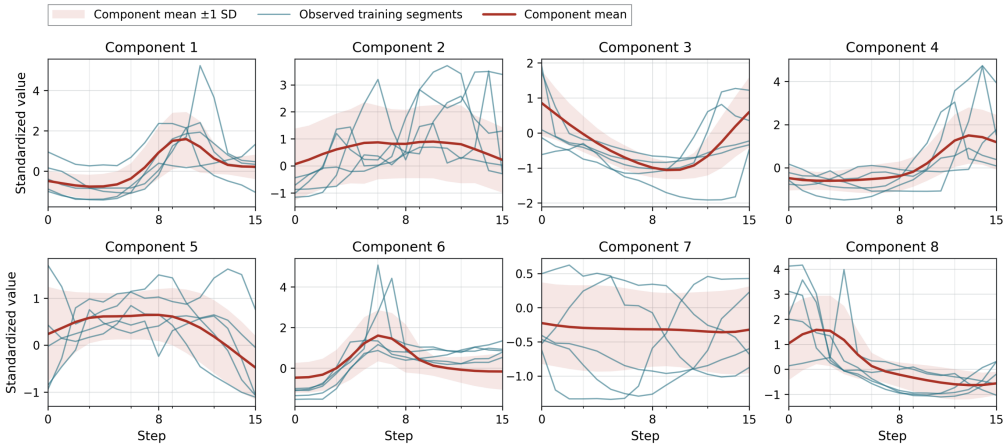


Figure E.7: Real observed **Traffic** training segments and the fitted reference with eight components.

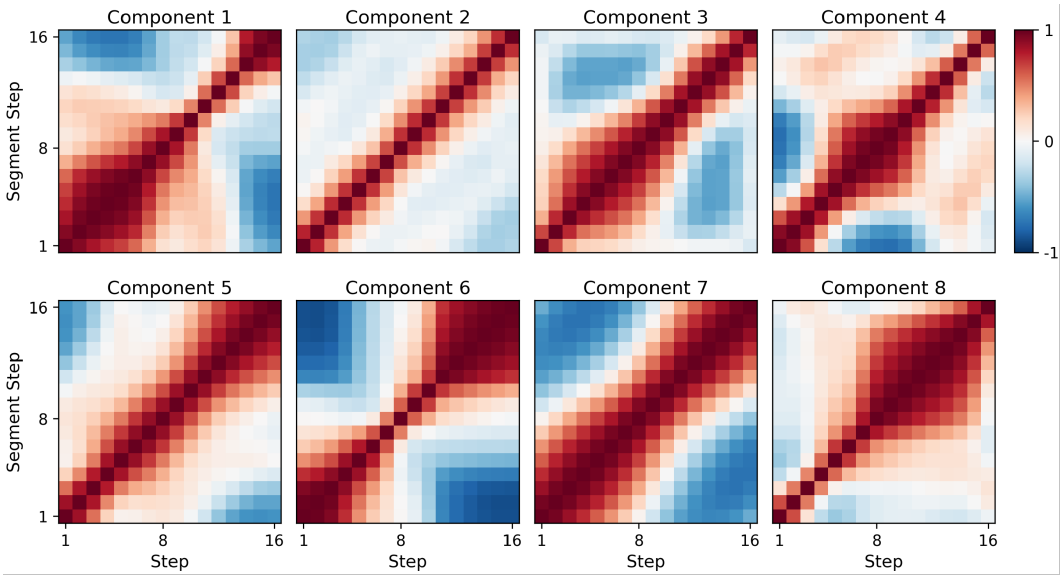


Figure E.8: Correlations among the 16 segment positions for each of the eight fitted **Traffic** components.

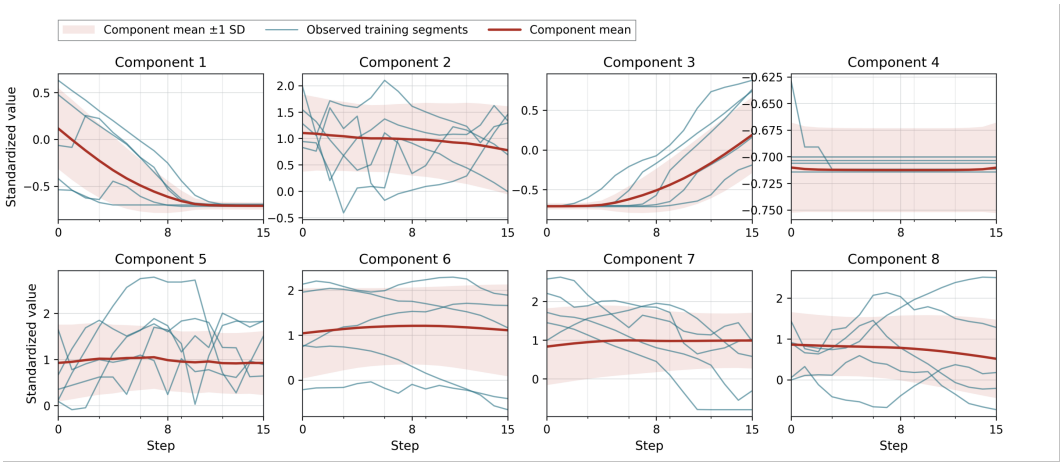


Figure E.9: Real observed **Solar** training segments and the fitted reference with eight components.

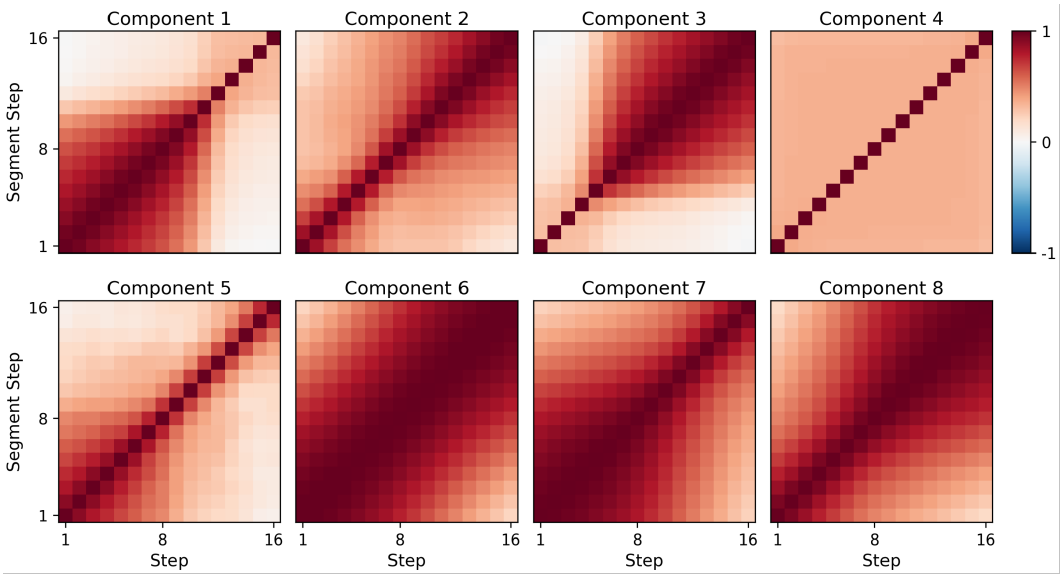


Figure E.10: Correlations among the 16 segment positions for each of the eight fitted **Solar** components.