FLOW MATCHING IN HIERARCHICAL RESIDUAL COORDINATES FOR NON-STATIONARY PROBABILISTIC FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Probabilistic forecasting for non-stationary time series is difficult because the conditional target distribution entangles drifting location-scale statistics with residual stochastic dynamics. Existing probabilistic forecasting methods therefore ask a single transport map to learn macroscopic distribution drift and microscopic stochastic dynamics simultaneously, while overlooking the optimization difficulty caused by this coupling. We propose hierarchical residual coordinates: context-dependent invertible maps remove predictable distributional components from the full future trajectory before generation, while a residual generator learns the remaining joint law and preserves temporal and cross-variate dependence. The inverse coordinate map then transports the learned residual flow back to the original space. We establish conditional pushforward consistency and a stability guarantee for estimated residual coordinates. As a second-order instantiation, LS-Flow uses predicted location and scale to define the residual coordinate and trains Conditional Flow Matching there. Across eight real-world non-stationary multivariate benchmarks, LS-Flow improves probabilistic forecasting over strong generative baselines; order-wise comparisons and a third-order conditional-skewness extension further examine the framework's scope. Code is available at this repository: `https://anonymous.4open.science/r/LS-Flow-66`

1 INTRODUCTION

Probabilistic time series forecasting models the conditional distribution of future observations, rather than only their expected values. This makes it important whenever uncertainty and risk matter (Salinas et al., 2020; Rasul et al., 2021b). The task is difficult because real-world data are non-stationary: the target distribution changes with context. Its lower-order moments, such as location and scale, often drift over time (Du et al., 2021). Higher-order structure then appears as microscopic stochastic dynamics within each regime (Song et al., 2021). A useful forecasting model must therefore separate these statistical roles instead of fitting them implicitly with one entangled mapping.

Deep generative models, especially Diffusion Models (Ho et al., 2020) and Flow Matching (Lipman et al., 2023), are effective tools for probabilistic forecasting. Early methods such as TimeGrad (Rasul et al., 2021a) and CSDI (Tashiro et al., 2021) estimate conditional densities from historical context. Standard architectures, however, transport a static prior directly to a target that changes across contexts (Chen et al., 2025). Recent methods provide more informative source distributions: TMDM (Li et al., 2024) injects predictive means, NsDiff (Ye et al., 2025) uses context-dependent location-scale noise, and D^3U (Li et al., 2025) separates deterministic forecasts from residual uncertainty. Their generative dynamics still largely evolve in the original observation space. Macroscopic drift and microscopic stochastic dynamics therefore remain coupled, so one neural vector field must learn both at once. This coupling creates an avoidable optimization burden.

To address this entanglement, we formulate *hierarchical residual coordinates* for conditional probabilistic forecasting. For order k, context-predictable components define a composition of conditional invertible maps on the full future trajectory. The generator is trained on the resulting residual joint law, and the inverse coordinate map returns samples to the observation space. This construction acts on the

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.

target trajectory itself, rather than using moments only as conditioning features or source-distribution parameters. It therefore preserves temporal and cross-variate dependence in the residual law.

LS-Flow is the second-order instantiation. It predicts future location and scale because a first-order residual removes location drift but leaves volatility in the transport target. The resulting standardized residual coordinate gives Flow Matching a better-conditioned target while preserving all remaining joint structure. A third-order conditional-skewness extension tests how the same coordinate construction can incorporate an additional invertible shape component. Our main contributions are:

- **A Novel Generative Paradigm**: We introduce the LS-Flow framework, rethinking probabilistic forecasting for non-stationary time series through hierarchical probabilistic decomposition. Context-dependent invertible residual coordinates progressively remove predictable location-scale and higher-order components from the full future trajectory before generation, shifting the transport problem from a volatile observation space to a better-conditioned residual space. The residual generator still models temporal and cross-variate dependence without an independence or Gaussian assumption.

- **Architecture and Theoretical Guarantee**: We design a systematic architecture that predicts context-dependent future location and scale and trains a Conditional Flow Matching generator in the resulting residual coordinate. We establish a conditional pushforward identity for arbitrary estimated invertible coordinates and a Wasserstein stability bound that quantifies residual-generation error after reconstruction.

- **Extensive Empirical Validation**: We conduct comprehensive experiments across eight real-world benchmarks. LS-Flow improves CRPS, $\mathrm{CRPS}_{\mathrm{sum}}$, Energy Score, and point forecasting accuracy against strong generative baselines, while order-wise comparisons, geometry analyses, third-order extension results, and efficiency studies identify when and why each residualization order helps.

2 Related Work

Probabilistic Forecasting under Non-Stationarity Deep generative models, primarily Diffusion Models (Sohl-Dickstein et al., 2015; Ho et al., 2020) and Normalizing Flows (Papamakarios et al., 2021), have emerged as important tools for multivariate probabilistic forecasting. Early adaptations, such as TimeGrad (Rasul et al., 2021a) and autoregressive normalizing flows (Rasul et al., 2021b), established practical conditional density estimation. Subsequent work has improved efficiency and dependence modeling through diffusion approaches such as CSDI (Tashiro et al., 2021) and TimeDiff (Shen & Kwok, 2023), and flow architectures such as MANF (Feng et al., 2024). For non-stationary data, TMDM (Li et al., 2024) injects predictive information into diffusion, NsDiff (Ye et al., 2025) uses a location-scale noise formulation, and $\mathrm{D}^3\mathrm{U}$ (Li et al., 2025) separates deterministic prediction from residual uncertainty. These methods use predictive statistics in different roles, including conditioning, source distributions, diffusion endpoints, or deterministic residualization. Our formulation uses such components to define hierarchical conditional residual coordinates for the full future joint distribution and trains Flow Matching on the resulting residual target.

Non-stationary Time Series Forecasting In deterministic forecasting, extensive efforts have been made to address non-stationarity. Methods like AdaRNN (Du et al., 2021) and Non-stationary Transformers (Liu et al., 2022) employ domain adaptation and de-stationary attention to account for statistical shifts, while Koopman-based approaches (Wang et al., 2023; Liu et al., 2024b) capture multi-scale dynamics. A widely adopted technique for mitigating distribution shift is reversible normalization (RevIN) (Kim et al., 2022), which standardizes inputs based on historical statistics. To capture more complex non-stationary patterns, recent works have diversified their approaches: exploring finer-grained trend decomposition (Fan et al., 2023; Liu et al., 2024c), preserving invariant measures to stabilize chaotic dynamics (Jiang et al., 2023), and leveraging frequency-domain transformations to tackle spectral shifts (Ye et al., 2024). However, historical normalization inherently lags behind future drift, which makes directly translating these techniques to generative probabilistic models non-trivial.

3 Problem Formulation and Hierarchical Residual Coordinates

3.1 Probabilistic Forecasting and the Macro-Micro Entanglement

Let $\mathbf{x}_{1:T}$ be a multivariate time series with d variables. Given the context window $\mathbf{X} := \mathbf{x}_{t-L+1:t} \in \mathbb{R}^{L\times d}$, we model the predictive distribution of the future horizon $\mathbf{Y} := \mathbf{x}_{t+1:t+H} \in \mathbb{R}^{H\times d}$. Here, L and H are the context and prediction lengths.

Under non-stationarity, $p(\mathbf{Y} \mid \mathbf{X})$ changes with the context (Liu et al., 2022; Kim et al., 2022). We describe this change through conditional statistics. The target contains two components:

- **Macroscopic Drifting Moments:** The lower-order moments, specifically the conditional expectation $\mathbb{E}[\mathbf{Y} \mid \mathbf{X}]$ and standard deviation $\mathrm{Std}(\mathbf{Y} \mid \mathbf{X})$. In non-stationary data, these dictate the absolute location and scale of the predictive space, undergoing severe shifts across varying contexts.
- **Microscopic Stochastic Dynamics:** The intrinsic topological structure of the distribution (e.g., skewness, heavy tails, and multi-modality). This represents the higher-order residual uncertainty after accounting for the macroscopic drift.

Most probabilistic models estimate $p(\mathbf{Y} \mid \mathbf{X})$ directly in the observation space (Salinas et al., 2020; Rasul et al., 2021a). Continuous-time generators instead work best when the transport target is stable (Lipman et al., 2023). The raw target violates this condition: its macroscopic drift is entangled with microscopic dynamics. Mapping a static prior to this shifting target is consequently difficult to optimize (Lee et al., 2021; Han et al., 2022).

3.2 General Hierarchical Residual Coordinates

To define the coordinates explicitly, let $S^{(0)}_{\mathbf{X}} = \mathrm{Id}$ and let $R_{j,\mathbf{X}}$ be a context-dependent invertible residualization map for the jth distributional component. The order-k hierarchical residual coordinate is the composition

$$S^{(k)}_{\mathbf{X}} = R_{k,\mathbf{X}} \circ \cdots \circ R_{1,\mathbf{X}}, \qquad T^{(k)}_{\mathbf{X}} = (S^{(k)}_{\mathbf{X}})^{-1}. \tag{1}$$

For a future trajectory $\mathbf{Y}$, we define

$$\mathbf{Z}^{(k)} = S^{(k)}_{\mathbf{X}}\big(\mathbf{Y}; \boldsymbol{\Theta}^{(k)}(\mathbf{X})\big), \qquad \mathbf{Y} = T^{(k)}_{\mathbf{X}}\big(\mathbf{Z}^{(k)}; \boldsymbol{\Theta}^{(k)}(\mathbf{X})\big). \tag{2}$$

Here $\boldsymbol{\Theta}^{(k)}(\mathbf{X})$ contains the context-predictable components removed through order k. This is a conditional coordinate change for the full predictive distribution, not an independence assumption: the residual law may still contain temporal dependence, cross-variate dependence, skewness, heavy tails, and multimodality. The goal is a better-conditioned residual target, not an assertion that it is exactly stationary or Gaussian.

First-order decomposition removes only the conditional location shift, with $\boldsymbol{\Theta}^{(1)}(\mathbf{X}) = \{\boldsymbol{\mu}(\mathbf{X})\}$. The residual is $\mathbf{Z}^{(1)} = \mathbf{Y} - \boldsymbol{\mu}(\mathbf{X})$. This location residual leaves context-dependent scale variation in the generative target.

Second-order decomposition additionally removes conditional scale, with $\boldsymbol{\Theta}^{(2)}(\mathbf{X}) = \{\boldsymbol{\mu}(\mathbf{X}), \boldsymbol{\sigma}(\mathbf{X})\}$. The residual and reconstruction are

$$\mathbf{Z}^{(2)} = (\mathbf{Y} - \boldsymbol{\mu}(\mathbf{X})) \oslash \boldsymbol{\sigma}(\mathbf{X}), \qquad \mathbf{Y} = \boldsymbol{\mu}(\mathbf{X}) + \boldsymbol{\sigma}(\mathbf{X}) \odot \mathbf{Z}^{(2)}. \tag{3}$$

LS-Flow is this second-order coordinate with a Conditional Flow Matching residual generator. A higher-order instance can append an invertible conditional shape map when its additional parameter is estimable and useful; our third-order conditional-skewness extension in Appendix E demonstrates this construction.

By removing context-predictable components, the residual generator focuses on the remaining stochastic structure while the inverse coordinate map preserves the full predictive law. The present paper studies the location-scale case because it offers a practical balance between distributional expressiveness and estimation stability.

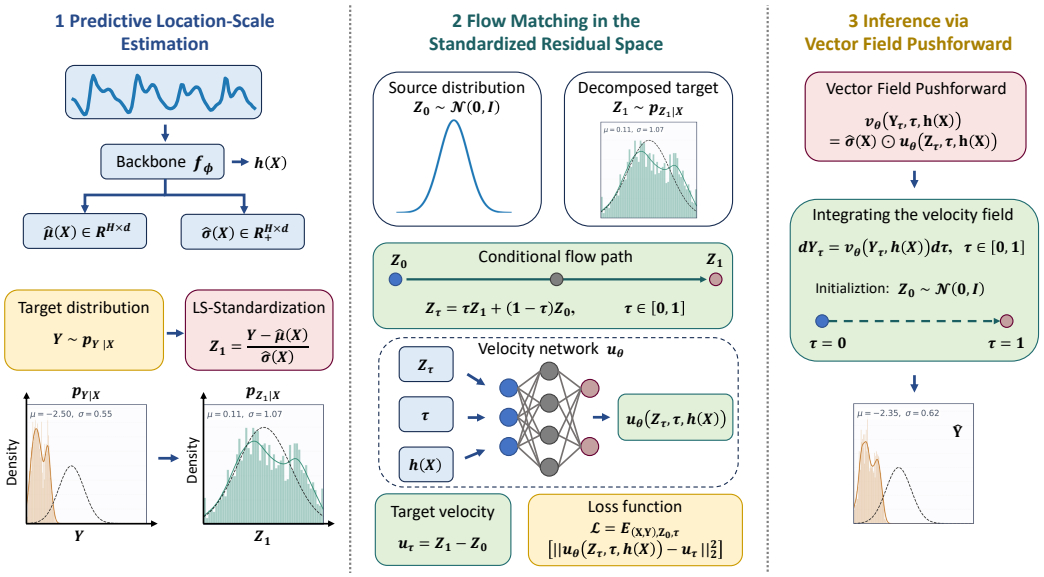


Figure 1: An overview of the proposed second-order LS-Flow instantiation. Left: A predictive module extracts context-aware embeddings to estimate macroscopic location and scale, projecting the target into a standardized residual space. Middle: A conditional flow path is constructed to exclusively learn the microscopic stochastic dynamics. Right: During inference, the learned vector field is pushed forward to directly simulate continuous-time physical trajectories, yielding the final probabilistic forecasts.

4 Second-Order Location-Scale Flow Matching

LS-Flow is the second-order location-scale Conditional Flow Matching instantiation of the hierarchical residual-coordinate formulation defined in Section 3. It predicts future location and scale, trains the flow on the standardized residual target, and pushes the learned continuous-time residual dynamics through the coordinate map to produce the full probabilistic forecast. The general pushforward consistency result is given in Section 4.3.

4.1 Predictive Location-Scale Estimation

To instantiate the context-dependent statistics $\boldsymbol{\mu}(\mathbf{X}) \in \mathbb{R}^{H\times d}$ and $\boldsymbol{\sigma}(\mathbf{X}) \in \mathbb{R}^{H\times d}$, we need to predict them based on $\mathbf{X}$. First, we employ a deterministic forecasting architecture, iTransformer (Liu et al., 2024a), as our backbone, which can model cross-variate dependencies. This backbone captures the temporal dynamics of each variate to yield a context-aware shared hidden embedding $\mathbf{h}(\mathbf{X}) \in \mathbb{R}^{d\times d_{\text{model}}}$, where d_{model} denotes the hidden embedding dimension. This representation is subsequently fed into a projection module, consisting of two parallel two-layer Multi-Layer Perceptrons (MLPs) that predict the future expected location $\hat{\boldsymbol{\mu}}(\mathbf{X})$ and future scale $\hat{\boldsymbol{\sigma}}(\mathbf{X})$:

$$\hat{\boldsymbol{\mu}}(\mathbf{X}) = \text{MLP}_\mu(\mathbf{h}(\mathbf{X})), \quad \hat{\boldsymbol{\sigma}}(\mathbf{X}) = \text{Softplus}\big(\text{MLP}_\sigma(\mathbf{h}(\mathbf{X}))\big). \tag{4}$$

The softplus activation is applied at the output of the scale estimator to strictly guarantee the positivity of the predicted scale ($\hat{\boldsymbol{\sigma}}(\mathbf{X}) > 0$). We optimize these estimators jointly during training using a single Gaussian Negative Log-Likelihood (NLL) objective (Nix & Weigend, 1994). This unified objective optimizes both heads simultaneously, allowing the network to accurately estimate the expected location while capturing the heteroscedastic uncertainty:

$$\mathcal{L}_{\text{NLL}} = \frac{1}{2}\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\left[\sum_{i=1}^{H}\left(\frac{(\mathbf{Y}_i - \hat{\boldsymbol{\mu}}_i(\mathbf{X}))^2}{\hat{\boldsymbol{\sigma}}_i(\mathbf{X})^2} + 2\log\hat{\boldsymbol{\sigma}}_i(\mathbf{X})\right)\right]. \tag{5}$$

4.2 Conditional Flow Matching in the Second-Order Residual Coordinate

With the predictable macroscopic statistics decoupled, the core generative task is completely offloaded to the standardized residual space. As depicted in the "LS-Standardization" block of Figure 1, we map the target into the standardized residual space:

$$\mathbf{Z}_1 = (\mathbf{Y} - \hat{\boldsymbol{\mu}}(\mathbf{X})) \oslash \hat{\boldsymbol{\sigma}}(\mathbf{X}), \tag{6}$$

where $\oslash$ denotes element-wise division. By applying this operation, $\mathbf{Z}_1$ inherently captures the microscopic stochastic dynamics, stripped of the macroscopic drift. Within this better-conditioned residual coordinate, we formulate Conditional Flow Matching.

We sample a prior from a standard Gaussian, $\mathbf{Z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, and define the probability path using the Optimal Transport (OT) linear schedule (McCann, 1997; Lipman et al., 2023):

$$\mathbf{Z}_\tau = \tau \mathbf{Z}_1 + (1 - \tau)\mathbf{Z}_0, \tag{7}$$

where $\tau \in [0, 1]$ represents the continuous integration time of the flow process, deliberately chosen to avoid notational confusion with the discrete time steps of the underlying time series.

Loss Function: The generative objective is to train a neural vector field u_θ, conditioned on the shared backbone embedding $\mathbf{h}(\mathbf{X})$, to match the target velocity in the standardized residual space. Under the linear OT schedule, the target velocity simplifies elegantly to $u_\tau = \mathbf{Z}_1 - \mathbf{Z}_0$. Thus, our Flow Matching loss is formulated as:

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{(\mathbf{X},\mathbf{Y}),\mathbf{Z}_0,\tau}\left[\left\|u_\theta(\mathbf{Z}_\tau, \tau, \mathbf{h}(\mathbf{X})) - u_\tau\right\|_2^2\right]. \tag{8}$$

By computing the loss strictly within the standardized space, the neural network θ is completely relieved from extrapolating macroscopic drift and focuses exclusively on modeling the complex higher-order variations.

Inference via Vector Field Pushforward: During the inference phase, LS-Flow explicitly leverages the change of variables to reconstruct the continuous-time dynamics directly in the physical space. Since the predicted macroscopic statistics $\hat{\boldsymbol{\mu}}(\mathbf{X})$ and $\hat{\boldsymbol{\sigma}}(\mathbf{X})$ are invariant to the flow integration time τ, the neural vector field u_θ learned in the standardized space naturally induces a corresponding physical vector field v_θ in the original observation space. Let $\mathbf{Y}_\tau$ denote the physical-space state at flow time τ. By differentiating the relation $\mathbf{Y}_\tau = \hat{\boldsymbol{\mu}}(\mathbf{X}) + \hat{\boldsymbol{\sigma}}(\mathbf{X}) \odot \mathbf{Z}_\tau$ with respect to τ, we obtain:

$$v_\theta(\mathbf{Y}_\tau, \tau, \mathbf{h}(\mathbf{X})) = \hat{\boldsymbol{\sigma}}(\mathbf{X}) \odot u_\theta(\mathbf{Z}_\tau, \tau, \mathbf{h}(\mathbf{X})). \tag{9}$$

By solving the ordinary differential equation (ODE) defined by v_θ from $\tau = 0$ to $\tau = 1$ (Chen et al., 2018), LS-Flow directly simulates the continuous-time physical trajectories, yielding the final probabilistic forecast $\hat{\mathbf{Y}}$ (see Appendix B for the detailed inference process). The conditional pushforward consistency result is established in Section 4.3.

This is a pushforward of the entire residual flow rather than an endpoint-only denormalization: every intermediate physical state and velocity is induced by the same conditional coordinate map, so the physical trajectory remains consistent with the conditional change of variables. The pushforward formulation makes this law-preserving construction explicit and extends directly to nonlinear higher-order coordinates, where the Jacobian of the coordinate map determines the corresponding physical-space dynamics.

The equivalence concerns the induced ODE and probability path. It does not identify the two Euclidean Flow Matching losses. In the affine case, a physical-space velocity error is weighted by $\mathrm{diag}(\hat{\boldsymbol{\sigma}}^2)$ relative to the residual-space error, which is one reason to optimize in the standardized coordinate.

4.3 Conditional Pushforward under Estimated Coordinates

Conditional on a context $\mathbf{X}$, the fitted residual coordinate is a change of coordinates for the full predictive law. The result below treats the fitted statistics as arbitrary estimated quantities. It does not require them to equal the population moments, or the residual law to be stationary or Gaussian.

Theorem 1 (Conditional Pushforward under Estimated Residual Coordinates)**.** *For almost every context* $\mathbf{x}$*, let* $\widehat{S}_{\mathbf{x}}^{(k)} : \mathcal{Y} \to \mathcal{Z}$ *be any context-measurable* C^1 *diffeomorphism induced by estimated*

components $\widehat{\mathbf{\Theta}}^{(k)}(\mathbf{x})$*, and let* $\widehat{T}_{\mathbf{x}}^{(k)} = (\widehat{S}_{\mathbf{x}}^{(k)})^{-1}$*. Let* $\widetilde{p}_\tau^Z(\mathbf{z} \mid \mathbf{x})$ *solve the residual-space continuity equation with vector field* $\widetilde{u}_\tau(\mathbf{z} \mid \mathbf{x})$*, and define* $\widetilde{q}_\tau^Y = (\widehat{T}_{\mathbf{x}}^{(k)})_{\#}\widetilde{p}_\tau^Z$*. Then*

$$\widetilde{q}_\tau^Y(\mathbf{y} \mid \mathbf{x}) = \widetilde{p}_\tau^Z\big(\widehat{S}_{\mathbf{x}}^{(k)}(\mathbf{y}) \mid \mathbf{x}\big)\left|\det D\widehat{S}_{\mathbf{x}}^{(k)}(\mathbf{y})\right|, \tag{10}$$

and $\widetilde{q}_\tau^Y$ *satisfies the continuity equation with induced vector field*

$$\widetilde{v}_\tau(\mathbf{y} \mid \mathbf{x}) = D\widehat{T}_{\mathbf{x}}^{(k)}\big(\widehat{S}_{\mathbf{x}}^{(k)}(\mathbf{y})\big)\widetilde{u}_\tau\big(\widehat{S}_{\mathbf{x}}^{(k)}(\mathbf{y}) \mid \mathbf{x}\big). \tag{11}$$

For any generated endpoint law, $\widetilde{q}_1^Y = (\widehat{T}_{\mathbf{x}}^{(k)})_{\#}\widetilde{p}_1^Z$*. In the special case* $\widetilde{p}_1^Z = (\widehat{S}_{\mathbf{x}}^{(k)})_{\#}p_{\mathrm{data}}(\mathbf{Y} \mid \mathbf{x})$*, exact recovery follows.*

Proof Sketch. The density identity follows from the conditional change-of-variables formula. Differentiating $\mathbf{Y}_\tau = \widehat{T}_{\mathbf{x}}^{(k)}(\mathbf{Z}_\tau)$ gives the induced vector field. Substitution into the residual continuity equation gives the physical-space continuity equation, and the endpoint identity follows from push-forward composition. The theorem is conditional on the fitted maps; it makes no ideal-moment assumption.

Corollary 1 (Affine Location-Scale Case)**.** *For* $\widehat{T}_{\mathbf{X}}^{(2)}(\mathbf{Z}) = \hat{\boldsymbol{\mu}}(\mathbf{X}) + \hat{\boldsymbol{\sigma}}(\mathbf{X}) \odot \mathbf{Z}$ *with strictly positive predicted scale, the theorem gives*

$$\widetilde{v}_\tau(\mathbf{y} \mid \mathbf{X}) = \hat{\boldsymbol{\sigma}}(\mathbf{X}) \odot \widetilde{u}_\tau\left(\frac{\mathbf{y} - \hat{\boldsymbol{\mu}}(\mathbf{X})}{\hat{\boldsymbol{\sigma}}(\mathbf{X})}\middle|\mathbf{X}\right). \tag{12}$$

No equality between $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}$ *and the population conditional moments is required. Thus, residual-space ODE integration and affine reconstruction induce the same physical probability path. The detailed algebra is given in Appendix A.*

Proposition 1 (Approximate Distributional Recovery)**.** *Assume that* $\widehat{T}_{\mathbf{x}}^{(k)}$ *is* $L_{\mathbf{x}}$*-Lipschitz and that the relevant distributions have finite* p*th moments. If the generated residual endpoint satisfies*

$$W_p\Big(\widetilde{p}_1^Z, (\widehat{S}_{\mathbf{x}}^{(k)})_{\#}p_{\mathrm{data}}(\mathbf{Y} \mid \mathbf{x})\Big) \le \varepsilon_{\mathbf{x}}, \tag{13}$$

then

$$W_p\Big((\widehat{T}_{\mathbf{x}}^{(k)})_{\#}\widetilde{p}_1^Z, p_{\mathrm{data}}(\mathbf{Y} \mid \mathbf{x})\Big) \le L_{\mathbf{x}}\varepsilon_{\mathbf{x}}. \tag{14}$$

Writing $\varepsilon_{\mathbf{x}} = \varepsilon_{\mathrm{stat},\mathbf{x}} + \varepsilon_{\mathrm{model},\mathbf{x}}$ *separates residual-target estimation error from generator and ODE approximation error. For affine LS-Flow,* $L_{\mathbf{x}} = \|\hat{\boldsymbol{\sigma}}(\mathbf{x})\|_\infty$*.*

The bound formalizes the role of imperfect statistics. Using the same fitted coordinate for residualization and reconstruction preserves the population pushforward identity; estimation error instead changes the residual target and can affect the Lipschitz amplification factor.

5 Experiments

In this section, we evaluate the second-order LS-Flow instantiation on eight real-world non-stationary forecasting benchmarks against strong generative baselines. We report both probabilistic and point-based results, together with analyses of the residual-coordinate construction.

5.1 Experiment Setup

Datasets. Eight popular real-world datasets with diverse characteristics are selected, including Electricity Transformer Temperature datasets (ETTh1, ETTh2, ETTm1, and ETTm2), Weather, SolarEnergy (Solar), Electricity (ECL), and Traffic. Table 1 summarizes basic statistics for these datasets; specifically, the reported variation is the ratio of test

Table 1: Summary of dataset statistics.

Dataset	Dim.	Freq.	Time steps	Variation
ETTh1	7	1 Hour	14,307	3.69
ETTh2	7	1 Hour	14,307	1.01
ETTm1	7	15 Min	69,680	2.53
ETTm2	7	15 Min	69,680	1.27
Weather	21	10 Min	52,695	0.39
Solar	137	1 Hour	52,560	0.92
ECL	321	1 Hour	26,304	3.94
Traffic	862	1 Hour	17,544	14.23

Table 2: Performance comparisons on 8 real-world datasets regarding CRPS and $\mathrm{CRPS}_{\mathrm{sum}}$. The **best**/second results are highlighted in **bold**/underline, respectively.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Solar		ECL		Traffic	
Method	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$	CRPS	CRPS$_{\mathrm{sum}}$
TimeGrad	0.616	1.043	0.973	0.549	0.665	0.996	0.785	1.051	0.482	0.503	0.783	1.167	0.503	1.452	0.657	1.683
CSDI	1.118	1.577	0.946	0.270	0.773	0.852	0.625	0.782	0.508	0.465	0.649	0.681	0.465	0.823	0.612	1.275
TimeDiff	0.453	0.822	0.503	0.245	0.454	0.846	0.316	0.180	0.293	0.400	0.900	1.164	0.475	0.594	0.671	0.823
TMDM	0.433	0.762	0.384	0.186	0.349	0.679	0.308	0.171	0.226	0.292	0.350	0.267	0.465	0.136	0.561	0.181
NsDiff	0.377	0.703	0.425	0.237	0.359	0.670	0.258	0.140	0.260	0.309	0.292	0.369	0.265	0.137	0.363	0.223
$\mathrm{D}^3\mathrm{U}$	0.340	0.678	0.381	0.223	**0.286**	0.569	0.246	0.142	0.212	**0.283**	0.186	0.266	0.217	0.174	0.256	0.243
LS-Flow	**0.328**	**0.637**	**0.317**	**0.165**	0.298	**0.556**	**0.233**	**0.133**	**0.205**	0.309	**0.185**	**0.228**	**0.189**	**0.121**	**0.229**	**0.178**

variance to train variance, selecting the highest value across dimensions to capture non-stationary uncertainty (Ye et al., 2025). For dataset splits, we follow previous time series prediction works (Wu et al., 2022; Li et al., 2024): the ETT datasets are split in a 6:2:2 ratio for train/val/test, while the others are split 7:1:2. Details can be found in Appendix C.1.

Baselines. We selected six strong probabilistic forecasting baselines for comparison, including: (i) Standard Gaussian Prior: TimeGrad (Rasul et al., 2021a), CSDI (Tashiro et al., 2021), and TimeDiff (Shen & Kwok, 2023); (ii) Informed Prior: TMDM (Li et al., 2024), NsDiff (Ye et al., 2025), and $\mathrm{D}^3\mathrm{U}$ (Li et al., 2025). Detailed descriptions are provided in Appendix C.2.

Settings. Experiments are conducted following prior work (Li et al., 2024), with an input length of 96 and a prediction length of 192. The deterministic backbone used in LS-Flow is iTransformer (Liu et al., 2024a). To solve the ODE, we employ a 5-step Euler method, consistent with the setup in Rectified Flow (Liu et al., 2023). During inference, 100 samples are generated to estimate the predictive distribution. All reported test results are averaged over five independent runs on a single NVIDIA GeForce RTX 5090 GPU. Optimization is performed using Adam (Kingma & Ba, 2014). Further experimental details are provided in Appendix C.3.

Metrics. Following prior work (Li et al., 2025), we evaluate marginal and aggregate probabilistic accuracy with the Continuous Ranked Probability Score (CRPS) (Gneiting & Raftery, 2007) and $\mathrm{CRPS}_{\mathrm{sum}}$ (Rasul et al., 2021a). We additionally report the Energy Score (ES), a proper scoring rule for the full multivariate predictive distribution. Point forecasting performance is assessed with Mean Squared Error (MSE) and Mean Absolute Error (MAE). Lower values are better for all metrics; definitions are provided in Appendix C.4.

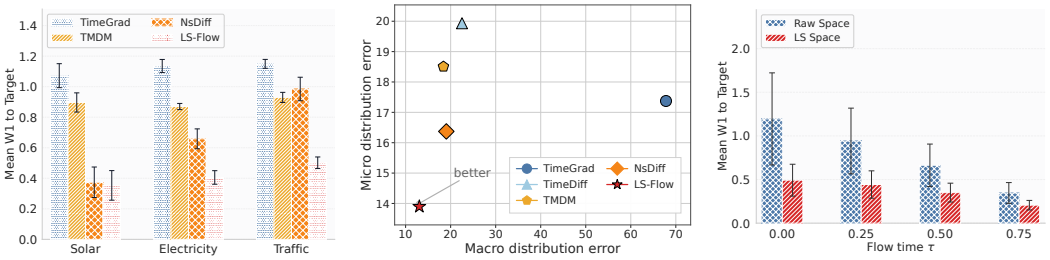
5.2 Forecasting Results

Probabilistic evaluation. Table 2 shows LS-Flow leading in probabilistic calibration, securing the best $\mathrm{CRPS}_{\mathrm{sum}}$ on 7 of 8 datasets and CRPS on 7 of 8 datasets. It notably reduces $\mathrm{CRPS}_{\mathrm{sum}}$ by 14.29% on Solar and 11.03% on ECL compared to the strongest baselines. This advantage arises from projecting targets into a standardized residual space. Unlike NsDiff, which couples drift and dynamics in the volatile observation space, or $\mathrm{D}^3\mathrm{U}$, which separates deterministic and residual components, LS-Flow isolates macroscopic location-scale drift to simplify the transport task. This decomposition improves uncertainty estimation on these benchmarks. The Energy Score provides a joint multivariate check: LS-Flow is best on 7 of the 8 datasets in the available comparison, while $\mathrm{D}^3\mathrm{U}$ is best on Weather. Full ES values are reported in Appendix C.5.

Point-based evaluation. Table 3 shows that LS-Flow achieves leading point accuracy, securing the best MSE on 5 and best MAE on all 8 datasets. Its consistency across high-drift scenarios (e.g., Solar, Traffic) validates that the location-scale coordinate system provides a robust foundation for trajectory expectation. Unlike baselines limited to mean-prior injection (e.g., TMDM), LS-Flow's hierarchical decomposition enables the deterministic backbone to capture macroscopic trends without interference from volatile scale shifts. This synergy ensures the sample mean remains accurate and robust against non-stationary fluctuations, confirming that our second-order decoupling effectively bridges point and probabilistic modeling.

Table 3: Performance comparisons on 8 real-world datasets regarding MSE and MAE.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Solar		ECL		Traffic	
Method	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
TimeGrad	1.125	0.845	2.591	1.191	1.716	1.057	1.385	0.732	0.885	0.551	1.211	1.004	0.645	0.723	0.932	0.807
CSDI	1.913	0.972	1.979	1.100	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	0.467	0.469	0.533	0.516	0.796	0.577	0.284	0.342	0.277	0.331	1.169	0.936	0.730	0.690	1.465	0.851
TMDM	0.684	0.570	0.502	0.488	0.516	0.448	0.363	0.394	0.244	0.288	0.295	0.317	0.226	0.334	0.695	0.412
NsDiff	0.547	0.508	0.614	0.538	0.567	0.474	0.248	0.334	0.302	0.321	0.288	0.354	0.195	0.300	0.630	0.371
D^3U	0.457	0.451	0.506	0.504	**0.367**	0.390	**0.241**	0.320	**0.232**	0.292	0.237	0.270	0.196	0.282	0.535	0.341
LS-Flow	**0.444**	**0.437**	**0.407**	**0.423**	0.375	**0.384**	0.252	**0.310**	0.233	**0.250**	**0.187**	**0.249**	**0.153**	**0.250**	**0.451**	**0.293**



(a) Source–target W1 gap **(b) Macro–micro error** **(c) Transport W1 over flow time**

Figure 2: **Hierarchical analysis of LS-Flow decomposition.** (a) Source–target 1-Wasserstein distance (W_1) gap across datasets for priors with increasing predictive information. (b) Probabilistic decoupling quality on ECL, showing the alignment of macroscopic statistics and standardized residual dynamics (see Appendix G.2 for detailed scatter plots). (c) Evolution of W_1 distance along the transport path, demonstrating the geometric stability of the standardized residual space.

5.3 Model Analysis

This section analyzes LS-Flow from three complementary perspectives: source-target alignment before transport, probabilistic decoupling quality , and the source-target distance over flow time.

Source distribution alignment. Figure 2 (a) compares four source priors, from a static Gaussian (TimeGrad) to mean-informed (TMDM), mean-variance-informed (NsDiff), and LS-Flow. Across Solar, Electricity, and Traffic, adding predictive location-scale information generally reduces the source-target 1-Wasserstein distance (W_1) gap (Rubner et al., 1997), with LS-Flow achieving the lowest distance. NsDiff's slightly higher W_1 on the Traffic dataset stems from its reliance on sliding-window statistics, which struggles to capture the strong periodicity inherent in such series. We also compare the performance of the prior distribution in LS-Flow with that after the Flow-based transport mapping in the Appendix I.4.

Probabilistic decoupling quality. Figure 2 (b) decomposes each method's predictive distribution on ECL into a macro component (μ, σ) and a standardized residual component (refer to Appendix G.2 for window-level scatter plot visualizations). We observe that (i) **Standard Gaussian Prior** models (TimeGrad and TimeDiff) suffer from large errors in both statistics and residual matching, while (ii) **Informed Prior** models (TMDM and NsDiff) improve the macro estimation but still struggle to simplify the residual dynamics. LS-Flow lies closest to the lower-left corner, showing that it simultaneously improves future statistic estimation and simplifies the residual distribution that the generative model must match.

Transport Geometry With and Without Decomposition. Figure 2 (c) measures the mean W_1 distance to the target distribution along the transport path in raw and standardized spaces. The standardized base space exhibits a consistently smaller distance to the target compared to the raw physical space at every flow time. This indicates that removing macroscopic location-scale drift yields a more homogeneous transport geometry, significantly easing the learning of the vector field. Details of the path visualizations are provided in Figure 4 in Appendix.

Table 4: Compact ablation on the location-scale decomposition. Values are CRPS; lower is better. The full CRPS/$\mathrm{CRPS}_{\mathrm{sum}}$ table is in Appendix D.

Design	ETTm1	ETTm2	Weather	ECL
LS-Flow	0.298	0.233	0.205	0.189
Pred. source, w/o LSD	0.326	0.267	0.217	0.193
w/o source and LSD	0.586	1.032	0.483	0.445

Table 5: Order-wise residualization results. Each arrow compares the lower-order coordinate with the next order on the same dataset.

Dataset and comparison	CRPS	$\mathrm{CRPS}_{\mathrm{sum}}$
ETTm1: L1 $\rightarrow$ L2	$0.324 \rightarrow 0.298$	$0.574 \rightarrow 0.556$
ETTm2: L1 $\rightarrow$ L2	$0.273 \rightarrow 0.233$	$0.135 \rightarrow 0.133$
ECL: L1 $\rightarrow$ L2	$0.205 \rightarrow 0.189$	$0.140 \rightarrow 0.121$
Traffic: L2 $\rightarrow$ L3	$0.229 \rightarrow 0.227$	$0.178 \rightarrow 0.154$
Solar: L2 $\rightarrow$ L3	$0.185 \rightarrow 0.178$	$0.228 \rightarrow 0.190$
ETTh1: L2 $\rightarrow$ L3	$0.331 \rightarrow 0.337$	$0.630 \rightarrow 0.632$

5.4 Ablation and Order-wise Residualization

Table 4 isolates the effects of predicted statistics and the location-scale decomposition. Predicted future statistics outperform historical statistics because they track the drift between context and target. Removing the decomposition further degrades performance, confirming that the residual coordinate reduces a source of optimization difficulty for Flow Matching. The complete table is given in Appendix D.

Order-wise residualization. The order-wise results are summarized in Table 5. Removing only location leaves scale variation in the residual target, while the second-order coordinate improves both CRPS and $\mathrm{CRPS}_{\mathrm{sum}}$ on the three mean-only comparisons. The third-order conditional-skewness extension improves both probabilistic metrics on Traffic and Solar, where asymmetric regimes are prominent, while the additional shape parameter is unnecessary on the smoother ETTh1 data. The complete five-metric results and the definition of the third-order coordinate are given in Appendix E.

5.5 Training Strategy and Inference Efficiency

We compare a pre-trained backbone with joint end-to-end optimization. Both variants are competitive; we use the pre-trained setting because it is slightly more stable. The visual comparison and full results are in Appendix F and Appendix I.

Inference Efficiency. Figure 3(b) compares CRPS, MSE, and inference time. LS-Flow combines lower error with shorter inference time because the residual coordinate reduces the source-target discrepancy and simplifies ODE integration.

6 Conclusion

This paper formulates hierarchical residual coordinates for conditional probabilistic forecasting under non-stationarity. Fitted invertible maps remove predictable components before generation, and an inverse pushforward returns residual dynamics to the observation space. The theorem and stability bound cover imperfect fitted coordinates. Across eight benchmarks, the second-order LS-Flow instantiation improves probabilistic and point forecasting, while order-wise results demonstrate higher-order extensibility.

Limitations and Future Work. We use location and scale because they balance expressiveness and estimation stability. The third-order extension targets conditional skewness; tail weight remains a separate direction for future work.

REFERENCES

Chen Chen, Pengsheng Guo, Liangchen Song, Jiasen Lu, Rui Qian, Xinze Wang, Tsu-Jui Fu, Wei Liu, Yinfei Yang, and Alex Schwing. Car-flow: Condition-aware reparameterization aligns source and target for better flow matching. *arXiv preprint arXiv:2509.19300*, 2025.

Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

Yuntao Du, Jindong Wang, Wenjie Feng, Sinno Pan, Tao Qin, Renjun Xu, and Chongjun Wang. Adarnn: Adaptive learning and forecasting of time series. In *Proceedings of the 30th ACM international conference on information & knowledge management*, pp. 402–411, 2021.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-ts: a general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 7522–7529, 2023.

Shibo Feng, Chunyan Miao, Ke Xu, Jiaxiang Wu, Pengcheng Wu, Yang Zhang, and Peilin Zhao. Multi-scale attention flow for probabilistic time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(5):2056–2068, 2024. doi: 10.1109/TKDE.2023.3319672.

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

Xizewen Han, Huangjie Zheng, and Mingyuan Zhou. Card: Classification and regression diffusion models. *Advances in Neural Information Processing Systems*, 35:18100–18115, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Ruoxi Jiang, Peter Y Lu, Elena Orlova, and Rebecca Willett. Training neural operators to preserve invariant measures of chaotic attractors. *Advances in Neural Information Processing Systems*, 36: 27645–27669, 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.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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*, SIGIR '18, pp. 95–104. ACM, 2018. doi: 10.1145/3209978.3210006. URL https://doi.org/10.1145/3209978.3210006.

Sang-gil Lee, Heeseung Kim, Chaehun Shin, Xu Tan, Chang Liu, Qi Meng, Tao Qin, Wei Chen, Sungroh Yoon, and Tie-Yan Liu. Priorgrad: Improving conditional denoising diffusion models with data-dependent adaptive prior. *arXiv preprint arXiv:2106.06406*, 2021.

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 *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=HdUkF1Qk7g.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=qae04YACHs.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=XVjTT1nw5z.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in Neural Information Processing Systems*, 35: 9881–9893, 2022.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024a. URL https://openreview.net/forum?id=JePfAI8fah.

Yong Liu, Chenyu Li, Jianmin Wang, and Mingsheng Long. Koopa: Learning non-stationary time series dynamics with koopman predictors. *Advances in Neural Information Processing Systems*, 36, 2024b.

Zhiding Liu, Mingyue Cheng, Zhi Li, Zhenya Huang, Qi Liu, Yanhu Xie, and Enhong Chen. Adaptive normalization for non-stationary time series forecasting: A temporal slice perspective. *Advances in Neural Information Processing Systems*, 36, 2024c.

Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1): 153–179, 1997.

D.A. Nix and A.S. Weigend. Estimating the mean and variance of the target probability distribution. In *Proceedings of 1994 IEEE International Conference on Neural Networks (ICNN'94)*, pp. 55–60 vol.1, 1994. doi: 10.1109/ICNN.1994.374138.

George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL http://jmlr.org/papers/v22/19-1028.html.

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.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/rasul21a.html.

Kashif Rasul, Abdul-Saboor Sheikh, Ingmar Schuster, Urs M. Bergmann, and Roland Vollgraf. Multivariate probabilistic time series forecasting via conditioned normalizing flows. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=WiGQBFuVRv.

Yossi Rubner, Leonidas J Guibas, and Carlo Tomasi. The earth mover's distance, multi-dimensional scaling, and color-based image retrieval. In *Proceedings of the ARPA image understanding workshop*, volume 661, pp. 668. Springer, 1997.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 2020.

Lifeng Shen and James Kwok. Non-autoregressive conditional diffusion models for time series prediction. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31016–31029. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shen23d.html.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

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. URL https://openreview.net/forum?id=PxTIG12RRHS.

Yusuke Tashiro, Jiaming Song, Yang Song, and Stefano Ermon. CSDI: conditional score-based diffusion models for probabilistic time series imputation. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 24804–24816, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/cfe8504bda37b575c70ee1a8276f3486-Abstract.html.

R Wang, Y Dong, SO Arik, and R Yu. Koopman neural forecaster for time series with temporal distribution shifts. In *International Conference on Learning Representations (ICLR)*, 2023.

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, pp. 22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. Timesnet: Temporal 2d-variation modeling for general time series analysis. *arXiv preprint arXiv:2210.02186*, 2022.

Weiwei Ye, Songgaojun Deng, Qiaosha Zou, and Ning Gui. Frequency adaptive normalization for non-stationary time series forecasting. *Advances in Neural Information Processing Systems*, 37: 31350–31379, 2024.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=afpc1MFMYU.

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.

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 Proof of the Affine Location-Scale Corollary (Corollary 1)

In this section, we provide the proof of the affine location-scale corollary in Corollary 1. The proof relies on the continuity equation governing the transport of probability densities.

Definition of Transformations:
Given the historical context $\mathbf{X}$, our method estimates a location $\hat{\boldsymbol{\mu}} := \hat{\boldsymbol{\mu}}(\mathbf{X})$ and a strictly positive scale $\hat{\boldsymbol{\sigma}} := \hat{\boldsymbol{\sigma}}(\mathbf{X})$. The affine transformation T and its inverse T^{-1} mapping between the standardized residual space $\mathcal{Z}$ and the original physical space $\mathcal{Y}$ are defined as:

$$\mathbf{Y} = T(\mathbf{Z}) = \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot \mathbf{Z}, \tag{15}$$

$$\mathbf{Z} = T^{-1}(\mathbf{Y}) = (\mathbf{Y} - \hat{\boldsymbol{\mu}}) \oslash \hat{\boldsymbol{\sigma}}. \tag{16}$$

Because $\hat{\boldsymbol{\sigma}} > 0$ strictly holds, T is a global diffeomorphism. The absolute value of the determinant of its spatial Jacobian is constant with respect to both $\mathbf{Z}$ and the integration time τ:

$$J_T = |\det \nabla_{\mathbf{Z}} T(\mathbf{Z})| = \prod_j \hat{\boldsymbol{\sigma}}_j, \tag{17}$$

where the product iterates over all $H \times d$ elements of the predictive horizon.

Part 1: Equivalence of the Probability Path (Boundary Matching)
In the residual space $\mathcal{Z}$, we define the base prior distribution $p_0(\mathbf{Z}) = \mathcal{N}(\mathbf{0}, \mathbf{I})$ and the standardized target distribution $p_1(\mathbf{Z})$. The linear Optimal Transport (OT) path is:

$$\mathbf{Z}_\tau = \tau \mathbf{Z}_1 + (1 - \tau)\mathbf{Z}_0. \tag{18}$$

Pushing this path forward to the original space $\mathcal{Y}$ using T, we obtain:

$$\begin{aligned} \mathbf{Y}_\tau &= T(\mathbf{Z}_\tau) \\ &= \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot (\tau \mathbf{Z}_1 + (1 - \tau)\mathbf{Z}_0) \\ &= \tau(\hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot \mathbf{Z}_1) + (1 - \tau)(\hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot \mathbf{Z}_0) \\ &= \tau \mathbf{Y} + (1 - \tau)\mathbf{Y}_0. \end{aligned} \tag{19}$$

This demonstrates that applying the affine transformation to the residual OT path $\mathbf{Z}_\tau$ perfectly reconstructs a linear OT path $\mathbf{Y}_\tau$ in the original space. This induced path smoothly interpolates between the predicted Gaussian prior $\mathbf{Y}_0 \sim \mathcal{N}(\hat{\boldsymbol{\mu}}, \mathrm{diag}(\hat{\boldsymbol{\sigma}}^2))$ and the true target $\mathbf{Y} \sim p(\mathbf{Y}|\mathbf{X})$. At the boundary $\tau = 1$, $\mathbf{Y}_1 = T(\mathbf{Z}_1) = \mathbf{Y}$, meaning the pushed-forward marginal distribution $q_1(\mathbf{Y})$ exactly recovers the target distribution $p(\mathbf{Y}|\mathbf{X})$. The linearity of this physical path is specific to the affine second-order instance. For a higher-order nonlinear coordinate, the same pushforward is generally nonlinear, and the Jacobian-induced field in Theorem 1 defines the corresponding physical dynamics.

Part 2: Equivalence of the Continuity Equation
Let $p_\tau(\mathbf{Z})$ be the marginal density of $\mathbf{Z}_\tau$, driven by the vector field $u_\tau(\mathbf{Z})$. According to continuous normalizing flow theory, p_τ satisfies the continuity equation:

$$\frac{\partial p_\tau(\mathbf{Z})}{\partial \tau} + \nabla_{\mathbf{Z}} \cdot \big(p_\tau(\mathbf{Z}) u_\tau(\mathbf{Z})\big) = 0. \tag{20}$$

By the standard change of variables formula for probability densities under the diffeomorphism T, the induced density $q_\tau(\mathbf{Y})$ in the original space is:

$$q_\tau(\mathbf{Y}) = p_\tau(T^{-1}(\mathbf{Y})) \left|\det \nabla_{\mathbf{Y}} T^{-1}(\mathbf{Y})\right| = p_\tau(T^{-1}(\mathbf{Y})) J_T^{-1}. \tag{21}$$

Thus, $p_\tau(\mathbf{Z}) = q_\tau(T(\mathbf{Z}))J_T$. Substituting this relation into Equation 20:

$$\frac{\partial \big(q_\tau(T(\mathbf{Z}))J_T\big)}{\partial \tau} + \nabla_{\mathbf{Z}} \cdot \big(q_\tau(T(\mathbf{Z}))J_T u_\tau(\mathbf{Z})\big) = 0. \tag{22}$$

Since the spatial Jacobian J_T is entirely independent of $\mathbf{Z}$ and τ, it acts as a scalar constant under the differential operators. Factoring J_T out and dividing the entire equation by J_T, we obtain:

$$\frac{\partial q_\tau(\mathbf{Y})}{\partial \tau} + \nabla_{\mathbf{Z}} \cdot \big(q_\tau(\mathbf{Y}) u_\tau(\mathbf{Z})\big) = 0. \tag{23}$$

Next, we apply the chain rule for the divergence operator under the coordinate transformation $\mathbf{Z} \to \mathbf{Y}$. Since $\mathbf{Y} = \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot \mathbf{Z}$, the spatial differential operator maps precisely as $\nabla_{\mathbf{Z}} = \hat{\boldsymbol{\sigma}} \odot \nabla_{\mathbf{Y}}$. Thus, the divergence term becomes:

$$\frac{\partial q_\tau(\mathbf{Y})}{\partial \tau} + \nabla_{\mathbf{Y}} \cdot \big(q_\tau(\mathbf{Y})(\hat{\boldsymbol{\sigma}} \odot u_\tau(\mathbf{Z}))\big) = 0. \tag{24}$$

This is exactly the continuity equation governing the continuous-time dynamics in the original space $\mathcal{Y}$:

$$\frac{\partial q_\tau(\mathbf{Y})}{\partial \tau} + \nabla_{\mathbf{Y}} \cdot \big(q_\tau(\mathbf{Y}) v_\tau(\mathbf{Y})\big) = 0, \tag{25}$$

where the induced physical vector field is defined as:

$$v_\tau(\mathbf{Y}) = \hat{\boldsymbol{\sigma}} \odot u_\tau(T^{-1}(\mathbf{Y})) = \hat{\boldsymbol{\sigma}} \odot u_\tau(\mathbf{Z}). \tag{26}$$

This establishes the affine corollary: the pushed-forward physical dynamics and residual-space Flow Matching describe the same conditional probability path.

B Algorithms for Training and Inference

In this section, we provide the formal algorithmic procedures for training and inference in the second-order LS-Flow instantiation. By explicitly decoupling macroscopic statistics from microscopic stochastic dynamics, this instantiation trains a residual-space transport model for the full conditional predictive distribution.

B.1 Training

The training of LS-Flow is conducted in a supervised manner, where we simultaneously or sequentially optimize the deterministic backbone for location-scale prediction and the neural vector field for residual dynamics. As established in Section 4, the objective is to match the conditional vector field in the standardized residual space. The complete training procedure is summarized in Algorithm 1.

Algorithm 1 LS-Flow Training Procedure

Input: Time series dataset $\mathcal{D}$, Encoder f_ϕ, Velocity Network u_θ
Pre-train or freeze Encoder f_ϕ (optional but recommended)
while not converged **do**
  Sample batch of context-target pairs $(\mathbf{X}, \mathbf{Y}) \sim \mathcal{D}$
  $[\mathbf{h}(\mathbf{X}), \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}] \leftarrow f_\phi(\mathbf{X})$ {Extract embedding and predict statistics}
  Sample initial noise $\mathbf{Z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  Compute normalized target: $\mathbf{Z}_1 \leftarrow (\mathbf{Y} - \hat{\boldsymbol{\mu}}) \oslash \hat{\boldsymbol{\sigma}}$
  Sample flow time $\tau \sim \mathcal{U}[0, 1]$
  Compute interpolated state: $\mathbf{Z}_\tau \leftarrow \tau \mathbf{Z}_1 + (1 - \tau)\mathbf{Z}_0$
  Predict residual velocity: $u_\theta \leftarrow u_\theta(\mathbf{Z}_\tau, \tau, \mathbf{h}(\mathbf{X}))$
  Target residual velocity: $u_\tau \leftarrow \mathbf{Z}_1 - \mathbf{Z}_0$
  Loss $\mathcal{L}_{\text{FM}} \leftarrow \|u_\theta - u_\tau\|^2$
  Update θ (and optionally ϕ) via gradient descent $\nabla \mathcal{L}_{\text{FM}}$
end while

B.2 Inference

During the inference phase, LS-Flow explicitly leverages the change of variables to reconstruct the continuous-time dynamics directly in the physical observation space. Since the predicted macroscopic statistics $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\sigma}}$ are invariant to the flow integration time τ, we directly employ the physical vector field v_θ as defined in Eq. equation 9 to govern the generative process. Thus, inference is a coordinate-consistent pushforward along the full path, rather than a post-hoc inverse normalization applied only to the final samples.

By solving the ordinary differential equation (ODE) defined by v_θ from $\tau = 0$ to $\tau = 1$, LS-Flow simulates the physical trajectories to yield the final probabilistic forecast. The residual coordinate

reduces the macroscopic variation seen by the transport model, and the resulting path can be integrated accurately with a small number of steps in our experiments. The complete inference procedure is summarized in Algorithm 2.

Algorithm 2 LS-Flow Inference

Input: Context window $\mathbf{X}$, Number of integration steps N
$[\mathbf{h}(\mathbf{X}), \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}] \leftarrow f_\phi(\mathbf{X})$ {Predict context-aware macroscopic statistics}
Sample initial noise $\mathbf{Z}_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
$\mathbf{Y}_0 \leftarrow \hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\sigma}} \odot \mathbf{Z}_0$ {Map prior to initial physical state}
if $N = 1$ (One-Step Generation) **then**
    $\hat{\mathbf{Y}} \leftarrow \mathbf{Y}_0 + \hat{\boldsymbol{\sigma}} \odot u_\theta(\mathbf{Z}_0, 0, \mathbf{h}(\mathbf{X}))$ {Single Euler step estimation}
else
    At each ODE evaluation, compute $\mathbf{Z}_\tau \leftarrow (\mathbf{Y}_\tau - \hat{\boldsymbol{\mu}}) \oslash \hat{\boldsymbol{\sigma}}$
    Define physical velocity: $v_\theta(\mathbf{Y}_\tau, \tau, \mathbf{h}(\mathbf{X})) = \hat{\boldsymbol{\sigma}} \odot u_\theta(\mathbf{Z}_\tau, \tau, \mathbf{h}(\mathbf{X}))$
    $\hat{\mathbf{Y}} \leftarrow \text{ODESolver}(\mathbf{Y}_0, v_\theta, \tau \in [0, 1], \text{steps} = N)$
end if
Return Probabilistic forecast $\hat{\mathbf{Y}}$

C Implementation Details

C.1 Dataset Descriptions

We conduct experiments on 8 real-world datasets to evaluate the second-order LS-Flow instantiation, including: (1) **ETT** (Zhou et al., 2021) contains 7 factors of electricity transformer from July 2016 to July 2018. There are four subsets where ETTh1 and ETTh2 are recorded every hour, and ETTm1 and ETTm2 are recorded every 15 minutes. (2) **Weather** (Wu et al., 2021) includes 21 meteorological factors collected every 10 minutes from the Weather Station of the Max Planck Biogeochemistry Institute in 2020. (3) **ECL** (Wu et al., 2021) records the hourly electricity consumption data of 321 clients. (4) **Traffic** (Wu et al., 2021) collects hourly road occupancy rates measured by 862 sensors of San Francisco Bay area freeways from January 2015 to December 2016. (5) **Solar** (Lai et al., 2018) records the solar power production of 137 PV plants in 2006, which are sampled every 10 minutes.

We follow the same data processing and train-validation-test set split protocol widely used in previous time series prediction works (Wu et al., 2022; Li et al., 2024). The datasets are strictly divided according to chronological order to ensure there are no data leakage issues. Specifically, the ETT datasets are split chronologically in a 6:2:2 ratio for the train, validation, and test sets. For Weather, SolarEnergy, ECL, and Traffic, we use a 7:1:2 chronological split. All features are standardized with statistics computed from the training partition only. The input context length is fixed to 96 and the prediction horizon is fixed to 192 for all experiments. The detailed characteristics and sample splits of all evaluated datasets are summarized in Table 6.

Table 6: Detailed descriptions and sample splits of the evaluated datasets.

Dataset	Dim	Prediction Length	Dataset Size (Train, Val, Test)	Frequency	Information
ETTh1, ETTh2	7	192	(8545, 2881, 2881)	Hourly	Electricity
ETTm1, ETTm2	7	192	(34465, 11521, 11521)	15 Min	Electricity
Weather	21	192	(36792, 5271, 10540)	10 Min	Weather
ECL	321	192	(18317, 2633, 5261)	Hourly	Electricity
Traffic	862	192	(12185, 1757, 3509)	Hourly	Transportation
Solar	137	192	(36601, 5161, 10417)	10 Min	Energy

C.2 Baselines

We compare LS-Flow against six representative probabilistic forecasting baselines, which can be categorized into three distinct paradigms:

- **Standard Gaussian Prior:**

  - **TimeGrad** (Rasul et al., 2021a): An autoregressive framework that leverages energy-based models to iteratively estimate data gradients. Code: `https://github.com/zalandoresearch/pytorch-ts`
  - **CSDI** (Tashiro et al., 2021): A conditional score-based diffusion model designed for robust probabilistic imputation and forecasting. Code: `https://github.com/ermongroup/csdi`
  - **TimeDiff** (Shen & Kwok, 2023): A non-autoregressive conditional diffusion model that utilizes future mixup mechanisms to stabilize generative prediction. (No publicly available code; we use the implementation code in NsDiff).
- **Informed Prior:**
  - **TMDM** (Li et al., 2024): A Transformer-modulated diffusion model that extracts temporal priors from a deterministic backbone to guide the generative process. Code: `https://github.com/LiYuxin321/TMDM`
  - **NsDiff** (Ye et al., 2025): A framework that replaces the standard additive noise assumption with a location-scale noise model to address non-stationarity. Code: `https://github.com/wwy155/NsDiff`
  - D^3U (Li et al., 2025): A decoupled framework that separates deterministic prediction from residual uncertainty generation. Code: `https://github.com/zhangzhenBrave/D3U`

For fairness, each baseline uses its official implementation or the configuration reported in the original paper when available. All methods are trained and evaluated under the same data split and forecasting horizon. All baselines and LS-Flow are implemented using PyTorch (Paszke et al., 2019).

C.3 Reproducibility

To facilitate reproducibility and future research, the source code, pre-trained models, and detailed environment configurations for LS-Flow will be made publicly available. During the review process, an anonymous repository is provided at: `https://anonymous.4open.science/r/LS-Flow-66`.

Training and Evaluation Protocol. For all experiments, we report the test results averaged over five specific random seeds: $\{2022, 2023, 2024, 2025, 2026\}$. For each seed, the model is trained for a maximum of 40 epochs. We select the model checkpoint that achieves the best performance on the validation set for the final evaluation on the test set. All models are trained and evaluated on a single NVIDIA GeForce RTX 5090 GPU to ensure consistent measurement of inference latency. For a fair comparison, all baseline models are evaluated strictly using their default hyperparameter settings provided in their official implementations. Detailed hyperparameter configurations for LS-Flow across all evaluated datasets are provided in Table 7.

Table 7: Hyperparameter configurations for LS-Flow across different datasets.

Dataset	batch_size	d_model	d_ff	n_heads	flow_layers	num_sampling_steps
ECL	16	512	512	8	3	5
ETTh1	32	256	256	8	3	5
ETTh2	32	128	128	8	3	5
ETTm1	32	128	128	8	3	5
ETTm2	32	128	128	8	3	5
Solar	32	512	512	8	3	5
Traffic	16	512	512	8	3	5
Weather	32	512	512	8	3	5

C.4 Metrics

Let $\{\hat{y}_{t+h}^{(s)}\}_{s=1}^{S}$ be S generated samples at horizon step h, and y_{t+h} the ground-truth value.

CRPS. For predictive distribution F, we compute

$$\mathrm{CRPS}(F, y) = \int_{-\infty}^{\infty} (F(z) - \mathbf{1}\{y \le z\})^2 \, dz \tag{27}$$

Table 8: Energy Score (ES) on the eight benchmarks. Lower is better; bold denotes the best result within each dataset. The table reports the available results for LS-Flow, D^3U, and NsDiff.

Dataset	LS-Flow	D^3U	NsDiff
ETTh1	**17.499**	17.642	19.738
ETTh2	**15.766**	17.947	19.050
ETTm1	**15.149**	15.211	20.811
ETTm2	**12.325**	12.833	12.658
Weather	22.532	**20.906**	23.632
Solar	**51.130**	52.306	57.096
ECL	**71.847**	85.595	85.218
Traffic	**198.713**	248.114	236.051

$\mathrm{CRPS}_{\mathrm{sum}}$. We compute

$$\mathrm{CRPS}_{\mathrm{sum}} = \mathbb{E}_t \left[\mathrm{CRPS} \left(\hat{F}_{\mathrm{sum}}(t), \sum_{i=1}^{D} y_{i,t} \right) \right]. \tag{28}$$

MSE and MAE. Using the sample mean prediction $\bar{y}_{t+h} = \frac{1}{S} \sum_{s=1}^{S} \hat{y}_{t+h}^{(s)}$,

$$\mathrm{MSE} = \frac{1}{H} \sum_{h=1}^{H} (\bar{y}_{t+h} - y_{t+h})^2, \quad \mathrm{MAE} = \frac{1}{H} \sum_{h=1}^{H} |\bar{y}_{t+h} - y_{t+h}|. \tag{29}$$

Energy Score. For each forecast window, flatten the complete multivariate future trajectory into $y \in \mathbb{R}^{Hd}$ and let $\{\hat{y}^{(s)}\}_{s=1}^{S}$ be generated trajectories. We compute

$$\mathrm{ES} = \frac{1}{S} \sum_{s=1}^{S} \|\hat{y}^{(s)} - y\|_2 - \frac{1}{2S^2} \sum_{s=1}^{S} \sum_{s'=1}^{S} \|\hat{y}^{(s)} - \hat{y}^{(s')}\|_2. \tag{30}$$

The score is averaged over test windows. Lower values indicate better performance for all metrics; ES values should only be compared across methods within the same dataset.

C.5 Energy Score Results

D Full Location-Scale Ablation

Table 9: Ablations on LS-Flow. We replace different components on the source distribution and location-scale decomposition (LSD), in addition to component removal. "Pred." uses predicted future statistics $(\hat{\mu}, \hat{\sigma})$, "Hist." uses historical statistics (μ_X, σ_X) from the input X.

Design	Source	LSD	ETTm1		ETTm2		Weather		ECL	
			CRPS	$\mathrm{CRPS}_{\mathrm{sum}}$	CRPS	$\mathrm{CRPS}_{\mathrm{sum}}$	CRPS	$\mathrm{CRPS}_{\mathrm{sum}}$	CRPS	$\mathrm{CRPS}_{\mathrm{sum}}$
LS-Flow	**Pred.**	**Pred.**	**0.298**	**0.556**	**0.233**	**0.133**	**0.205**	**0.309**	**0.189**	**0.121**
Replace	Pred.	Hist.	0.308	0.558	0.239	0.133	0.209	0.313	0.193	0.129
	Hist.	Hist.	0.367	0.576	0.237	0.135	0.243	0.365	0.343	0.151
w/o	Pred.	w/o	0.326	0.588	0.267	0.145	0.217	0.316	0.193	0.127
	w/o	w/o	0.586	0.721	1.032	0.769	0.483	1.027	0.445	0.207

E Third-Order Conditional-Skewness Extension

The third-order extension augments the location-scale coordinate with a context-dependent shape parameter. We use the element-wise sinh-arcsinh map

$$h_\alpha(z) = \sinh(\operatorname{asinh}(z) + \alpha), \qquad h_\alpha^{-1}(y) = \sinh(\operatorname{asinh}(y) - \alpha). \tag{31}$$

It is strictly increasing for every $\alpha \in \mathbb{R}$, and h_0 is the identity. Its derivative is

$$h'_\alpha(z) = \frac{\cosh(\operatorname{asinh}(z) + \alpha)}{\sqrt{1 + z^2}}. \tag{32}$$

Table 10: Complete second-order (L2) and third-order (L3) comparison. Values are mean ± standard deviation over five runs; bold denotes the better result within each dataset and metric.

Dataset	Design	CRPS	CRPS_{sum}	ES	MSE	MAE
Traffic	L2	0.229 ± 0.002	0.178 ± 0.005	207.652 ± 3.760	0.451 ± 0.002	0.293 ± 0.001
	L3	$\mathbf{0.227} \pm 0.001$	$\mathbf{0.154} \pm 0.001$	$\mathbf{203.732} \pm 0.946$	$\mathbf{0.444} \pm 0.002$	$\mathbf{0.291} \pm 0.001$
Solar	L2	0.185 ± 0.006	0.228 ± 0.010	51.833 ± 1.680	$\mathbf{0.193} \pm 0.013$	0.249 ± 0.006
	L3	$\mathbf{0.178} \pm 0.005$	$\mathbf{0.190} \pm 0.007$	$\mathbf{50.516} \pm 0.244$	0.194 ± 0.005	$\mathbf{0.240} \pm 0.011$
ETTh1	L2	$\mathbf{0.331} \pm 0.004$	$\mathbf{0.630} \pm 0.006$	$\mathbf{17.903} \pm 0.620$	$\mathbf{0.449} \pm 0.007$	$\mathbf{0.437} \pm 0.005$
	L3	0.337 ± 0.003	0.632 ± 0.007	18.306 ± 0.412	0.457 ± 0.006	0.448 ± 0.004



(a) Training strategy comparison

(b) Inference time comparison

Figure 3: Training strategies and inference efficiency. (a) Performance with end-to-end and pre-trained backbones. (b) Performance versus inference time on Traffic with 100 samples.

For a context $\mathbf{X}$, the third-order reconstruction and residual coordinate are

$$\mathbf{Y} = \widehat{T}^{(3)}_{\mathbf{X}}(\mathbf{Z}) = \hat{\boldsymbol{\mu}}(\mathbf{X}) + \hat{\boldsymbol{\sigma}}(\mathbf{X}) \odot h_{\hat{\boldsymbol{\alpha}}(\mathbf{X})}(\mathbf{Z}), \tag{33}$$

$$\mathbf{Z} = h^{-1}_{\hat{\boldsymbol{\alpha}}(\mathbf{X})}\left((\mathbf{Y} - \hat{\boldsymbol{\mu}}(\mathbf{X})) \oslash \hat{\boldsymbol{\sigma}}(\mathbf{X})\right). \tag{34}$$

The Jacobian is diagonal, with determinant

$$\left|\det D\widehat{T}^{(3)}_{\mathbf{X}}(\mathbf{Z})\right| = \prod_j \hat{\boldsymbol{\sigma}}_j(\mathbf{X})\, h'_{\hat{\alpha}_j(\mathbf{X})}(Z_j). \tag{35}$$

The shape head predicts $\hat{\boldsymbol{\alpha}}(\mathbf{X})$ from the shared context representation. We train the residual Conditional Flow Matching objective in the third-order coordinate and include the above Jacobian in the conditional likelihood used to calibrate the shape head. The location-scale backbone and its predicted $\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\sigma}}$ are kept fixed for the order-wise comparison. At inference, the residual ODE is solved in $\mathbf{Z}$ and mapped back with $\widehat{T}^{(3)}_{\mathbf{X}}$; the physical vector field is obtained from the Jacobian pushforward in Theorem 1.

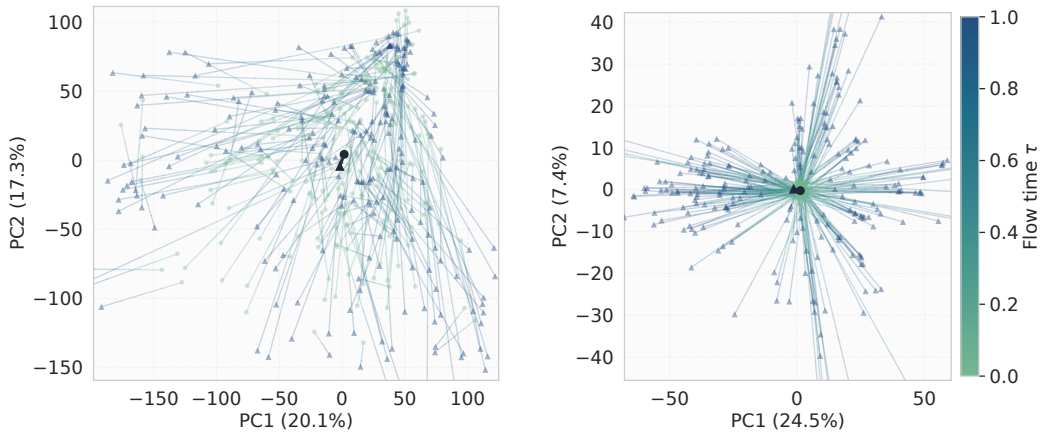
Table 10 reports the five-metric comparison on Traffic, Solar, and ETTh1. The extension improves all reported metrics on Traffic, and improves four of five metrics on Solar. On ETTh1, the additional shape degree of freedom slightly degrades performance, illustrating that higher-order residualization is data dependent.

F Training Strategy and Inference Efficiency Visualization

G Additional Model Analysis Visualizations

G.1 Geometry Transportation in Standardized Space

To complement the quantitative transport-dispersion result in Figure 2 (c), Figure 4 visualizes the same OT paths before and after location-scale decomposition for representative high-drift test windows. These panels make the geometric effect of standardization explicit: raw-space trajectories are widely scattered because they inherit instance-specific mean and scale drift, whereas the normalized trajectories become substantially more concentrated.



(a) Raw physical-space transport paths **(b) Standardized residual-space transport paths**

Figure 4: Supplementary visualization of transport geometry for the same high-drift test windows discussed in Figure 2. Left: OT paths in the raw physical space Y_τ. Right: the same paths after location-scale decomposition, i.e., $Z_\tau = (Y_\tau - \hat{\mu})/\hat{\sigma}$. The qualitative contraction of trajectories complements the quantitative dispersion reduction shown in Figure 2 (c).

G.2 Window-level Probabilistic Decoupling Analysis

To provide a more granular view of the probabilistic decoupling quality discussed in Section 5.3, we visualize window-level scatter plots for the ECL dataset in Figure 5. For each test window i, we consider S generated future samples $\{y_{i,s}\}_{s=1}^{S}$ and the corresponding ground-truth future x_i^f. We quantify the decoupling quality along two orthogonal dimensions:

Macro Distribution Error (X-axis). This metric measures whether the generated samples correctly capture the macroscopic level and scale of the target distribution. We first extract the window-level statistics for each sample and the ground truth:

$$m_{i,s} = \text{mean}(y_{i,s}), \quad v_{i,s} = \text{std}(y_{i,s}) \tag{36}$$

$$m_i^f = \text{mean}(x_i^f), \quad v_i^f = \text{std}(x_i^f) \tag{37}$$

The macro distribution error for window i is then defined as:

$$D_{\text{macro}}^{(i)} = \left(\frac{1}{S} \sum_{s=1}^{S} \left[\left(\frac{m_{i,s} - m_i^f}{v_i^f + \epsilon} \right)^2 + (\log v_{i,s} - \log v_i^f)^2 \right] \right)^{1/2} \tag{38}$$

where ϵ is a small constant for numerical stability. This error is computed as an expectation over the predictive distribution, measuring how accurately the model places the level and scale of the future samples without being a simple point-prediction metric.

Standardized Residual Distribution Error (Y-axis). This metric evaluates the model's ability to reconstruct the residual stochastic dynamics after factoring out macroscopic shifts. We first standardizedize both the samples and the ground truth to a standardized space:

$$r_{i,s} = \frac{y_{i,s} - m_{i,s}}{v_{i,s}}, \quad r_i^f = \frac{x_i^f - m_i^f}{v_i^f} \tag{39}$$

The residual distribution error is then defined as:

$$D_{\text{res}}^{(i)} = \left(\frac{1}{S} \sum_{s=1}^{S} \frac{1}{H} \| r_{i,s} - r_i^f \|_2^2 \right)^{1/2} \tag{40}$$

where H is the prediction horizon. This term captures the model's fidelity in recovering standardized microscopic dynamics once window-specific location and scale are removed.

In Figure 5, each point represents a $(D_{\text{macro}}^{(i)}, D_{\text{res}}^{(i)})$ pair for a single forecasting window. Models with better decoupling capabilities should cluster points toward the lower-left corner. LS-Flow demonstrates the most significant concentration in this region, confirming its effectiveness in isolating macroscopic non-stationarity from microscopic stochastic dynamics.

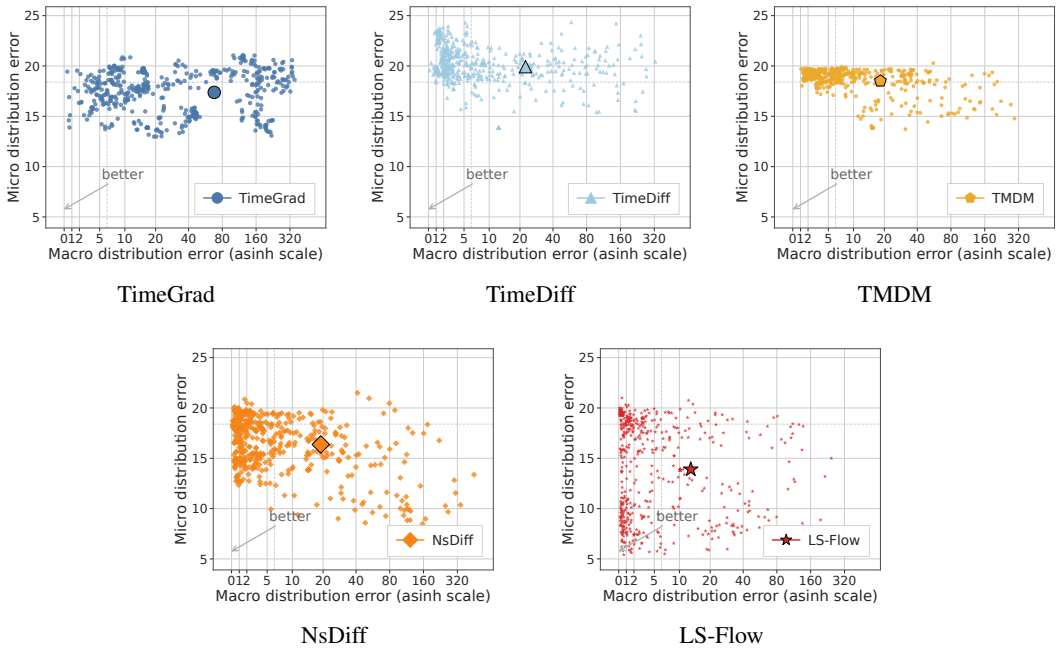


Figure 5: Window-level probabilistic decoupling scatter plots on the ECL dataset for different generative forecasting methods. Points closer to the lower-left corner indicate superior disentanglement of distribution drift from stochastic dynamics.

G.3 Qualitative Sample Showcases

To provide deeper intuition behind our quantitative findings, Figure 6 presents a qualitative visualization of the generated trajectories on the Traffic dataset. As depicted, the generated samples exhibit high fidelity; they not only accurately track the overarching macroscopic trend shifts but also faithfully preserve the microscopic stochastic fluctuations. This visual alignment validates the model's robust capacity to capture complex temporal dynamics, thereby reinforcing the quantitative superiority demonstrated in the main text.

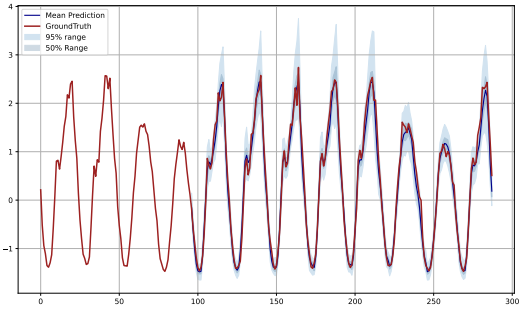
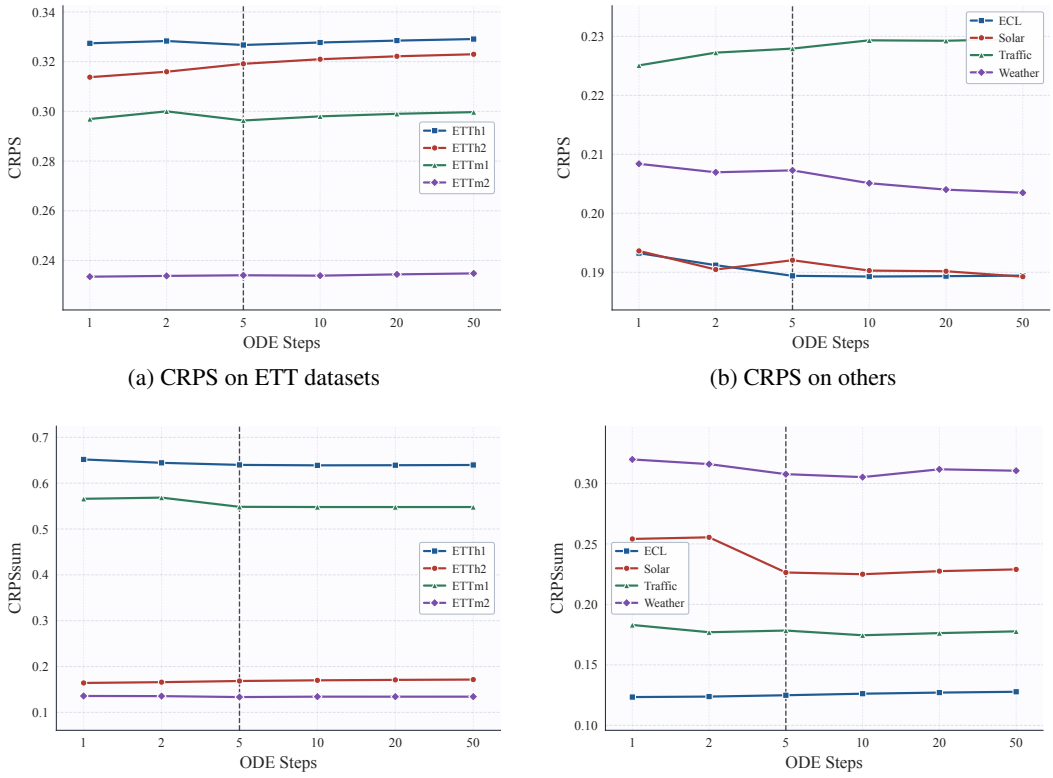


Figure 6: Qualitative forecasting showcase on the Traffic dataset.

H HYPERPARAMETER SENSITIVITY

In this section, we analyze the sensitivity of LS-Flow to the number of ODE integration steps (N) during the inference phase. Figure 7 illustrates the impact of varying the ODE steps ($N \in \{1, 2, 5, 10, 20, 50\}$) on the probabilistic forecasting performance, evaluated by CRPS and $\mathrm{CRPS}_{\mathrm{sum}}$ across all eight real-world datasets.



(a) CRPS on ETT datasets

(b) CRPS on others

(c) $\mathrm{CRPS}_{\mathrm{sum}}$ on ETT datasets

(d) $\mathrm{CRPS}_{\mathrm{sum}}$ on others

Figure 7: Hyperparameter sensitivity analysis of LS-Flow with respect to the number of ODE inference steps. The vertical dashed line marks the default setting ($N = 5$).

As shown in Figure 7, LS-Flow demonstrates remarkable robustness to the choice of integration steps. For most datasets, the performance curves are extremely flat, indicating that generating high-quality samples does not strictly require a large number of function evaluations. Notably, even with a single step ($N = 1$), LS-Flow achieves highly competitive CRPS and $\mathrm{CRPS}_{\mathrm{sum}}$ scores that often surpass those of state-of-the-art diffusion baselines. This extreme efficiency is a direct benefit of our location-scale decomposition: by mapping the complex non-stationary target into a standardized base space, the optimal transport plan is highly linearized. Consequently, the neural vector field can be accurately simulated with minimal integration steps. Throughout the main experiments, we consistently employ $N = 5$ (marked by the dashed lines), which offers a favorable balance between high predictive accuracy and inference speed.

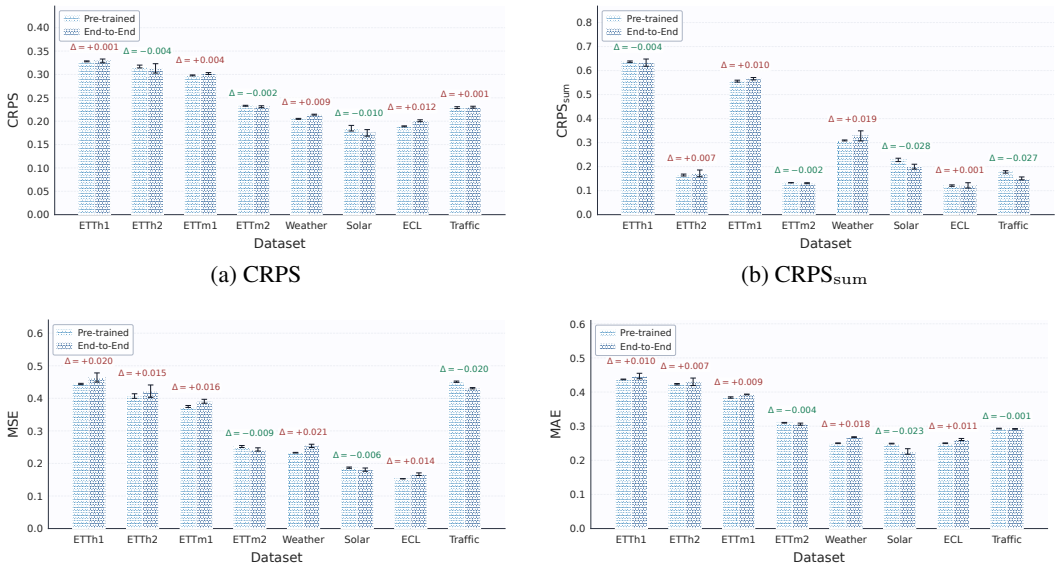
I EXTENDED ANALYSIS

I.1 TRAINING STRATEGY

We provide the complete performance breakdown for both Pre-trained and End-to-End training strategies across all evaluated metrics (CRPS, $\mathrm{CRPS}_{\mathrm{sum}}$, MSE, and MAE) in Table 11. Additionally, Fig. 8 visualizes the results for the remaining metrics.

Table 11: Comparison between Pre-Train and End-to-End training strategies of LS-Flow across 8 real-world datasets. Results are reported as mean and standard deviation.

Metric	Training Strategy	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Solar	ECL	Traffic
CRPS	Pre-trained	**0.328** ± 0.001	0.317 ± 0.003	**0.298** ± 0.001	0.233 ± 0.001	**0.205** ± 0.001	0.185 ± 0.006	**0.189** ± 0.001	**0.229** ± 0.002
	End-to-End	0.329 ± 0.004	**0.313** ± 0.010	0.302 ± 0.002	**0.231** ± 0.002	0.214 ± 0.001	**0.175** ± 0.007	0.201 ± 0.002	0.230 ± 0.002
$CRPS_{sum}$	Pre-trained	0.637 ± 0.003	**0.165** ± 0.004	**0.556** ± 0.004	0.133 ± 0.000	**0.309** ± 0.002	0.228 ± 0.007	**0.121** ± 0.003	0.178 ± 0.005
	End-to-End	**0.633** ± 0.015	0.172 ± 0.014	0.566 ± 0.005	**0.131** ± 0.002	0.328 ± 0.021	**0.200** ± 0.010	0.122 ± 0.011	**0.151** ± 0.006
MSE	Pre-trained	**0.444** ± 0.002	**0.407** ± 0.007	**0.375** ± 0.003	0.252 ± 0.003	**0.233** ± 0.001	0.187 ± 0.002	**0.153** ± 0.000	0.451 ± 0.002
	End-to-End	0.464 ± 0.014	0.422 ± 0.019	0.391 ± 0.006	**0.243** ± 0.005	0.254 ± 0.005	**0.181** ± 0.005	0.167 ± 0.004	**0.431** ± 0.002
MAE	Pre-trained	**0.437** ± 0.001	**0.423** ± 0.002	**0.384** ± 0.002	0.310 ± 0.001	**0.250** ± 0.001	0.249 ± 0.001	**0.250** ± 0.001	0.293 ± 0.001
	End-to-End	0.447 ± 0.008	0.430 ± 0.011	0.393 ± 0.001	**0.306** ± 0.003	0.268 ± 0.001	**0.226** ± 0.008	0.261 ± 0.003	**0.292** ± 0.001



(a) CRPS

(b) $CRPS_{sum}$

(c) MSE

(d) MAE

Figure 8: Training strategy ablation results for additional metrics.

I.2 Complexity Analysis

As discussed in the main text, the inference time of LS-Flow is highly efficient due to the linearized transport path in the standardized residual space. Detailed empirical comparison on the Traffic dataset is provided in Fig. 3 in the main text.

I.3 Robustness to Deterministic Backbones

To verify that our framework is robust to the choice of the underlying deterministic feature extractor, we evaluate LS-Flow using two structurally distinct backbones: iTransformer (Liu et al., 2024a) and DLinear (Zeng et al., 2023). As shown in Table 12, both configurations achieve highly competitive probabilistic and point forecasting performances across the majority of datasets.

While iTransformer's attention mechanism excels at capturing complex cross-variate dependencies—resulting in marginal gains on datasets with strong inter-channel correlations (e.g., Solar and Traffic)—DLinear's simple linear mapping still provides excellent baseline representations. This demonstrates that the high performance of LS-Flow is not merely a byproduct of a powerful backbone. Instead, it is fundamentally attributed to our hierarchical location-scale decomposition design.

Regardless of the backbone's complexity, as long as it can extract reasonable representations for the macroscopic statistics $\hat{\boldsymbol{\mu}}(\mathbf{X})$ and $\hat{\boldsymbol{\sigma}}(\mathbf{X})$, the target distribution is mapped into a more standardized residual space. The neural vector field is consequently relieved from modeling the predictable macroscopic variation, while still learning the residual temporal and cross-variate dynamics. This observed robustness across backbones supports the practical value of the explicit decomposition.

Table 12: Comparison between LS-Flow with iTransformer and DLinear backbones across 8 real-world datasets.

Metric	Backbone	ETTh1	ETTh2	ETTm1	ETTm2	Weather	Solar	ECL	Traffic
CRPS	iTransformer	**0.328**	**0.317**	**0.298**	0.233	**0.205**	**0.185**	**0.189**	**0.229**
	DLinear	0.336	0.319	0.303	**0.230**	0.217	0.269	0.222	0.281
CRPS_{sum}	iTransformer	0.637	**0.165**	0.556	**0.133**	0.309	**0.228**	**0.121**	**0.178**
	DLinear	**0.606**	0.173	**0.541**	**0.133**	**0.303**	0.269	0.151	0.197
MSE	iTransformer	**0.444**	**0.407**	**0.375**	0.252	**0.233**	**0.187**	**0.153**	**0.451**
	DLinear	0.462	0.411	**0.375**	**0.250**	0.253	0.283	0.200	0.523
MAE	iTransformer	**0.437**	**0.423**	0.384	0.310	**0.250**	**0.249**	**0.250**	**0.293**
	DLinear	0.447	0.428	**0.379**	**0.305**	0.264	0.337	0.294	0.361

I.4 Effectiveness of Flow-based Transport

Table 13 compares LS-Flow's prior distribution with the final distribution obtained after the learned Flow-based transport mapping on three real-world datasets. LS-Flow consistently achieves better results across all metrics. This indicates that although the prior distribution provides a reasonable predictive basis, the Flow-based transport mapping further refines the generated samples and moves them closer to the target future distribution, thereby improving both probabilistic and point forecasting performance.

Table 13: Comparison between LS-Flow and Prior distributions across 3 real-world datasets. Best results are in **bold**.

Methods	ETTm1				Solar				ECL			
	CRPS	CRPS_{sum}	MSE	MAE	CRPS	CRPS_{sum}	MSE	MAE	CRPS	CRPS_{sum}	MSE	MAE
LS-Flow	**0.298**	**0.556**	**0.375**	**0.384**	**0.185**	**0.228**	**0.187**	**0.249**	**0.189**	**0.121**	**0.153**	**0.250**
Prior	0.315	0.561	0.381	0.394	0.204	0.281	0.256	0.280	0.197	0.133	0.162	0.253

J BROADER IMPACTS

The second-order LS-Flow model provides a generative approach for probabilistic forecasting in non-stationary environments, with several broader societal implications. Positively, it enhances decision-making in critical infrastructure (e.g., energy grids, supply chains) and improves public safety and climate resilience by better predicting rare, high-impact events. However, potential negative impacts include the reinforcement of historical biases from training data in applications such as credit scoring or insurance, the risk of malicious manipulation in financial markets, and the danger of over-reliance on automated outputs without human oversight, particularly in unforeseen "black swan" scenarios. To mitigate these risks, we recommend integrating LS-Flow into human-in-the-loop systems, using its outputs as a reference rather than the sole basis for critical actions, and conducting rigorous bias audits alongside stress-testing on residual distributions to avoid encoding harmful historical artifacts.