FLOWOC: HISTORY-INFORMED FLOW MATCHING WITH OPTIMAL CONTROL CALIBRATION FOR PROBABILISTIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Generative models provide a natural framework for probabilistic multivariate time series forecasting by modeling the conditional distribution of future trajectories rather than producing a single deterministic estimate. However, improving the alignment of generated forecasts with future observations while preserving meaningful predictive variability remains challenging. To address this issue, FlowOC adopts a two-stage training procedure. In the first stage, Conditional Flow Matching learns the predictive distribution of future trajectories from informative historical-patch source states. In the second stage, the pretrained flow is frozen, and an additive controller performs optimal control-inspired calibration using ensemble-level accuracy and distribution-shape preservation objectives, thereby improving the predictive center while maintaining the stochastic structure of the generated trajectories. Experiments on six multivariate forecasting benchmarks demonstrate that FlowOC improves point forecasting accuracy while preserving or enhancing probabilistic forecasting quality, thereby achieving a balance between forecast accuracy and faithful uncertainty characterization. Extensive ablation and qualitative analysis further validate the effectiveness of the historical-patch source distribution and the proposed optimal control calibration strategy. Code is available at `https://anonymous.4open.science/r/Flow_OC-E8DA/`.

1 INTRODUCTION

Multivariate time series forecasting plays a critical role in many real-world applications (Jin et al., 2023), which include energy management (Ma et al., 2023), traffic monitoring (Shaygan et al., 2022), and weather prediction (Alley et al., 2019). Existing forecasting methods can be broadly categorized into deterministic forecasting (Hu et al., 2025; Liu et al., 2025b) and probabilistic forecasting (Rasul et al., 2021; Wu et al., 2025). Deterministic forecasting produces a single estimate of the future trajectory, whereas probabilistic forecasting characterizes the conditional distribution of possible future outcomes. As real-world time series are often affected by stochastic dynamics and unobserved disturbances, achieving accurate forecasts while faithfully representing predictive uncertainty is essential for reliable downstream decision-making.

Deterministic forecasting methods generate a single future trajectory and typically optimize the discrepancy between the predicted and observed future trajectories. Deep learning has led to strong deterministic forecasting models, including linear models such as DLinear (Zeng et al., 2023), patch-based Transformers such as PatchTST (Nie et al., 2023), and variate-token Transformers such as iTransformer (Liu et al., 2024). Although these methods achieve competitive point forecasting accuracy, a single deterministic trajectory does not directly characterize predictive uncertainty or the range of plausible future outcomes. Probabilistic forecasting aims to characterize the conditional distribution of future trajectories by modeling the conditional distribution given historical observations. Representative approaches include latent variable-based and autoencoder-based methods, such as K^2VAE (Wu et al., 2025) and TLAE (Nguyen & Quanz, 2021), as well as diffusion-based methods (Su et al., 2025; Tashiro et al., 2021; Li et al., 2025) which introduce stochastic generative processes to capture diverse future outcomes. Nevertheless, the ability to generate multiple future trajectories does not necessarily ensure that the resulting forecasts are well aligned with fu-

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.

ture observations. A calibration mechanism holds great potential to improve forecast accuracy while preserving predictive variability needed to represent uncertainty.

In this work, we propose FlowOC, a two-stage framework for probabilistic multivariate time series forecasting. FlowOC integrates history-informed flow matching with optimal control-inspired calibration. In the first stage, a Flow Matching model learns to transport samples from a source distribution constructed from historical patches to the distribution of future trajectories, thereby generating an ensemble of future trajectories that captures predictive uncertainty. The generation process is conditioned on contextual information extracted from the observed history. In the second stage, the pretrained flow is frozen, and an additive controller is trained using ensemble-level forecasting objectives to refine the generated trajectories, improving forecast accuracy while retaining the uncertainty structure captured by the pretrained flow. The contributions of this work include the following aspects:

- We develop a history-informed Flow Matching model that combines historical-patch source initialization with cross-channel contextual conditioning to learn the conditional distribution of future trajectories.
- We introduce an optimal control-inspired post-training calibration strategy that derives value-gradient guidance from an ensemble-level forecasting objective, reducing bias in the predictive mean while constraining changes to the learned distributional structure.
- We evaluate FlowOC against deterministic and probabilistic baselines, showing improved point accuracy without sacrificing probabilistic forecasting quality.

2 Related works

Probabilistic Time Series Forecasting. In contrast to deterministic forecasting, probabilistic time series forecasting aims to model the conditional distribution of future trajectories, allowing multiple possible future outcomes to be generated and providing a characterization of predictive uncertainty (Gneiting & Katzfuss, 2014; Salinas et al., 2020). Among these approaches, VAE-based methods introduce stochastic latent variables and generate multiple future trajectories by sampling from the learned latent distribution (Wu et al., 2025). Following the widespread success of diffusion models in image and video generation (Rombach et al., 2022; Ho et al., 2022), diffusion-based approaches have been increasingly adopted for probabilistic time series forecasting, which leverages their inherent ability to model complex predictive distributions. CSDI (Tashiro et al., 2021) introduces a conditional score-based diffusion framework for time series imputation and forecasting, where observed values are incorporated as conditioning information to guide the generation process. TimeGrad (Rasul et al., 2021), in contrast, integrates diffusion models into an autoregressive forecasting framework, generating future values sequentially conditioned on historical observations and previously generated values. TimeDART (Wang et al., 2025) combines autoregressive and diffusion modeling for self-supervised time series representation learning. Flow Matching provides an alternative formulation by learning a continuous velocity field that transports samples between source and target distributions (Lipman et al., 2024; Albergo et al., 2025). D3U (Li et al., 2025) decouples relatively deterministic and uncertain temporal components by combining a point forecaster with conditional diffusion, whereas PPM (Li et al., 2026) constructs dynamic parametric priors to guide the learning of predictive distributions through a hybrid objective. (Lim et al., 2025) design forecasting-oriented probability paths and interpret their construction from a Schrödinger Bridge perspective, which enables Flow Matching to transport samples from a history-informed source distribution toward the target future distribution. TSFlow (Kollovieh et al., 2025) further incorporates Gaussian Processes as informed prior distributions to generate future trajectories. Although some probabilistic forecasting models incorporate auxiliary or hybrid objectives, existing Flow Matching-based approaches primarily focus on learning the transport from source to future distributions. This motivates post-training adaptation to improve forecast accuracy while retaining the uncertainty structure captured by the pretrained model.

Calibration Methods. To further adapt pretrained generative models toward task-specific objectives, recent studies have explored various fine-tuning and guidance strategies. A straightforward approach is to directly fine-tune the pretrained generative model with task-specific objectives. However, updating the entire model may substantially distort the learned predictive distribution and potentially reduce the diversity of generated samples. Fan & Lee (2023) propose Shortcut Fine-Tuning

(SFT) to fine-tune pretrained diffusion models for accelerated sampling while retaining the original generative capability. Optimal control-inspired methods have attracted increasing attention because they provide a mathematically grounded framework for steering continuous-time generative dynamics toward task-specific objectives. The adjoint matching method (Domingo i Enrich et al., 2025) enables efficient optimization of continuous-time generative dynamics by propagating the influence of the final objective backward through the generation process, providing guidance for the control of intermediate states. Value Gradient Guidance (VGG) (Liu et al., 2025c) further provides a direct guidance mechanism by learning the value gradient, without the need to explicitly solve the adjoint dynamics during generation. Optimal control-inspired methods provide a framework for refining generated trajectories while constraining deviations from the base model, thereby improving forecast accuracy without substantially distorting the learned predictive distribution.

3 BACKGROUND

3.1 PROBLEM FORMULATION

Given historical observations $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_L] \in \mathbb{R}^{L \times C}$, we aim to model the conditional distribution of $\mathbf{Y} = [\mathbf{x}_{L+1}, \ldots, \mathbf{x}_{L+H}] \in \mathbb{R}^{H \times C}$, where L, C, and H denote the look-back length, number of channels, and prediction horizon, respectively. Specifically, probabilistic forecasting aims to learn the conditional predictive distribution $p(\mathbf{Y}|\mathbf{X})$, but not to provide one single future trajectory prediction. Once the conditional distribution is learned, multiple future trajectories $\{\widehat{\mathbf{Y}}^{(k)}\}_{k=1}^{K}$ can be sampled to characterize the uncertainty associated with future outcomes.

3.2 FLOW MATCHING

Flow Matching is a generative modeling approach that learns a continuous vector field to transport samples from a source distribution q_0 to a target distribution q_1 along a predefined probability path (Lipman et al., 2023). Given a source sample $\mathbf{z}_0 \sim q_0$ and a target sample $\mathbf{z}_1 \sim q_1$, an intermediate state $\mathbf{z}_\tau$ is constructed along the probability path, with the corresponding target velocity $\mathbf{w}_\tau(\mathbf{z}_\tau)$:

$$\frac{\mathrm{d}\mathbf{z}_\tau}{\mathrm{d}\tau} = \mathbf{w}_\tau(\mathbf{z}_\tau), \qquad \tau \in [0, 1]. \tag{1}$$

A neural velocity field $\mathbf{v}_\theta$ is trained to approximate the target velocity through

$$\mathcal{L}_{\mathrm{FM}} = \mathbb{E}_{\tau, \mathbf{z}_0 \sim q_0, \mathbf{z}_1 \sim q_1} \left[\|\mathbf{v}_\theta(\mathbf{z}_\tau, \tau) - \mathbf{w}_\tau(\mathbf{z}_\tau)\|_2^2 \right]. \tag{2}$$

Notably, the standard Flow Matching objective supervises the local velocity field along the probability path rather than directly minimizing a terminal forecasting error.

3.3 OPTIMAL CONTROL PERSPECTIVE

Optimal control (Pontryagin, 2018; Liberzon, 2012) seeks to optimize control actions that minimize a pre-defined cost. Inspired by the concept of optimal control, let us consider the following dynamic system with additive control $\mathbf{u}$:

$$\frac{\mathrm{d}\mathbf{z}_\tau}{\mathrm{d}\tau} = f(\mathbf{z}_\tau, \tau) + \mathbf{u}_\tau, \tag{3}$$

Let us formulate the following optimization problem, which is analogous to an optimal control formulation:

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \left[\int_0^1 \ell(\mathbf{z}_\tau, \mathbf{u}_\tau, \tau)\, \mathrm{d}\tau + \Phi(\mathbf{z}_1) \right], \tag{4}$$

where ℓ denotes the running cost; Φ denotes the terminal cost. Then, the optimal cost-to-go from state $\mathbf{z}$ at time τ is

$$V(\mathbf{z}, \tau) = \min_{\mathbf{u}} \left[\int_\tau^1 \ell(\mathbf{z}_s, \mathbf{u}_s, s)\, \mathrm{d}s + \Phi(\mathbf{z}_1) \,\middle|\, \mathbf{z}_\tau = \mathbf{z} \right], \tag{5}$$

which satisfies the Hamilton–Jacobi–Bellman (HJB) equation with terminal condition $V(\mathbf{z}, 1) = \Phi(\mathbf{z})$. For the quadratic running cost $\ell(\mathbf{z}, \mathbf{u}, \tau) = \|\mathbf{u}\|_2^2/(2\beta)$, where $\beta > 0$ is a control-strength parameter, minimizing the Hamiltonian over $\mathbf{u}$ gives the closed-form feedback law

$$\mathbf{u}^\star(\mathbf{z}, \tau) = -\beta \nabla_{\mathbf{z}} V(\mathbf{z}, \tau). \tag{6}$$

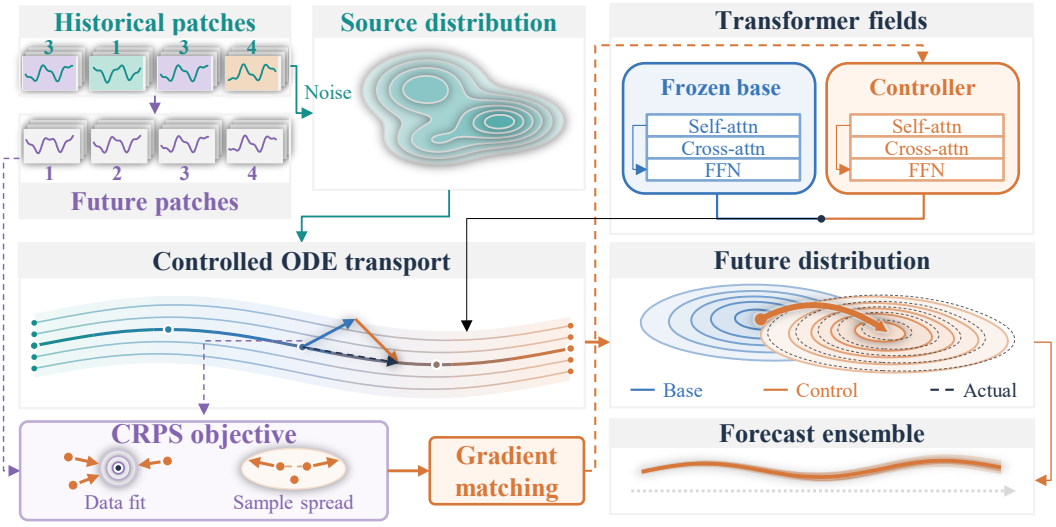


Figure 1: Overview of the FlowOC framework, which consists of two training stages: history-informed Flow Matching and optimal control-inspired refinement.

The drift f does not appear explicitly in Eq. (6); it affects the optimal control only through V. The detailed derivation is provided in Appendix B.

4 Proposed Method

The proposed method consists of two main components. First, we develop a history-informed Flow Matching framework by redesigning the sampling strategy of Flow Matching (Lipman et al., 2023) and incorporating additional information from historical observations. Second, we calibrate the Flow Matching model through an optimal control-inspired refinement strategy.

4.1 History-informed Flow Matching

In the first stage of FlowOC, a conditional flow is learned to transport samples from a history-informed source distribution to the target distribution over future trajectories. Historical patches serve as informative source states. The encoded historical context conditions the learned velocity field.

Historical Patch Construction and Sampling. Following the patching strategy of PatchTST (Nie et al., 2023), the historical observations $\mathbf{X} \in \mathbb{R}^{L\times C}$ and future observations $\mathbf{Y} \in \mathbb{R}^{H\times C}$ are divided into non-overlapping patches of length T. The i-th historical patch and the j-th future patch are defined as:

$$\mathbf{P}_i^x = \mathbf{X}_{(i-1)T+1:iT,:}, \qquad \mathbf{P}_j^y = \mathbf{Y}_{(j-1)T+1:jT,:}, \tag{7}$$

where $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$, $N_x = L/T$, and $N_y = H/T$, assuming that T divides both L and H. In this way, both $\mathbf{P}_i^x$ and $\mathbf{P}_j^y$ lie in $\mathbb{R}^{T\times C}$. Flow Matching is then performed at the patch level, with historical patches serving as source states and future patches as target states. Instead of drawing source states from a generic Gaussian distribution, a historical patch is randomly selected for each future patch, providing a data-dependent initialization related to the observed dynamics. Specifically, for each future patch j, we independently sample a historical patch index

$$i_j \sim \mathrm{Uniform}\left(\{1, \ldots, N_x\}\right), \qquad j = 1, \ldots, N_y, \tag{8}$$

and construct the source–target pair $\left(\mathbf{P}_{i_j}^x, \mathbf{P}_j^y\right)$.

Historical Context Encoding. To capture cross-channel dependencies in the historical observations, we employ an iTransformer-style variate encoder (Liu et al., 2024). Given a historical window $\mathbf{X} \in \mathbb{R}^{L\times C}$, we project the lookback sequence of each channel into a variate token:

$$\mathbf{g}_c = \mathbf{W}_h \mathbf{X}_{:,c} + \mathbf{e}_c, \qquad c = 1, \cdots, C, \tag{9}$$

where $\mathbf{g}_c \in \mathbb{R}^{d_h}$, $\mathbf{W}_h \in \mathbb{R}^{d_h \times L}$ is a learnable projection matrix, $\mathbf{e}_c \in \mathbb{R}^{d_h}$ is a learnable channel embedding, and d_h is the token dimension. The variate tokens are then processed jointly using self-attention across the channel dimension:

$$\mathbf{G} = \mathrm{Transformer}_{\mathrm{var}}\left([\mathbf{g}_1, \ldots, \mathbf{g}_C]^\top\right), \qquad \mathbf{G} \in \mathbb{R}^{C \times d_h}, \tag{10}$$

where $\mathrm{Transformer}_{\mathrm{var}}$ denotes a Transformer encoder (Vaswani et al., 2017) which applies self-attention across channel tokens, and $\mathbf{G}$ collects the resulting channel-specific historical representations used to condition the flow velocity field.

Conditional Flow Matching. To train the velocity field in Eq. (1), we construct a stochastic patch-state interpolant for each sampled source–target patch pair $(\mathbf{P}^x_{i_j}, \mathbf{P}^y_j)$. Specifically, we adopt the linear interpolation used in rectified flow (Liu et al., 2022b) and incorporate additive Gaussian noise to obtain:

$$\mathbf{Z}_{\tau,j} = (1-\tau)\mathbf{P}^x_{i_j} + \tau \mathbf{P}^y_j + \sigma(\tau)\boldsymbol{\epsilon}_j, \quad \tau \sim \mathrm{Uniform}(0,1), \quad j = 1, \cdots, N_y \tag{11}$$

where $\boldsymbol{\epsilon}_j \in \mathbb{R}^{T \times C}$ represents independent standard Gaussian noise, and $\sigma : [0,1] \to \mathbb{R}_{\geq 0}$ denotes a differentiable time-dependent noise schedule. The corresponding pathwise velocity target is

$$\mathbf{W}_{\tau,j} = \frac{\mathrm{d}\mathbf{Z}_{\tau,j}}{\mathrm{d}\tau} = \mathbf{P}^y_j - \mathbf{P}^x_{i_j} + \sigma'(\tau)\boldsymbol{\epsilon}_j. \tag{12}$$

Following Conditional Flow Matching (CFM) (Lipman et al., 2023), we parameterize the conditional velocity field $\mathbf{v}_\theta$ using a Transformer (Vaswani et al., 2017) and train it to predict the corresponding velocity targets:

$$\theta^\star = \arg\min_\theta \mathbb{E}\left[\frac{1}{N_y}\sum_{j=1}^{N_y} \left\|\mathbf{v}_\theta\left(\mathbf{Z}_{\tau,j}, \tau \mid \mathbf{G}\right) - \mathbf{W}_{\tau,j}\right\|_F^2\right]. \tag{13}$$

where $\|\cdot\|_F$ denotes the Frobenius norm. The velocity field $\mathbf{v}_\theta$ takes the state $\mathbf{Z}_{\tau,j}$, flow time τ, and cross-channel historical context $\mathbf{G}$ as inputs. The parameter vector θ collectively represents the parameters of all trainable modules, including the Transformer-based velocity network and the historical context encoder. After integrating the patch-wise dynamics to $\tau = 1$, the generated future patches are concatenated along the temporal dimension to obtain the complete base forecast:

$$\widehat{\mathbf{Y}}_{\mathrm{base}} = \mathrm{Concat}\left(\mathbf{Z}_{1,1}, \ldots, \mathbf{Z}_{1,N_y}\right) \in \mathbb{R}^{H \times C}. \tag{14}$$

4.2 Optimal-Control-Inspired Calibration

The CFM objective in Eq. (13) regresses the velocity field against pathwise targets at sampled points along the prescribed probability path, but the quality of the forecast is determined by the terminal state. To bridge this gap, we freeze the pretrained field and train an additive controller, which can improve the alignment of generated forecasts with observed future trajectories.

Calibration Problem Formulation. Using the frozen velocity field $\mathbf{v}_{\theta^\star}$ trained with the CFM objective in Eq. (13), we generate an ensemble of K future trajectories by integrating the flow from K initial states constructed from sampled historical patches and source noise $\sigma(0)\boldsymbol{\epsilon}^{(k)}$. For the k-th trajectory, we stack the N_y patch states into $\mathbf{Z}^{(k)}_\tau = (\mathbf{Z}^{(k)}_{\tau,1}, \ldots, \mathbf{Z}^{(k)}_{\tau,N_y})$ and introduce an additive control signal $\mathbf{U}^{(k)}_\tau$, formulating the optimization problem as

$$\mathcal{U}^\star = \arg\min_{\mathcal{U}} \mathbb{E}\left[\frac{1}{2\beta K}\int_0^1 \sum_{k=1}^{K} \left\|\mathbf{U}^{(k)}_\tau\right\|_F^2 \mathrm{d}\tau + \Phi\left(\widehat{\mathcal{Y}}, \mathbf{Y}\right)\right], \tag{15a}$$

$$\text{s.t.} \qquad \frac{\mathrm{d}\mathbf{Z}^{(k)}_\tau}{\mathrm{d}\tau} = \mathbf{v}_{\theta^\star}\left(\mathbf{Z}^{(k)}_\tau, \tau \mid \mathbf{G}\right) + \mathbf{U}^{(k)}_\tau, \qquad k = 1, \ldots, K, \tag{15b}$$

$$\widehat{\mathcal{Y}} = \left\{\widehat{\mathbf{Y}}^{(k)}\right\}_{k=1}^K, \qquad \widehat{\mathbf{Y}}^{(k)} = \mathrm{Concat}\left(\mathbf{Z}^{(k)}_{1,1}, \ldots, \mathbf{Z}^{(k)}_{1,N_y}\right) \in \mathbb{R}^{H \times C}. \tag{15c}$$

Here, $\mathcal{U} = \{\mathbf{U}^{(k)}_\tau\}_{k=1}^K$ denotes the control signals of all trajectories. The first term in Eq. (15a) penalizes control energy, with a smaller β keeping the controlled dynamics closer to the pretrained flow. The second term is the terminal cost, which assesses the quality of the predictive distribution. Eq. (15b) states that the transport is governed by the frozen field $\mathbf{v}_{\theta^\star}$ together with the added control. Eq. (15c) concatenates the terminal patch state of each sample into a complete forecast. A detailed notation table is provided in Appendix C.

Terminal Cost. To assess forecast accuracy and ensemble spread with a single objective, we define the terminal cost as the empirical Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007), averaged over the horizon H and channel C:

$$\Phi\left(\widehat{\mathcal{Y}}, \mathbf{Y}\right) = \frac{1}{KHC} \sum_{k=1}^{K} \left\| \widehat{\mathbf{Y}}^{(k)} - \mathbf{Y} \right\|_1 - \frac{1}{2K^2HC} \sum_{k=1}^{K} \sum_{l=1}^{K} \left\| \widehat{\mathbf{Y}}^{(k)} - \widehat{\mathbf{Y}}^{(l)} \right\|_1, \tag{16}$$

where $\|\cdot\|_1$ denotes the entrywise ℓ_1 norm. The first term measures the average discrepancy between the generated forecasts and the observed future, while the second rewards diversity among generated forecasts through pairwise distances. Accordingly, minimizing the expected CRPS balances forecast accuracy and ensemble dispersion.

Analytical Solution. As the running cost in Eq. (15a) is quadratic in the control signal $\mathbf{U}_\tau^{(k)}$, the formulation in Eq. (6) can be directly applied. Since the terminal cost couples all K trajectories, the cost-to-go is a function of the joint ensemble state $\mathcal{Z}_\tau = (\mathbf{Z}_\tau^{(1)}, \ldots, \mathbf{Z}_\tau^{(K)})$, rather than of any individual trajectory. The resulting optimal control solution can be expressed as

$$\begin{aligned} \mathbf{U}^{\star(k)}\left(\mathcal{Z}_\tau, \tau \mid \mathbf{X}\right) &= -\beta K \, \nabla_{\mathbf{Z}^{(k)}} V\left(\mathcal{Z}_\tau, \tau \mid \mathbf{X}\right), \qquad k = 1, \ldots, K, \\ V\left(\mathcal{Z}, \tau \mid \mathbf{X}\right) &= \min_{\mathcal{U}} \mathbb{E}\left[\frac{1}{2\beta K} \int_\tau^1 \sum_{k=1}^{K} \left\| \mathbf{U}_s^{(k)} \right\|_F^2 \mathrm{d}s + \Phi\left(\widehat{\mathcal{Y}}, \mathbf{Y}\right) \,\middle|\, \mathcal{Z}_\tau = \mathcal{Z},\ \mathbf{X} \right]. \end{aligned} \tag{17}$$

where $\mathbf{U}^{\star(k)}$ denotes the optimal additive control signal for the k-th trajectory.

Gradient Matching. Eq. (17) provides an analytic solution but is not directly usable: evaluating the value gradient at a given state requires solving the remaining control problem on $[\tau, 1]$, which is intractable for a learned velocity field and a non-quadratic terminal cost. Therefore, we train a neural controller $\mathbf{u}_\phi$ to approximate $\mathbf{U}^{\star(k)}$. Following VGG-Flow (Liu et al., 2025c), we approximate the terminal state with a single forward Euler step using the current velocity, thereby avoiding numerical integration over the interval $[\tau_n, 1]$. Let $0 = \tau_0 < \cdots < \tau_N = 1$ denote the N-step flow-time grid used for numerical integration. At τ_n, the endpoint estimate of trajectory k is

$$\widehat{\mathbf{Z}}_n^{(k)} = \mathbf{Z}_{\tau_n}^{(k)} + (1 - \tau_n)\, \mathrm{sg}\left[\mathbf{v}_{\theta^\star}\left(\mathbf{Z}_{\tau_n}^{(k)}, \tau_n \mid \mathbf{G}\right) + \mathbf{u}_\phi\left(\mathbf{Z}_{\tau_n}^{(k)}, \tau_n \mid \mathbf{X}\right) \right], \tag{18}$$

where sg denotes the stop-gradient operator. Applying the formulation in Eq. (15c) to $\{\widehat{\mathbf{Z}}_n^{(k)}\}_{k=1}^K$ gives the estimated forecast ensemble $\widehat{\mathcal{Y}}_n$. Following the form of the ideal control law in Eq. (17), we define the target for the control signal by differentiating the terminal cost with respect to the estimated endpoint:

$$\mathbf{q}_n^{(k)} = -\beta K \, \eta(\tau_n) \, \nabla_{\widehat{\mathbf{Z}}_n^{(k)}} \Phi\left(\widehat{\mathcal{Y}}_n, \mathbf{Y}\right), \qquad \eta(\tau_n) \geq 0. \tag{19}$$

Since the velocity is not constant over $[\tau_n, 1]$, the frozen-velocity extrapolation is only a local approximation. Under smoothness assumptions, the endpoint error decreases as τ_n approaches 1. We set the guidance schedule to $\eta(\tau) = \tau^2$, which downweights targets constructed at early flow times. Then, the controller is regressed onto the detached targets:

$$\mathcal{L}_{\mathrm{match}}(\phi) = \mathbb{E}\left[\frac{1}{NK} \sum_{n=0}^{N-1} \sum_{k=1}^{K} \left\| \mathbf{u}_\phi\left(\mathrm{sg}\left[\mathbf{Z}_{\tau_n}^{(k)}\right], \tau_n \mid \mathbf{X}\right) - \mathrm{sg}\left[\mathbf{q}_n^{(k)}\right] \right\|_F^2 \right]. \tag{20}$$

The resulting controller parameters are denoted by $\phi^\star$. Unlike the full VGG-Flow formulation, we do not introduce a separate value-gradient network or additional consistency objectives. Instead, with the base flow frozen, the controller refines the pretrained transport using only the endpoint-derived target, keeping the calibration stage lightweight. Eq. (20) therefore provides a tractable gradient-matching surrogate to Eq. (15), rather than an exact solution of the HJB equation. A detailed comparison with VGG-Flow is provided in Appendix D. At inference time, the calibrated model generates forecasts by integrating

$$\frac{\mathrm{d}\mathbf{Z}_\tau^{(k)}}{\mathrm{d}\tau} = \mathbf{v}_{\theta^\star}\left(\mathbf{Z}_\tau^{(k)}, \tau \mid \mathbf{G}\right) + \mathbf{u}_{\phi^\star}\left(\mathbf{Z}_\tau^{(k)}, \tau \mid \mathbf{X}\right), \qquad k = 1, \ldots, K, \tag{21}$$

where each trajectory uses only its own state and the observed history, without access to future or the other trajectories.

4.3 Training and Inference

Training has two stages. First, the history-conditioned base field is optimized with Eq. (13). Second, we freeze the base model and optimize the controller using Eq. (20). At inference time, each initial patch state $\mathbf{Z}_0$ is constructed from a sampled historical patch and the source noise. The resulting states are then transported by the combined field $v_{\theta^\star} + u_\phi$ following the dynamics in Eq. (21).

5 Experiments

5.1 Experimental setting

Dataset. We conduct experiments on six real-world multivariate time series datasets covering diverse temporal patterns and application domains: ETTm1, ETTm2, Weather, Solar-Energy, Electricity, and Traffic. Detailed descriptions of these datasets are provided in Appendix E.

Baselines. We select representative, strong-performing methods from different years as baseline methods. Specifically, the deterministic baselines include PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), TimeFilter (Hu et al., 2025), and TimeMosaic (Ding et al., 2026), while the probabilistic baselines include CSDI (Tashiro et al., 2021), TimeDiff (Shen & Kwok, 2023), D3U (Li et al., 2025), and PPM (Li et al., 2026). We also report results for FlowOC (base), which corresponds to the history-informed Flow Matching model trained in the first stage.

Metrics. We evaluate point forecasting accuracy using Mean Squared Error (MSE) and Mean Absolute Error (MAE), and assess probabilistic forecasting quality using the Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007). Lower values indicate better performance for all three metrics. A detailed description of these three metrics is in Appendix F.

Implementation Details. All experiments are implemented in PyTorch (Paszke et al., 2019) and trained on two NVIDIA H200 GPUs. The lookback length L and prediction horizon H are set to 96 and 192, respectively, following the settings in Li et al. (2025; 2026). All hyperparameters are shared across the six datasets. We use $N = 4$ Euler discretization steps and set the patch size T to 24. During inference, 100 samples are generated to approximate the predictive distribution. The time-dependent noise scale is defined by a cosine function: $\sigma(\tau) = 0.75 \cos\left(\frac{\pi}{2}\tau\right)$. Further implementation details are provided in Appendix G.

5.2 Results

The point forecasting performance of both deterministic and probabilistic models is presented in Table 1. As shown in the table, FlowOC achieves the best overall performance among probabilistic forecasting models, obtaining the lowest MSE and MAE on most datasets. Compared with state-of-the-art probabilistic forecasting approaches, FlowOC consistently improves forecasting accuracy. In particular, compared with D3U, FlowOC reduces the MSE by 14.41% on Weather and the MSE by 10.47% on Traffic, demonstrating improvements in point forecasting performance. Although deterministic models are specifically optimized for point prediction, FlowOC remains highly competitive. In particular, FlowOC achieves comparable or superior performance on ETTm1, ETTm2, Weather, and Electricity datasets. The probabilistic forecasting performance evaluated by CRPS is presented in Table 2. Compared with D3U, the proposed method reduces CRPS by 15.25% on the Traffic dataset and 12.86% on the Weather dataset, indicating that FlowOC learns predictive distributions that are better aligned with the underlying future outcomes. As shown in Tables 1 and 2, the base Flow Matching model already achieves competitive performance, particularly on the ETTm2 dataset. Incorporating the proposed calibration mechanism consistently improves both point forecasting accuracy and probabilistic forecasting quality across all evaluated datasets. Specifically, the calibrated model reduces MSE on ETTm1 by 3.74%, while achieving a 4.01% reduction in CRPS on the Weather dataset. As shown in Figure 2, the optimal-control-inspired calibration mechanism drives the generated trajectories closer to the future observations. These improvements indicate that the proposed calibration mechanism effectively refines the predictive distribution, resulting in forecasts that are better aligned with the observed data distribution. Extra visualization results are presented in Appendix K.

Table 1: Multivariate forecasting results (MSE and MAE). The best and second-best results are highlighted in **bold red** and blue underline, respectively.

Type	Method	ETTm1		ETTm2		Weather		Solar-Energy		Electricity		Traffic	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
Deterministic Model	PatchTST (Nie et al., 2023)	0.390	0.390	0.252	0.313	0.233	0.260	0.269	0.328	0.200	0.307	0.464	0.314
	iTransformer (Liu et al., 2024)	0.377	0.391	0.250	0.309	0.221	0.254	0.233	0.261	0.164	0.255	0.418	0.284
	TimeFilter (Hu et al., 2025)	0.356	0.380	0.235	0.299	0.202	0.246	0.226	0.249	0.154	0.248	**0.395**	0.262
	TimeMosaic (Ding et al., 2026)	0.364	**0.368**	0.236	0.291	0.215	0.243	0.238	0.272	0.174	0.269	0.448	0.279
Probabilistic Model	CSDI (Tashiro et al., 2021)	0.867	0.690	1.291	0.576	0.842	0.523	0.848	0.818	0.553	0.795	0.921	0.678
	TimeDiff (Shen & Kwok, 2023)	0.796	0.577	0.284	0.342	0.277	0.331	1.169	0.936	0.730	0.690	1.465	0.851
	D3U (Li et al., 2025)	0.365	0.389	0.241	0.302	0.229	0.276	0.237	0.270	0.177	0.265	0.468	0.299
	PPM (Li et al., 2026)	0.386	0.393	0.246	0.306	0.238	0.269	0.264	0.283	0.182	0.267	0.486	0.316
	FlowOC (base)	0.362	0.381	0.230	0.289	0.199	0.239	0.232	0.249	0.155	0.246	0.424	0.270
	FlowOC (with controller)	**0.349**	0.374	**0.227**	**0.286**	**0.196**	**0.231**	**0.224**	**0.241**	**0.151**	**0.241**	0.419	**0.260**

Table 2: Probabilistic forecasting results measured by CRPS. The best and second-best results are highlighted in **bold red** and blue underline, respectively.

Method	ETTm1	ETTm2	Weather	Solar-Energy	Electricity	Traffic
CSDI (Tashiro et al., 2021)	0.773	0.625	0.508	0.649	0.465	0.612
TimeDiff (Shen & Kwok, 2023)	0.454	0.316	0.293	0.900	0.475	0.671
D3U (Li et al., 2025)	0.285	0.243	0.210	0.186	0.202	0.236
PPM (Li et al., 2026)	0.316	0.236	0.215	0.202	0.208	0.243
FlowOC (base)	0.285	0.227	0.191	0.188	0.184	0.205
FlowOC (with controller)	**0.279**	**0.224**	**0.183**	**0.177**	**0.183**	**0.200**

6 Ablation Studies

6.1 Effectiveness of the Source Distribution

In this section, we conduct an ablation study to investigate the contribution of historical patches as source states. Instead of sampling from historical patches as $\left(\mathbf{P}^x_{i_j}, \mathbf{P}^y_j\right)$, we replace $\mathbf{P}^x_{i_j}$ with samples drawn from a standard Gaussian distribution. The results are reported in Table 3. Although the Gaussian initialization achieves slightly better MSE on ETTm1, the historical-patch initialization obtains better MAE and CRPS, suggesting that it produces predictive distributions with better calibrated centers and shapes. Overall, these results demonstrate that a data-dependent source distribution can facilitate more effective transport toward the future distribution compared with an unstructured Gaussian source. Extra results can be found in Appendix H.

Table 3: Ablation study on the choice of the source distribution.

	ETTm1			Weather			Traffic		
	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
Historical patches	0.3622	**0.3808**	**0.2847**	**0.1991**	**0.2388**	**0.1908**	**0.4239**	**0.2699**	**0.2048**
Standard Gaussian	**0.3554**	0.3859	0.2909	0.2002	0.2416	0.1926	0.4437	0.2738	0.2128

6.2 Effectiveness of Optimal Control Calibration

The ablation study evaluates whether trajectory-level, optimal control-inspired calibration provides benefits over simpler or more direct ways of injecting the forecast score into the forecaster. We compare FlowOC with two alternatives. The first, Joint CFM–CRPS Training (Joint-CRPS), directly adds the objective in Eq. (16) to the Flow Matching loss, eliminating the separate post-training stage. The second, Direct Rollout Calibration (Direct), retains the two-stage structure and uses the same frozen base flow and controller architecture as FlowOC. Instead of the value-gradient matching rule in Eq. (20), it optimizes the terminal CRPS loss by backpropagating through the entire rollout. To ensure fair comparison, Direct and FlowOC use controllers with the same number of parameters. As shown in Table 4, only FlowOC improves upon FlowOC (base) across all datasets and metrics.

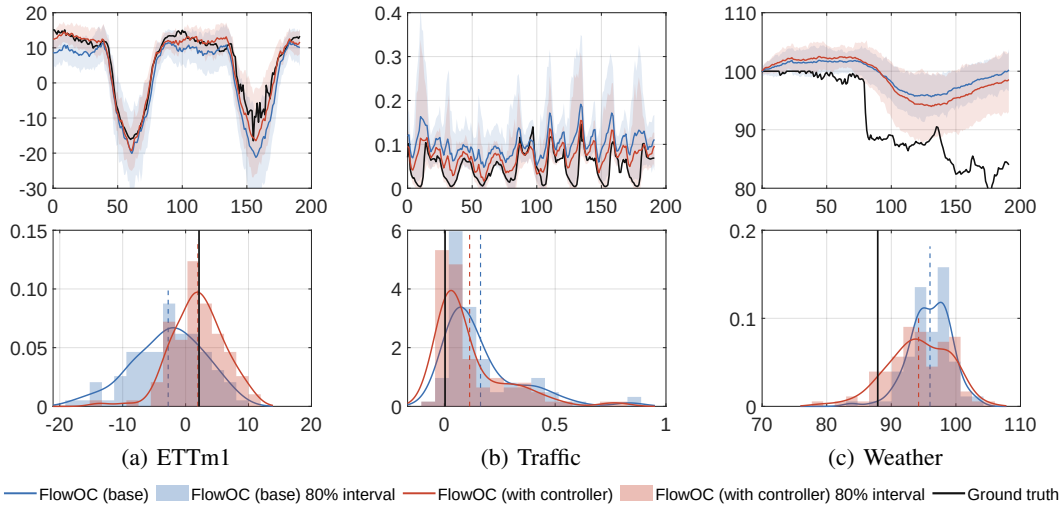


Figure 2: Comparison between the base model and the controlled model on ETTm1, Traffic, and Weather. The top panel shows the predictive trajectories and their uncertainty, and the bottom panel presents the distribution of predictive samples at a representative forecast step.

The other two strategies are competitive in selected settings, but do not consistently outperform the base model. Detailed results can be found in Appendix I. To evaluate robustness, we repeat the four-method comparison using three training seeds on ETTm1, Electricity, and Traffic. As reported in Appendix J, FlowOC achieves the lowest mean MSE and CRPS on all three datasets.

Table 4: Ablation results for the calibration strategy. Each entry is the relative change against the frozen Base, $(m_{\text{Base}} - m_{\text{strategy}})/m_{\text{Base}} \times 100\%$; ↑ marks an improvement and ↓ a degradation.

Metric	Strategy	ETTm1	ETTm2	Weather	Solar-Energy	Electricity	Traffic	Improved
MSE	Joint-CRPS	↑2.69%	↓1.25%	↓1.94%	↑2.56%	↑0.63%	↓0.98%	3/6
	Direct	↓0.68%	↓0.29%	↑0.20%	↓0.44%	↑1.94%	↑0.85%	3/6
	FlowOC	↑3.74%	↑1.29%	↑1.65%	↑3.09%	↑2.06%	↑1.26%	**6/6**
MAE	Joint-CRPS	↑0.18%	↓1.10%	↓1.24%	↓0.68%	↑0.53%	↑0.10%	3/6
	Direct	↓0.29%	↓0.22%	↑1.79%	↑1.51%	↑1.64%	↑5.26%	4/6
	FlowOC	↑1.89%	↑0.85%	↑3.24%	↑3.26%	↑1.92%	↑3.65%	**6/6**
CRPS	Joint-CRPS	↑0.41%	↓0.09%	↓0.17%	↑2.33%	↑0.88%	↑0.21%	4/6
	Direct	↓1.95%	↓1.79%	↑2.72%	↑3.71%	↓0.74%	↓0.34%	2/6
	FlowOC	↑2.05%	↑1.02%	↑4.01%	↑5.97%	↑0.87%	↑2.35%	**6/6**

7 Conclusion

In this paper, we propose FlowOC, an optimal control-inspired Flow Matching framework for multivariate probabilistic forecasting. The proposed framework first learns a conditional future distribution through a history-informed Flow Matching formulation, which effectively exploits historical observations to construct plausible future trajectories. To further improve forecasting accuracy while preserving the learned predictive distribution, we introduce an optimal control calibration strategy that adjusts the predictive center and maintains the structure of generated trajectories through a distribution-aware objective. Extensive experiments demonstrate that FlowOC achieves competitive performance against both probabilistic and point forecasting baselines. Moreover, comprehensive ablation studies verify the effectiveness of the proposed source initialization, Flow Matching design, and optimal control calibration strategy. In future work, we plan to apply the proposed calibration to large-scale industrial systems, where calibrated probabilistic forecasts are directly applied into downstream decision-making and control.

AI USE STATEMENT

In this work, we used generative AI tools to assist with the refinement of parts of the method implementation and debugging, translation and manuscript editing, literature search and summarization, qualitative analysis of visualizations, and interpretation of experimental results. We did not use generative AI to generate the conceptual framework and research hypotheses, mathematical formulation and derivations, methodology and experimental design, generate synthetic datasets, or to clean or reformat the experimental datasets. All final research decisions were made by the authors. Mathematical claims and derivations were independently checked, AI-assisted literature suggestions were verified against the original sources, and AI-assisted code was reviewed, tested, and validated by the authors. All reported results were produced using the authors' verified experimental pipeline. We reviewed all AI-assisted text, code, and analyses and take full responsibility for the final content of this work, including its claims, implementations, results, and artifacts.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the source code, experiment configurations, and training and evaluation scripts in an anonymous repository. The README describes environment setup, dataset access, and execution instructions. The configuration files specify model architectures, preprocessing settings, training hyperparameters, and random seeds for reproducing the base forecasting and post-training experiments.

REFERENCES

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

Richard B Alley, Kerry A Emanuel, and Fuqing Zhang. Advances in weather prediction. *Science*, 363(6425):342–344, 2019.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. TimeMosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026.

Carles Domingo i Enrich, Michal Drozdzal, Brian Karrer, and Ricky TQ Chen. Adjoint matching: Fine-tuning flow and diffusion generative models with memoryless stochastic optimal control. In *International Conference on Learning Representations*, volume 2025, pp. 53791–53846, 2025.

Ying Fan and Kangwook Lee. Optimizing DDPM sampling with shortcut fine-tuning. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9623–9639. PMLR, 2023.

Tilmann Gneiting and Matthias Katzfuss. Probabilistic forecasting. *Annual Review of Statistics and Its Application*, 1(1):125–151, 2014.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24893–24911. PMLR, 2025.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. TimeBase: The power of minimalism in efficient long-term time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 26227–26246, 2025.

Ming Jin, Yu Zheng, Yuan-Fang Li, Siheng Chen, Bin Yang, and Shirui Pan. Multivariate time series forecasting with dynamic graph neural odes. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):9168–9180, 2023.

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.

Jinglin Li, Jun Tan, QI Fang, and Ning Gui. Parametric prior mapping framework for non-stationary probabilistic time series forecasting. *International Conference on Machine Learning*, 2026.

Qi Li, Zhenyu Zhang, Lei Yao, Zhaoxia Li, Tianyi Zhong, and Yong Zhang. Diffusion-based decoupled deterministic and uncertain framework for probabilistic multivariate time series forecasting. In *International Conference on Learning Representations*, volume 2025, pp. 67475–67503, 2025.

Daniel Liberzon. *Calculus of Variations and Optimal Control Theory: A Concise Introduction*. Princeton University Press, 2012.

Soon Hoe Lim, Yijin Wang, Annan Yu, Emma Hart, Michael W Mahoney, Xiaoye S Li, and N Benjamin Erichson. Elucidating the design choice of probability paths in Flow Matching for forecasting. *Transactions on Machine Learning Research*, 2025.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Feiyu Zhao, Ruichao Mo, and Haotong Zhang. SegRNN: Segment recurrent neural network for long-term time series forecasting. *IEEE Internet of Things Journal*, 13(5):9861–9871, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow Matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Yaron Lipman, Marton Havasi, Peter Holderrieth, Neta Shaul, Matt Le, Brian Karrer, Ricky TQ Chen, David Lopez-Paz, Heli Ben-Hamu, and Itai Gat. Flow Matching guide and code. *arXiv preprint arXiv:2412.06264*, 2024.

Minhao Liu, Ailing Zeng, Muxi Chen, Zhijian Xu, Qiuxia Lai, Lingna Ma, and Qiang Xu. SCINet: Time series modeling and forecasting with sample convolution and interaction. *Advances in Neural Information Processing Systems*, 35:5816–5828, 2022a.

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. PMLR, 2025a.

Qinglong Liu, Cong Xu, Wenhao Jiang, Kaixuan Wang, Lin Ma, and Haifeng Li. TimeStacker: A novel framework with multilevel observation for capturing nonstationary patterns in time series forecasting. In *International Conference on Machine Learning*, volume 267, pp. 39929–39946, 2025b.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 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*, volume 2024, pp. 11116–11140, 2024.

Zhen Liu, Tim Xiao, Carles Domingo i Enrich, Weiyang Liu, and Dinghuai Zhang. Value gradient guidance for Flow Matching alignment. *Advances in Neural Information Processing Systems*, 38: 11372–11404, 2025c.

Ping Ma, Shuhui Cui, Mingshuai Chen, Shengzhe Zhou, and Kai Wang. Review of family-level short-term load forecasting and its application in household energy management system. *Energies*, 16(15):5809, 2023.

Nam Nguyen and Brian Quanz. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 9117–9125, 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*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. PyTorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.

Lev Semenovich Pontryagin. *Mathematical theory of optimal processes*. Routledge, 2018.

Yao Qin, Dongjin Song, Haifeng Chen, Wei Cheng, Guofei Jiang, and Garrison Cottrell. A dual-stage attention-based recurrent neural network for time series prediction. *arXiv preprint arXiv:1704.02971*, 2017.

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*, pp. 8857–8868. PMLR, 2021.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10674–10685. ieee, 2022.

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.

Maryam Shaygan, Collin Meese, Wanxin Li, Xiaoliang George Zhao, and Mark Nejad. Traffic prediction using artificial intelligence: Review of recent advances and emerging opportunities. *Transportation Research Part C: Emerging Technologies*, 145:103921, 2022.

Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. In *International Conference on Machine Learning*, pp. 31016–31029. PMLR, 2023.

Chen Su, Zhengzhou Cai, Yuanhe Tian, Zhuochao Chang, Zihong Zheng, and Yan Song. Diffusion models for time series forecasting: A survey. *arXiv preprint arXiv:2507.14507*, 2025.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: Conditional score-based diffusion models for probabilistic time series imputation. *Advances in Neural Information Processing Systems*, 34:24804–24816, 2021.

Guangpo Tian, Yunyang Xu, Xiang Ma, Xuemei Li, and Caiming Zhang. MDWConv: CNN based on multi-scale atrous pyramid and depthwise separable convolution for long time series forecasting. *Neural Networks*, 185:107139, 2025.

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

Daoyu Wang, Mingyue Cheng, Zhiding Liu, and Qi Liu. TimeDART: A diffusion autoregressive transformer for self-supervised time series representation. In *International Conference on Machine Learning*, volume 267, pp. 62627–62651, 2025.

Yuxuan Wang, Haixu Wu, Jiaxiang Dong, Guo Qin, Haoran Zhang, Yong Liu, Yunzhong Qiu, Jianmin Wang, and Mingsheng Long. TimeXer: Empowering transformers for time series forecasting with exogenous variables. *Advances in Neural Information Processing Systems*, 37:469–498, 2024.

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, pp. 67562–67583, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 11121–11128, 2023.

A Multivariate Time Series Forecasting

Recent advances in multivariate time series forecasting have explored a wide range of architectures to improve the modeling of temporal dynamics and cross-variable dependencies. Early approaches primarily relied on statistical models, such as ARIMA (Box et al., 2015), which model temporal dependencies based on predefined statistical assumptions. With the development of deep learning, neural forecasting approaches have attracted increasing attention due to their ability to capture complex temporal patterns directly from historical observations. Lightweight feed-forward models (Zeng et al., 2023; Huang et al., 2025) have demonstrated that simple yet efficient architectures can achieve competitive forecasting performance. Convolutional neural networks (Liu et al., 2022a; Tian et al., 2025) and recurrent neural networks (Qin et al., 2017; Lin et al., 2026) have also been explored to model temporal dependencies in time series data. More recently, Transformer-based architectures (Vaswani et al., 2017) have been widely adopted due to their ability to capture long-range dependencies and process large-scale temporal information. PatchTST (Nie et al., 2023) introduces patching to transform historical time series segments into tokens, enabling efficient modeling of temporal dependencies while reducing the computational burden associated with long sequences. TimeXer (Wang et al., 2024) further incorporates exogenous variables through dedicated embeddings, allowing additional contextual information to be integrated into the forecasting process. TimeBridge (Liu et al., 2025a) addresses the interaction between non-stationarity and dependency modeling through a patch-based Transformer architecture. Overall, these methods primarily focus on deterministic forecasting, aiming to minimize the discrepancy between a single predicted trajectory and the ground-truth future trajectory. This line of research has established strong foundations for forecasting accuracy and motivates subsequent efforts toward probabilistic forecasting.

B Hamilton–Jacobi–Bellman Derivation

This section derives the optimal feedback law used in Eq. (6). Consider the controlled continuous-time dynamics

$$\frac{\mathrm{d}\mathbf{z}_\tau}{\mathrm{d}\tau} = f(\mathbf{z}_\tau, \tau) + \mathbf{u}_\tau, \qquad \tau \in [0, 1], \tag{22}$$

where f denotes the uncontrolled dynamics and $\mathbf{u}_\tau$ is an additive control signal. Given a running cost ℓ and a terminal cost Φ, the optimal control is defined as

$$\mathbf{u}^\star = \arg\min_{\mathbf{u}} \left[\int_0^1 \ell(\mathbf{z}_\tau, \mathbf{u}_\tau, \tau)\, \mathrm{d}\tau + \Phi(\mathbf{z}_1) \right]. \tag{23}$$

Value function. For an intermediate state $\mathbf{z}$ at time τ, the optimal cost-to-go can be formulated as:

$$V(\mathbf{z}, \tau) = \min_{\mathbf{u}} \left[\int_\tau^1 \ell(\mathbf{z}_s, \mathbf{u}_s, s)\, \mathrm{d}s + \Phi(\mathbf{z}_1) \,\middle|\, \mathbf{z}_\tau = \mathbf{z} \right], \qquad V(\mathbf{z}, 1) = \Phi(\mathbf{z}). \tag{24}$$

By the dynamic programming principle and a first-order expansion over an infinitesimal time interval, V satisfies the Hamilton–Jacobi–Bellman equation

$$\partial_\tau V(\mathbf{z}, \tau) + \min_{\mathbf{u}} \left\{ \ell(\mathbf{z}, \mathbf{u}, \tau) + \nabla_{\mathbf{z}} V(\mathbf{z}, \tau)^\top \left[f(\mathbf{z}, \tau) + \mathbf{u} \right] \right\} = 0. \tag{25}$$

Quadratic control cost. We use the quadratic running cost

$$\ell(\mathbf{z}, \mathbf{u}, \tau) = \frac{1}{2\beta} \|\mathbf{u}\|_2^2, \qquad \beta > 0. \tag{26}$$

The control-dependent part of Eq. (25) is strictly convex in $\mathbf{u}$, which can formulate the control part in Eq. (25) as:

$$\nabla_{\mathbf{u}} \left(\frac{1}{2\beta} \|\mathbf{u}\|_2^2 + \nabla_{\mathbf{z}} V(\mathbf{z}, \tau)^\top \mathbf{u} \right) = \frac{1}{\beta} \mathbf{u} + \nabla_{\mathbf{z}} V(\mathbf{z}, \tau) = 0. \tag{27}$$

Solving this condition gives

$$\mathbf{u}^\star(\mathbf{z}, \tau) = -\beta \nabla_{\mathbf{z}} V(\mathbf{z}, \tau). \tag{28}$$

Since the control-dependent objective is strictly convex in $\mathbf{u}$, this stationary point is the unique global minimizer and therefore gives the optimal feedback control.

C CONTROLLED SYSTEM IN PATCH SPACE

The notation used in the patch-space control formulation is summarized in Table 5. The batch dimension is omitted for clarity.

Table 5: Notation for the patch-space ensemble control formulation.

Symbol	Shape	Description
$\mathbf{Z}_{\tau,j}^{(k)}$	$\mathbb{R}^{T\times C}$	State of the j-th future patch in the k-th trajectory at flow time τ.
$\mathbf{Z}_{\tau}^{(k)}$	$\mathbb{R}^{N_y\times T\times C}$	Stacked future-patch state of the k-th trajectory, containing all N_y future patches.
$\mathbf{U}_{\tau,j}^{(k)}$	$\mathbb{R}^{T\times C}$	Ideal control applied to the j-th patch of the k-th trajectory.
$\mathbf{U}_{\tau}^{(k)}$	$\mathbb{R}^{N_y\times T\times C}$	Stacked ideal control acting on all future patches of the k-th trajectory.
$\mathbf{v}_{\theta^\star}(\mathbf{Z}_{\tau}^{(k)}, \tau \mid \mathbf{G})$	$\mathbb{R}^{N_y\times T\times C}$	Frozen base velocity evaluated on the k-th stacked patch state.
$\mathbf{u}_{\phi}(\mathbf{Z}_{\tau}^{(k)}, \tau \mid \mathbf{X})$	$\mathbb{R}^{N_y\times T\times C}$	Parameterized controller output for the k-th trajectory.
$\widehat{\mathbf{Y}}^{(k)}$	$\mathbb{R}^{H\times C}$	Complete forecast obtained by concatenating the terminal patches of the k-th trajectory.
$\widehat{\mathcal{Y}}$	$\mathbb{R}^{K\times H\times C}$	Forecast ensemble formed by stacking the K complete forecasts.
$\mathbf{G}$	$\mathbb{R}^{C\times d_h}$	Cross-channel historical context used by the frozen velocity field.

Here, $H = N_y T$. The state and control evolve in the stacked patch space, whereas the terminal forecast cost is evaluated on the reconstructed ensemble $\widehat{\mathcal{Y}}$. For each trajectory, the complete forecast $\widehat{\mathbf{Y}}^{(k)}$ is obtained by concatenating $\{\mathbf{Z}_{1,j}^{(k)}\}_{j=1}^{N_y}$ along the temporal dimension according to Eq. (14).

D COMPARISON WITH VGG FLOW

FlowOC shares the optimal-control motivation of VGG-Flow (Liu et al., 2025c): in both methods, the pretrained velocity field defines the reference dynamics, and an additive controller is introduced to improve a terminal objective under a quadratic cost on control energy.

VGG-Flow is designed for reward alignment of pretrained generative flow models. Given a differentiable sample-level reward, it parameterizes a value-gradient field using the reward gradient evaluated at an extrapolated endpoint together with a learnable residual correction. The value-gradient model is trained with value-consistency and terminal-boundary objectives, and the generative velocity field is subsequently updated by matching its residual velocity to the learned value-gradient guidance.

FlowOC instead addresses post-training calibration for probabilistic forecasting. The objective for the designed controller is to reduce the empirical CRPS of a forecast ensemble rather than an independent reward assigned to each generated sample. Consequently, the ideal value function in Eq. (17) is defined over the sampled trajectories, and the guidance target for each trajectory depends on both the discrepancy from the observed future and the relative position within the generated ensemble.

To reduce the additional model and optimization overhead of the full VGG-Flow formulation, FlowOC does not introduce a separate value-gradient network or optimize HJB consistency and terminal-boundary losses. At each selected flow state, the current controlled velocity is extrapolated to obtain an estimated terminal state in Eq. (18), and the gradient of the CRPS at this estimated endpoint is used directly as a local surrogate for the unavailable value gradient:

$$\mathbf{q}_n^{(k)} = -\beta K\, \eta(\tau_n) \nabla_{\widehat{\mathbf{Z}}_n^{(k)}} \Phi\left(\widehat{\mathcal{Y}}_n, \mathbf{Y}\right). \tag{29}$$

The parameterized controller is then trained to match the detached targets $\mathbf{q}_n^{(k)}$ following Eq. (20), while the pretrained forecasting flow remains frozen. This design converts the value-gradient estimation problem into direct CRPS-gradient supervision and requires only the controller to be optimized during the calibration stage.

The resulting method should therefore be viewed as a lightweight, forecasting-specific gradient-matching surrogate rather than an exact implementation of VGG-Flow. Moreover, while the ideal

Table 6: Comparison between VGG-Flow and FlowOC.

Aspect	VGG-Flow	FlowOC
Primary objective	Generic reward alignment	Probabilistic forecast calibration
Terminal signal	Differentiable sample-level reward	Ensemble empirical CRPS
State used by the ideal value function	Individual generative state	Full forecast ensemble
Base-model treatment	Finetunes the velocity field relative to a frozen reference field	Freezes the base forecasting flow and trains an additive controller
Value-gradient approximation	Endpoint reward gradient plus a learned residual correction	Endpoint CRPS gradient without a learned correction field
Auxiliary objectives	Value consistency and terminal-boundary losses	No additional HJB-consistency objective
Learned component	Finetuned velocity field	Lightweight calibration controller
Inference dynamics	Finetuned velocity field	Frozen base field plus controller

control may depend on the complete ensemble state, the implemented controller operates independently on each trajectory using its current state and historical context. These approximations reduce training overhead and allow forward-only inference, but they do not provide an exact HJB solution or a global optimality guarantee.

E DATASETS

All datasets are split into training, validation, and test sets without shuffling to preserve temporal order and avoid information leakage from the future. Following the standard experimental protocols, ETTm1 and ETTm2 are divided with a ratio of approximately 6:2:2, while Weather, Solar-Energy, Electricity, and Traffic follow an approximately 7:1:2 split. The exact numbers of time points in each split are reported in Table 7.

Table 7: Summary of the datasets and forecasting settings used in our experiments. Dataset size reports the number of time points in the training, validation, and test splits, respectively.

Dataset	#Var.	Freq.	Dataset Size	Setting	Description
ETTm1	7	15 Min	(34,465, 11,521, 11,521)	$96 \rightarrow 192$	Electricity transformer temperature
ETTm2	7	15 Min	(34,465, 11,521, 11,521)	$96 \rightarrow 192$	Electricity transformer temperature
Weather	21	10 Min	(36,792, 5,271, 10,540)	$96 \rightarrow 192$	Local climatological measurements
Solar-Energy	137	10 Min	(36,697, 5,257, 10,513)	$96 \rightarrow 192$	Solar power production records
Electricity	321	1 Hour	(18,317, 2,633, 5,261)	$96 \rightarrow 192$	Electricity consumption
Traffic	862	1 Hour	(12,185, 1,757, 3,509)	$96 \rightarrow 192$	Road occupancy rates

F METRICS

We evaluate the forecasting performance using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007). MSE and MAE evaluate point forecasting accuracy, while CRPS measures the quality of the entire predictive distribution. Lower values indicate better performance for all three metrics.

Mean Squared Error (MSE). Given the ground-truth future sequence $\mathbf{Y} \in \mathbb{R}^{H \times C}$ and the point prediction $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times C}$, MSE is defined as

$$\mathrm{MSE} = \frac{1}{HC} \sum_{h=1}^{H} \sum_{c=1}^{C} \left(Y_{h,c} - \hat{Y}_{h,c} \right)^2 . \tag{30}$$

For probabilistic forecasting models, the point prediction is obtained by averaging the generated forecast samples:

$$\hat{\mathbf{Y}} = \frac{1}{S} \sum_{s=1}^{S} \hat{\mathbf{Y}}^{(s)}, \tag{31}$$

where $S = 100$ denotes the number of generated samples.

Mean Absolute Error (MAE). MAE measures the average absolute deviation between the point prediction and the ground-truth future sequence:

$$\mathrm{MAE} = \frac{1}{HC} \sum_{h=1}^{H} \sum_{c=1}^{C} \left| Y_{h,c} - \hat{Y}_{h,c} \right| . \tag{32}$$

Continuous Ranked Probability Score (CRPS). CRPS (Gneiting & Raftery, 2007) evaluates the discrepancy between the predictive distribution and the observed outcome. For a predictive cumulative distribution function F and an observation y, CRPS is defined as

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} \left(F(z) - \mathbb{I}\{y \leq z\} \right)^2 \mathrm{d}z, \tag{33}$$

where $\mathbb{I}(\cdot)$ denotes the indicator function. CRPS is a proper scoring rule and jointly evaluates the location and dispersion of the predictive distribution.

Given S forecast samples $\{\hat{y}^{(s)}\}_{s=1}^{S}$, CRPS can be estimated empirically as

$$\mathrm{CRPS} = \frac{1}{S}\sum_{s=1}^{S}\left|\hat{y}^{(s)} - y\right| - \frac{1}{2S^2}\sum_{s=1}^{S}\sum_{s'=1}^{S}\left|\hat{y}^{(s)} - \hat{y}^{(s')}\right|. \tag{34}$$

The final CRPS is averaged over all prediction steps, variables, and test instances.

G Implementation Details

The base flow and controller are trained separately using AdamW. The base flow uses an initial learning rate of 5×10^{-4} with an exponential decay of 0.97 per epoch, while the controller uses a fixed learning rate of 10^{-3}. The maximum numbers of training epochs are 24 and 10, respectively. Validation is performed every two epochs, and early stopping is applied after five validation checks without improvement. The base model is frozen throughout controller training. Both stages are selected using validation MSE computed from the ensemble mean with 20 samples. The principal hyperparameters are summarized in Table 8.

We use full-window RevIN (Kim et al., 2022) and identity patch encoding and decoding. The controller output layer is initialized to zero, and a moving-average filter with kernel size 15 is applied to its output along the forecast axis. The guidance strength in Eq. (19) is set to $\beta = 6.4 \times 10^5$ and shared across all datasets and forecasting horizons. Test metrics are computed using 100 samples.

We use the same architecture, optimization, sampling, and control hyperparameters across all datasets, without dataset-specific hyperparameter tuning.

Table 8: Hyperparameter settings of the base flow and controller. Batch size denotes the number of historical windows before sample expansion.

Hyperparameter	Base flow	Controller
Hidden dimension	128	64
Transformer layers	3	3
Attention heads	8	8
Feed-forward dimension	512	256
Variate-encoder layers	1	1
Dropout	0.1	0.1
Batch size	4	16
Training samples per window	1	4
Learning rate	5×10^{-4}	10^{-3}
Maximum epochs	24	10
Guidance strength β	–	6.4×10^5

H ABLATION ON THE SOURCE DISTRIBUTION

Figure 3 and Figure 4 compare the probabilistic forecasting results obtained with historical patch-based initialization and standard Gaussian initialization. The results show that initializing the flow from historical source patches produces forecasts with better alignment to the observed future trajectories, indicating that incorporating historical information into the source distribution provides a more informative starting point for future generation.

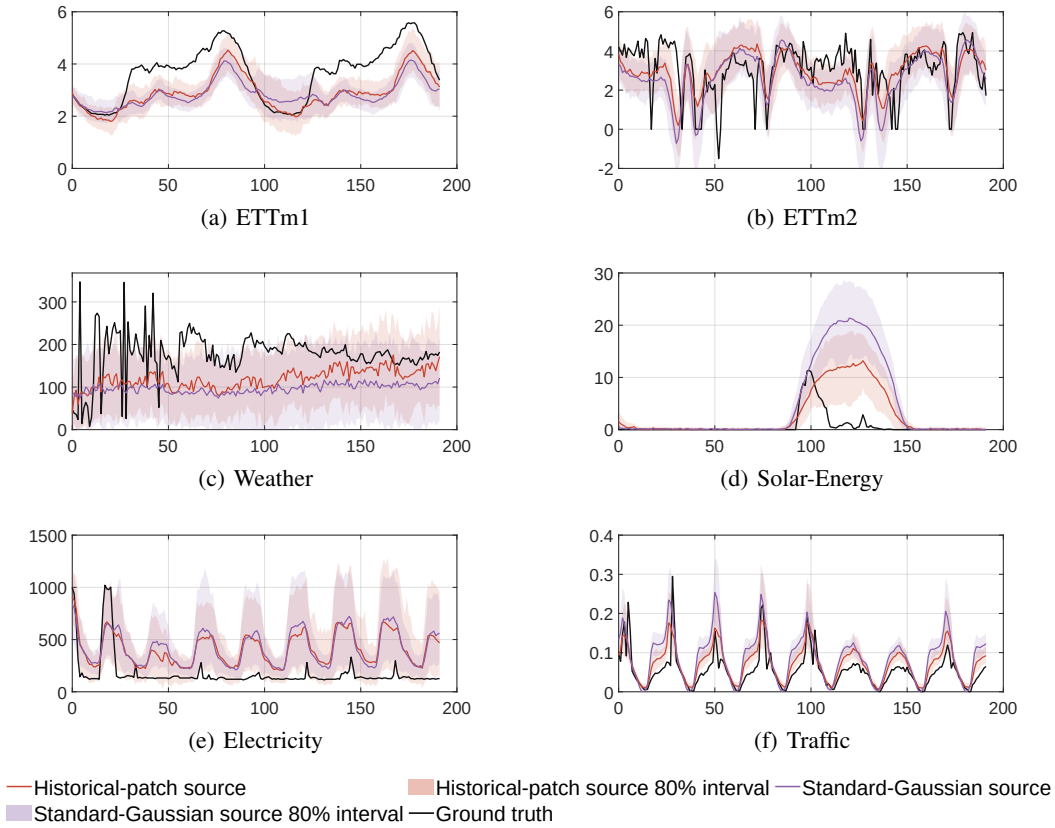


Figure 3: Predictive trajectories and the uncertainty for the historical-patch source and the standard-Gaussian source across six benchmark datasets.

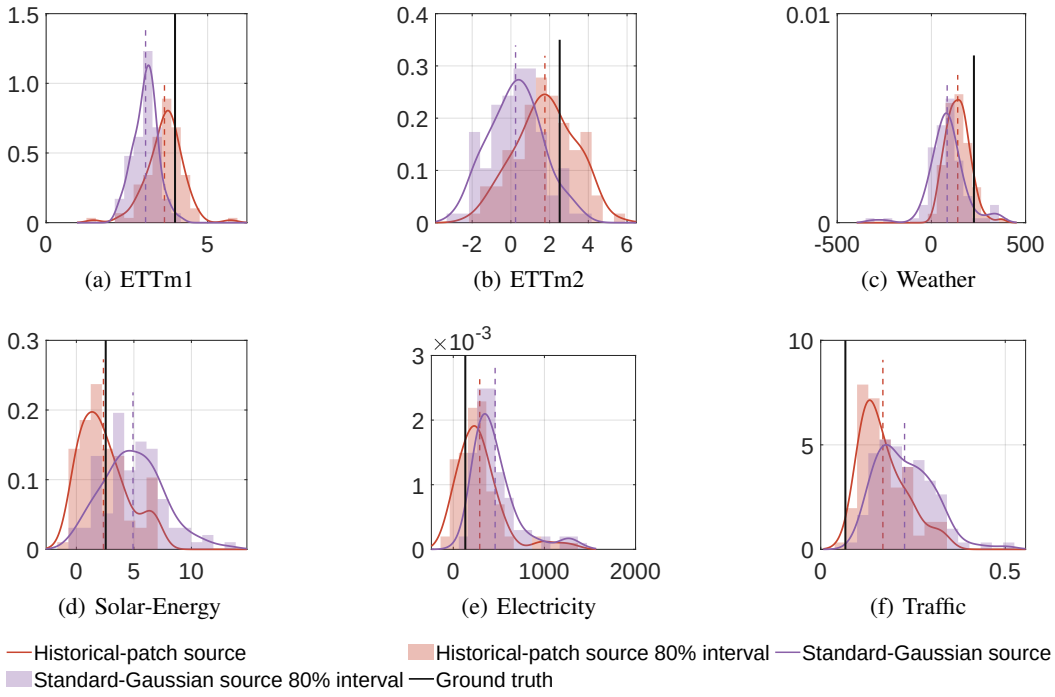


Figure 4: Distribution of predictive samples at a representative forecast step for the historical-patch source and the standard-Gaussian source across six benchmark datasets.

I ABLATION ON THE CALIBRATION METHOD

The absolute results are reported in Table 9. FlowOC achieves the best performance among the calibration strategies on most datasets and evaluation metrics. A more intuitive comparison is provided in Figure 5 and Figure 6. As shown in Figure 5, FlowOC effectively calibrates the forecasts generated by the pretrained Flow Matching model, bringing the predicted trajectories closer to the observed futures. Compared with the Direct strategy, which directly optimizes a neural controller using the forecasting loss, FlowOC preserves the diversity of the generated trajectories while improving forecasting accuracy. Figure 6 further illustrates the distributional behavior of different calibration strategies. FlowOC maintains a broader predictive distribution with lower deviation from the observed future, demonstrating its ability to improve accuracy without overly collapsing the stochastic forecasts. Other results are presented in Figure 7- Figure 10.

Table 9: Absolute results for the calibration-strategy ablation reported as relative changes in Table 4. Lower is better for every metric; the best of the four strategies is shown in **bold** for each dataset.

Dataset	Strategy	MSE	MAE	CRPS
ETTm1	Base	0.3622	0.3808	0.2847
	Joint-CRPS	0.3525	0.3801	0.2835
	Direct	0.3646	0.3819	0.2903
	FlowOC	**0.3486**	**0.3736**	**0.2789**
ETTm2	Base	0.2304	0.2885	0.2267
	Joint-CRPS	0.2332	0.2917	0.2269
	Direct	0.2310	0.2892	0.2308
	FlowOC	**0.2274**	**0.2861**	**0.2244**
Weather	Base	0.1991	0.2388	0.1908
	Joint-CRPS	0.2029	0.2417	0.1912
	Direct	0.1987	0.2345	0.1857
	FlowOC	**0.1958**	**0.2310**	**0.1832**
Solar-Energy	Base	0.2316	0.2488	0.1884
	Joint-CRPS	0.2257	0.2505	0.1840
	Direct	0.2326	0.2450	0.1814
	FlowOC	**0.2244**	**0.2407**	**0.1772**
Electricity	Base	0.1546	0.2455	0.1843
	Joint-CRPS	0.1536	0.2442	**0.1827**
	Direct	0.1516	0.2415	0.1857
	FlowOC	**0.1514**	**0.2408**	0.1827
Traffic	Base	0.4239	0.2699	0.2048
	Joint-CRPS	0.4280	0.2697	0.2044
	Direct	0.4203	**0.2557**	0.2055
	FlowOC	**0.4185**	0.2601	**0.2000**

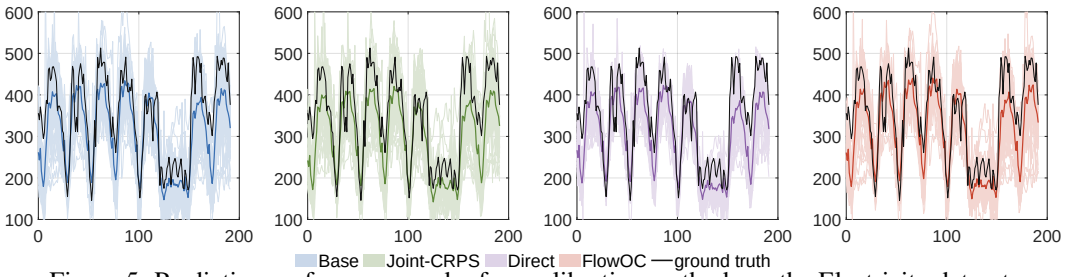


Figure 5: Prediction performance under four calibration methods on the Electricity dataset.

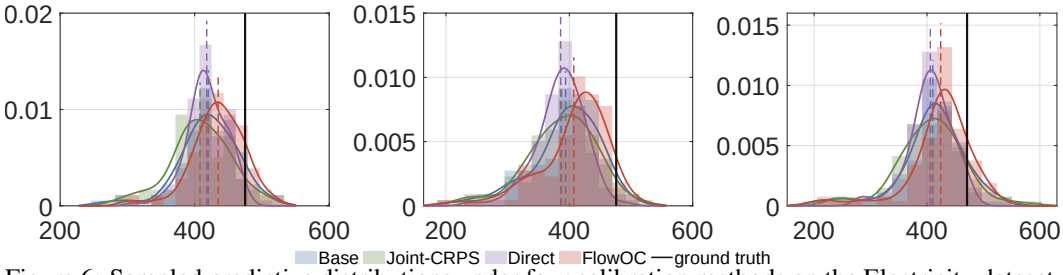


Figure 6: Sampled predictive distributions under four calibration methods on the Electricity dataset.

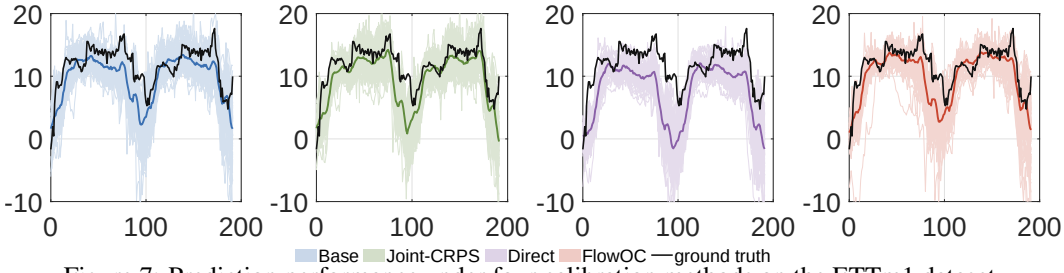


Figure 7: Prediction performance under four calibration methods on the ETTm1 dataset.

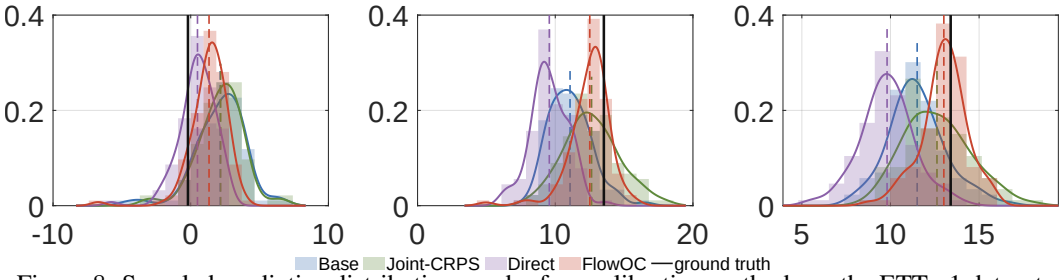


Figure 8: Sampled predictive distributions under four calibration methods on the ETTm1 dataset.

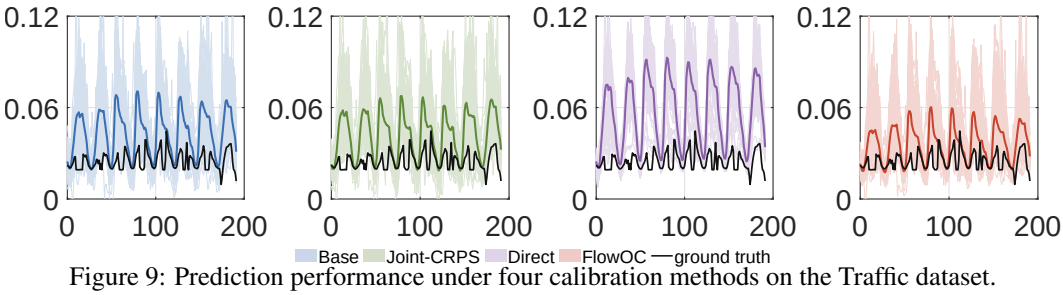


Figure 9: Prediction performance under four calibration methods on the Traffic dataset.

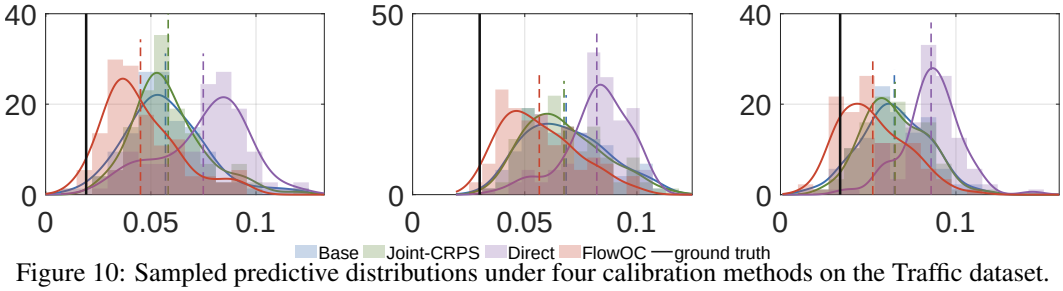


Figure 10: Sampled predictive distributions under four calibration methods on the Traffic dataset.

J ROBUSTNESS EVALUATION

Table 10 reports the comparisons over three training seeds on ETTm1, Electricity, and Traffic. FlowOC achieves the lowest mean MSE and CRPS on all three datasets, and the lowest mean MAE on ETTm1 and Electricity. Direct achieves a lower mean MAE on Traffic, but the mean CRPS exceeds that of Base on every dataset. Joint-CRPS improves all three mean scores on ETTm1 but does not retain these gains on Electricity or Traffic. Overall, these results support the consistency of the improvements from FlowOC across the evaluated datasets.

Table 10: Calibration-strategy ablation over three training seeds. Results are mean ± sample standard deviation. Bold indicates the lowest mean for each dataset and metric.

Metric	Strategy	ETTm1	Electricity	Traffic
MSE	Base	0.3677 ± 0.0130	0.1543 ± 0.0009	0.4250 ± 0.0024
	Joint-CRPS	0.3577 ± 0.0040	0.1570 ± 0.0029	0.4296 ± 0.0035
	Direct	0.3712 ± 0.0077	0.1517 ± 0.0004	0.4212 ± 0.0007
	FlowOC	$\mathbf{0.3530} \pm 0.0053$	$\mathbf{0.1512} \pm 0.0002$	$\mathbf{0.4190} \pm 0.0008$
MAE	Base	0.3863 ± 0.0076	0.2461 ± 0.0011	0.2690 ± 0.0009
	Joint-CRPS	0.3781 ± 0.0048	0.2470 ± 0.0025	0.2692 ± 0.0024
	Direct	0.3849 ± 0.0031	0.2409 ± 0.0006	$\mathbf{0.2575} \pm 0.0017$
	FlowOC	$\mathbf{0.3769} \pm 0.0037$	$\mathbf{0.2406} \pm 0.0002$	0.2599 ± 0.0005
CRPS	Base	0.2890 ± 0.0057	0.1834 ± 0.0011	0.2039 ± 0.0014
	Joint-CRPS	0.2828 ± 0.0020	0.1846 ± 0.0017	0.2045 ± 0.0010
	Direct	0.2921 ± 0.0019	0.1856 ± 0.0005	0.2060 ± 0.0005
	FlowOC	$\mathbf{0.2819} \pm 0.0029$	$\mathbf{0.1824} \pm 0.0004$	$\mathbf{0.1994} \pm 0.0005$

K COMPARISON WITH BASELINES

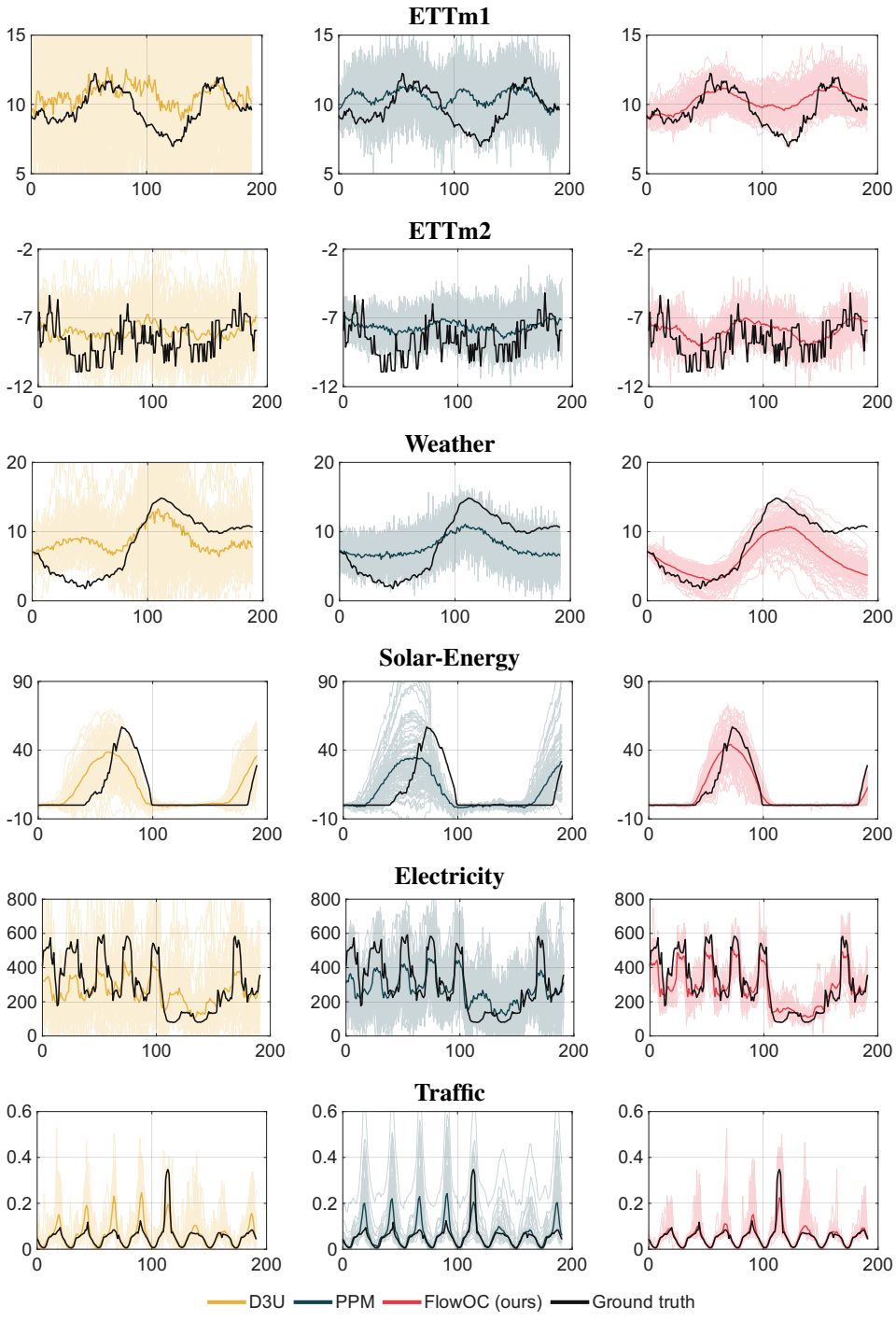


Figure 11: Comparison of the generated trajectories with the baselines.

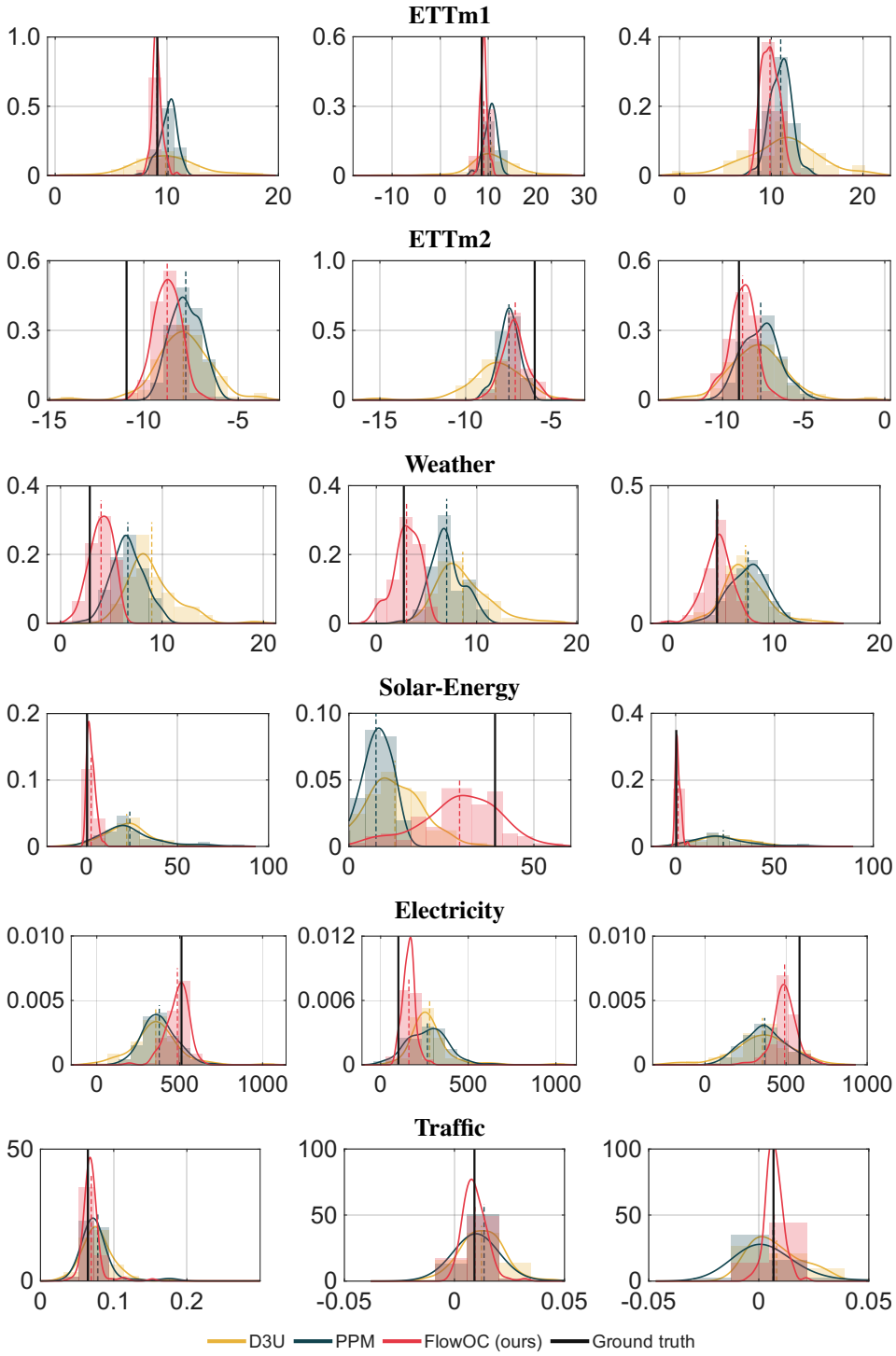


Figure 12: Comparison of the distributions on representative samples with the benchmarks.

L SHOWCASE

Figure 13 visualizes the probabilistic forecasting results generated by FlowOC across six benchmark datasets. The sampled trajectories closely follow the observed future patterns while maintaining diverse uncertainty ranges.

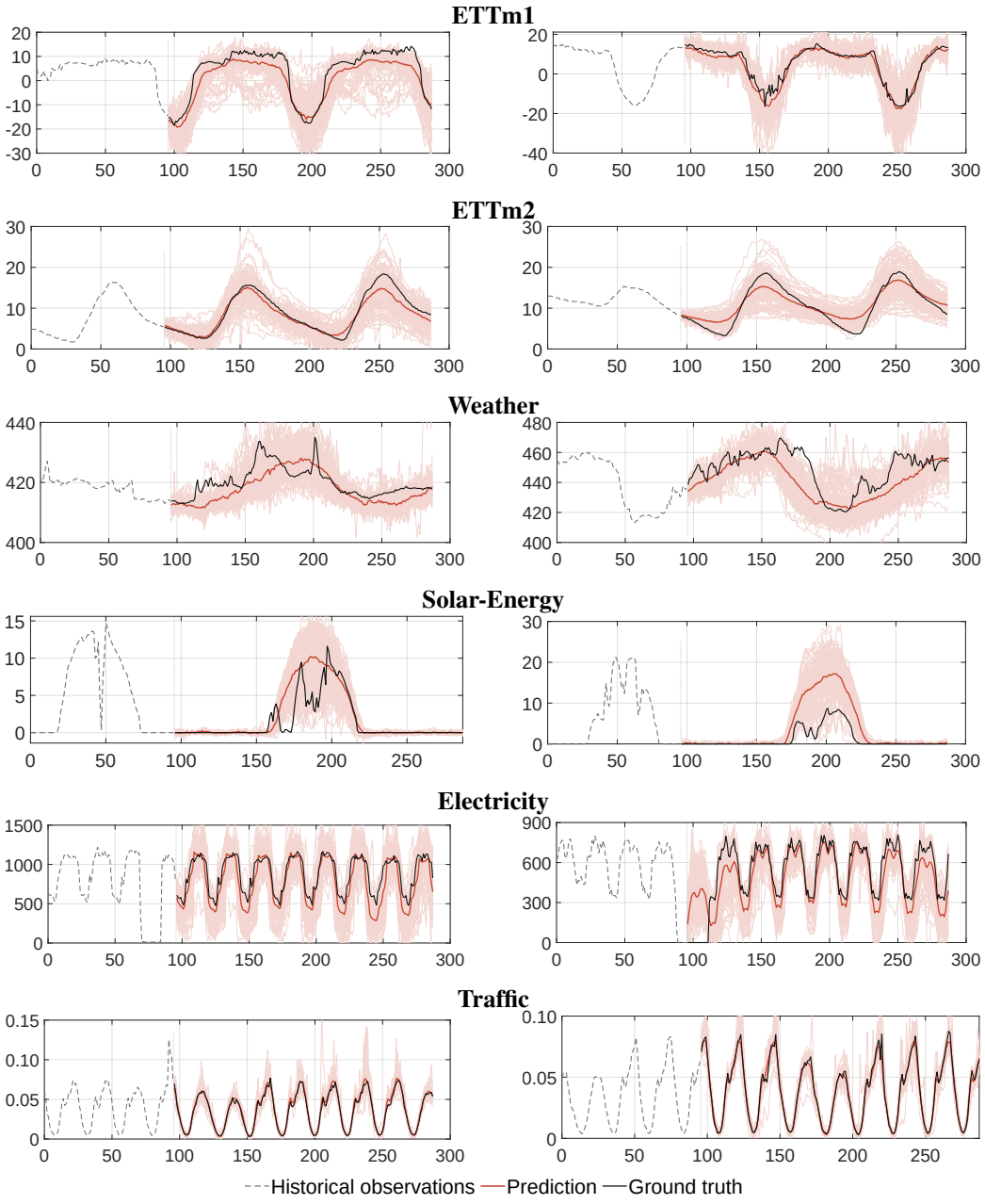


Figure 13: Visualization of probabilistic forecasting results generated by FlowOC across six benchmark datasets. Each row corresponds to one dataset and shows representative sampled trajectories with predictive uncertainty.