LAMINAR FLOW MATCHING: GEOMETRIC MULTI-SCALE TRANSPORT FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Real-world time series exhibit stochastic dynamics across multiple temporal scales, posing a challenge for continuous-time generative forecasting. In standard Flow Matching (FM), a single vector field simultaneously models coarse global evolution and fine-scale fluctuations, which increases transport curvature and leads to inefficient dynamics. We propose Laminar Flow Matching (LFM), a geometry-aware framework that decouples multiscale probability transport into two hierarchically conditioned flows. The Macro-Flow learns a low-curvature global transport path in latent space, while the Micro-Flow models conditionally centered stochastic fluctuations in data space. This hierarchical factorization reduces cross-scale interference and improves transport smoothness and numerical efficiency. Extensive experiments on six real-world datasets show that LFM achieves state-of-the-art performance, reducing MSE by up to 15% and CRPS by 5%, while requiring fewer integration steps than existing generative baselines.

1 INTRODUCTION

Real-world time series exhibit multiscale stochastic dynamics, with long-term structure intertwined with short-term fluctuations Rahvar et al. (2024); Liu et al. (2025c). While discriminative models are computationally efficient Wen et al. (2022); Liang et al. (2024), point-wise objectives such as MSE often bias forecasts toward the conditional mean, making it difficult to model multi-modal distributions and extreme tail risks Barnett et al. (2005); Kneib et al. (2023), as illustrated in Fig.1(a). Consequently, recent advances in time-series modeling have shifted from point estimation to probabilistic generative frameworks, such as diffusion models Yuan & Qiao (2024); Yang et al. (2026) and flow matching (FM) Lipman et al. (2022); Tong et al. (2024); Kollovieh et al. (2025), which aim to model the full data distribution.

However, directly applying standard continuous-time generative models, such as FM, to time-series forecasting introduces a geometric mismatch Cao et al. (2026). FM is often designed using optimal transport (OT)-inspired probability paths that are approximately straight and low-curvature Benamou & Brenier (2000), which facilitates stable numerical integration and reduces the number of function evaluations (NFEs) Lipman et al. (2022); Tong et al. (2024). In contrast, time series exhibit multiscale dynamics. Real-world sequences consist of macroscopic regime transitions, largely driven by historical context, intertwined with high-frequency micro-oscillations that are predominantly stochastic Nason (2008); Bassamzadeh & Ghanem (2017); Wang et al. (2024a).

Forcing a single global vector field to simultaneously model multiscale dynamics induces a scale conflict Rahaman et al. (2019); Finlay et al. (2020); Cao et al. (2026). To simultaneously extrapolate structured macro-trends and reconstruct stochastic micro-oscillations, the learned transport trajectory must depart from the near-straight OT path. Thus, the learned dynamics become more curved and less smooth, increasing the stiffness of the ordinary differential equation (ODE) and making stable numerical integration more difficult Chen et al. (2018); Kelly et al. (2020). This leads to accumulated integration errors, degraded stability near extrema (Fig.1(b)), and a substantial increase in NFEs. Kelly et al. (2020); Kim et al. (2021).

To address this geometric bottleneck, we propose Laminar Flow Matching (LFM), a geometry-aware generative framework that factorizes continuous probability transport into two hierarchically conditioned flows through block-wise sequence partitioning. Instead of relying on a single ODE

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

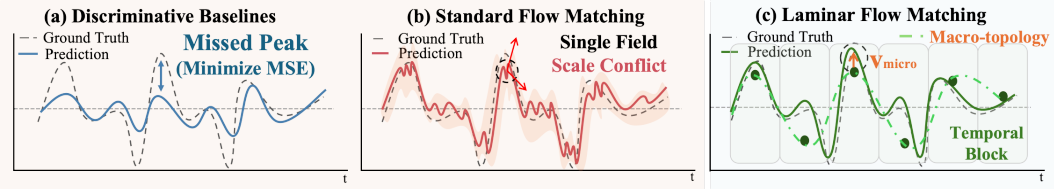


Figure 1: Comparison of time-series methods.(a) Discriminative baselines regress to the conditional mean. (b) Standard Flow Matching uses a single field, causing scale conflict. (c) Laminar Flow Matching decouples macro transitions and micro fluctuations, resulting in more stable dynamics.

vector field to capture coupled multiscale dynamics, LFM assigns distinct roles to each flow, thereby enhancing the fidelity of transport geometry.

Specifically, the *Outer Macro-Flow* evolves sequences in a latent space by formulating inter-block macroscopic dynamics as an optimal transport problem, producing straight and low-curvature global trajectories. Conditioned on this macroscopic structure, the *Inner Micro-Flow* operates in data space to model intra-block fine-grained stochastic fluctuations. By separating structured macro-trends from micro-scale volatility, LFM alleviates the scale conflict in the vector field and yields smooth, low-curvature trajectories reminiscent of laminar flow, as illustrated in Fig.1(c).

We evaluate LFM on six real-world benchmarks. Extensive experiments demonstrate that LFM achieves state-of-the-art results in both point forecasting and probabilistic modeling. Quantitatively, LFM outperforms the best discriminative methods by up to 15% in MSE and surpasses leading generative baselines by 5% in CRPS, while requiring significantly fewer function evaluations.

Our contributions are summarized as follows:

- We provide theoretical evidence for a geometric bottleneck in time-series flow matching, showing that multiscale dynamics induce high-curvature transport trajectories, which degrade transport geometry and numerical stability.
- We propose Laminar Flow Matching (LFM), a geometry-aware dual-flow framework that decouples probability transport into a low-curvature Macro-Flow and a stochastic Micro-Flow, thereby alleviating scale conflict and stabilizing ODE integration.
- Across six real-world benchmarks, LFM achieves state-of-the-art performance, reducing MSE by up to 15% over discriminative models and improving CRPS by 5% over generative baselines, while requiring fewer function evaluations and maintaining robustness.

2 Related Work

Discriminative Time-Series Modeling. Deep learning has advanced time-series forecasting Lim & Zohren (2021); Kong et al. (2025), evolving from early recurrent networks Bengio et al. (1994); Pascanu et al. (2013) to specialized architectures for complex temporal dynamics Wen et al. (2022); Ahmed et al. (2023). Early works improved long-range modeling via efficient attention, decomposition, and frequency methods, as in Informer Zhou et al. (2021), Autoformer Wu et al. (2021), and FEDformer Zhou et al. (2022). Subsequent approaches enhanced local and multivariate modeling with patch-based and structural designs, including PatchTST Nie et al. (2023) and iTransformer Liu et al. (2024), while linear models such as DLinear Zeng et al. (2023) and FITS Xu et al. (2024) offered strong efficiency. Recent methods move beyond 1D sequences, with TimesNet Wu et al. (2022), TimeMixer Wang et al. (2024a), and TimeXer Wang et al. (2024c) capturing multi-periodicity, multiscale structure, and exogenous dependencies. More recently, scaling laws have further driven large-scale time-series foundation models (TSFMs) Liang et al. (2024); Miller et al. (2024), including TimesFM Das et al. (2024), Chronos Ansari et al. (2024), Moirai Woo et al. (2024), Timer-XL Liu et al. (2025a), Time-MoE Shi et al. (2025), and Sundial Liu et al. (2025b), which leverage generalized tokenization, patching, and large-scale pre-training for strong zero-shot performance. *However, most models remain discriminative with point-wise objectives, biasing predictions toward the conditional mean and over-smoothing outputs, thereby failing to capture high-*

frequency stochasticity Barnett et al. (2005); Kneib et al. (2023); Karaouli et al. (2025); Xu et al. (2025); Khanal et al. (2026).

Generative Models for Time-Series. To model complex predictive distributions, continuous-time generative frameworks have shifted from point estimation to full distribution learning. Early efforts adapted score-based diffusion models Lin et al. (2024); Yang et al. (2026) to capture temporal stochasticity and multivariate dependencies, as in TimeGrad Rasul et al. (2021), CSDI Tashiro et al. (2021), and Diffusion-TS Yuan & Qiao (2024). Subsequent Flow Matching (FM) methods Lipman et al. (2022); Tong et al. (2024) provided a simulation-free alternative by constructing straighter optimal transport (OT) paths Benamou & Brenier (2000); Peyré & Cuturi (2019), as in CGFM Xu et al. (2025), TSFlow Kollovieh et al. (2025), and Sundial Liu et al. (2025b). While path-straightening techniques such as Rectified Flow Liu et al. (2022) aim to linearize transport, the intrinsic multi-frequency nature of time series Nason (2008); Wang et al. (2024a), together with neural spectral bias Rahaman et al. (2019), often drives the vector field away from low-curvature trajectories. To mitigate this, specialized designs attempt to disentangle stable trends from stochastic residuals, including frequency-aware filters Yang et al. (2024) and time-gated multi-scale velocity heads Wang & Sugiyama (2026). *However, a single global ODE still entangles regime shifts with local fluctuations, increasing transport curvature and numerical cost.* LFM instead separates macro and micro transport without predefined frequency bands, targeting cross-scale interference rather than source–target alignment as in TSFlow.

3 PRELIMINARIES

3.1 PROBLEM DEFINITION

We consider a multivariate time series $\boldsymbol{x} \in \mathbb{R}^{T \times D}$ with T time steps and D features:

$$\boldsymbol{x} = [\boldsymbol{x}^h, \boldsymbol{x}^f] = \Big[\underbrace{x_1, \ldots, x_{T_h}}_{\text{history}}, \underbrace{x_{T_h+1}, \ldots, x_{T_h+T_f}}_{\text{future}}\Big], \tag{1}$$

where $\boldsymbol{x}^h$ and $\boldsymbol{x}^f$ denote the historical context and future target of lengths T_h and T_f, respectively. Our objective is to model the conditional distribution $p_\theta(\boldsymbol{x}^f \mid \boldsymbol{x}^h)$ via a continuous probability transport formulation, capturing the multiscale stochasticity in temporal dynamics.

3.2 CONDITIONAL FLOW MATCHING (CFM)

CFM Lipman et al. (2022); Tong et al. (2024) defines a probability path p_t from a tractable prior p_0 at $t = 0$ to the target distribution p_1 at $t = 1$, governed by the ODE:

$$d\boldsymbol{z}_t = v(\boldsymbol{z}_t, t)dt, \quad t \in [0, 1] \tag{2}$$

where v is a time-dependent vector field. FM learns v via simulation-free supervision from per-sample paths. Under the OT coupling, interpolation between $\boldsymbol{z}_0 \sim p_0$ and $\boldsymbol{z}_1 \sim p_1$ is linear:

$$\boldsymbol{z}_t = (1 - t)\boldsymbol{z}_0 + t\boldsymbol{z}_1 \tag{3}$$

which induces a constant target field $v_t(\boldsymbol{z}_t \mid \boldsymbol{z}_0, \boldsymbol{z}_1) = \boldsymbol{z}_1 - \boldsymbol{z}_0$. In the conditional setting, the vector field is conditioned on $\boldsymbol{x}^h$. A neural network v_θ is trained to match this field:

$$\mathcal{L}_{\text{CFM}} = \mathbb{E}_{t, \boldsymbol{z}_0, \boldsymbol{z}_1}\left[\|v_\theta(\boldsymbol{z}_t, t \mid \boldsymbol{x}^h) - (\boldsymbol{z}_1 - \boldsymbol{z}_0)\|^2\right] \tag{4}$$

Formally, we cast the time series forecasting problem into the CFM framework by defining the target distribution $p_1(\cdot \mid \boldsymbol{x}^h)$ as the conditional distribution of future $\boldsymbol{x}^f$ given historical context $\boldsymbol{x}^h$.

3.3 GEOMETRIC MOTIVATION

To explain why standard FM struggles on multiscale time series, we study the geometry induced by targets that simultaneously exhibit macro-trends and rapid micro-scale fluctuations. Such multiscale structure creates a scale conflict: a single vector field must accommodate both coarse transport and rapid local variation, increasing geometric complexity and potentially inducing stiff dynamics.

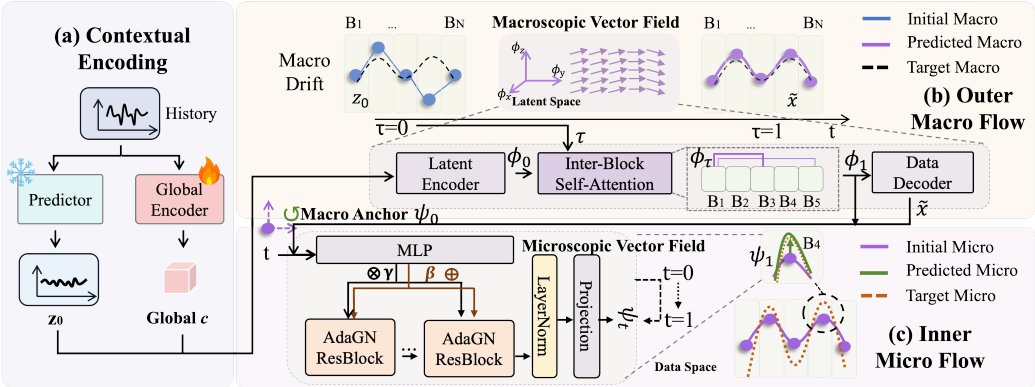


Figure 2: Laminar Flow Matching (LFM). (a) Contextual Encoding builds a prior and global context. (b) Outer Macro-Flow models low-curvature transitions in latent space. (c) Inner Micro-Flow captures stochastic fluctuations conditioned on macro-states, enabling stable transport.

Scale Conflict. To capture this geometric tension, consider an abstract state-space trajectory parameterized by $s \in [0, 1]$:

$$\gamma(s) = \gamma^{(L)}(s) + a \sin(\omega s)\boldsymbol{e}, \tag{5}$$

where $\gamma^{(L)}$ denotes the smooth macroscopic component, s denotes the transport parameter rather than the temporal index of the observed sequence, $\|\boldsymbol{e}\| = 1$, $a > 0$, and ω controls the oscillation rate of the fine-scale component along the transport path. Eq. 5 is an analytical construction rather than a statistical model of the observed time series. The sinusoidal term simply provides an interpretable instance of fine-scale variation that induces large trajectory acceleration.

We assume that the macroscopic component $\gamma^{(L)}$ is smooth with bounded derivatives, i.e., $\|\gamma^{(L)\prime}(s)\| \le M$ and $\|\gamma^{(L)\prime\prime}(s)\| \le B$, while the associated transport field has uniformly bounded explicit variation along the trajectory: $\sup_{s\in[0,1]} \|\partial_s v(z(s), s)\| \le C_s$.

Proposition 1 (Scale Conflict in Single-Field FM)**.** *Under the above setting,*

$$\sup_s \|\gamma''(s)\| \ge a\omega^2 - B, \quad \sup_s \|\nabla_z v(z(s), s)\|_{\text{op}} \ge \frac{a\omega^2 - B - C_s}{M + a\omega}. \tag{6}$$

Proof is provided in Appendix B. The connection to scale conflict follows from $\gamma''(s) = \partial_s v(\gamma(s), s) + \nabla_z v(\gamma(s), s)\gamma'(s)$: when explicit variation in s is bounded, larger trajectory acceleration requires larger spatial variation of the shared vector field, as quantified by Proposition 1.

Thus, faster fine-scale variation requires greater local variation in the shared transport field, increasing its tendency toward stiffness. This motivates LFM to separate smooth macro transport from fine-scale corrections. The proposition establishes this tendency as a sufficient condition rather than a tight universal bound.

4 Methodology

Standard continuous-time generative models struggle to jointly model macroscopic trends and microscopic fluctuations within a single vector field, leading to a scale conflict in the induced dynamics. To address this, Laminar Flow Matching (LFM) introduces a geometry-aware decoupling of multiscale dynamics, implemented as two coordinated flows over distinct integration axes (Fig. 2). (a) *Contextual Encoding* first constructs the conditioning prior and global context (Sec. 4.1). Conditioned on this context, the (b) *Outer Macro-Flow* evolves along a latent axis $\tau \in [0, 1]$ to capture low-curvature macroscopic transport across blocks(Sec. 4.2). The resulting macroscopic anchor then conditions the (c) *Inner Micro-Flow*, which operates on the data-space axis $t \in [0, 1]$ to model fine-scale stochastic fluctuations locally within each block(Sec. 4.3).

4.1 Contextual Prior Construction

Unlike standard CFM, which starts from a Gaussian prior, we construct an informative prior from the historical context $\boldsymbol{x}^h$. To capture global temporal patterns, we use a pre-trained deterministic

predictor $\mathcal{F}_\phi$ Zeng et al. (2023), to extrapolate the historical sequence into a coarse forecast:

$$\boldsymbol{\mu} = \mathcal{F}_\phi(\boldsymbol{x}^h) \in \mathbb{R}^{T_f \times D}. \tag{7}$$

During training, we initialize the prior as

$$\boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I}). \tag{8}$$

Meanwhile, the historical context is encoded into a high-dimensional global representation $\boldsymbol{c}$, which serves as the conditioning for the macroscopic vector field.

4.2 Outer Macro-Flow

The Outer Macro-Flow models low-curvature large-scale transport, isolating global regime transitions from high-frequency fluctuations. Since individual time steps contain limited information for regime-level evolution, we partition each sequence into contiguous blocks of length L to form coarser temporal units. To suppress micro-scale variability, macro-level transport is performed in a compact latent space, where large-scale regime transitions become geometrically smoother.

We construct a compact block-level latent space using an encoder $\mathcal{E}$ and a decoder $\mathcal{B}$. The deterministic encoder uses convolutional token embedding, an AutoCorrelation-based temporal encoder, and a linear projection to produce one latent token per block, attenuating micro-scale variability. Stochasticity arises from the context-informed prior rather than variational sampling. During training, $\mathcal{E}$ maps both the initial prior $\boldsymbol{z}_0$ and the ground-truth future $\boldsymbol{x}^f$ into the latent space:

$$\boldsymbol{\phi}_0 = \mathcal{E}(\boldsymbol{z}_0), \quad \boldsymbol{\phi}_1 = \mathcal{E}(\boldsymbol{x}^f), \tag{9}$$

where $\boldsymbol{\phi}_0, \boldsymbol{\phi}_1 \in \mathbb{R}^{N \times d_l}$, N is the number of blocks and d_l is the latent dimension. This projection constrains the macroscopic evolution to a noise-robust manifold.

Latent Optimal Transport. Latent abstraction reduces local variability but preserves uncertainty in macro-level evolution. We therefore model latent evolution via conditional flow matching rather than direct regression. Specifically, over $\tau \in [0, 1]$, we define

$$\boldsymbol{\phi}_\tau = (1 - \tau)\boldsymbol{\phi}_0 + \tau \boldsymbol{\phi}_1, \tag{10}$$

which induces the constant target velocity $\boldsymbol{u}^* = \boldsymbol{\phi}_1 - \boldsymbol{\phi}_0$. The macro-flow is trained by the flow matching objective

$$\mathcal{L}_{\text{macro}} = \mathbb{E}_{\tau, \boldsymbol{\phi}_0, \boldsymbol{\phi}_1} \left[\| u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}) - \boldsymbol{u}^* \|^2 \right], \tag{11}$$

where $\boldsymbol{c}$ denotes the encoded global representation.

Inter-Block Coordination Network. To parameterize u_θ, we model dependencies across latent blocks $\boldsymbol{\phi}_\tau \in \mathbb{R}^{N \times d_l}$. Since independent block-wise prediction cannot capture coherent macro-trends, we employ self-attention Vaswani et al. (2017) to coordinate information across blocks. Let $\mathbf{H}_\tau \in \mathbb{R}^{N \times d_h}$ denote the joint embedding of $\boldsymbol{\phi}_\tau$, τ, $\boldsymbol{c}$, where d_h is the hidden dimension. The vector field is parameterized as

$$\mathbf{H}_\tau = \text{MLP}([\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}]), \tag{12}$$

$$u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}) = \text{SelfAttn}(\mathbf{H}_\tau)\boldsymbol{W}, \tag{13}$$

where $\boldsymbol{W} \in \mathbb{R}^{d_h \times d_l}$ is a learnable projection matrix, τ and $\boldsymbol{c}$ are broadcasted and concatenated to each block feature in $\boldsymbol{\phi}_\tau$. Conditioning on $\boldsymbol{c}$ enables adaptation to macroscopic regime shift, while self-attention aggregates block-level signals to parameterize a coherent macro vector field.

Under the OT formulation, the target velocity $\boldsymbol{u}^*$ along the linear path is constant. Hence, from any intermediate state $\boldsymbol{\phi}_\tau$ at time τ, the terminal state can be estimated by extrapolating the predicted velocity over the remaining interval $(1 - \tau)$. We thus estimate the terminal macro state as

$$\hat{\boldsymbol{\phi}}_1 = \boldsymbol{\phi}_\tau + (1 - \tau)\, u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}). \tag{14}$$

The decoder $\mathcal{B}$ maps the estimated macro state back to the data space:

$$\tilde{\boldsymbol{x}} = \mathcal{B}(\hat{\boldsymbol{\phi}}_1). \tag{15}$$

To ensure that $\tilde{\boldsymbol{x}}$ captures the global trend, we introduce an anchoring loss. A smoothed target $\boldsymbol{x}_{\text{s}}^f$ is constructed from $\boldsymbol{x}^f$ using 1D average pooling, and the decoder is trained to match it:

$$\mathcal{L}_{\text{anchor}} = \left\| \tilde{\boldsymbol{x}} - \boldsymbol{x}_{\text{s}}^f \right\|^2, \tag{16}$$

which encourages $\tilde{\boldsymbol{x}}$ to capture the global trend, leaving high-frequency details to the Micro-Flow.

4.3 INNER MICRO-FLOW

The Inner Micro-Flow models stochastic fine-scale dynamics conditioned on the macro anchor, without interfering with macro-scale transport. Since global inter-block dependencies are already captured by the Outer Macro-Flow, the Inner Micro-Flow operates locally within each block to model high-frequency fluctuations in data space.

Given the prediction $\tilde{\boldsymbol{x}}$, the Inner Micro-Flow refines the macro prediction while modeling fine-scale stochastic variations. Anchoring training on the ground-truth macro state would introduce a train–inference mismatch, since inference is conditioned on imperfect macro estimates produced by the Outer Macro-Flow. We therefore construct a dynamic base using the predicted macro trend:

$$\boldsymbol{\psi}_0 = \mathrm{sg}(\tilde{\boldsymbol{x}}), \quad \boldsymbol{\psi}_1 = \boldsymbol{x}^f, \quad \boldsymbol{\psi}_t = (1-t)\boldsymbol{\psi}_0 + t\boldsymbol{\psi}_1. \tag{17}$$

Although $\tilde{\boldsymbol{x}}$ evolves across training iterations, the stop-gradient operator $\mathrm{sg}(\cdot)$ Van Den Oord et al. (2017) ensures the path is fixed within each optimization step and blocks gradients from the Micro-Flow to the Macro-Flow. This ensures that the Micro-Flow is trained on the same class of imperfect anchors encountered during inference. We warm up the Macro-Flow for E_w epochs before jointly training the Micro-Flow, which stabilizes the dynamic base during the early stages of optimization.

The dynamics are governed by $d\boldsymbol{\psi}_t = v_\theta(\boldsymbol{\psi}_t, t, \tilde{\boldsymbol{\phi}})\, dt$, where v_θ is conditioned on a latent macro representation $\tilde{\boldsymbol{\phi}}$ derived from the predicted latent macro state. During training, we define:

$$\tilde{\boldsymbol{\phi}} = \mathrm{sg}(\hat{\boldsymbol{\phi}}_1) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \eta^2 \mathbf{I}), \tag{18}$$

where $\hat{\boldsymbol{\phi}}_1$ is the one-step latent estimate from Eq.14. The injected noise aligns training with stochastic inference and acts as noise-conditioned augmentation Ho et al. (2022), reducing reliance on exact macro estimates and improving robustness to extrapolation errors.

We parameterize the Micro-Flow vector field v_θ with a 1D convolutional backbone modulated by Adaptive Group Normalization (AdaGN) Dhariwal & Nichol (2021), enabling parallel modeling of intra-block fine-scale dynamics. The integration-time embedding and predicted macro representation $\tilde{\boldsymbol{\phi}}$ jointly generate feature-wise scale and shift parameters for the local Conv1D features, allowing macro-conditioned refinement while reducing cross-scale interference.

Given the macro anchor $\boldsymbol{\psi}_0$, the Micro-Flow primarily models the remaining fine-scale stochastic variations. The target field is $\boldsymbol{v}^* = \boldsymbol{\psi}_1 - \boldsymbol{\psi}_0$, and the model is trained via

$$\mathcal{L}_{\text{micro}} = \mathbb{E}_{t, \boldsymbol{\psi}_0, \boldsymbol{\psi}_1} \left[\| v_\theta(\boldsymbol{\psi}_t, t, \tilde{\boldsymbol{\phi}}) - \boldsymbol{v}^* \|^2 \right]. \tag{19}$$

Here, $\boldsymbol{v}^*$ represents the residual discrepancy after macro prediction and include both fine-scale variations and residual macro prediction errors.

4.4 JOINT TRAINING AND PROBABILISTIC INFERENCE

Training. To capture structured macro-trends and stochastic micro-oscillations without scale conflict, LFM is optimized with

$$\mathcal{L} = \mathcal{L}_{\text{macro}} + \mathcal{L}_{\text{micro}} + \lambda \mathcal{L}_{\text{anchor}}. \tag{20}$$

Probabilistic Inference. At inference, unlike the single-step estimate used during training, we integrate the Outer Macro-Flow for improved numerical stability. Specifically, we discretize $\tau \in [0, 1]$ into P steps and apply Euler integration:

$$\boldsymbol{\phi}_{\tau+\Delta\tau} = \boldsymbol{\phi}_\tau + \Delta\tau \cdot u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}), \tag{21}$$

yielding the macro-anchor $\hat{\boldsymbol{\phi}}_1 = \boldsymbol{\phi}_{\tau=1}$. Unlike training, no latent perturbation is applied at inference, and the conditioning variable reduces to $\tilde{\boldsymbol{\phi}} = \hat{\boldsymbol{\phi}}_1$. The Inner Micro-Flow is then integrated via an explicit Euler solver over K steps ($t \in [0, 1]$):

$$\boldsymbol{\psi}_{t+\Delta t} = \boldsymbol{\psi}_t + \Delta t \cdot v_\theta(\boldsymbol{\psi}_t, t, \tilde{\boldsymbol{\phi}}). \tag{22}$$

The terminal state $\hat{\boldsymbol{x}}^f = \boldsymbol{\psi}_1$ is taken as the generated sequence. Repeating this procedure S times yields an ensemble of forecasts. The full procedure of training and inference is in Appendix C.

Table 1: Forecasting results on representative standard benchmarks. $T_h = 96$ and $T_f \in \{96, 192, 336, 720\}$. Averaged results of four prediction lengths are reported here. Full results on all datasets and prediction lengths are provided in Appendix Table 13 and 14.

Method	Electricity			Traffic			Exchange			ETTm1			Average		
	MSE ↓	MAE ↓	CRPS ↓	MSE ↓	MAE ↓	CRPS ↓	MSE ↓	MAE ↓	CRPS ↓	MSE ↓	MAE ↓	CRPS ↓	MSE ↓	MAE ↓	CRPS ↓
AF (2021)	0.214	0.290	0.287	0.617	0.637	0.633	0.610	0.345	0.346	0.514	0.303	0.293	0.489	0.394	0.390
FED (2022)	0.213	0.295	0.288	0.579	0.631	0.636	0.510	0.356	0.341	0.519	0.310	0.291	0.455	0.398	0.389
PTST (2023)	0.249	0.283	0.280	0.448	0.588	0.583	0.447	0.338	0.360	0.444	0.280	0.274	0.397	0.372	0.374
DLinear (2023)	0.259	0.285	0.284	0.516	0.584	0.584	0.491	0.373	0.371	0.456	0.285	0.284	0.431	0.382	0.381
iTrans (2024)	0.175	0.280	0.277	0.469	0.600	0.597	0.456	0.341	0.338	0.452	0.288	0.283	0.388	0.377	0.374
TimeXer (2024c)	0.171	0.274	0.268	0.467	0.598	0.594	0.445	0.337	0.331	0.382	0.269	0.262	0.366	0.370	0.364
FITS (2024)	0.259	0.272	0.277	0.496	0.577	0.573	0.510	0.377	0.372	0.445	0.284	0.286	0.428	0.378	0.377
CSDI (2021)	0.343	0.284	0.268	0.987	0.713	0.582	0.711	0.434	0.346	0.822	0.349	0.263	0.716	0.445	0.365
TimeGrad (2021)	0.334	0.275	0.267	1.003	0.623	0.568	0.701	0.455	0.347	0.812	0.361	0.256	0.713	0.429	0.360
DTS (2024)	0.330	0.270	0.267	0.973	0.710	0.530	0.602	0.421	0.315	0.693	0.349	0.244	0.650	0.438	0.339
CGFM (2025)	0.342	0.278	0.270	0.966	0.667	0.531	0.623	0.413	0.303	0.435	0.355	0.249	0.592	0.428	0.338
TSFlow (2025)	0.329	0.274	0.269	0.975	0.656	0.527	0.625	0.399	0.321	0.695	0.345	0.243	0.656	0.419	0.340
LFM(Ours)	**0.129**	**0.258**	**0.257**	**0.390**	**0.571**	**0.495**	**0.385**	**0.311**	**0.299**	**0.332**	**0.264**	**0.234**	**0.309**	**0.351**	**0.321**
Rel. Improv.	24.6%	4.4%	3.7%	12.9%	1.0%	6.1%	13.5%	7.7%	1.3%	13.1%	1.9%	3.7%	15.6%	5.1%	5.0%

Table 2: Comparison on AEMO with $T_h = 96$ and $T_f \in \{96, 192, 336, 720\}$. Results are averaged across all regions and forecasting horizons; full results are reported in Appendix Table 15.

Metric	Discriminative Methods							Generative Methods					Our Method
	AF (2021)	FED (2022)	PTST (2023)	DLinear (2023)	iTrans (2024)	TimeXer (2024c)	FITS (2024)	CSDI (2021)	TimeGrad (2021)	DTS (2024)	CGFM (2025)	TSFlow (2025)	LFM (Ours)
MSE(↓)	0.807	0.794	0.774	0.767	0.733	0.726	0.769	0.823	0.810	0.790	0.779	0.761	**0.618** (14.9% ↓)
MAE(↓)	0.698	0.693	0.682	0.675	0.664	0.659	0.670	0.715	0.700	0.684	0.674	0.664	**0.525** (20.3% ↓)
CRPS(↓)	0.680	0.673	0.661	0.655	0.653	0.646	0.644	0.684	0.674	0.656	0.633	0.565	**0.518** (8.3% ↓)

5 EXPERIMENT

5.1 EXPERIMENTAL SETUP

Datasets. We evaluate Laminar Flow Matching (LFM) on six diverse real-world benchmarks: (a) **Standard Benchmarks** Lai et al. (2018); Zhou et al. (2021); Wu et al. (2021): **Traffic**, **Electricity**, **Weather**, **ETT**, and **Exchange**, spanning diverse temporal patterns and forecasting regimes. (b) **High-Volatility Benchmarks** Wang et al. (2024b): the **AEMO** electricity demand datasets across five Australian regions (TAS, VIC, NSW, QLD, SA), characterized by short-term variability and stochastic dynamics. Dataset statistics are summarized in Appendix D.

Baselines. We compare LFM against a comprehensive suite of state-of-the-art models: (1) **Discriminative methods**: Autoformer Wu et al. (2021), FEDformer Zhou et al. (2022), PatchTST Nie et al. (2023), DLinear Zeng et al. (2023), iTransformer Liu et al. (2024), TimeXer Wang et al. (2024c) and FITS Xu et al. (2024). (2) **Generative methods**: CSDI Tashiro et al. (2021), TimeGrad Rasul et al. (2021), Diffusion-TS Yuan & Qiao (2024), CGFM Xu et al. (2025),and TSFlow Kollovieh et al. (2025). Method details are in the related work section.

Evaluation Metrics. We evaluate forecasting performance from two perspectives. For **point forecasting**, we report Mean Squared Error (MSE) and Mean Absolute Error (MAE), where point forecasts are obtained via deterministic inference from a fixed initial state ($\sigma = 0$ in Eq.8). For **probabilistic quality**, we adopt the Continuous Ranked Probability Score (CRPS) Gneiting & Raftery (2007), computed from $S = 100$ samples generated by initializing $\mathbf{z}_0 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I})$ in Eq.8 and propagating them through the generative model. Lower values indicate better performance for all three metrics.

Implementation Details. All experiments are conducted on a single NVIDIA H100 GPU. We pretrain $\mathcal{F}_\phi$ for 20 epochs (lr 10^{-3}, bs 32) and freeze it, then warm-start the macro-flow u_θ for $E_w = 5$ epochs (lr 10^{-4}), followed by 15 epochs of joint training with the micro-flow v_θ. The model uses latent and hidden dimensions $d_l = 24$ and $d_h = 128$, with 0.72M parameters. We use $L = 24$ and $\lambda = 0.5$ across all datasets, with sensitivity analyses provided in Appendix J. At inference, we employ a ODE solver with $P = 5$ macro steps and $K = 10$ micro steps, and draw $S = 100$ samples. In comparison, Diffusion-TS requires $K = 500$ denoising steps. Additional efficiency results are provided in Appendix E, and code is included in the supplementary material.

5.2 Main Results on Time Series Benchmarks

Results on standard benchmarks. As reported in Table 1, LFM achieves state-of-the-art performance across all benchmarks, improving MSE and CRPS by 15.6% and 5.0% on average. On datasets with multiscale dynamics, such as Traffic and Electricity, standard flow-based models (e.g., CGFM and TSFlow) often struggle, as a single global vector field induces curved trajectories and unstable integration. In contrast, LFM's geometry-aware dual-flow design better separates heterogeneous dynamics, leading to improved accuracy.

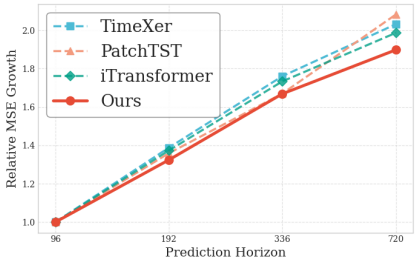


Figure 3: Relative MSE growth across forecasting horizons, normalized to the base horizon $T_f = 96$.

As shown in Fig.3, prediction errors increase with the forecast horizon for all baselines, whereas LFM degrades more gradually. Compared with iTrans, whose MSE rises by 99% from $T_f = 96$ to 720, LFM increases by 89%, indicating stronger long-horizon robustness.

To investigate LFM's stability, we compare the empirical distribution of vector-field magnitudes in Fig.4, which serves as a proxy for transport complexity. Compared with the single-field baseline CGFM (blue), LFM's Inner Micro-Flow (red) concentrates in a lower-magnitude regime, indicating smoother local transport dynamics. This supports our hypothesis that decoupling multiscale dynamics improves transport geometry. Direct trajectory measurements further show that latent Macro-Flow produces straighter and smoother paths than data-space macro transport (Appendix K).

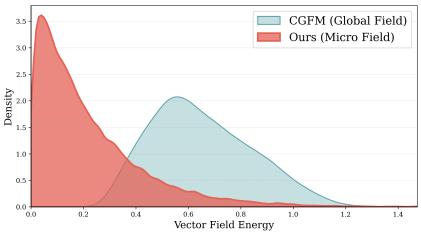


Figure 4: Distribution of vector-field magnitudes.

Results on the AEMO dataset. Table 2 reports the results on AEMO. LFM closes the gap between discriminative and generative paradigms, outperforming TimeXer by 14.9% in MSE (0.618 vs. 0.726) and 20.3% in MAE (0.525 vs. 0.659), while surpassing TSFlow by 8.3% in CRPS (0.518 vs. 0.565). These results highlight LFM's ability to provide both accurate point forecasts and well-calibrated uncertainty estimates in real-world markets.

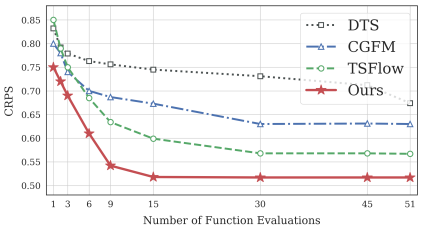


Figure 5: Inference efficiency.

By decoupling multiscale dynamics, LFM reduces trajectory stiffness and enables stable integration with fewer solver steps. Using a 1:2 Outer–Inner step allocation, LFM achieves its best CRPS with only 15 steps (5 macro, 10 micro), whereas the strongest baseline, TSFlow, requires 30 steps, as shown in Fig.5. This highlights LFM's simultaneous gains in efficiency and accuracy.

Appendix F demonstrates LFM's cross-domain transferability against recent foundation models.

5.3 Visual Analysis and Case Studies

Visual Analysis. As shown in Fig.6, LFM (red) tracks the ground truth (blue) more closely across diverse datasets, capturing both periodic peaks and abrupt fluctuations. In contrast, TimeXer (green) and TSFlow (orange) tend to oversmooth or underestimate highly volatile segments. These results suggest that LFM's dual-flow design better preserves complex temporal structure.

Spectral Analysis. To assess fine-scale fidelity, we compare the power spectral density (PSD) of predicted sequences in Fig.7. LFM (red) more closely matches the ground-truth spectral profile (black), particularly in high-frequency regions that are often attenuated by baseline models. This indicates that decoupled transport enables more accurate modeling of stochastic fine-scale dynamics.

Tail-Risk Analysis. We further evaluate forecasting performance on the top 5% most extreme target values. LFM substantially reduces Tail-MSE over TimeXer by 20.5% on ETTm1 and 17.2% on SA, indicating improved robustness to extreme dynamics. These results are consistent with the in-

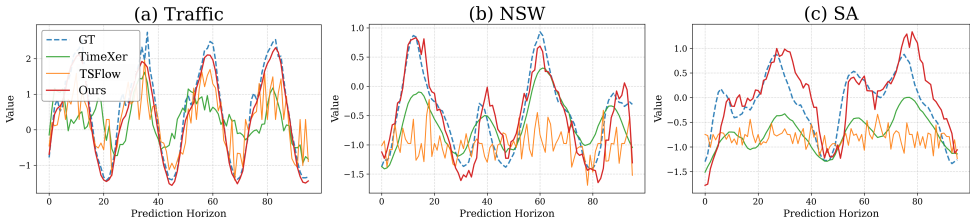


Figure 6: Visual comparison of prediction results in the time domain.

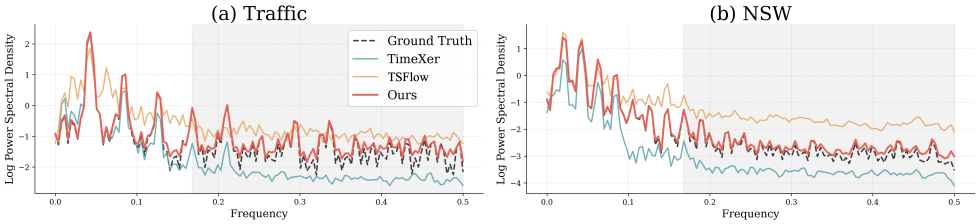


Figure 7: Power spectral density analysis highlighting high-frequency recovery.

tended Macro–Micro decomposition, which separates global regime evolution from local stochastic refinement. Detailed results are reported in Appendix G.

5.4 Ablation Study

We validate the design choices of LFM through ablations in Table 3. Additional architectural comparisons are provided in Appendix H.

Prior and Context. Removing the informed prior or global context consistently degrades performance, confirming the importance of history-informed initialization and conditioning.

Framework Decoupling. Removing either flow causes clear degradation, with w/o Macro-Flow increasing the average MSE by 12.1% (0.497 vs. 0.557). Performing macro transport directly in data space also degrades performance, supporting the latent-space decomposition. A capacity-matched TSFlow-Large (0.70M vs. LFM's 0.72M) remains substantially worse, ruling out parameter count as the primary source of improvement (Appendix I).

Training Mechanisms. Removing the dynamic base, noise augmentation, or anchor loss consistently degrades performance, validating their roles in train–inference alignment and stable transport.

Table 3: Ablation studies of LFM on key architectural and mechanistic designs. We categorize the variants into three groups: Prior and Context, Framework Decoupling, and Training Mechanisms.

Category	Method Variant	ETTm1			NSW			SA			AVG		
		MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
Prior	*w/o Prior (Eq.7)*	0.391	0.302	0.240	0.641	0.572	0.445	0.622	0.547	0.503	0.551	0.474	0.396
	w/o Global context (c)	0.375	0.288	0.241	0.637	0.539	0.440	0.593	0.550	0.510	0.535	0.459	0.397
Decoupling	*w/o Macro-Flow (Sec. 4.2)*	0.395	0.317	0.253	0.643	0.579	0.461	0.633	0.580	0.511	0.557	0.492	0.408
	w/o Micro-Flow (Sec. 4.3)	0.410	0.339	0.249	0.640	0.581	0.472	0.625	0.557	0.513	0.558	0.492	0.411
	Macro in Data-Space	0.370	0.285	0.243	0.627	0.593	0.451	0.616	0.550	0.514	0.538	0.476	0.403
Mechanism	*w/o Dynamic Base* (Eq.17)	0.371	0.280	0.260	0.611	0.582	0.477	0.610	0.543	0.520	0.531	0.468	0.419
	w/o Noise Augment (Eq.18)	0.375	0.281	0.259	0.613	0.580	0.472	0.615	0.545	0.517	0.534	0.469	0.416
	w/o Anchor Loss (Eq.16)	0.393	0.305	0.241	0.637	0.582	0.451	0.620	0.547	0.505	0.550	0.478	0.399
LFM (Full Model)		**0.332**	**0.264**	**0.234**	**0.589**	**0.441**	**0.427**	**0.569**	**0.489**	**0.486**	**0.497**	**0.398**	**0.382**

6 Conclusion

In this paper, we study the scale conflict in continuous-time generative forecasting caused by multiscale dynamics. We propose Laminar Flow Matching (LFM), a geometry-aware framework that decouples probability transport into latent-space macro evolution and data-space micro refinement, enhancing transport smoothness and stabilizing ODE integration. Extensive experiments across diverse real-world benchmarks demonstrate its effectiveness and efficiency. In future work, we aim to extend geometry-aware transport mechanisms to broader multivariate dynamic systems.

AI USE STATEMENT

In this work, we used generative AI tools to assist with code refinement and provide feedback on experimental design. We did not use generative AI tools to formulate the core research ideas, methodological contributions or theoretical claims. Additionally, generative AI tools were used to improve the clarity and readability of the manuscript. All AI-assisted code, experimental suggestions, and text were reviewed and verified by the authors. We take responsibility for the final content of this work, including any text, code, or other artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To support reproducibility, we provide the implementation code in the supplementary material. Model configurations, training details, and hyperparameters are documented in Sec. 5.1. The proof of Proposition 1 is given in Appendix B, and dataset statistics and descriptions are provided in Appendix D.

REFERENCES

Sabeen Ahmed, Ian E Nielsen, Aakash Tripathi, Shamoon Siddiqui, Ravi P Ramachandran, and Ghulam Rasool. Transformers in time-series analysis: A tutorial. *Circuits, systems, and signal processing*, 42(12):7433–7466, 2023.

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Adrian G Barnett, Jolieke C van der Pols, and Annette J Dobson. Regression to the mean: what it is and how to deal with it. *International Journal of Epidemiology*, 34(1):215–220, 2005.

Nastaran Bassamzadeh and Roger Ghanem. Multiscale stochastic prediction of electricity demand in smart grids using bayesian networks. *Applied Energy*, 193:369–380, 2017.

Jean-David Benamou and Yann Brenier. A computational fluid mechanics solution to the monge-kantorovich mass transfer problem. *Numerische Mathematik*, 84:375–393, 2000.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

Haozhuo Cao, Xiaorui Wang, Liyang Yu, and Wangcai Ding. Geomflow: Geometry-aware adaptive diffusion model via hessian information. *Neurocomputing*, 676:132934, 2026.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Forty-first International Conference on Machine Learning*, 2024.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Chris Finlay, Jörn-Henrik Jacobsen, Levon Nurbekyan, and Adam Oberman. How to train your neural ode: The world of jacobian and kinetic regularization. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3154–3164. PMLR, 2020.

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, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022.

Nouha Karaouli, Denis Coquenet, Elisa Fromont, Martial Mermillod, and Marina Reyboz. How foundational are foundation models for time series forecasting? *arXiv preprint arXiv:2510.00742*, 2025.

Jacob Kelly, Jesse Bettencourt, Matthew J. Johnson, and David K. Duvenaud. Learning differential equations that are easy to solve. In *Advances in Neural Information Processing Systems*, volume 33, pp. 4370–4380, 2020.

Subina Khanal, Seshu Tirupathi, Merim Dzaferagic, Marco Ruffini, and Torben Bach Pedersen. Bridging the high-frequency data gap: A millisecond-resolution network dataset for advancing time series foundation models. *arXiv preprint arXiv:2603.16497*, 2026.

Suyong Kim, Weiqi Ji, Sili Deng, Yingbo Ma, and Christopher Rackauckas. Stiff neural ordinary differential equations. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 31(9):093122, 2021.

Thomas Kneib, Alexander Silbersdorff, and Benjamin Säfken. Rage against the mean – a review of distributional regression approaches. *Econometrics and Statistics*, 26:99–123, 2023.

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

Xiangjie Kong, Zhenghao Chen, Weiyao Liu, Kaili Ning, Lechao Zhang, Syauqie Muhammad Marier, Yichen Liu, Yuhao Chen, and Feng Xia. Deep learning for time series forecasting: a survey. *International Journal of Machine Learning and Cybernetics*, 16(7):5079–5112, 2025.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling long-and short-term temporal patterns with deep neural networks. In *The 41st international ACM SIGIR conference on research & development in information retrieval*, pp. 95–104, 2018.

Yuxuan Liang, Haomin Wen, Yuqi Nie, Yushan Jiang, Ming Jin, Dongjin Song, Shirui Pan, and Qingsong Wen. Foundation models for time series analysis: A tutorial and survey. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 6555–6565, 2024.

Bryan Lim and Stefan Zohren. Time-series forecasting with deep learning: a survey. *Philosophical transactions of the royal society a: mathematical, physical and engineering sciences*, 379(2194), 2021.

Lequan Lin, Zhengkun Li, Ruikun Li, Xuliang Li, and Junbin Gao. Diffusion models for time-series applications: a survey. *Frontiers of Information Technology & Electronic Engineering*, 25(1): 19–41, 2024.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022.

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*, 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*, 2024.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-XL: Long-context transformers for unified time series forecasting. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Yong Liu, Guo Qin, Zhiyuan Shi, Zhi Chen, Caiyin Yang, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Sundial: A family of highly capable time series foundation models. In *Forty-second International Conference on Machine Learning*, 2025b.

Yuansan Liu, Sudanthi Wijewickrema, Dongting Hu, Christofer Bester, Stephen O'Leary, and James Bailey. Stochastic diffusion: A diffusion based model for stochastic time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 1939–1950, 2025c.

John A Miller, Mohammed Aldosari, Farah Saeed, Nasid Habib Barna, Subas Rana, I Budak Arpinar, and Ninghao Liu. A survey of deep learning and foundation models for time series forecasting. *arXiv preprint arXiv:2401.13912*, 2024.

G. P. Nason. Multiscale time series analysis. In *Wavelet Methods in Statistics with R*, pp. 167–199. Springer, New York, NY, 2008.

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.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pp. 1310–1318. Pmlr, 2013.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 5301–5310. PMLR, 2019.

Sepehr Rahvar, Erfan S. Reihani, Amirhossein N. Golestani, Abolfazl Hamounian, Fatemeh Aghaei, Muhammad Sahimi, Pouya Manshour, Milan Paluš, Ulrike Feudel, Jan A. Freund, Klaus Lehnertz, Thorsten Rings, and M. Reza Rahimi Tabar. Characterizing time-resolved stochasticity in non-stationary time series. *Chaos, Solitons & Fractals*, 185:115069, 2024.

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*, 2021.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. In *The Thirteenth International Conference on Learning Representations*, 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.

Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

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.

Hangtian Wang and Mahito Sugiyama. Time-gated multi-scale flow matching for time-series imputation. In *The Fourteenth International Conference on Learning Representations*, 2026.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024a.

Xinlei Wang, Maike Feng, Jing Qiu, Jinjin Gu, and Junhua Zhao. From news to forecast: Integrating event analysis in llm-based time series forecasting with reflection. *Advances in Neural Information Processing Systems*, 37:58118–58153, 2024b.

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, 2024c.

Qingsong Wen, Tian Zhou, Chaoli Zhang, Weiqi Chen, Ziqing Ma, Junchi Yan, and Liang Sun. Transformers in time series: A survey. *arXiv preprint*, arXiv:2202.07125, 2022.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 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.

Huibo Xu, Runlong Yu, Likang Wu, Xianquan Wang, and Qi Liu. Bridging the last mile of prediction: Enhancing time series forecasting with conditional guided flow matching. *arXiv preprint arXiv:2507.07192*, 2025.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with \$10k\$ parameters. In *The Twelfth International Conference on Learning Representations*, 2024.

Xinyu Yang, Yu Sun, Xiaojie Yuan, and Xinyang Chen. Frequency-aware generative models for multivariate time series imputation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Yiyuan Yang, Ming Jin, Haomin Wen, Chaoli Zhang, Yuxuan Liang, Lintao Ma, Yi Wang, Chenghao Liu, Bin Yang, Zenglin Xu, Shirui Pan, and Qingsong Wen. A survey on diffusion models for time series and spatio-temporal data. *ACM Comput. Surv.*, 58(8), February 2026.

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

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*, 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.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International Conference on Machine Learning*, pp. 27268–27286. PMLR, 2022.

A Appendix

B Proof of Proposition 1

In this section, we derive the scaling behavior stated in Proposition 1.

Setup. We consider the abstract trajectory

$$\gamma(s) = \gamma^{(L)}(s) + a\sin(\omega s)\boldsymbol{e},$$

where $s \in [0,1]$ is an abstract state-space coordinate. Let $\|\boldsymbol{e}\| = 1$, $a > 0$, and ω denote the dominant oscillation frequency. We assume that the low-frequency component is smooth with bounded first- and second-order derivatives:

$$\|\gamma^{(L)\prime}(s)\| \le M, \qquad \|\gamma^{(L)\prime\prime}(s)\| \le B.$$

We also assume that the variation of the induced vector field along the trajectory is uniformly bounded in the transport parameter:

$$\sup_{s\in[0,1]} \|\partial_s v(\gamma(s), s)\| \le C_s,$$

for some constant $C_s > 0$.

Step 1: Bounding the acceleration. Differentiating $\gamma(s)$, we obtain

$$\gamma'(s) = \gamma^{(L)\prime}(s) + a\omega\cos(\omega s)\boldsymbol{e},$$

and

$$\gamma''(s) = \gamma^{(L)\prime\prime}(s) - a\omega^2\sin(\omega s)\boldsymbol{e}.$$

For $\omega \ge \pi/2$, there exists $s^\star \in [0,1]$ such that $|\sin(\omega s^\star)| = 1$. Hence,

$$\sup_{s\in[0,1]} \|\gamma''(s)\| \ge \|\gamma''(s^\star)\| \ge a\omega^2 - \|\gamma^{(L)\prime\prime}(s^\star)\| \ge a\omega^2 - B.$$

Step 2: Linking trajectory acceleration to vector-field variation. Assume that γ is generated by the ODE

$$\frac{d\gamma}{ds} = v(\gamma, s).$$

Differentiating with respect to s yields

$$\frac{d^2\gamma}{ds^2} = \partial_s v(\gamma, s) + \nabla_z v(\gamma, s)\,\frac{d\gamma}{ds}.$$

Taking norms and using submultiplicativity of the induced operator norm, we have

$$\left\|\frac{d^2\gamma}{ds^2}\right\| \le \|\partial_s v(\gamma, s)\| + \|\nabla_z v(\gamma, s)\|_{\mathrm{op}} \left\|\frac{d\gamma}{ds}\right\|.$$

Therefore, whenever $\frac{d\gamma}{ds} \neq 0$,

$$\|\nabla_z v(\gamma, s)\|_{\mathrm{op}} \ge \frac{\left\|\frac{d^2\gamma}{ds^2}\right\| - \|\partial_s v(\gamma, s)\|}{\left\|\frac{d\gamma}{ds}\right\|}.$$

Step 3: Bounding the terms. Evaluating the above inequality at $s^\star$ and using

$$\left\|\frac{d\gamma}{ds}(s^\star)\right\| \le \left\|\gamma^{(L)\prime}(s^\star)\right\| + a\omega \le M + a\omega,$$

together with the bound $\|\partial_s v(\gamma(s^\star), s^\star)\| \le C_s$, we obtain

$$\|\nabla_z v(\gamma(s^\star), s^\star)\|_{\mathrm{op}} \ge \frac{a\omega^2 - B - C_s}{M + a\omega}.$$

Consequently,

$$\sup_{s\in[0,1]} \|\nabla_z v(\gamma(s), s)\|_{\mathrm{op}} \ge \frac{a\omega^2 - B - C_s}{M + a\omega}.$$

Discussion. The above derivation establishes the scaling behavior in Proposition 1. The result is not intended as a tight universal bound, but rather formalizes the intuition that high-frequency components increase trajectory acceleration and require larger spatial variation in the vector field, leading to stiffer dynamics in single-field flow matching. This excludes the degenerate case where arbitrarily large oscillatory acceleration is entirely absorbed by explicit variation in the transport parameter.

A more comprehensive functional analysis of decoupled transport under general distributions is left for future work.

C Training Procedure

The complete training step for Laminar Flow Matching (LFM) is detailed in Algorithm 1 and inference step in Algorithm 2.

Algorithm 1 LFM Training Step

Input: $\boldsymbol{x}^h, \boldsymbol{x}^f, \boldsymbol{c}; \tau, t \sim \mathcal{U}(0,1)$
Output: $\nabla_\theta \mathcal{L}$
$\boldsymbol{\mu} = \mathcal{F}_\phi(\boldsymbol{x}^h); \quad \boldsymbol{z}_0 \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2 \mathbf{I})$
$\boldsymbol{\phi}_0 = \mathcal{E}(\boldsymbol{z}_0), \quad \boldsymbol{\phi}_1 = \mathcal{E}(\boldsymbol{x}^f)$
$\boldsymbol{u}^* = \boldsymbol{\phi}_1 - \boldsymbol{\phi}_0; \quad \boldsymbol{\phi}_\tau = (1-\tau)\boldsymbol{\phi}_0 + \tau \boldsymbol{\phi}_1$
$\mathcal{L}_{\text{macro}} = \|u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c}) - \boldsymbol{u}^*\|^2$
$\hat{\boldsymbol{\phi}}_1 = \boldsymbol{\phi}_\tau + (1-\tau)\, u_\theta(\boldsymbol{\phi}_\tau, \tau, \boldsymbol{c})$
$\tilde{\boldsymbol{x}} = \mathcal{B}(\hat{\boldsymbol{\phi}}_1); \quad \mathcal{L}_{\text{anchor}} = \|\tilde{\boldsymbol{x}} - \boldsymbol{x}^f_{\text{s}}\|^2$
$\tilde{\boldsymbol{\phi}} = \text{sg}(\hat{\boldsymbol{\phi}}_1) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \eta^2 \mathbf{I})$
$\boldsymbol{\psi}_0 = \text{sg}(\tilde{\boldsymbol{x}}), \quad \boldsymbol{\psi}_1 = \boldsymbol{x}^f$
$\boldsymbol{v}^* = \boldsymbol{\psi}_1 - \boldsymbol{\psi}_0; \quad \boldsymbol{\psi}_t = (1-t)\boldsymbol{\psi}_0 + t\boldsymbol{\psi}_1$
$\mathcal{L}_{\text{micro}} = \|v_\theta(\boldsymbol{\psi}_t, t, \tilde{\boldsymbol{\phi}}) - \boldsymbol{v}^*\|^2$
$\mathcal{L} = \mathcal{L}_{\text{macro}} + \mathcal{L}_{\text{micro}} + \lambda \mathcal{L}_{\text{anchor}}$
return $\nabla_\theta \mathcal{L}$

D Dataset Statistics

Table 4 summarizes the statistical characteristics of all benchmarks. For the ETT datasets, we divide them by ratio $\{0.6, 0.2, 0.2\}$ into train set, validation set, and test set. For Electricity, Traffic, and Weather, Exchange and AEMO, we follow the same split ratio of $\{0.7, 0.1, 0.2\}$

Table 4: Details of the dataset statistics.

Dataset	Frequency	Total Steps	Prediction Task ($T_H \rightarrow T_F$)	Feature Dimensions D
Australian Energy (AEMO)	30 min	70,128	$96 \rightarrow 96/192/336/720$	7
Traffic	1 hour	17,544	$96 \rightarrow 96/192/336/720$	862
Electricity (ECL)	1 hour	26,304	$96 \rightarrow 96/192/336/720$	321
Weather	10 min	52,696	$96 \rightarrow 96/192/336/720$	21
Exchange	1 day	7,588	$96 \rightarrow 96/192/336/720$	8
ETTh1	1 hour	17,420	$96 \rightarrow 96/192/336/720$	7
ETTh2	1 hour	17,420	$96 \rightarrow 96/192/336/720$	7
ETTm1	15 min	69,680	$96 \rightarrow 96/192/336/720$	7
ETTm2	15 min	69,680	$96 \rightarrow 96/192/336/720$	7

Algorithm 2 LFM Probabilistic Inference

Input: Historical $\boldsymbol{x}^h$, context $\boldsymbol{c}$, predictor $\mathcal{F}_\phi$, ODE steps P, K, samples S
Output: Ensemble forecasts $\{\hat{\boldsymbol{x}}^{f,(s)}\}_{s=1}^{S}$
$\boldsymbol{\mu} = \mathcal{F}_\phi(\boldsymbol{x}^h)$
for $s = 1$ **to** S **parallel do**
  $\boldsymbol{z}_0^{(s)} \sim \mathcal{N}(\boldsymbol{\mu}, \sigma^2\mathbf{I}), \quad \boldsymbol{\phi}_0^{(s)} = \mathcal{E}(\boldsymbol{z}_0^{(s)})$
  Initialize $\tau = 0, \quad \Delta\tau = 1/P$
  for $p = 0$ **to** $P - 1$ **do**
    $\boldsymbol{\phi}_{\tau+\Delta\tau}^{(s)} = \boldsymbol{\phi}_\tau^{(s)} + \Delta\tau \cdot u_\theta(\boldsymbol{\phi}_\tau^{(s)}, \tau, \boldsymbol{c})$
    $\tau \leftarrow \tau + \Delta\tau$
  end for
  $\hat{\boldsymbol{\phi}}_1^{(s)} = \boldsymbol{\phi}_{\tau=1}^{(s)}, \quad \tilde{\boldsymbol{\phi}}^{(s)} = \hat{\boldsymbol{\phi}}_1^{(s)}$
  $\boldsymbol{\psi}_0^{(s)} = \mathcal{B}(\hat{\boldsymbol{\phi}}_1^{(s)})$
  Initialize $t = 0, \quad \Delta t = 1/K$
  for $k = 0$ **to** $K - 1$ **do**
    $\boldsymbol{\psi}_{t+\Delta t}^{(s)} = \boldsymbol{\psi}_t^{(s)} + \Delta t \cdot v_\theta(\boldsymbol{\psi}_t^{(s)}, t, \tilde{\boldsymbol{\phi}}^{(s)})$
    $t \leftarrow t + \Delta t$
  end for
  $\hat{\boldsymbol{x}}^{f,(s)} = \boldsymbol{\psi}_{t=1}^{(s)}$
end for
return $\{\hat{\boldsymbol{x}}^{f,(s)}\}_{s=1}^{S}$

E Computational Efficiency

We compare the computational efficiency of LFM against representative discriminative and generative baselines in Table 5, evaluating model size, integration steps, and inference latency.

Continuous-time generative models often incur substantial inference overhead due to iterative sampling. For example, Diffusion-TS requires 500 denoising steps, resulting in 620.2 ms latency. Standard flow matching (TSFlow) reduces this to 32 steps (36.5 ms), but the scale conflict within its single vector field still limits stable integration at larger step sizes. By decoupling macro- and micro-scale transport, LFM produces smoother trajectories that can be accurately integrated with only 15 total function evaluations ($P + K = 15$), reducing inference latency to 20.3 ms. Although deterministic one-step models such as TimeXer (2.2 ms) remain faster, LFM substantially narrows the efficiency gap while retaining state-of-the-art probabilistic forecasting performance, making it practical for real-world deployment.

Table 5: Model Complexity and Inference Cost.

Method	Class	Params (M)	Inf. Steps	Time (ms)
Autoformer	Transformer	0.67	1	2.9
TimeXer	Transformer	0.45	1	2.2
Diffusion-TS	Diffusion	0.29	500	620.2
TSFlow	Flow Matching	0.18	32	36.5
LFM	**Block FM**	**0.72**	**15**	20.3

F Comparison to Foundation Models

We evaluate the structural transferability of LFM in a cross-domain few-shot setting on the AEMO dataset. The model is trained on a source domain (e.g., VIC) and adapted to a target domain (e.g., SA) using only 3% of target data with a standard context length ($T_H = 96$). We compare against state-of-the-art time-series foundation models (TSFMs), which are pre-trained on large-scale data and evaluated in a zero-shot setting with long context ($T_H = 1000$).

As shown in Table 6, LFM consistently outperforms TSFMs across both transfer scenarios, achieving up to 2.3% MSE and 3.1% MAE reduction, despite using significantly fewer parameters. These results support our geometric hypothesis: by decoupling macroscopic transport from microscopic adaptation, LFM alleviates scale conflict and enables efficient cross-domain generalization without large-scale pre-training.

Table 6: Cross-domain transfer forecasting performance on AEMO. Results are averaged over prediction lengths $\{96, 192, 336, 720\}$. Foundation models utilize a context length of $T_H = 1000$ Woo et al. (2024), whereas LFM uses a history length of $T_H = 96$.

Model	VIC → SA			TAS → NSW		
	MSE	MAE	CRPS	MSE	MAE	CRPS
TimesFM (2024)	0.665	0.494	0.469	0.658	0.552	0.560
Chronos (2024)	0.659	0.490	0.464	0.667	0.553	0.557
Moirai (2024)	0.660	0.491	0.464	0.656	0.552	0.560
Timer-XL(2025a)	0.657	0.490	0.465	0.665	0.550	0.558
Time-MoE(2025)	0.659	0.489	0.468	0.661	0.550	0.555
Sundial (2025b)	0.667	0.491	0.464	0.660	0.554	0.553
LFM	**0.642**	**0.476**	**0.453**	**0.646**	**0.533**	**0.539**
Rel. Improv.	**2.3%**	**2.7%**	**2.4%**	**1.5%**	**3.1%**	**2.5%**

G Tail-Risk Analysis

Table 7 reports performance on the top 5% most extreme target values.

Table 7: Performance on the top 5% most extreme target values.

Dataset	**Model**	Tail-MSE↓	Tail-MAE↓	**Dataset**	**Model**	Tail-MSE↓	Tail-MAE↓
ETTm1	TimeXer	0.492	**0.361**	**SA**	TimeXer	0.796	0.690
	TSFlow	0.722	0.507		TSFlow	0.823	0.749
	LFM	**0.391**	0.382		**LFM**	**0.659**	**0.517**

H Impact of Network Architectures

Beyond the framework-level ablations in the main text, Table 8 evaluates the contribution of specific architectural choices within the dual-flow framework.

Replacing inter-block self-attention in the Outer Macro-Flow with an independent block-wise MLP (*w/o SelfAttn*) increases the average MSE by 6.3% (0.496 vs. 0.527), highlighting the importance of explicit cross-block coordination for coherent macroscopic transport.

Replacing the 1D convolutional backbone in the Inner Micro-Flow with a point-wise feed-forward network (*w/o Conv1D*) degrades average MSE by 4.4% (0.496 vs. 0.518), indicating that local temporal receptive fields are beneficial for modeling high-frequency stochastic dynamics.

Overall, these architectural changes yield smaller degradations than removing the core dual-flow decomposition in the main text, confirming that LFM's effectiveness primarily stems from resolving scale conflict through structural transport decomposition rather than architectural refinement.

Table 8: Additional ablation studies on network architectures.

Method Variant	**ETTm1**			**NSW**			**SA**			**AVG**		
	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
w/o SelfAttn (Macro Backbone)	0.372	0.291	0.253	0.617	0.472	0.450	0.593	0.511	0.522	0.527	0.425	0.408
w/o Conv1D (Micro Backbone)	0.356	0.302	0.267	0.610	0.457	0.440	0.588	0.500	0.493	0.518	0.420	0.400
LFM (Full Model)	**0.332**	**0.264**	**0.234**	**0.588**	**0.441**	**0.427**	**0.569**	**0.489**	**0.486**	**0.496**	**0.398**	**0.382**

I CAPACITY-CONTROLLED COMPARISON

To separate the effect of model capacity from transport decomposition, we construct TSFlow-Large by increasing the residual-channel width of TSFlow from 64 to 128 while keeping the remaining training and inference settings unchanged. This increases its parameter count from 0.18M to 0.70M, closely matching LFM's 0.72M parameters.

Table 9: Capacity-controlled comparison on ETTm1.

Model	Params.	MSE ↓	MAE ↓	CRPS ↓
TSFlow	0.18M	0.695	0.345	0.243
TSFlow-Large	0.70M	0.684	0.331	0.240
LFM	0.72M	**0.332**	**0.264**	**0.234**

Increasing TSFlow to a comparable parameter budget yields only modest gains, while a substantial gap to LFM remains. This indicates that LFM's improvement cannot be explained by increased model capacity alone.

J HYPERPARAMETER SENSITIVITY

We evaluate the sensitivity of LFM to the block length L and the anchor-loss weight λ on ETTm1, Traffic, and Exchange, which have different sampling frequencies. Results are averaged over four forecasting horizons.

Block Length. Table 10 reports results for $L \in \{6, 12, 24, 48, 96\}$, ranging from $0.25\times$ to $4\times$ the default $L = 24$. Performance remains stable across this range, with $L = 24$ achieving the best average MSE. Halving and doubling the default block length lead to only 1.4% and 1.0% degradation, while the more extreme settings lead to 2.3% and 3.9% degradation, respectively.

Table 10: Sensitivity to block length L in terms of MSE. Results are averaged over four forecasting horizons.

Dataset	Frequency	$L = 6$	$L = 12$	$L = 24$	$L = 48$	$L = 96$
ETTm1	15 min	0.344	0.337	**0.332**	0.341	0.349
Traffic	1 hour	0.398	0.399	**0.390**	0.395	0.402
Exchange	1 day	0.390	0.386	0.385	**0.382**	0.399

Anchor-Loss Weight. Table 11 shows that LFM is also stable across the tested range of λ. Moderate weighting generally performs best, while very small or large values can weaken macro-level supervision or over-constrain the macro prediction, respectively.

Table 11: Sensitivity to the anchor-loss weight λ in terms of MSE.

Dataset	$\lambda = 0.1$	$\lambda = 0.25$	$\lambda = 0.5$	$\lambda = 1.0$
ETTm1	0.337	0.340	**0.332**	0.337
Traffic	0.402	0.398	**0.390**	0.392
Exchange	0.391	**0.385**	**0.385**	0.389

Overall, these results indicate that LFM does not rely on a narrowly tuned choice of temporal granularity or anchor-loss weight.

K MACRO-FLOW TRANSPORT GEOMETRY

To directly quantify the effect of latent-space macro transport, we evaluate the inference trajectories using two geometric measures. Straightness is defined as the ratio between endpoint displacement

and accumulated path length, while the normalized turning angle measures directional variation between consecutive trajectory segments. Higher straightness and a lower turning angle indicate smoother transport.

Table 12: Geometry of Macro-Flow inference trajectories.

Transport	Straightness ↑	Normalized Turning Angle ↓
Macro in Data-Space	0.92	0.061
LFM Macro-Flow	**0.96**	**0.032**

Latent-space transport yields straighter trajectories and reduces the normalized turning angle by 47.5% relative to data-space macro transport, providing direct evidence that the latent formulation simplifies macro-level transport geometry.

L FULL RESULTS

In the main text, we report results averaged over four horizons; here we provide the full results. Tables 13, 14 present results on standard benchmarks with $T_f \in \{96, 192, 336, 720\}$. Table 15 reports detailed results on AEMO. LFM consistently outperforms baselines across all horizons, confirming its ability to handle regime shifts and stochastic variability.

Table 13: Full results on ETT datasets. Comparison of MSE, MAE and CRPS on ETTh1, ETTh2, ETTm1, and ETTm2. $T_h = 96$ and $T_f \in \{96, 192, 336, 720\}$ **Bold** indicates the best result, and underlined indicates the second best.

		ETTh1			ETTh2			ETTm1			ETTm2		
Model	Len	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
AF. (2021)	96	0.436	0.295	0.293	0.346	0.267	0.263	0.510	0.300	0.288	0.309	0.235	0.233
	192	0.456	0.301	0.295	0.379	0.274	0.270	0.514	0.301	0.293	0.317	0.252	0.246
	336	0.485	0.305	0.299	0.401	0.284	0.273	0.510	0.302	0.295	0.325	0.251	0.247
	720	0.512	0.305	0.300	0.466	0.277	0.270	0.521	0.308	0.297	0.334	0.255	0.251
FED. (2022)	96	0.441	0.290	0.291	0.321	0.260	0.265	0.516	0.299	0.285	0.300	0.228	0.221
	192	0.455	0.319	0.290	0.362	0.270	0.255	0.510	0.312	0.290	0.325	0.250	0.227
	336	0.480	0.312	0.290	0.415	0.280	0.272	0.529	0.315	0.290	0.322	0.250	0.235
	720	0.510	0.312	0.305	0.460	**0.257**	0.277	0.519	0.312	0.299	0.336	0.259	0.259
PTST. (2023)	96	0.471	0.288	0.279	0.352	0.255	0.257	0.395	0.280	0.274	0.295	0.251	0.224
	192	0.512	0.270	0.266	0.361	0.283	0.279	0.385	0.271	0.265	0.287	0.232	0.225
	336	0.522	0.531	0.279	0.391	0.290	0.285	0.470	0.283	0.277	0.306	0.231	0.228
	720	0.647	0.280	0.280	0.400	0.291	0.288	0.525	0.285	0.279	0.310	0.244	0.235
DLinear (2023)	96	0.477	0.282	0.276	0.337	0.243	0.240	0.399	0.283	0.277	0.288	**0.215**	0.221
	192	0.465	0.280	0.281	0.371	0.254	0.253	0.435	0.286	0.280	0.293	0.227	0.225
	336	0.499	0.285	0.283	0.388	0.270	0.272	0.470	0.282	0.285	0.306	0.229	0.237
	720	0.572	0.291	0.285	0.397	0.280	0.276	0.520	0.290	0.293	0.325	0.241	0.237
iTran. (2024)	96	0.405	0.282	0.272	0.331	0.244	0.233	0.390	0.281	0.265	0.288	0.225	0.230
	192	0.461	0.276	0.285	0.370	0.254	0.251	0.430	0.289	0.290	0.290	0.228	0.235
	336	0.488	0.281	0.280	0.385	0.266	0.270	0.475	0.284	0.280	0.309	0.237	0.231
	720	0.575	0.290	0.293	0.396	0.272	0.270	0.511	0.298	0.295	0.326	0.233	0.239
TimeXer (2024c)	96	**0.382**	0.264	0.260	0.295	0.275	0.221	0.319	0.270	0.257	0.257	0.231	0.224
	192	0.428	0.280	0.266	0.304	0.263	0.255	0.357	0.266	0.259	**0.266**	0.230	0.231
	336	0.588	0.274	0.270	0.335	0.265	0.259	0.399	**0.268**	0.262	0.279	0.255	0.245
	720	0.690	**0.274**	0.272	0.393	0.273	0.266	0.452	0.271	**0.270**	0.294	0.266	0.257
FITS (2024)	96	0.489	0.302	0.279	0.366	0.259	0.262	0.405	0.281	0.295	0.306	0.249	0.225
	192	0.512	0.278	0.270	0.362	0.290	0.265	0.393	0.271	0.279	0.298	0.240	0.253
	336	0.490	0.511	0.275	0.380	0.277	0.275	0.450	0.293	0.277	0.312	0.229	0.230
	720	0.647	0.289	0.270	0.402	0.295	0.277	0.533	0.290	0.293	0.343	0.241	0.245
CSDI (2021)	96	0.823	0.283	0.244	0.711	0.250	0.227	0.811	0.277	0.250	0.593	0.238	0.229
	192	0.820	0.301	0.247	0.733	0.298	0.249	0.818	0.360	0.254	0.607	0.245	0.238
	336	0.831	0.315	0.258	0.761	0.296	0.258	0.825	0.372	0.267	0.665	0.250	0.249
	720	0.838	0.323	0.254	0.792	0.291	0.266	0.833	0.385	0.281	0.692	0.253	0.285
TimeGrad (2021)	96	0.820	0.340	0.241	0.695	0.299	0.210	0.800	0.348	0.243	0.572	0.229	0.215
	192	0.810	0.351	0.240	0.726	0.292	0.237	0.813	0.359	0.250	0.583	0.231	0.227
	336	0.830	0.357	0.251	0.735	0.292	0.240	0.815	0.355	0.253	0.597	0.248	0.239
	720	0.830	0.355	0.252	0.741	0.294	0.259	0.821	0.380	0.279	0.615	0.263	0.257
DTS. (2024)	96	0.720	0.350	0.218	0.654	0.295	0.207	0.643	0.327	0.222	0.491	0.251	0.195
	192	0.725	0.314	0.217	0.665	0.289	0.218	0.649	0.330	0.232	0.503	**0.220**	0.211
	336	0.722	0.395	**0.220**	0.674	0.281	**0.225**	0.647	0.332	0.229	0.519	0.225	0.218
	720	0.830	0.364	0.257	0.692	0.301	0.236	0.832	0.405	0.291	0.572	0.285	0.220
CGFM (2025)	96	0.520	0.320	0.215	0.460	0.250	0.235	0.350	0.325	0.224	0.295	0.250	0.205
	192	0.571	0.329	0.230	0.502	0.298	0.240	0.357	0.349	0.229	0.322	0.302	0.195
	336	0.675	0.344	0.253	0.595	0.302	0.297	0.447	0.335	0.241	0.438	0.305	0.212
	720	0.627	0.360	0.277	0.507	0.331	0.247	0.585	0.409	0.301	0.457	0.308	0.211
TSFlow (2025)	96	0.724	0.305	0.205	0.669	0.257	0.225	0.640	0.315	0.214	0.496	0.254	0.199
	192	0.724	0.315	0.210	0.693	0.288	0.226	0.646	0.334	0.230	0.502	0.299	0.195
	336	0.725	0.340	0.223	0.705	0.289	0.227	0.657	0.330	0.232	0.539	0.290	0.207
	720	0.811	0.352	**0.250**	0.699	0.303	**0.230**	0.838	0.400	0.295	0.540	0.295	**0.210**
LFM	96	0.388	**0.250**	**0.195**	**0.288**	**0.240**	**0.206**	**0.280**	**0.257**	**0.210**	**0.256**	0.227	**0.190**
	192	**0.420**	0.277	**0.205**	**0.302**	**0.251**	**0.212**	**0.319**	**0.260**	**0.225**	0.272	0.230	**0.192**
	336	**0.457**	**0.268**	0.256	**0.332**	**0.250**	0.237	**0.342**	0.269	**0.227**	**0.277**	0.235	**0.201**
	720	**0.489**	0.282	0.280	**0.367**	0.261	0.240	**0.385**	**0.269**	0.273	**0.290**	**0.227**	**0.210**

Table 14: Full results on benchmark datasets. Comparison of MSE, MAE, and CRPS on Electricity, Traffic, Weather, and Exchange. $T_h = 96$ and $T_f \in \{96, 192, 336, 720\}$ **Bold** indicates the best result, and underlined indicates the second best.

		Electricity			**Traffic**			**Weather**			**Exchange**		
Model	Len	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
AF. (2021)	96	0.196	0.275	0.272	0.597	0.622	0.615	0.245	0.370	0.356	0.205	0.295	0.291
	192	0.212	0.295	0.291	0.609	0.637	0.630	0.323	0.361	0.363	0.315	0.300	0.305
	336	0.214	0.301	0.290	0.623	0.637	0.641	0.353	0.370	0.371	0.541	0.365	0.374
	720	0.235	0.290	0.294	0.640	0.651	0.644	0.412	0.377	0.372	1.377	0.419	0.412
FED. (2022)	96	0.200	0.285	0.276	0.490	0.612	0.605	0.221	0.357	0.349	0.200	0.288	0.271
	192	0.216	0.290	0.292	0.601	0.632	0.657	0.320	0.365	0.344	0.310	0.320	0.312
	336	0.210	0.317	0.295	0.633	0.657	0.640	0.332	0.375	0.370	0.532	0.369	0.370
	720	0.227	0.286	0.290	0.592	0.621	0.640	0.433	0.356	0.377	0.998	0.445	0.409
PTST. (2023)	96	0.169	0.280	0.279	0.370	0.579	0.572	0.189	0.338	0.331	**0.145**	0.290	0.288
	192	0.267	0.275	0.274	0.399	0.580	0.581	0.294	0.350	0.339	0.271	0.290	0.390
	336	0.263	0.284	0.280	0.492	0.587	0.589	0.345	0.370	0.374	0.492	0.365	0.356
	720	0.297	0.293	0.288	0.532	0.605	0.590	0.414	0.381	0.377	0.879	0.405	0.405
DLinear (2023)	96	0.240	0.280	0.278	0.470	0.574	0.575	0.276	0.340	0.332	0.149	0.291	0.287
	192	0.253	0.285	0.282	0.493	0.593	0.580	0.280	0.372	0.335	0.305	0.310	0.311
	336	0.269	0.284	0.285	0.536	0.573	0.592	0.365	0.349	0.340	0.598	0.420	0.415
	720	0.273	0.291	0.289	0.566	0.595	0.590	0.383	0.357	0.351	0.911	0.470	0.472
iTran. (2024)	96	0.145	0.271	0.265	0.433	0.599	0.593	0.172	0.335	0.330	0.189	0.292	0.290
	192	**0.157**	0.274	0.270	0.445	0.593	0.590	0.222	0.347	0.355	0.298	0.312	0.317
	336	0.176	0.279	0.275	0.478	0.603	0.605	0.262	0.339	0.335	0.442	0.387	0.375
	720	0.221	0.295	0.298	0.520	0.606	0.600	0.357	0.347	0.336	0.895	0.374	0.369
TimeXer (2024c)	96	0.140	0.269	0.263	0.428	0.595	0.588	0.157	0.325	0.320	0.182	0.291	0.285
	192	**0.157**	0.272	0.265	0.449	0.595	0.592	**0.204**	0.330	0.331	0.277	0.305	0.301
	336	0.176	0.271	0.265	0.471	0.599	0.597	0.261	0.337	0.332	0.435	0.381	0.372
	720	0.211	0.285	**0.278**	0.520	0.604	0.600	0.340	0.337	0.335	0.885	0.372	0.365
FITS (2024)	96	0.224	0.250	0.289	0.416	**0.560**	0.574	0.262	0.340	0.331	**0.145**	0.290	0.272
	192	0.245	0.279	0.270	0.485	0.593	0.577	0.289	0.359	0.325	0.322	0.330	0.342
	336	0.271	0.279	0.270	0.515	**0.560**	0.565	0.387	0.339	0.310	0.586	0.415	0.400
	720	0.295	0.280	0.280	0.566	0.594	0.575	0.381	0.322	0.324	0.988	0.472	0.475
CSDI (2021)	96	0.337	0.291	0.262	0.915	0.773	0.560	0.450	0.572	0.319	0.331	0.442	0.287
	192	0.336	0.295	0.260	0.989	0.884	0.577	0.472	0.681	0.320	0.365	0.497	0.301
	336	0.343	0.270	0.267	1.022	0.595	0.593	0.480	0.627	0.325	0.695	0.380	0.384
	720	0.355	0.281	0.282	1.022	0.600	0.596	0.486	0.635	0.329	1.453	0.417	0.411
TimeGrad (2021)	96	0.335	0.265	0.260	0.988	0.625	0.521	0.432	0.427	0.320	0.341	0.395	0.286
	192	0.312	0.271	0.265	0.977	0.672	0.565	0.489	0.517	0.313	0.379	0.405	0.299
	336	0.340	0.267	0.260	1.015	0.592	0.590	0.472	0.520	0.313	0.690	0.501	0.392
	720	0.350	0.297	0.281	1.033	0.603	0.597	0.485	0.527	0.315	1.394	0.520	0.410
DTS. (2024)	96	0.316	0.262	0.260	0.944	0.707	0.500	0.405	0.409	0.303	0.388	0.392	0.285
	192	0.319	0.263	0.263	0.945	0.712	0.508	0.409	0.419	0.310	0.391	0.390	0.293
	336	0.319	0.265	0.260	0.948	0.712	0.505	0.405	0.419	0.308	0.655	0.445	0.342
	720	0.364	0.290	0.286	1.055	0.710	0.607	0.492	0.430	0.325	0.974	0.455	0.340
CGFM (2025)	96	0.315	0.270	0.261	0.955	0.610	0.512	0.410	0.359	0.225	0.382	0.345	0.259
	192	0.320	0.266	0.265	0.940	0.711	0.510	0.413	0.439	0.320	0.429	0.381	0.293
	336	0.362	0.290	0.273	0.981	0.695	0.504	0.405	0.501	0.317	0.730	0.429	**0.311**
	720	0.372	0.285	0.280	0.989	0.653	0.597	0.501	0.442	0.339	0.950	0.497	0.347
TSFlow (2025)	96	0.312	0.272	0.265	0.956	0.611	0.505	0.415	0.409	0.205	0.392	0.355	0.254
	192	0.318	0.264	0.260	0.949	0.620	0.512	0.408	0.422	0.315	0.415	0.375	0.373
	336	0.322	0.271	0.265	0.949	0.690	**0.492**	0.401	0.511	0.308	0.736	0.425	0.319
	720	0.363	0.290	0.285	1.045	0.703	0.600	0.495	0.555	0.328	0.955	0.441	0.337
LFM	96	**0.109**	**0.249**	**0.247**	**0.332**	**0.560**	**0.491**	**0.155**	**0.318**	**0.203**	0.157	**0.257**	**0.251**
	192	**0.127**	**0.258**	**0.253**	**0.365**	**0.571**	**0.490**	0.208	**0.329**	**0.275**	**0.244**	**0.289**	**0.287**
	336	**0.133**	**0.261**	**0.247**	**0.401**	0.566	0.493	**0.257**	**0.330**	**0.299**	**0.425**	**0.330**	0.324
	720	**0.145**	**0.264**	0.280	**0.460**	**0.588**	**0.507**	**0.332**	**0.313**	**0.314**	**0.712**	**0.369**	**0.335**

Table 15: Full forecasting results on AEMO. History length is 96. Corresponding prediction lengths include $\{96, 192, 336, 720\}$. **Bold** indicates the best result, and underlined indicates the second best.

		NSW			QLD			SA			VIC			TAS		
		MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
AF. (2021)	96	0.660	0.451	0.454	0.499	0.423	0.457	0.523	0.475	0.474	0.628	0.571	0.595	0.542	0.497	0.488
	192	0.629	0.623	0.575	0.727	0.617	0.599	0.612	0.552	0.544	0.749	0.588	0.590	0.699	0.623	0.599
	336	0.844	0.687	0.640	0.823	0.821	0.740	0.804	0.799	0.712	0.988	0.898	0.778	0.877	0.715	0.722
	720	1.092	0.898	0.875	1.045	0.836	0.839	1.129	0.949	1.002	1.180	0.968	0.939	1.093	0.970	0.980
FED. (2022)	96	0.592	0.441	0.432	0.495	0.420	0.421	0.511	0.470	0.470	0.627	0.570	0.591	0.572	0.494	0.480
	192	0.620	0.621	0.570	0.739	0.612	0.590	0.605	0.550	0.540	0.723	0.581	0.588	0.705	0.620	0.585
	336	0.836	0.680	0.642	0.825	0.821	0.745	0.788	0.790	0.705	0.984	0.892	0.770	0.871	0.710	0.721
	720	1.088	0.895	0.874	1.040	0.832	0.837	1.120	0.945	0.997	1.054	0.958	0.922	1.090	0.957	0.972
PTST. (2023)	96	0.463	0.411	0.410	0.472	0.427	0.433	0.503	0.471	0.464	0.588	0.501	0.495	0.522	0.477	0.471
	192	0.622	0.603	0.571	0.722	0.609	0.593	0.605	0.532	0.521	0.733	0.585	0.572	0.691	0.613	0.599
	336	0.815	0.677	0.639	0.810	0.811	0.742	0.800	0.792	0.711	0.959	0.880	0.774	0.872	0.703	0.706
	720	0.992	0.880	0.863	1.012	0.830	0.837	1.199	0.940	0.932	1.110	0.938	0.919	0.993	0.960	0.965
DLinear (2023)	96	0.465	0.427	0.423	0.457	0.417	0.413	0.493	0.460	0.445	0.597	0.505	0.500	0.534	0.492	0.483
	192	0.615	0.593	0.577	0.667	0.569	0.568	0.612	0.533	0.525	0.740	0.590	0.586	0.697	0.578	0.593
	336	0.792	0.647	0.611	0.811	0.793	0.764	0.822	0.741	0.735	0.933	0.889	0.790	0.885	0.712	0.709
	720	0.984	0.883	0.857	1.095	0.833	0.812	1.027	0.949	0.911	1.115	0.937	0.900	0.992	0.947	0.908
iTran. (2024)	96	0.412	0.411	0.419	0.425	0.413	0.397	0.489	0.412	0.455	0.569	0.497	0.478	0.510	0.479	0.466
	192	0.604	0.588	0.593	0.692	0.588	0.591	0.592	0.521	0.522	0.725	0.590	0.585	0.690	0.599	0.572
	336	0.753	0.637	0.620	0.781	0.773	0.765	0.812	0.722	0.730	0.935	0.831	0.789	0.889	0.721	0.710
	720	0.972	0.890	0.855	0.941	0.824	0.791	0.952	0.925	0.904	0.990	0.919	0.901	0.923	0.941	0.916
TimeXer (2024c)	96	0.405	0.407	0.407	0.423	0.409	0.397	0.457	0.402	0.440	0.565	0.489	0.472	0.500	0.471	0.465
	192	0.597	0.578	0.563	0.691	0.575	0.570	0.593	0.521	0.520	0.721	0.588	0.575	0.660	0.593	0.573
	336	0.742	0.623	0.600	0.766	0.769	0.760	0.809	0.721	0.730	0.933	0.829	0.780	0.888	0.715	0.712
	720	0.972	0.880	0.850	0.933	0.825	0.800	0.954	0.922	0.900	0.993	0.917	0.900	0.923	0.941	0.915
FITS (2024)	96	0.458	0.409	0.400	0.454	0.411	0.420	0.547	0.470	0.445	0.565	0.493	0.488	0.510	0.465	0.460
	192	0.621	0.593	0.570	0.715	0.601	0.593	0.590	0.522	0.510	0.715	0.580	0.562	0.683	0.600	0.581
	336	0.822	0.670	0.633	0.801	0.805	0.738	0.790	0.775	0.703	0.942	0.850	0.775	0.863	0.697	0.705
	720	0.993	0.860	0.672	1.005	0.812	0.831	1.156	0.920	0.927	1.153	0.927	0.905	0.998	0.940	0.965
CSDI (2021)	96	0.675	0.593	0.572	0.581	0.493	0.465	0.594	0.459	0.424	0.648	0.572	0.575	0.592	0.500	0.498
	192	0.711	0.600	0.584	0.729	0.637	0.559	0.632	0.582	0.547	0.798	0.542	0.591	0.705	0.625	0.603
	336	0.892	0.701	0.624	0.820	0.825	0.742	0.845	0.893	0.732	0.972	0.891	0.773	0.865	0.722	0.719
	720	1.115	0.902	0.881	1.003	0.825	0.819	1.002	0.997	1.059	1.180	0.968	0.939	1.093	0.970	0.980
TimeGrad (2021)	96	0.665	0.590	0.542	0.521	0.413	0.405	0.584	0.451	0.420	0.642	0.570	0.566	0.572	0.503	0.499
	192	0.705	0.593	0.581	0.720	0.601	0.542	0.626	0.580	0.544	0.793	0.537	0.590	0.700	0.622	0.595
	336	0.890	0.693	0.611	0.827	0.815	0.733	0.840	0.887	0.729	0.970	0.884	0.765	0.860	0.717	0.709
	720	1.093	0.889	0.885	1.000	0.793	0.805	1.001	0.963	1.042	1.097	0.960	0.941	1.099	0.943	0.980
DTS. (2024)	96	0.621	0.557	0.502	0.497	0.403	0.399	0.554	0.450	0.411	0.605	0.514	0.492	0.522	0.445	0.439
	192	0.693	0.590	0.551	0.709	0.592	0.522	0.591	0.577	0.540	0.792	0.523	0.580	0.705	0.621	0.590
	336	0.840	0.633	0.605	0.811	0.793	0.723	0.835	0.872	0.700	0.915	0.880	0.754	0.842	0.700	0.712
	720	1.095	0.872	0.880	0.995	0.790	0.792	1.000	0.973	1.022	1.090	0.955	0.937	1.089	0.945	0.973
CGFM (2025)	96	0.473	0.520	0.495	0.427	0.415	0.400	0.550	0.427	0.401	0.593	0.521	0.490	0.519	0.440	0.428
	192	0.621	0.615	0.490	0.699	0.582	0.520	0.585	0.570	0.537	0.790	**0.515**	0.572	0.700	0.615	0.588
	336	0.891	0.628	0.541	0.893	0.799	0.651	0.830	0.852	0.690	0.910	0.872	0.750	0.833	0.659	0.710
	720	1.112	0.795	0.663	0.974	0.786	0.789	1.004	0.970	1.057	1.088	0.943	0.921	1.080	0.957	0.970
TSFlow (2025)	96	0.499	0.515	0.478	0.433	0.410	0.411	0.554	0.420	0.395	0.590	0.517	0.488	0.520	0.447	**0.415**
	192	0.665	0.642	0.489	0.612	0.580	**0.505**	0.573	0.560	0.521	0.788	0.520	0.562	0.693	0.622	0.587
	336	0.986	0.657	0.488	0.973	0.790	0.642	0.822	0.850	0.650	0.920	0.872	0.750	0.833	0.659	0.710
	720	1.211	0.663	0.492	1.002	0.783	**0.653**	0.827	0.841	0.633	0.907	0.985	0.732	0.806	0.947	0.699
LFM	96	**0.374**	**0.383**	**0.376**	**0.369**	**0.376**	**0.369**	**0.382**	**0.391**	**0.383**	**0.499**	**0.451**	**0.443**	**0.445**	**0.439**	0.434
	192	**0.523**	**0.415**	**0.409**	**0.611**	**0.538**	0.530	**0.552**	**0.477**	**0.486**	**0.683**	0.547	**0.538**	**0.551**	**0.532**	**0.533**
	336	**0.698**	**0.477**	**0.459**	**0.705**	**0.600**	**0.592**	**0.631**	**0.507**	**0.501**	**0.729**	**0.642**	**0.640**	**0.690**	**0.600**	**0.592**
	720	**0.759**	**0.488**	**0.463**	**0.821**	**0.688**	0.663	**0.709**	**0.580**	**0.572**	**0.889**	**0.693**	**0.695**	**0.749**	**0.667**	**0.680**