MIXTURES OF LOCALIZED DIFFUSION PROCESSES FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

In many real-world forecasting tasks, we observe only a limited subset of the variables governing a high-dimensional system, yet must predict its future evolution from these incomplete measurements. Under such partial observation, some localized temporal and frequency components of the future can be predicted reliably from the observed history, while others retain substantial residual variation. We propose MELD, an end-to-end forecasting framework that adapts stochastic modeling to this localized unresolved structure. MELD constructs a localized time-frequency frame over the observed history and uses context-dependent routing to identify components that are well explained by the available observations and those with larger unresolved uncertainty. The resolved components provide the conditional structure of the forecast, while the unresolved components define the subspace on which diffusion is applied as the central stochastic modeling mechanism. Across standard long-horizon forecasting benchmarks, we show that stochastic variation is not uniformly distributed across these components, and that diffusion becomes substantially more effective with concentrated residual.

1 INTRODUCTION

In long-horizon time-series forecasting, uncertainty within the forecast window is non-uniform: some components of the time series are determined by historic observations, while others are not conditioned on them. In real-world time series, predictable deterministic and unpredictable stochastic components can be distributed across temporal and spectral structure of time series (Nguyen & Quanz, 2021). In electricity forecasting, slowly varying periodic components are often supported by recent context, while high-frequency fluctuations reflect unobserved factors (Lee et al., 2018). In financial time series, medium-scale trends may be supported by recent history, whereas abrupt local movements and shocks can remain unresolved under the available observations (Gajamannage et al., 2023). A challenge that time-series forecasters face is to model both the components that can be resolved by context and the components that are not conditioned on it.

Recent approaches to jointly capturing the deterministic and stochastic nature of time series are represented by diffusion models. Diffusion forecasters view future trajectories of time series as stochastic conditional rollouts based on observed history of time series variation (Kollovieh et al., 2023; Lin et al., 2024). Existing methods apply diffusion to the entirety of the time-series context window or its latent representation, without separating its deterministic and stochastic components (Li et al., 2022; Feng et al., 2024; Suh et al., 2024; Shen & Kwok, 2023; Li et al., 2024). As a result, the diffusion processes, which are inherently stochastic, are forced to model deterministic components of time-series and not just their stochastic components, which motivated the use of diffusion in the first place (Fan et al., 2024; Ye et al., 2025; Yang et al., 2024; Gao et al., 2025; Rishi et al., 2025).

The separation of deterministic and stochastic components is standard in other fields. For example, in dynamical systems, a reduced-dynamics view provides a useful perspective on this problem (Mori, 1965; Venturi & Li, 2023): under partial observation, the evolution of measured variables contains both history-dependent effects and forcing induced by unobserved degrees of freedom. We argue that, similarly, a time-series forecasting model can first extract the predictive structure available in the observed history, then separately account for the variation that cannot be captured.

Motivated by this view, we propose MELD (MixturE of Localized Diffusions), an end-to-end time-series forecasting framework that combines localized deterministic prediction with stochastic residual

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.

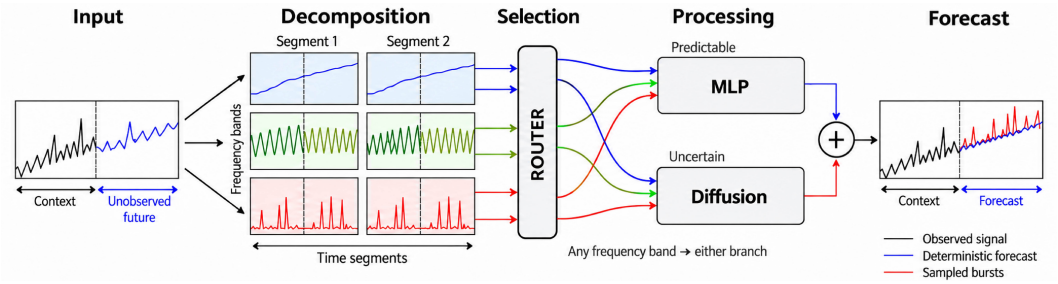


Figure 1: **Overview of MELD.** Input time series data is decomposed into localized time-frequency segments. The segments are passed through the learnable router that marks the components as "predictable" or "uncertain". "Predictable" components are passed through a deterministic MLP forecasters, while "uncertain" components are passed through stochastic diffusion forecasters. Deterministic and stochastic partial forecasts are then summed to form the joint forecast.

modeling. MELD breaks time series down into localized time-frequency components and uses context-dependent routing of these components to form the *anchor* - a deterministic forecast produced before stochastic residual refinement. A conditional diffusion model then learns the distribution of the time series signal's residual around this anchor. Since unresolved variation differs across localized components of the time series, the routing information also determines the components on which diffusion is placed. We evaluate MELD on standard long-horizon forecasting benchmarks using both deterministic and probabilistic forecasting metrics. Across datasets, MELD achieves strong performance competitive with state-of-the-art deterministic and diffusion-based baselines. Ablations further isolate the roles of localized forecasting, context-dependent routing, and stochastic residual refinement, showing the benefit of the mixture size, the selection rule, and each architectural component.

The contributions of this work are summarized as follows:

- We formulate the problem of time-series forecasting under partial observation as the problem of finding the localized time-frequency structure of the time series resolved from the observed history and modeling the residual variation of the signal that remains unresolved.
- We propose MELD, a time-series forecaster that learns an adaptive localized time-frequency frame, constructs a history-conditioned anchor through context-dependent routing, and models the anchor-relative residual as a routed mixture of localized diffusion processes.
- We show that MELD achieves state-of-the-art accuracy across standard long-horizon forecasting benchmarks while improving probabilistic forecast quality. We further show that unresolved variation in standard benchmarks is concentrated in time-frequency features learned by MELD, validating the proposed design.

2 Related Work

Representations in time-series forecasting methods. Existing time-series forecasting models exploit temporal and spectral structure in several ways. Some methods combine time- and frequency-domain information, such as TimesNet (Wu et al., 2023), FEDformer (Zhou et al., 2022b), ETSformer (Woo et al., 2022b), TFDNet (Luo et al., 2025), and CoST (Woo et al., 2022a), typically using compact spectral views to improve efficiency or capture seasonal structure. Others, including N-HiTS (Challu et al., 2023), DLinear (Zeng et al., 2023), TimeMixer (Wang et al., 2024), SCINet (Liu et al., 2022a), N-BEATS (Oreshkin et al., 2019), Autoformer (Wu et al., 2021), MICN (Wang et al., 2023), and Pyraformer (Liu et al., 2022b), build multiscale hierarchies primarily in the time domain. A third line of works, including FreTS (Yi et al., 2023), FiLM (Zhou et al., 2022a), and D-PAD (Yuan & Chen, 2024), emphasizes spectral decomposition, selection, or disentanglement. These methods are, however, focused on point forecasts for time-series.

Diffusion forecasting and distributional evaluation. Recent work casts time-series forecasting as a conditional generation task and samples future trajectories of time series using diffusion (Li et al., 2022; Shen & Kwok, 2023; Li et al., 2024; Suh et al., 2024; Fan et al., 2024; Ye et al., 2025; Yang et al., 2024; Gao et al., 2025; Rishi et al., 2025). D^3VAE, latent diffusion transformers and TimeAutoDiff

perform the diffusion over learned latents (Li et al., 2022; Feng et al., 2024; Suh et al., 2024); TimeDiff and TMDM learn latent-state construction and denoising jointly (Shen & Kwok, 2023; Li et al., 2024); more recent variants add multi-granularity guidance, uncertainty-aware schedules, Brownian bridges, continuous sequential diffusion or nonlinear conditional transformations (Fan et al., 2024; Ye et al., 2025; Yang et al., 2024; Gao et al., 2025; Rishi et al., 2025). These models, however, do not identify which localized components of time series are unresolved after conditioning on history, instead modeling the deterministic and the stochastic parts of time series jointly.

Reduced dynamics under partial observation. When a dynamical system is only partially observed, the presence of unobserved variables generally induces non-Markovian dynamics in the observed process (Mori, 1965; Venturi & Li, 2023). For the observed context $\mathbf{X}_t \in \mathcal{F}_L$ and forecast lead $h \in \{1, \ldots, H\}$, the corresponding window-level reduced dynamics can be written as:

$$\mathbf{X}_{t+h} = \underbrace{\sum_{\tau=0}^{L-1} \mathcal{K}_{\tau,h}^{L}(\mathbf{X}_{t-\tau})}_{\text{resolved}} + \underbrace{\xi_{t+h}^{(L)}}_{\text{unresolved}}, \tag{1}$$

where $\mathcal{K}_{0,h}^{L} : \mathcal{F}_L \to \mathcal{F}_L$ denotes the instantaneous resolved dynamics operator (Markovian component) and $\{\mathcal{K}_{\tau,h}^{L}\}_{\tau=1}^{L-1}$ are the memory operators capturing history-dependent effects in the observed context (non-Markovian component). The term $\xi_{t+h}^{(L)} \in \mathcal{F}_L$ represents the unresolved component, reflecting unobserved variables. This view, standard for the modeling of dynamical systems, however, has not been applied in diffusion modeling of time series.

Position of MELD. MELD brings the separate modeling of resolved and unresolved structures to the field of time-series modeling and uses the best available approaches to model each structure. It represents time series in learnable localized time-frequency frames, forms deterministic anchor forecasts capturing history-resolved structures, and applies conditional diffusion to anchor-relative residuals to form rollouts within distributions of unresolved structures.

3 METHODS

Problem setup and MELD overview. The goal of time-series forecasting is, given an observed context window $\mathbf{X}_t = X_{t-L+1:t}$ of a time series at forecast origin t, to predict the future time-series signal $\mathbf{Y}_t = X_{t+1:t+H}$ for the forecast window. To do that, in MELD, we first define a localized time-frequency frame $\mathcal{B}_\psi$ consisting of temporal segments m, frequency bands k with learnable scales ψ, and channels c (Sec. 3.1). We use the frame $\mathcal{B}_\psi$ to represent the contents of the context window of the time series: for each tuple (m, k, c), the analysis operator A_ψ returns the corresponding component of the time series, localized in time, frequency, and channel: $V_t^\psi = A_\psi^L(\mathbf{X}_t)$ (Fig. 2). We separate components V_t^ψ into resolvable and unresolvable: a router with parameters θ assigns each context component a weight $w_{t,m,k,c}$, reflecting how much it informs the forecast. We denote the resolvable components as $\widetilde{V}_t^\psi$. A deterministic forecaster F_η maps the resolvable components $\widetilde{V}_t^\psi$ of the time-series context window to the deterministic *anchor* forecast $\widehat{\mathbf{Y}}_t^{\text{anc}} = F_\eta(\widetilde{V}_t^\psi)$ for the forecast window. To model the anchor-relative residual $R_t^\psi = U_t^\psi - U_t^{\text{anc}}$, we apply conditional diffusion to κ components of the time-series context window that show the highest product of the router scores and component's signal variances $a_{t,i} = w_{t,i} \operatorname{Var}_r(U_{t,i}^{\text{anc}})$ (Sec. 3.2). We summarize MELD's training and inference in Algorithm 1 and detail our design in the sections below.

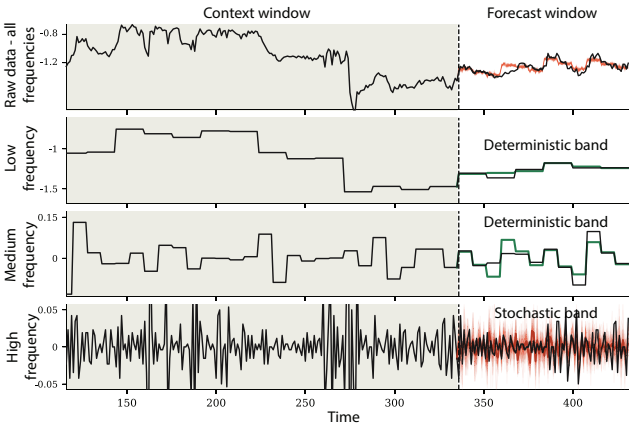


Figure 2: **Time-series decomposition in MELD.** (A) Raw time series. (B-D) components identified by learnable Gaussian scale, classified as deterministic vs. stochastic by importance scoring.

Algorithm 1 MELD overview

Require: context $\mathbf{x}_t$; future $\mathbf{y}_t$ (train only); scales ψ; forecaster F_η; router g_θ; denoiser ϵ_ω; mixture size κ

1: Split $\mathbf{x}_t$ into overlapping segments $\{\mathbf{x}_t^{(m)}\}$ and compute multiscale responses $z_{t,m,k,c} = \Phi_k^\psi[\mathbf{x}_t^{(m)}]_c$.
2: Compute routing logits $s_{t,m,k,c} = g_\theta(d_{t,m,k,c})$, and normalize to obtain routing weights $w_{t,m,k,c}$.
3: Aggregate within each segment, $\widetilde{z}_{t,m,c} = \sum_k w_{t,m,k,c} z_{t,m,k,c}$, and collect into $\widetilde{V}_t^\psi$.
4: Predict the anchor $\widehat{\mathbf{y}}_t^{\mathrm{anc}} = F_\eta(\widetilde{V}_t^\psi)$ and analyze it, $U_t^{\mathrm{anc}} = A_\psi^H(\widehat{\mathbf{y}}_t^{\mathrm{anc}})$.
5: Compute importance scores $a_{t,i} = w_{t,i}\,\mathrm{Var}_r(U_{t,i}^{\mathrm{anc}})$; let S_t be the κ components with the largest $a_{t,i}$.
6: **Train:** for each $i \in \mathcal{S}_t$, form $R_{t,i}^\psi = A_\psi^H(\mathbf{y}_t)_i - U_{t,i}^{\mathrm{anc}}$, noise it by Eq. 6, and accumulate Eq. 7.
7: **Infer:** sample $\{\widehat{R}_{t,i}^\psi\}_{i\in\mathcal{S}_t}$ by reverse diffusion; **return** $\widehat{\mathbf{y}}_t = A_\psi^{H\dagger}(\widehat{U}_t^\psi)$ with $\widehat{U}_t^\psi$ from Eq. 3.2.

3.1 DETERMINISTIC FORECASTING FROM LOCALIZED HISTORY

Definition 1 (Localized time-frequency frame)**.** *Let $\mathcal{F}_T = \mathbb{R}^{T\times C}$ denote the Hilbert space of C-variate time-series segments of length T, equipped with the standard inner product. Let the time axis of $s \in \mathcal{F}_T$ be covered by M_T overlapping segments $\mathcal{I} = \{I_m\}_{m=1}^{M_T}$, each of length P_T, with local segment $s^{(m)} = s[I_m, :]$. Let ψ parameterize the temporal scales and let $\phi_{m,k,c,r}^\psi$, with $k = 1,\dots,K$ and $r = 1,\dots,P_T$, denote a frame element localized to segment m, frequency component k, channel c, and local position r. The localized time-frequency frame is $\mathcal{B}_\psi = \{\phi_{m,k,c,r}^\psi\}_{m,k,c,r}$.*

Definition 1 specifies a set of localized time-frequency features $\mathcal{B}_\psi^T$ on $\mathcal{F}_T = \mathbb{R}^{T\times C}$, where ψ parameterize a learnable set of frequencies to be extracted from the data. For any context window $s \in \mathcal{F}_T$ of the time series, the extracted features $\langle s, \phi_{m,k,c,r}^\psi\rangle$ define its localized time-frequency components, to be processed by the operator below.

Definition 2 (Analysis operator)**.** *The analysis operator associated with $\mathcal{B}_\psi^T$ is*

$$A_\psi^T : \mathcal{F}_T \to \mathfrak{C}_T, \qquad A_\psi^T(s) = \big(\langle s, \phi_{m,k,c,r}^\psi\rangle_{\mathcal{F}_T}\big)_{m,k,c,r}, \tag{2}$$

where $\mathfrak{C}_T = \mathbb{R}^{M_T\times K\times C\times P_T}$ is the localized coefficient space. We call $A_\psi^T(s)$ the localized coefficient map of s. For $T \in \{L, H\}$, let $\mathcal{J}_T = \{1,\dots,M_T\}\times\{1,\dots,K\}\times\{1,\dots,C\}$, and index localized components by $i = (m,k,c) \in \mathcal{J}_T$, so that $A_\psi^T(s)_i \in \mathbb{R}^{P_T}$. We require that A_ψ^T is injective and admits a synthesis left inverse $A_\psi^{T\dagger}$ satisfying $A_\psi^{T\dagger}A_\psi^T = I_{\mathcal{F}_T}$.

At forecast origin t, let $V_t^\psi = A_\psi^L(\mathbf{X}_t)$ and $U_t^\psi = A_\psi^H(\mathbf{Y}_t)$ denote the context-window and forecast-window coefficient maps. The context-window coefficients are available at both training and inference. The forecast-window coefficients U_t^ψ are computed only during training; they are unavailable and do not participate in the predictive support selection at inference.

Since $A_\psi^{L\dagger}A_\psi^L = I_{\mathcal{F}_L}$, we have $\mathbf{X}_t = A_\psi^{L\dagger}V_t^\psi$. Applying A_ψ^L to Eq. 1 therefore gives

$$V_{t+h}^\psi = \underbrace{\textstyle\sum_{\tau=0}^{L-1} A_\psi^L \mathcal{K}_\tau^L A_\psi^{L\dagger} V_{t-\tau}^\psi}_{\text{resolved}} + \underbrace{A_\psi^L \xi_{t+h}^{(L)}}_{\text{unresolved}}. \tag{3}$$

Eq. 3 separates the history-supported contribution to the forecast from unresolved contribution in MELD's localized time-frequency frame. The transform operators $A_\psi^L\mathcal{K}_\tau^L A_\psi^{L\dagger}$ describe localized Markovian (instantaneous) and non-Markovian (memory) contributions from the observed context.

Learnable localized frequency in decomposition of time-series data. MELD instantiates the analysis operator A_ψ^L as a finite set of temporal operators $\{\Phi_k^\psi\}_{k=1}^K$ with learnable scales $\psi = \{\sigma_1,\dots,\sigma_{K-1}\}$, applied per channel within each segment. Let $\mathcal{G}_\sigma[s] = g_\sigma * s$ denote Gaussian smoothing (with reflection padding), and define $\mathcal{L}_0^\psi = I$ and $\mathcal{L}_k^\psi = \mathcal{G}_{\sigma_k}\mathcal{L}_{k-1}^\psi$ for $k = 1,\dots,K-1$. We then set $\Phi_1^\psi = I - \mathcal{L}_1^\psi$, $\Phi_k^\psi = \mathcal{L}_{k-1}^\psi - \mathcal{L}_k^\psi$ for $2 \le k \le K-1$, and $\Phi_K^\psi = \mathcal{L}_{K-1}^\psi$. By construction $\sum_{k=1}^K \Phi_k^\psi = I$, giving an exact multiscale decomposition whose components progress from high-pass (Φ_1^ψ) through intermediate band-pass operators to low-pass (Φ_K^ψ), so that k indexes progressively lower-frequency structure. Each Φ_k^ψ acts independently along channels: $z_{t,i} = \Phi_k^\psi[s^{(m)}]_c \in \mathbb{R}^{P_T}$, so that $A_\psi^T(s)_i = z_{t,i}$; nonlinear interactions are left to the forecaster F_η.

Context-dependent deterministic forecasting. A learnable router assigns weights $w_{t,m,k,c}$ to the localized time-frequency representations of context-window time-series data and aggregates them within each time-frequency segment and channel, $\widetilde{z}_{t,m,c} = \sum_k w_{t,m,k,c} z_{t,m,k,c} = \big(\sum_k w_{t,m,k,c} \Phi_k^{\psi}\big)[\mathbf{X}_t^{(m)}]_c$, thus forming a finite context-dependent realization of the "resolved" term in Eq. 3. Jointly, the terms $\{\widetilde{z}_{t,m,c}\}$ form $\widetilde{V}_t^{\psi}$, from which the deterministic forecaster F_η predicts the contents of the forecast window $U_t^{\mathrm{anc}} = F_\eta(\widetilde{V}_t^{\psi})$. The forecaster F_η is instantiated as a collection of two-layer MLP heads (Linear-ReLU-Linear, separate for each time-frequency segment of the context-window data). The outputs of these MLPs are rescaled by a learned gate and then summed to form a joint forecast. U_t^{anc} approximates the future coefficient map supported by the observed history; the remaining variation is modeled in Sec. 3.2.

3.2 A MIXTURE OF LOCALIZED RESIDUAL DIFFUSIONS

Definition 3 (Conditional unresolved energy)**.** *For $i \in \mathcal{J}_H$, let $\mu_{t,i}^{\psi} = \mathbb{E}[U_{t,i}^{\psi} \mid \widetilde{V}_t^{\psi}]$. The conditional unresolved energy of component i is $\rho_{t,i} = \mathbb{E}\big[\|U_{t,i}^{\psi} - \mu_{t,i}^{\psi}\|_2^2 \;\big|\; \widetilde{V}_t^{\psi}\big]$.*

Since $U_{t,i}^{\mathrm{anc}}$ is a function of $\widetilde{V}_t^{\psi}$, the residual energy splits into unresolved variation and anchor error,

$$\mathbb{E}\big[\|R_{t,i}^{\psi}\|_2^2 \;\big|\; \widetilde{V}_t^{\psi}\big] = \rho_{t,i} + \big\|\mu_{t,i}^{\psi} - U_{t,i}^{\mathrm{anc}}\big\|_2^2. \tag{4}$$

Empirically, $\rho_{t,i}$ may differ across time-frequency components of the time series: a few components may carry the most of residual energy, while the majority of the components may be nearly fully determined by the context. MELD selects κ components with high residual energy and applies diffusion processes to them.

Importance score. Ideally, the stochastic diffusion modeling would be applied to the κ components of the time-series time-frequency representation with the largest $\rho_{t,i}$, but, at inference, ρ depends on the unknown contents of the forecast window. To estimate it, we instead use two context-only quantities as follows. The routing weight $w_{t,i}$ shows how much the router relies on a time-frequency component k of the context. The anchor's variability within the component, $\mathrm{Var}_r(U_{t,i}^{\mathrm{anc}})$ over its P_H positions r, shows how much energy this component carries overall. We define their product,

$$a_{t,i} = w_{t,i}\,\mathrm{Var}_r\big(U_{t,i}^{\mathrm{anc}}\big), \qquad i \in \mathcal{J}_H, \tag{5}$$

as the *importance score* and denote $\mathcal{S}_t$ as the set of κ largest importance scores. The score is used for ranking only. The mixture size κ sets how much of the forecast is modeled as stochastic: at $\kappa = |\mathcal{J}_H|$ every time-frequency component of the time-series context window is assigned a diffusion process; at $\kappa = 1$ a single localized diffusion acts on the least resolved component and the forecast is deterministic elsewhere. At inference, the anchor and the support are computed from the context, reverse diffusion draws $\{\widehat{R}_{t,i}^{\psi}\}_{i\in\mathcal{S}_t}$, resulting in $\widehat{U}_{t,i}^{\psi} = U_{t,i}^{\mathrm{anc}} + \mathbf{1}[i \in \mathcal{S}_t]\,\widehat{R}_{t,i}^{\psi}$, $\widehat{\mathbf{Y}}_t = A_{\psi}^{H\dagger}\big(\widehat{U}_t^{\psi}\big)$.

Component-wise residual diffusion. Let $\mathcal{S}_t \subset \mathcal{J}_H$, $|\mathcal{S}_t| = \kappa$, index the components that carry a diffusion process. Each $i \in \mathcal{S}_t$ diffuses its own residual $R_{t,i}^{\psi} \in \mathbb{R}^{P_H}$; at step n,

$$R_{t,i}^{[n]} = \sqrt{\bar{\alpha}_n}\,R_{t,i}^{\psi} + \sqrt{1-\bar{\alpha}_n}\,\epsilon_i, \qquad \epsilon_i \sim \mathcal{N}(0, I), \tag{6}$$

where ϵ_i is artificial diffusion noise. The κ processes share one denoiser ϵ_ω conditioned on $\widetilde{V}_t^{\psi}$, U_t^{anc} and the component identity, so the mixture costs a component embedding rather than κ networks:

$$\mathcal{L}_{\mathrm{diff}} = \mathbb{E}_{n,\epsilon} \sum_{i\in\mathcal{S}_t} \big\|\epsilon_i - \epsilon_\omega\big(R_{t,i}^{[n]}, n, i, \widetilde{V}_t^{\psi}, U_t^{\mathrm{anc}}\big)\big\|_2^2. \tag{7}$$

Each residual is standardized by the scale of its band in the observed context, which is available at inference and, unlike the anchor's own scale, does not vanish on bands the anchor leaves flat.

3.3 FORECASTING AND EVALUATION

The full model is trained with $\mathcal{L} = \mathcal{L}_{\mathrm{det}} + \gamma\mathcal{L}_{\mathrm{diff}}$, where $\mathcal{L}_{\mathrm{det}}$ is the squared-error forecasting loss. We report CRPS, and the minimum-over-N (MoN) error to measure whether generated samples contain a realization close to the observed future. Here MoN is: $\mathrm{MoN}_N(\ell) = \mathbb{E}[\min_{1\le n\le N} \ell(\widehat{y}^{(n)}, y)]$.

Theorem 1. *Let $\{(x_q, y_q)\}_{q=1}^{Q}$ be a fixed reference set. For model j, let $y_{q,1}^{j}, \ldots, y_{q,N}^{j} \overset{\text{iid}}{\sim} p_j(\cdot \mid x_q)$, with $p_j(\cdot \mid x_q)$ bounded and continuous at y_q with $p_j(y_q \mid x_q) > 0$. Define $T_N^{q,j} = \min_{1 \le n \le N} \|y_{q,n}^{j} - y_q\|_2^2$ as the top-1 MSE value at test point i under model j. Then as $N \to \infty$,*

$$\mathbb{E}\big[\tau_N^{q,j} \mid x_q, y_q\big] = \frac{\Gamma(1+2/d)}{\big(V_d\, p_j(y_q \mid x_q)\big)^{2/d}} N^{-2/d}\big(1 + o(1)\big),$$

where V_d is the volume of the unit ball in $\mathbb{R}^d$. Hence $\mathbb{E}[\tau_N^{q,1}] < \mathbb{E}[\tau_N^{q,2}]$ if and only if $p_1(y_q \mid x_q) > p_2(y_q \mid x_q)$: MoN ranks models by their density at the observed future.

4 RESULTS

We evaluate MELD on a standard set of eight multivariate long-term forecasting benchmarks spanning diverse resolutions and domains: ETTh1, ETTh2, ETTm1, ETTm2 (Zhou et al., 2021); Electricity, Traffic and Exchange (Lai et al., 2018); and Weather (Zeng et al., 2023). We compare MELD with deterministic and diffusion (stochastic) baselines under protocols published with respective models.

4.1 MELD OUTPERFORMS DETERMINISTIC BASELINES ON STANDARD BENCHMARKS

To compare MELD with the strongest deterministic baselines, we considered D-PAD (Yuan & Chen, 2024), DLinear (Zeng et al., 2023), TimeMixer (Wang et al., 2024), N-HiTS (Challu et al., 2023), iTransformer (Liu et al., 2023), PatchTST (Nie et al., 2022) and XPatch (Stitsyuk & Choi, 2025). To match the evaluation settings of these works, we evaluated MELD at $L = 336$, $H \in \{96, 192, 336, 720\}$, on MSE and MAE and compared it with the published numbers for these works. (Of note, other models in the field are also typically evaluated under these settings, so an interested reader may compare the numbers directly to other literature without the need to rerun anything.) We report the results, aggregated over seeds and the forecast horizons H, in Table 1; the detailed per-horizon results are available in Table A1. To establish how MELD compares to deterministic baseline models, we considered the distribution of our model's performances (Table A2 shows the standard deviations) and performed the t-tests, followed by Benjamini-Hochberg false discovery rate (FDR) corrections (Table A5 reports the p-values and the FDR-corrected q-values). To perform the t-tests, we re-ran baseline models to obtain per-seed results (Table A3). We found that, compared to the strongest deterministic forecaster baselines, MELD has attained the lowest averaged error in 15 of 16 standard evaluation settings (Table A1; summary in Table 1). Overall, these results suggest that learnable decomposition of time-series signal into localized time-frequency features followed by learnable routing and selective diffusion is effective for making point forecasts for time-series data.

Table 1: **Comparison with deterministic baselines**: per-dataset averages over horizons $H \in \{96, 192, 336, 720\}$. All models use context length $L = 336$. MELD entries are means over 8 training seeds. Best (lowest) per-row per-metric results are shown in bold text.

Dataset	MELD		D-PAD		DLinear		TimeMixer		N-HiTS		iTrans.		PatchTST		XPatch	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.381**	**0.404**	0.386	0.407	0.423	0.437	0.450	0.447	0.529	0.556	0.454	0.448	0.413	0.434	0.439	0.436
ETTh2	**0.316**	**0.364**	0.331	0.373	0.431	0.447	0.366	0.404	0.390	0.422	0.383	0.406	0.331	0.381	0.346	0.384
ETTm1	**0.341**	**0.362**	0.343	0.364	0.357	0.379	0.362	0.392	0.444	0.501	0.407	0.410	0.353	0.382	0.353	0.373
ETTm2	**0.245**	**0.297**	0.250	0.306	0.268	0.332	0.264	0.321	0.292	0.334	0.288	0.332	0.256	0.316	0.255	0.309
Traffic	**0.383**	**0.250**	0.385	0.254	0.429	0.292	0.412	0.288	0.451	0.320	0.428	0.282	0.391	0.264	0.500	0.282
Electricity	**0.153**	**0.238**	0.160	0.249	0.164	0.261	0.164	0.256	0.191	0.294	0.175	0.260	0.159	0.253	0.179	0.264
Exchange	0.307	**0.370**	0.363	0.403	**0.297**	0.378	0.413	0.425	0.367	0.403	0.360	0.403	0.385	0.421	0.393	0.415
Weather	**0.216**	**0.244**	0.219	0.248	0.246	0.300	0.230	0.270	0.251	0.279	0.258	0.278	0.225	0.264	0.227	0.253
# best (of 16)	**15**		0		1		0		0		0		0		0	

4.2 MELD OUTPERFORMS DIFFUSION BASELINES ON STANDARD BENCHMARKS

To compare MELD with the strongest diffusion baselines (stochastic, as opposed to deterministic in the previous section), we considered TimeGrad, CSDI, SSSD, D^3VAE, TimeDiff and TMDM. To match the evaluation settings of these works, we followed (Li et al., 2024) and evaluated MELD using

Table 2: **Comparison with stochastic baselines.** Baseline MSE/MAE numbers are from Li et al. (2024), and baseline CRPS results are reported by TMDM. All models are evaluated at horizon $H = 192$ with context length $L = 96$. Best (lowest) per-column per-metric results are shown in bold text.

Model	Exchange			ETTm2			Electricity			Traffic			Weather		
	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS	MSE	MAE	CRPS
TimeGrad	2.43	0.90	0.66	1.36	0.74	0.54	0.69	0.74	0.40	0.96	0.81	0.39	0.90	0.57	0.43
CSDI	1.67	0.75	0.45	1.28	0.67	0.50	0.56	0.81	0.42	0.94	0.68	0.37	0.86	0.56	0.37
SSSD	0.90	0.86	0.56	0.97	0.56	0.57	0.47	0.60	0.46	0.81	0.50	0.41	0.67	0.49	0.44
D^3VAE	0.76	0.62	0.41	0.79	0.46	0.44	0.33	0.49	0.36	0.79	0.43	0.31	0.43	0.34	0.39
TimeDiff	0.48	0.43	0.38	0.41	0.42	0.40	0.27	0.32	0.38	0.68	0.47	0.28	0.36	0.37	0.33
TMDM	0.26	0.37	0.32	0.27	0.35	0.37	0.19	0.27	0.33	0.60	0.35	0.26	0.28	0.25	0.36
MELD	**0.17**	**0.29**	**0.25**	**0.23**	**0.29**	**0.23**	**0.18**	**0.26**	**0.22**	**0.49**	**0.31**	**0.24**	**0.21**	**0.24**	**0.21**

Table 3: **Impact of the diffusion mixture size κ on stochastic forecasting.** κ=0 is the deterministic anchor; κ=5 diffuses every component. Best CRPS per column is shown in bold text. L=336, H=96.

κ	ETTh1			ETTh2			ETTm1			ETTm2			Exchange		
	CRPS	MoN		CRPS	MoN		CRPS	MoN		CRPS	MoN		CRPS	MoN	
		MAE	MSE		MAE	MSE		MAE	MSE		MAE	MSE		MAE	MSE
0	0.3832	0.3933	0.3694	0.3205	0.3238	0.2658	0.3368	0.3395	0.2951	0.2446	0.2474	0.1645	0.2042	0.2093	0.0881
1	0.3083	**0.3891**	**0.3583**	0.2600	**0.3198**	**0.2594**	**0.3006**	**0.3391**	**0.2919**	0.1916	**0.2464**	**0.1619**	0.1896	**0.2085**	**0.0875**
2	0.3011	0.3926	0.3682	0.2561	0.3230	0.2630	0.3107	0.3395	0.2956	**0.1915**	0.2472	0.1640	**0.1573**	0.2090	0.0880
3	**0.2989**	0.3931	0.3692	0.2540	0.3232	0.2632	0.3145	0.3395	0.2950	0.1918	0.2473	0.1640	0.1574	0.2097	0.0885
4	0.2996	0.3929	0.3684	0.2533	0.3230	0.2628	0.3165	0.3397	0.2959	0.1921	0.2474	0.1640	0.1574	0.2092	0.0881
5	0.3001	0.3931	0.3695	**0.2522**	0.3234	0.2635	0.3188	0.3397	0.2960	0.1928	0.2473	0.1642	0.1575	0.2093	0.0881

$L = 96$, $H = 192$, using MSE, MAE and CRPS as metrics. To perform the comparison, we used the baseline numbers reported by (Li et al., 2024). For MELD, we averaged the metrics over 8 seeds. We report our results in Table 2. We found that, compared to the strong diffusion forecaster baselines, MELD has shown the best performance under all fifteen considered evaluation settings. We provide the statistical significance tests in Table A6. We found that our model is significantly better than any of the baselines in 12 out of 15 settings. Overall, these results suggest that selective diffusion is effective for capturing and forecasting stochastic components of time-series data.

4.3 DIFFUSION IN MELD IMPROVES STOCHASTIC FORECASTING METRICS

To analyze the role of MELD's selective diffusion in optimizing the metrics of stochastic forecasting, we varied the number of diffusion processes κ in MELD's mixture of diffusions. We considered the range of κ values from 0 (no diffusion) to $\kappa = 5$ (the maximum number of diffusion processes, matching the number of input components in our time series decomposition). To perform the comparison, we computed the CRPS for each of these settings. We report the results in Table 3. We found that the largest gain in the CRPS was related to introducing the first diffusion process: Moving from $\kappa = 0$ to $\kappa = 1$ has lowered the CRPS by 7-22% and, on four of the five datasets, already realized 89-100% of the best reduction attained at any κ. To further probe these results, we used our Theorem 1 (proof in Appendix A.1) and repeated the analysis using the MoN metric. For MoN, we used N=5000 samples and dropout (Table A4). We present our MoN results in the same Table 3, where we also observed the improvement of the metric between $\kappa = 0$ and $\kappa = 1$. Overall, these results suggest that diffusion plays a significant role in capturing the stochastic component of the data beyond the point forecast of the model's deterministic anchor.

4.4 MELD'S IMPORTANCE SCORE CAPTURES THE RESIDUAL ENERGY

Section 4.3 established that a single diffused component ($\kappa = 1$) can recover most of the attainable CRPS reduction. The remaining design choice is the rule that selects that component. Table 4 fixes $\kappa = 1$ and evaluates the importance score of Eq. 5 against three alternative ways of selection under the same budget. We considered: i) the variance-only selection that retains the anchor-variability factor $\mathrm{Var}_r(U^{\mathrm{anc}}_{t,i})$ and discards the routing weight $w_{t,i}$, isolating the contribution of the router; ii) Random selection that draws the components uniformly, isolating the contribution of ranking as such; iii) The oracle that selects the components to be diffused by their realized residual energy

Table 4: **Comparison of component selection mechanisms for the mixture of diffusions.** Experiments were performed under a fixed budget (κ=1, $L = 336$, H=192). The best CRPS values per row are shown in bold text.

	CRPS by selection rule				Agreement with oracle		
Dataset	Learned	Oracle	Variance	Random	Spearman learned	Spearman variance	Top-1 learned
ETTh1	**0.312**	**0.312**	0.326	0.340	0.213±.001	0.182±.001	1.000±.000
ETTh2	**0.288**	**0.288**	0.303	0.321	0.682±.015	0.613±.014	0.941±.002
ETTm1	0.291	**0.286**	0.293	0.295	0.668±.002	0.607±.001	0.996±.001
ETTm2	**0.193**	**0.193**	0.194	0.226	0.567±.044	0.312±.033	0.999±.000
Exchange	**0.188**	**0.188**	0.192	0.198	0.755±.009	0.721±.017	0.326±.084
Average	0.254	**0.253**	0.262	0.276	**0.577**	0.487	0.852

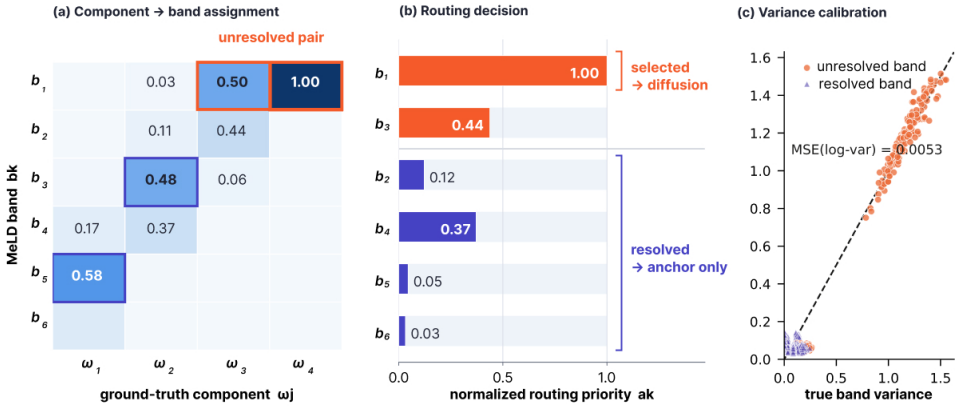


Figure 3: **Forecasts for synthetic data with known generative process.** (A) Ground-truth frequency bands vs. frequency bands learned by MELD. (B) Ground-truth stochastic and deterministic frequency bands vs. stochastic and deterministic bands identified by MELD. (C) Ground-truth variance of data's frequency bands vs. the variance predicted by MELD.

$\|R^{\psi}_{t,i}\|_2^2$. The oracle requires the observed future and serves as an upper-bound for the forecasting rather than a deployable method. We found that MELD's learned score captures the most of CRPS among the considered methods, mostly matching the oracle's numbers. Overall, these results suggest that MELD's router is effective for separating the deterministic and the stochastic components of time-series data.

4.5 MELD RECONSTRUCTS GROUND-TRUTH BANDS IN SYNTHETIC DATA

To ensure that our model captures data's stochastic generative processes, we tested our model on a synthetic dataset where ground-truth generative process is known. To generate a synthetic dataset, we randomly selected 4 frequency bands. We populated 2 of them with signal to model deterministic bands. We populated 2 other bands with noise to model stochastic bands. We then trained MELD on the resulting data and examined the trained model. We found that all the 4 frequency bands of the dataset were captured by the model (Figure 3A). The importance scores for the stochastic bands were higher than the scores for deterministic bands, leading to the correct classification (Figure 3B). The prediction for the data's variance in stochastic bands matched the ground truth (Figure 3C). Together, these results suggest that MELD captures generative stochastic processes of time-series data.

4.6 ABLATIONS

To examine the contribution of MELD's individual components to forecasting performance, we conducted an ablation study (Table 5). For temporal decomposition, we found that segmenting

Table 5: **Ablation of MELD.** Lower is better. We report averaged MSE/MAE over ETTh1, ETTh2, ETTm1, ETTm2, and Exchange. Each entry reports MSE/MAE; Avg. is the mean across the five datasets.

Ablation	**ETTh1**	**ETTh2**	**ETTm1**	**ETTm2**	**Exchange**	**Avg.**
Full MELD	0.381/0.404	0.316/0.364	0.341/0.362	0.245/0.297	0.307/0.370	0.318/0.359
uniform routing	0.408/0.421	0.320/0.366	0.346/0.369	0.249/0.304	0.310/0.375	0.327/0.367
w/o multi-step routing	0.418/0.425	0.328/0.369	0.350/0.370	0.252/0.306	0.337/0.387	0.337/0.371
w/o time overlaps	0.408/0.421	0.319/0.365	0.346/0.370	0.250/0.305	0.329/0.386	0.330/0.369
w/o segment-wise	0.711/0.580	0.363/0.410	0.698/0.555	0.337/0.377	0.572/0.542	0.536/0.493
w/o aggregation	0.424/0.431	0.327/0.368	0.349/0.370	0.250/0.305	0.311/0.377	0.332/0.370
w/o forecast seg	0.407/0.420	0.318/0.366	0.343/0.368	0.250/0.304	0.325/0.380	0.329/0.368
w/o forecast window	0.409/0.422	0.319/0.366	0.344/0.369	0.249/0.305	0.308/0.374	0.326/0.367
Fixed Haar bands	0.410/0.423	0.318/0.366	0.343/0.368	0.250/0.305	0.309/0.378	0.326/0.368
Fixed Gaussian scale	0.407/0.420	0.320/0.366	0.344/0.368	0.249/0.304	0.315/0.378	0.327/0.367

both the context and forecast windows improves MSE. For localized decomposition, the learnable Gaussian scale used by MELD outperformed their fixed-scale counterparts as well as alternative fixed feature sets, including Haar wavelets. For deterministic forecasting, importance scoring, segment-wise time-frequency forecasts, and their aggregation contributed to improved performance. Finally, for stochastic forecasting, initializing the forecast-window segments with noise also improved forecasting performance. Overall, these results support the importance of the model's design choices for improving the time-series forecasting metrics.

5 DISCUSSION

In this work, we introduced MELD, a generative forecasting framework that combines localized diffusion with a structured decomposition space for time-series prediction. By separating resolved and unresolved components of the forecast, MELD applies diffusion only to the subspace that remains uncertain, improving both forecasting accuracy and interpretability. Our results suggest that uncertainty in many benchmark forecasting datasets is concentrated in a relatively small subset of temporal components rather than being uniformly distributed across the signal. The strong performance of selective diffusion compared to both deterministic baselines and full-horizon diffusion further supports the view that much of the future is already constrained by the observed history, while only a limited subspace remains stochastic. Our results suggest that effective forecasting requires not only expressive generative models, but also mechanisms that isolate and model the uncertain components of the future.

Limitations and future work. Standard forecasting benchmarks are collected without records of the variables left unobserved, so the degree of partial observation differs across datasets and cannot be controlled by any method evaluated on them; this is why we complement benchmark results with a synthetic setting where the generative process is known. Probabilistic forecasting also lacks a single agreed evaluation protocol - sample counts, aggregation and metric variants differ across papers - so we report CRPS and MoN under the protocols published with the models we compare against, and state the settings in each table. These observations also point forward. Because MELD's routing reports where in time and frequency the unobserved part of a system leaves its trace, it applies wherever partial observation is intrinsic - physiological recordings, climate fields, reduced descriptions of physical systems - and to reconstruction tasks beyond forecasting - which can be studied in real-world cases in future research.

Outlook. MELD points toward a new class of forecasting systems that treat uncertainty in time series as a structured object to be isolated and modeled, rather than diffused over uniformly. This perspective motivates more scalable and adaptive generative models that dynamically allocate structured stochastic computation only to the components of the forecasts that remain unresolved.

6 AI DISCLOSURE

In this work, we used generative AI tools to aid and polish writing, for retrieval and discovery of related work, and for writing and debugging experiment code, including baseline reproduction scripts, cluster job scripts, evaluation utilities, and the scripts that aggregate per-seed results and compute the

reported statistical tests. We have not used generative AI tools for proving mathematical claims or for designing the MELD architecture, including the band-selective residual diffusion and the routing mechanism, and the remaining required-disclosure tasks are not applicable to this work. Additionally, we used generative AI tools to audit consistency between the manuscript and the implementation (e.g., checking that equations, evaluation protocols, and hyperparameters stated in the text match the code) and to cross-check reported tables against raw per-seed logs. We have reviewed all AI-assisted work. Every reported number was produced by running the code on our own hardware and is traceable to raw logs. AI-assisted experiment code was reviewed by the author(s), and baseline reproductions were checked against the official implementations and their published hyperparameters. Significance tests were recomputed directly from per-seed results, and AI-suggested text edits were checked against the implementation before inclusion. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Cristian Challu, Kin G Olivares, Boris N Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. Nhits: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 6989–6997, 2023.

Xinyao Fan, Yueying Wu, Chang Xu, Yuhao Huang, Weiqing Liu, and Jiang Bian. Mg-tsd: Multi-granularity time series diffusion models with guided learning process. *arXiv preprint arXiv:2403.05751*, 2024.

Shibo Feng, Chunyan Miao, Zhong Zhang, and Peilin Zhao. Latent diffusion transformer for probabilistic time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 11979–11987, 2024.

Kelum Gajamannage, Yonggi Park, and Dilhani I Jayathilake. Real-time forecasting of time series in financial markets using sequentially trained dual-lstms. *Expert Systems with Applications*, 223:119879, 2023.

Jiaxin Gao, Qinglong Cao, and Yuntian Chen. Auto-regressive moving diffusion models for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 16727–16735, 2025.

Marcel Kollovieh, Abdul Fatir Ansari, Michael Bohlke-Schneider, Jasper Zschiegner, Hao Wang, and Yuyang Bernie Wang. Predict, refine, synthesize: Self-guiding diffusion models for probabilistic time series forecasting. *Advances in Neural Information Processing Systems*, 36:28341–28364, 2023.

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.

YW Lee, KG Tay, and YY Choy. Forecasting electricity consumption using time series model. *International Journal of Engineering & Technology*, 7(4.30):218–223, 2018.

Yan Li, Xinjiang Lu, Yaqing Wang, and Dejing Dou. Generative time series forecasting with diffusion, denoise, and disentanglement. *Advances in Neural Information Processing Systems*, 35:23009–23022, 2022.

Yuxin Li, Wenchao Chen, Xinyue Hu, Bo Chen, Baolin Sun, and Mingyuan Zhou. Transformer-modulated diffusion models for probabilistic multivariate time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

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.

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

Shizhan Liu, Hang Yu, Cong Liao, Jianguo Li, Weiyao Lin, Alex X Liu, and Schahram Dustdar. Pyraformer: Low-complexity pyramidal attention for long-range time series modeling and forecasting. In *International Conference on Learning Representations*, 2022b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yuxiao Luo, Songming Zhang, Ziyu Lyu, and Yuhan Hu. Tfdnet: Time–frequency enhanced decomposed network for long-term time series forecasting. *Pattern Recognition*, 162:111412, 2025.

Hazime Mori. Transport, collective motion, and brownian motion. *Progress of theoretical physics*, 33(3): 423–455, 1965.

Nam Nguyen and Brian Quanz. Temporal latent auto-encoder: A method for probabilistic multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 9117–9125, 2021.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

J Rishi, GVS Mothish, and Deepak Subramani. Conditional diffusion model with nonlinear data transformation for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025.

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

Artyom Stitsyuk and Jaesik Choi. xpatch: Dual-stream time series forecasting with exponential seasonal-trend decomposition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20601–20609, 2025.

Namjoon Suh, Yuning Yang, Din-Yin Hsieh, Qitong Luan, Shirong Xu, Shixiang Zhu, and Guang Cheng. Timeautodiff: Combining autoencoder and diffusion model for time series tabular data synthesizing. *arXiv preprint arXiv:2406.16028*, 2024.

Daniele Venturi and Xiantao Li. The mori–zwanzig formulation of deep learning: D. venturi, x. li. *Research in the Mathematical Sciences*, 10(2):23, 2023.

Huiqiang Wang, Jian Peng, Feihu Huang, Jince Wang, Junhui Chen, and Yifei Xiao. MICN: Multi-scale local and global context modeling for long-term series forecasting. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=zt53IDUR1U.

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. *arXiv preprint arXiv:2405.14616*, 2024.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Cost: Contrastive learning of disentangled seasonal-trend representations for time series forecasting. *arXiv preprint arXiv:2202.01575*, 2022a.

Gerald Woo, Chenghao Liu, Doyen Sahoo, Akshat Kumar, and Steven Hoi. Etsformer: Exponential smoothing transformers for time-series forecasting. *arXiv preprint arXiv:2202.01381*, 2022b.

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. In *International Conference on Learning Representations*, 2023.

Hao Yang, Zhanbo Feng, Feng Zhou, Robert C Qiu, and Zenan Ling. Series-to-series diffusion bridge model. *arXiv preprint arXiv:2411.04491*, 2024.

Weiwei Ye, Zhuopeng Xu, and Ning Gui. Non-stationary diffusion for probabilistic time series forecasting. *arXiv preprint arXiv:2505.04278*, 2025.

Kun Yi, Qi Zhang, Wei Fan, Shoujin Wang, Pengyang Wang, Hui He, Ning An, Defu Lian, Longbing Cao, and Zhendong Niu. Frequency-domain mlps are more effective learners in time series forecasting. *Advances in Neural Information Processing Systems*, 36:76656–76679, 2023.

Xiaobing Yuan and Ling Chen. D-pad: Deep-shallow multi-frequency patterns disentangling for time series forecasting. *arXiv preprint arXiv:2403.17814*, 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*, 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.

Tian Zhou, Ziqing Ma, Qingsong Wen, Liang Sun, Tao Yao, Wotao Yin, Rong Jin, et al. Film: Frequency improved legendre memory model for long-term time series forecasting. *Advances in neural information processing systems*, 35:12677–12690, 2022a.

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

A Appendix

A.1 Proof of Theorem 1.

Theorem A.1. *Let f be a probability density on $\mathbb{R}^d$ that is bounded, continuous at y^* with $f(y^*) > 0$, and satisfies $\mathbb{E}_{y\sim f}\|y - y^*\|^2 < \infty$. Let $y_1, \dots, y_K \overset{iid}{\sim} f$ and define*

$$\tau_K \;=\; \min_{1\le k\le K} \|y_k - y^*\|^2.$$

Then

$$\mathbb{E}[\tau_K] \;=\; \frac{\Gamma(1+2/d)}{\big(V_d\, f(y^*)\big)^{2/d}} \cdot K^{-2/d}\,\big(1+o(1)\big) \qquad \text{as } K\to\infty,$$

where V_d denotes the volume of the unit ball in $\mathbb{R}^d$.

Proof. Define the cumulative distribution of squared distance from y^*,

$$F(t) \;=\; \Pr_{y\sim f}\big(\|y-y^*\|^2 \le t\big) \;=\; \int_{B(y^*,\sqrt{t})} f(y)\,dy, \qquad t\ge 0,$$

where $B(y^*, r)$ is the closed Euclidean ball of radius r centered at y^*.

By continuity of f at y^*, for every $\varepsilon > 0$ there exists $\delta > 0$ such that $|f(y) - f(y^*)| < \varepsilon$ for all $y \in B(y^*, \delta)$. For $t \le \delta^2$,

$$\big|F(t) - V_d\, t^{d/2}\, f(y^*)\big| \;=\; \left|\int_{B(y^*,\sqrt{t})} \big(f(y)-f(y^*)\big)\,dy\right| \;\le\; \varepsilon\, V_d\, t^{d/2}.$$

Since ε was arbitrary,

$$F(t) \;=\; V_d\, f(y^*)\, t^{d/2} + o\big(t^{d/2}\big) \qquad \text{as } t\to 0^+. \tag{$*$}$$

Since the samples are i.i.d., $\Pr(\tau_K > t) = (1-F(t))^K$. As $\tau_K \ge 0$ and $\mathbb{E}[\tau_K] \le \mathbb{E}\|y_1 - y^*\|^2 < \infty$,

$$\mathbb{E}[\tau_K] \;=\; \int_0^\infty \big(1-F(t)\big)^K\,dt.$$

Substitute $t = (s/K)^{2/d}$, so $dt = \frac{2}{d}\,K^{-2/d}\,s^{2/d-1}\,ds$. Then

$$\mathbb{E}[\tau_K] \;=\; K^{-2/d}\int_0^\infty \left(1 - F\left(\frac{s^{2/d}}{K^{2/d}}\right)\right)^K \cdot \frac{2}{d}\,s^{2/d-1}\,ds.$$

Define

$$g_K(s) \;:=\; \left(1 - F\left(\frac{s^{2/d}}{K^{2/d}}\right)\right)^K \cdot \frac{2}{d}\,s^{2/d-1}, \qquad g_\infty(s) \;:=\; e^{-V_d f(y^*)\,s}\cdot\frac{2}{d}\,s^{2/d-1}.$$

Our goal is to show $\int_0^\infty g_K(s)\,ds \to \int_0^\infty g_\infty(s)\,ds$.

Fix $s > 0$. By $(*)$,

$$F\left(\frac{s^{2/d}}{K^{2/d}}\right) \;=\; \frac{V_d f(y^*)\,s}{K} + o(1/K),$$

so $K\cdot F(s^{2/d}/K^{2/d}) \to V_d f(y^*)\,s$, and therefore

$$\big(1 - F(s^{2/d}/K^{2/d})\big)^K \;\longrightarrow\; e^{-V_d f(y^*)\,s}.$$

Hence $g_K(s) \to g_\infty(s)$ pointwise on $(0,\infty)$.

We construct an integrable envelope h with $g_K \le h$ for all sufficiently large K.

Since f is bounded and continuous at y^* with $f(y^*) > 0$, there exist constants $c > 0$ and $r_0 > 0$ such that

$$F(t) \;\geq\; c\,t^{d/2} \qquad \text{for all } 0 \leq t \leq r_0^2.$$

(Take $c = \frac{1}{2}V_d f(y^*)$ and r_0 small enough that $f(y) \geq \frac{1}{2}f(y^*)$ on $B(y^*, r_0)$.)

Split the integral at $s_0 := r_0^d \cdot K$. For $s \leq s_0$ (equivalently $s^{2/d}/K^{2/d} \leq r_0^2$),

$$\big(1 - F(s^{2/d}/K^{2/d})\big)^K \;\leq\; \big(1 - c\,s/K\big)^K \;\leq\; e^{-cs},$$

so $g_K(s) \leq \frac{2}{d}\, s^{2/d-1}\, e^{-cs}$, which is integrable on $(0, \infty)$.

For $s > s_0$, we use the second-moment bound. By Markov's inequality applied to $\|y - y^*\|^2$,

$$1 - F(t) \;=\; \Pr\big(\|y - y^*\|^2 > t\big) \;\leq\; \frac{M}{t}, \qquad M := \mathbb{E}\|y - y^*\|^2 < \infty.$$

Therefore, for $s > s_0$ and K large enough that $s^{2/d}/K^{2/d} > 0$,

$$\big(1 - F(s^{2/d}/K^{2/d})\big)^K \;\leq\; \left(\frac{MK^{2/d}}{s^{2/d}}\right)^K.$$

A cleaner bound suffices: since $1 - F(t) \leq 1$, and using the inequality above only when it improves on 1, we have for $s \geq s_1 := (2M)^{d/2}\,K$ that $1 - F(s^{2/d}/K^{2/d}) \leq 1/2$, so $(1-F)^K \leq 2^{-K}$. Combined with the polynomial factor $s^{2/d-1}$, this tail contribution vanishes faster than any power of $1/K$, and in particular

$$\int_{s_0}^{\infty} g_K(s)\, ds \;\longrightarrow\; 0 \qquad \text{as } K \to \infty.$$

Thus, on $(0, s_0]$ we dominate g_K by the integrable function $h(s) = \frac{2}{d}\, s^{2/d-1}\, e^{-cs}$, and the contribution from (s_0, ∞) vanishes in the limit. By the dominated convergence theorem applied to the first piece, together with the vanishing tail,

$$\int_0^{\infty} g_K(s)\, ds \;\longrightarrow\; \int_0^{\infty} g_\infty(s)\, ds.$$

Substitute $u = V_d f(y^*)\, s$ in $\int_0^\infty g_\infty(s)\, ds$:

$$\int_0^{\infty} e^{-V_d f(y^*)\, s} \cdot \frac{2}{d}\, s^{2/d-1}\, ds \;=\; \frac{1}{(V_d f(y^*))^{2/d}} \cdot \frac{2}{d} \int_0^{\infty} u^{2/d-1}\, e^{-u}\, du.$$

Using $\int_0^\infty u^{2/d-1}\, e^{-u}\, du = \Gamma(2/d)$ and $\frac{2}{d}\,\Gamma(2/d) = \Gamma(1 + 2/d)$,

$$\int_0^{\infty} g_\infty(s)\, ds \;=\; \frac{\Gamma(1 + 2/d)}{(V_d f(y^*))^{2/d}}.$$

Combining these results,

$$\mathbb{E}[\tau_K] \;=\; K^{-2/d} \cdot \frac{\Gamma(1 + 2/d)}{(V_d f(y^*))^{2/d}} \cdot \big(1 + o(1)\big) \qquad \text{as } K \to \infty,$$

which proves the theorem. □

A.2 Connection between Mori–Zwanzig memory and localized temporal operators

We clarify how the history-dependent memory term in the Mori–Zwanzig representation motivates the finite temporal operators used by MELD. Suppose the retained observable $x_t \in \mathbb{R}^C$ satisfies

$$x_{t+1} = \mathcal{R}(x_t) + \sum_{\tau \geq 1} \mathcal{K}_\tau(x_{t-\tau}) + \varepsilon_{t+1}^{\perp}. \tag{8}$$

For the retained delay window $\mathbf{X}_t = (x_{t-L+1}, \ldots, x_t) \in \mathcal{F}_L$, let $\pi_\tau \mathbf{X}_t = x_{t-\tau}$ denote the coordinate selector for $0 \le \tau < L$. The memory term can then be decomposed as

$$\sum_{\tau \ge 1} \mathcal{K}_\tau(x_{t-\tau}) = \underbrace{\sum_{\tau=1}^{L-1} \mathcal{K}_\tau(\pi_\tau \mathbf{X}_t)}_{\mathcal{M}_L(\mathbf{X}_t)} + \sum_{\tau \ge L} \mathcal{K}_\tau(x_{t-\tau}), \tag{9}$$

where $\mathcal{M}_L(\mathbf{X}_t) = \sum_{\tau=1}^{L-1} \mathcal{K}_\tau(\pi_\tau \mathbf{X}_t)$ collects the history dependence represented within the retained context. Absorbing the memory tail into $\xi_{t+1}^{(L)} = \varepsilon_{t+1}^{\perp} + \sum_{\tau \ge L} \mathcal{K}_\tau(x_{t-\tau})$, Eq. 8 becomes

$$x_{t+1} = \mathcal{R}(\pi_0 \mathbf{X}_t) + \mathcal{M}_L(\mathbf{X}_t) + \xi_{t+1}^{(L)}. \tag{10}$$

Thus, all Mori–Zwanzig memory terms whose lags lie inside the retained window are functions of the current delay state $\mathbf{X}_t$; only the memory beyond the retained window remains in $\xi_{t+1}^{(L)}$.

To express the corresponding delay-state dynamics, let $\mathcal{S}$ shift the window by one step and let $\mathcal{E}$ insert a new observation into its final coordinate. Then

$$\mathbf{X}_{t+1} = \mathcal{S}\mathbf{X}_t + \mathcal{E}[\mathcal{R}(\pi_0 \mathbf{X}_t) + \mathcal{M}_L(\mathbf{X}_t)] + \mathcal{E}\xi_{t+1}^{(L)}. \tag{11}$$

Since $V_t^\psi = A_\psi^L(\mathbf{X}_t)$ and $\mathbf{X}_t = A_\psi^{L\dagger} V_t^\psi$, the same finite-history dependence can be represented in MeLD's localized coefficient coordinates by applying A_ψ^L to Eq. 11.

This relation motivates, but does not identify, the learnable temporal operators used by MeLD. In particular, for segment m and channel c, the router forms

$$\widehat{\mathcal{H}}_{t,m,c}^{\psi} = \sum_{k=1}^{K} w_{t,m,k,c} \Phi_k^\psi, \qquad \widetilde{z}_{t,m,c} = \widehat{\mathcal{H}}_{t,m,c}^{\psi} [\mathbf{X}_t^{(m)}]_c. \tag{12}$$

Hence $\widehat{\mathcal{H}}_{t,m,c}^{\psi}$ provides a finite, context-dependent parameterization of temporal dependence within the retained window. We do not assume that $\widehat{\mathcal{H}}_{t,m,c}^{\psi}$ recovers the exact Mori–Zwanzig kernels $\{\mathcal{K}_\tau\}$; rather, the reduced-dynamics decomposition motivates learning localized temporal operators that summarize the history dependence available in $\mathbf{X}_t$.

For the special case of linear memory, $\mathcal{K}_\tau(x) = K_\tau x$, the finite-memory term is itself a single linear operator on the delay state:

$$\mathcal{M}_L(\mathbf{X}_t) = \sum_{\tau=1}^{L-1} K_\tau \pi_\tau \mathbf{X}_t = H_L \mathbf{X}_t, \qquad H_L = \sum_{\tau=1}^{L-1} K_\tau \pi_\tau. \tag{13}$$

This special case makes explicit why a finite family of learned temporal operators acting on the retained context provides a natural parameterization of Mori–Zwanzig-inspired history dependence.

A.3 Deterministic forecasting architecture

For each routed localized response, MeLD uses a band-specific Linear–ReLU–Linear prediction head to map the response directly to the full forecast horizon. The resulting K band forecasts are summarized by a pooled cross-band gating module implemented as a Linear–ReLU–Linear–sigmoid network. The resulting gates rescale the band forecasts before they are summed to produce the time-domain deterministic anchor $\widehat{\mathbf{Y}}_t^{\mathrm{anc}}$. The stochastic branch then re-analyzes this anchor with A_ψ^H so that anchor-relative residuals can be modeled in the same localized coefficient coordinates as the observed future.

Algorithm 2 MeLD training with selective residual diffusion

Input: Context window $x \in \mathbb{R}^{L \times C}$, future window $y \in \mathbb{R}^{H \times C}$
Input: Analysis operator A_ψ, anchor predictor f_{anc}, residual denoiser ϵ_θ
Input: Diffusion schedule $\{\bar{\alpha}_n\}_{n=1}^{N_{\text{diff}}}$, routing network g_θ, refinement budget κ (deployed default $\kappa = 1$)
Output: Training loss $\mathcal{L}$
Compute context coefficients (indexed by chunk m, band k, channel c) and future coefficients (indexed by (k, c) only, since $M_H = 1$): $h = A_\psi(x)$, $u_{k,c} = A_\psi(y)_{k,c}$
Predict anchor coefficients: $u^{\text{anc}}_{k,c} = f_{\text{anc}}(h)_{k,c}$
Compute residual: $\Delta_{k,c} = u_{k,c} - u^{\text{anc}}_{k,c}$
for each context chunk m, band k, and channel c **do**
    Compute local statistics $d_{m,k,c}$ from $h_{m,k,c}$
    Compute routing weights:

$$w_{m,k,c} = \frac{\exp(g_\theta(d_{m,k,c})/\tau)}{\sum_{j=1}^{K} \exp(g_\theta(d_{m,j,c})/\tau)}$$

end for
Aggregate context-side routing weights across chunks:

$$w_{k,c} = \frac{1}{M_L} \sum_{m=1}^{M_L} w_{m,k,c}.$$

for each band k and channel c **do**
    Compute anchor-band refinement score ($P_H = H$):

$$a_{k,c} = w_{k,c} \operatorname{Var}_{r \in \{1,\ldots,P_H\}}(u^{\text{anc}}_{k,c})$$

end for
Select unresolved coefficients: $M_{k,c} = \mathbf{1}\{(k, c) \in \text{top-}\kappa(a)\}$ (deployed default $\kappa = 1$)
Keep selected residual coefficients: $\Delta_M = M \odot \Delta$
Sample $n \sim \text{Unif}\{1, \ldots, N_{\text{diff}}\}$ and $\epsilon \sim \mathcal{N}(0, I)$
Corrupt selected residual coefficients:

$$\Delta_M^{(n)} = \sqrt{\bar{\alpha}_n}\Delta_M + \sqrt{1 - \bar{\alpha}_n}(\epsilon \odot M)$$

Predict diffusion noise:

$$\hat{\epsilon} = \epsilon_\theta(h, u^{\text{anc}}, \Delta_M^{(n)}, M, n)$$

Compute diffusion loss:

$$\mathcal{L}_{\text{diff}} = \|M \odot (\hat{\epsilon} - \epsilon)\|_2^2$$

Compute deterministic forecasting loss (decode the anchor alone to raw sequence space via the synthesis left inverse, $\hat{y}^{\text{anc}} = A_\psi^{H\dagger}(u^{\text{anc}})$):

$$\mathcal{L}_{\text{det}} = \|\hat{y}^{\text{anc}} - y\|_2^2$$

Return training loss:

$$\mathcal{L} = \mathcal{L}_{\text{det}} + \gamma\, \mathcal{L}_{\text{diff}}$$

Inference only (not part of the training loss): reconstruct $\hat{\Delta}_M^{(0)}$ from $\hat{\epsilon}$ using the DDPM x_0-prediction formula, form the coefficient forecast $\hat{u} = u^{\text{anc}} + M \odot \hat{\Delta}_M^{(0)}$, and decode $\hat{y} = A_\psi^{H\dagger}(\hat{u})$.

Table A1: **Published-reference comparison of multivariate long-term forecasting on standard benchmark datasets.** MELD results are means over eight independent training seeds; baseline results are published values from the corresponding reference works. Avg. is the per-dataset mean over $H \in \{96, 192, 336, 720\}$. Best (lowest) per-row per-metric result is in bold. The published baseline implementations use `drop_last=True` during evaluation, detailed protocol settings for MELD are summarized in Table A4. MELD error bars are reported in Table A2; protocol-controlled 8-seed baseline reproductions and significance tests are reported in Tables A3 and A5, respectively. The numbers for Xpatch are from Table A3 since there are no publication numbers under these settings.

Dataset	H	MELD		D-PAD		DLinear		TimeMixer		N-HiTS		iTrans.		PatchTST		XPatch	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.351**	**0.371**	0.357	0.376	0.375	0.399	0.388	0.410	0.475	0.498	0.386	0.405	0.370	0.400	0.377	0.398
	192	**0.382**	**0.400**	0.394	0.402	0.405	0.416	0.420	0.432	0.492	0.519	0.441	0.436	0.413	0.429	0.431	0.427
	336	**0.372**	**0.403**	0.374	0.406	0.439	0.443	0.487	0.466	0.550	0.564	0.487	0.458	0.422	0.440	0.461	0.443
	720	**0.419**	**0.441**	**0.419**	0.442	0.472	0.490	0.503	0.481	0.598	0.641	0.503	0.491	0.447	0.468	0.488	0.476
	Avg.	**0.381**	**0.404**	0.386	0.407	0.423	0.437	0.450	0.447	0.529	0.556	0.454	0.448	0.413	0.434	0.439	0.436
ETTh2	96	**0.260**	**0.319**	0.272	0.327	0.289	0.353	0.283	0.343	0.328	0.364	0.297	0.349	0.274	0.337	0.280	0.334
	192	**0.311**	**0.359**	0.331	0.368	0.383	0.418	0.350	0.390	0.372	0.408	0.380	0.400	0.341	0.382	0.345	0.377
	336	**0.308**	**0.360**	0.321	0.370	0.448	0.465	0.389	0.423	0.397	0.421	0.428	0.432	0.329	0.384	0.367	0.398
	720	0.383	**0.417**	0.399	0.426	0.605	0.551	0.443	0.458	0.461	0.497	0.427	0.445	**0.379**	0.422	0.393	0.426
	Avg.	**0.316**	**0.364**	0.331	0.373	0.431	0.447	0.366	0.404	0.390	0.422	0.383	0.406	0.331	0.381	0.346	0.384
ETTm1	96	**0.282**	**0.326**	0.285	0.328	0.299	0.343	0.299	0.352	0.370	0.468	0.334	0.368	0.293	0.346	0.289	0.333
	192	**0.322**	**0.347**	0.323	0.349	0.335	0.365	0.336	0.375	0.436	0.488	0.377	0.391	0.333	0.370	0.331	0.358
	336	**0.349**	**0.370**	0.351	0.372	0.369	0.386	0.379	0.406	0.483	0.510	0.426	0.420	0.369	0.392	0.365	0.381
	720	**0.410**	**0.404**	0.412	0.405	0.425	0.421	0.432	0.436	0.489	0.537	0.491	0.459	0.416	0.420	0.428	0.419
	Avg.	**0.341**	**0.362**	0.343	0.364	0.357	0.379	0.362	0.392	0.444	0.501	0.407	0.410	0.353	0.382	0.353	0.373
ETTm2	96	**0.158**	**0.233**	0.162	0.247	0.167	0.263	0.174	0.258	0.184	0.262	0.180	0.264	0.166	0.256	0.164	0.248
	192	**0.210**	**0.267**	0.218	0.283	0.224	0.303	0.239	0.304	0.260	0.293	0.250	0.309	0.223	0.296	0.221	0.287
	336	**0.261**	**0.317**	0.267	0.321	0.284	0.342	0.281	0.334	0.313	0.359	0.311	0.348	0.274	0.329	0.277	0.324
	720	**0.350**	**0.370**	0.353	0.372	0.397	0.421	0.363	0.388	0.411	0.421	0.412	0.407	0.362	0.385	0.358	0.376
	Avg.	**0.245**	**0.297**	0.250	0.306	0.268	0.332	0.264	0.321	0.292	0.334	0.288	0.332	0.256	0.316	0.255	0.309
Traffic	96	**0.355**	**0.233**	0.359	0.236	0.410	0.282	0.380	0.275	0.407	0.290	0.395	0.268	0.360	0.249	0.481	0.280
	192	**0.376**	**0.244**	0.377	0.245	0.423	0.287	0.401	0.282	0.423	0.302	0.417	0.276	0.379	0.256	0.484	0.275
	336	**0.390**	**0.252**	0.391	0.253	0.436	0.296	0.413	0.290	0.446	0.321	0.433	0.283	0.392	0.264	0.500	0.279
	720	**0.412**	**0.271**	0.413	0.282	0.446	0.305	0.455	0.305	0.528	0.369	0.467	0.302	0.432	0.286	0.534	0.293
	Avg.	**0.383**	**0.250**	0.385	0.254	0.429	0.292	0.412	0.288	0.451	0.320	0.428	0.282	0.391	0.264	0.500	0.282
Electricity	96	**0.120**	**0.217**	0.128	0.220	0.140	0.237	0.130	0.224	0.151	0.254	0.148	0.240	0.129	0.222	0.159	0.244
	192	**0.144**	**0.226**	0.148	0.236	0.153	0.247	0.152	0.246	0.170	0.273	0.149	0.231	0.147	0.240	0.160	0.248
	336	**0.158**	**0.242**	0.163	0.253	0.160	0.259	0.169	0.263	0.200	0.291	0.178	0.253	0.163	0.259	0.182	0.267
	720	**0.191**	**0.267**	0.201	0.286	0.203	0.301	0.204	0.293	0.244	0.356	0.225	0.317	0.197	0.290	0.216	0.298
	Avg.	**0.153**	**0.238**	0.160	0.249	0.164	0.261	0.164	0.256	0.191	0.294	0.175	0.260	0.159	0.253	0.179	0.264
Exchange	96	**0.081**	**0.197**	0.098	0.219	0.082	0.202	0.089	0.211	0.092	0.208	0.086	0.206	0.095	0.217	0.084	0.203
	192	0.169	0.295	0.204	0.321	**0.157**	**0.293**	0.194	0.315	0.208	0.300	0.177	0.299	0.201	0.322	0.181	0.304
	336	0.310	**0.392**	0.429	0.464	**0.305**	0.414	0.351	0.429	0.371	0.509	0.331	0.417	0.372	0.447	0.353	0.427
	720	0.669	**0.594**	0.721	0.607	**0.643**	0.601	1.019	0.744	0.798	0.596	0.847	0.691	0.873	0.699	0.954	0.725
	Avg.	0.307	**0.370**	0.363	0.403	**0.297**	0.378	0.413	0.425	0.367	0.403	0.360	0.403	0.385	0.421	0.393	0.415
Weather	96	0.146	0.186	**0.143**	**0.181**	0.176	0.237	0.152	0.206	0.160	0.197	0.174	0.214	0.149	0.198	0.150	0.187
	192	**0.185**	**0.221**	0.189	0.229	0.220	0.282	0.197	0.255	0.207	0.265	0.221	0.254	0.194	0.241	0.193	0.230
	336	**0.233**	**0.259**	0.239	0.268	0.265	0.319	0.249	0.283	0.273	0.301	0.278	0.296	0.245	0.282	0.245	0.271
	720	**0.302**	**0.311**	0.304	0.313	0.323	0.362	0.322	0.338	0.363	0.352	0.358	0.347	0.314	0.334	0.320	0.324
	Avg.	**0.216**	**0.244**	0.219	0.248	0.246	0.300	0.230	0.270	0.251	0.279	0.258	0.278	0.225	0.264	0.227	0.253
# best Avg. (of 16)		15		0		1		0		0		0		0		0	
# best per-horizon (of 64)		57		3		4		0		0		0		1		0	

Table A2: MELD forecasting performance with error bars. We report mean ± standard error over 8 independent runs.

Dataset	H	n	MSE	MAE
ETTh1	96	8	$0.351_{\pm 0.0007}$	$0.371_{\pm 0.0003}$
	192	8	$0.382_{\pm 0.0007}$	$0.400_{\pm 0.0003}$
	336	8	$0.372_{\pm 0.0017}$	$0.403_{\pm 0.0011}$
	720	8	$0.419_{\pm 0.0011}$	$0.441_{\pm 0.0007}$
ETTh2	96	8	$0.260_{\pm 0.0010}$	$0.319_{\pm 0.0005}$
	192	8	$0.311_{\pm 0.0008}$	$0.359_{\pm 0.0007}$
	336	8	$0.308_{\pm 0.0016}$	$0.360_{\pm 0.0010}$
	720	8	$0.383_{\pm 0.0015}$	$0.417_{\pm 0.0015}$
ETTm1	96	8	$0.282_{\pm 0.0014}$	$0.326_{\pm 0.0009}$
	192	8	$0.322_{\pm 0.0004}$	$0.347_{\pm 0.0003}$
	336	8	$0.349_{\pm 0.0010}$	$0.370_{\pm 0.0004}$
	720	8	$0.410_{\pm 0.0005}$	$0.404_{\pm 0.0004}$
ETTm2	96	8	$0.158_{\pm 0.0004}$	$0.233_{\pm 0.0002}$
	192	8	$0.210_{\pm 0.0007}$	$0.267_{\pm 0.0003}$
	336	8	$0.261_{\pm 0.0006}$	$0.317_{\pm 0.0006}$
	720	8	$0.350_{\pm 0.0003}$	$0.370_{\pm 0.0004}$
Traffic	96	8	$0.355_{\pm 0.0012}$	$0.233_{\pm 0.0008}$
	192	8	$0.376_{\pm 0.0014}$	$0.244_{\pm 0.0009}$
	336	8	$0.390_{\pm 0.0016}$	$0.252_{\pm 0.0010}$
	720	8	$0.412_{\pm 0.0021}$	$0.271_{\pm 0.0013}$
Electricity	96	8	$0.120_{\pm 0.0015}$	$0.217_{\pm 0.0016}$
	192	8	$0.144_{\pm 0.0016}$	$0.226_{\pm 0.0021}$
	336	8	$0.158_{\pm 0.0008}$	$0.242_{\pm 0.0009}$
	720	8	$0.191_{\pm 0.0018}$	$0.267_{\pm 0.0014}$
Exchange	96	8	$0.081_{\pm 0.0009}$	$0.197_{\pm 0.0011}$
	192	8	$0.169_{\pm 0.0026}$	$0.295_{\pm 0.0029}$
	336	8	$0.310_{\pm 0.0033}$	$0.392_{\pm 0.0026}$
	720	8	$0.669_{\pm 0.0150}$	$0.594_{\pm 0.0049}$
Weather	96	8	$0.146_{\pm 0.0007}$	$0.186_{\pm 0.0008}$
	192	8	$0.185_{\pm 0.0007}$	$0.221_{\pm 0.0006}$
	336	8	$0.233_{\pm 0.0004}$	$0.259_{\pm 0.0003}$
	720	8	$0.302_{\pm 0.0032}$	$0.311_{\pm 0.0030}$

Table A3: **Protocol-controlled 8-seed comparison at context length $L = 336$.** Each entry reports MSE/MAE averaged over $H \in \{96, 192, 336, 720\}$. All methods are evaluated using the same eight training seeds, data splits, normalization, evaluation protocol, and data-loader settings, with no drop last during evaluation. No published baseline values are used. Best (lowest) per-row per-metric result is in bold.

Dataset	MELD		D-PAD		DLinear		TimeMixer		XPatch		SCINet		WPMixer	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	**0.388**	**0.408**	0.404	0.422	0.423	0.438	0.440	0.447	0.439	0.436	0.626	0.570	0.411	0.423
ETTh2	**0.317**	**0.366**	0.342	0.380	0.456	0.458	0.365	0.403	0.346	0.384	3.347	1.398	0.343	0.387
ETTm1	**0.346**	**0.366**	0.356	0.378	0.358	0.380	0.359	0.389	0.353	0.373	0.656	0.579	0.352	0.377
ETTm2	**0.247**	**0.298**	0.264	0.314	0.281	0.343	0.262	0.321	0.255	0.309	1.980	0.921	0.258	0.314
Weather	**0.218**	**0.248**	0.226	0.254	0.244	0.297	0.229	0.268	0.227	0.253	0.263	0.294	0.224	0.257
Exchange	**0.320**	**0.374**	0.447	0.437	0.337	0.398	0.410	0.426	0.393	0.415	1.574	0.718	0.380	0.410

Table A4: **MeLD hyperparameters for the main six datasets.** All settings use look-back window 336 and learning rate 1e-4. Decomposition: H = Haar, LG = Learnable Gaussian, B = Binomial.

Hyperparameter	ETTh1				ETTh2				ETTm1				ETTm2				Exchange				Weather			
	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720	96	192	336	720
Batch size	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	512	128	128	128	128
Hidden size	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	336	256	336	336	336
Encoder layers	2	2	2	2	2	2	3	3	3	3	3	3	4	4	4	3	2	2	2	2	2	2	2	2
Number of components	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	7	5	5	5	5	5	5	5	5
Diffusion steps	4	4	4	4	4	4	4	4	3	3	3	3	3	3	3	3	2	2	3	3	3	2	3	5
Overlap	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
Dropout	0.5	0.5	0.5	0.5	0.5	0.5	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.2	0.5	0.5	0.5	0.5	0.25	0.5	0.5	0.5
Training epochs	120	120	120	120	100	85	60	100	100	35	100	48	100	100	100	100	95	200	123	200	100	100	100	100
Patience	10	10	10	10	10	10	15	15	10	10	10	10	10	10	10	25	25	25	25	25	15	10	10	10
Drop last	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Decomposition	H	LG	LG	LG	LG	LG	LG	H	LG	LG	LG	LG	LG	LG	LG	LG	B	LG	H	LG	LG	LG	LG	LG

Table A5: MELD vs. baselines: per-dataset significance. Δ = Baseline − MELD averaged over the four horizons (positive = MELD is better); MELD values are the 8-seed means and standard errors of Table A3, baseline values are computed from their per-seed runs. p is from a two-sided Welch t-test; CI_{95} is the half-width of the 95% interval of Δ. q-values via Benjamini–Hochberg FDR over all 60 tests. Bold: $p < 0.05$.

Dataset	Baseline	ΔMSE	CI_{95} MSE	p MSE	q MSE	ΔMAE	CI_{95} MAE	p MAE	q MAE
ETTh1	D-PAD	**+0.0160**	0.0013	**<.001**	**<.001**	**+0.0140**	0.0009	**<.001**	**<.001**
	DLinear	**+0.0350**	0.0016	**<.001**	**<.001**	**+0.0300**	0.0014	**<.001**	**<.001**
	SCINet	**+0.2380**	0.0223	**<.001**	**<.001**	**+0.1620**	0.0146	**<.001**	**<.001**
	TimeMixer	**+0.0520**	0.0070	**<.001**	**<.001**	**+0.0390**	0.0042	**<.001**	**<.001**
	WPMixer	**+0.0230**	0.0035	**<.001**	**<.001**	**+0.0150**	0.0018	**<.001**	**<.001**
	XPatch	**+0.0510**	0.0078	**<.001**	**<.001**	**+0.0280**	0.0039	**<.001**	**<.001**
ETTh2	D-PAD	**+0.0250**	0.0016	**<.001**	**<.001**	**+0.0140**	0.0011	**<.001**	**<.001**
	DLinear	**+0.1390**	0.0080	**<.001**	**<.001**	**+0.0920**	0.0039	**<.001**	**<.001**
	SCINet	**+3.0300**	0.3996	**<.001**	**<.001**	**+1.0320**	0.0761	**<.001**	**<.001**
	TimeMixer	**+0.0480**	0.0043	**<.001**	**<.001**	**+0.0370**	0.0024	**<.001**	**<.001**
	WPMixer	**+0.0260**	0.0021	**<.001**	**<.001**	**+0.0210**	0.0015	**<.001**	**<.001**
	XPatch	**+0.0290**	0.0016	**<.001**	**<.001**	**+0.0180**	0.0011	**<.001**	**<.001**
ETTm1	D-PAD	**+0.0100**	0.0012	**<.001**	**<.001**	**+0.0120**	0.0007	**<.001**	**<.001**
	DLinear	**+0.0120**	0.0012	**<.001**	**<.001**	**+0.0140**	0.0013	**<.001**	**<.001**
	SCINet	**+0.3100**	0.0392	**<.001**	**<.001**	**+0.2130**	0.0191	**<.001**	**<.001**
	TimeMixer	**+0.0130**	0.0019	**<.001**	**<.001**	**+0.0230**	0.0013	**<.001**	**<.001**
	WPMixer	**+0.0060**	0.0014	**<.001**	**<.001**	**+0.0110**	0.0009	**<.001**	**<.001**
	XPatch	**+0.0070**	0.0011	**<.001**	**<.001**	**+0.0070**	0.0007	**<.001**	**<.001**
ETTm2	D-PAD	**+0.0170**	0.0012	**<.001**	**<.001**	**+0.0160**	0.0007	**<.001**	**<.001**
	DLinear	**+0.0340**	0.0076	**<.001**	**<.001**	**+0.0450**	0.0060	**<.001**	**<.001**
	SCINet	**+1.7330**	0.2439	**<.001**	**<.001**	**+0.6230**	0.0500	**<.001**	**<.001**
	TimeMixer	**+0.0150**	0.0015	**<.001**	**<.001**	**+0.0230**	0.0011	**<.001**	**<.001**
	WPMixer	**+0.0110**	0.0018	**<.001**	**<.001**	**+0.0160**	0.0010	**<.001**	**<.001**
	XPatch	**+0.0080**	0.0011	**<.001**	**<.001**	**+0.0110**	0.0006	**<.001**	**<.001**
Exchange	D-PAD	**+0.1270**	0.0107	**<.001**	**<.001**	**+0.0630**	0.0043	**<.001**	**<.001**
	DLinear	**+0.0170**	0.0139	**.020**	**.020**	**+0.0240**	0.0104	**<.001**	**<.001**
	SCINet	**+1.2540**	0.7491	**.003**	**.003**	**+0.3440**	0.1131	**<.001**	**<.001**
	TimeMixer	**+0.0900**	0.0195	**<.001**	**<.001**	**+0.0520**	0.0070	**<.001**	**<.001**
	WPMixer	**+0.0600**	0.0154	**<.001**	**<.001**	**+0.0360**	0.0061	**<.001**	**<.001**
	XPatch	**+0.0730**	0.0145	**<.001**	**<.001**	**+0.0410**	0.0047	**<.001**	**<.001**

Table A6: Significance of MELD vs. each stochastic baseline (L=96, H=192). Each cell reports the p-value and the FDR-corrected q-value: the p-value from a two-tailed one-sample t-test of MELD's per-seed values (n=8) against the published baseline value, and the q-values reflecting its Benjamini-Hochberg adjusted values.

Dataset	Metric	TimeGrad	CSDI	SSSD	D^3VAE	TimeDiff	TMDM
Exchange	MSE	4.8e-27 / 1.1e-25	8.5e-26 / 9.6e-25	1.3e-23 / 9.3e-23	6.0e-23 / 3.4e-22	5.8e-21 / 1.4e-20	4.4e-17 / 6.3e-17
	MAE	9.9e-24 / 7.5e-23	7.3e-23 / 3.7e-22	1.6e-23 / 1.0e-22	7.7e-22 / 2.5e-21	3.6e-19 / 6.5e-19	2.2e-17 / 3.3e-17
	CRPS	1.2e-19 / 2.2e-19	1.7e-17 / 2.6e-17	8.2e-19 / 1.4e-18	8.1e-17 / 1.1e-16	3.4e-16 / 4.7e-16	2.4e-14 / 2.9e-14
ETTm2	MSE	9.9e-24 / 7.5e-23	1.7e-23 / 1.0e-22	2.0e-22 / 8.9e-22	1.5e-21 / 4.1e-21	5.5e-18 / 8.7e-18	1.2e-12 / 1.5e-12
	MAE	1.2e-21 / 3.7e-21	4.1e-21 / 1.1e-20	4.7e-20 / 9.6e-20	1.3e-18 / 2.2e-18	9.1e-18 / 1.4e-17	3.0e-15 / 4.0e-15
	CRPS	4.0e-22 / 1.7e-21	1.1e-21 / 3.4e-21	2.1e-22 / 8.9e-22	6.6e-21 / 1.6e-20	3.0e-20 / 6.5e-20	1.2e-19 / 2.3e-19
Electricity	MSE	8.7e-28 / 3.6e-26	7.4e-27 / 1.3e-25	5.5e-26 / 7.0e-25	8.3e-24 / 7.5e-23	5.4e-22 / 2.0e-21	1.1e-14 / 1.4e-14†
	MAE	1.2e-27 / 3.6e-26	4.5e-28 / 3.6e-26	1.5e-26 / 2.2e-25	2.8e-25 / 2.8e-24	2.1e-20 / 4.6e-20	5.4e-15 / 7.1e-15†
	CRPS	1.1e-19 / 2.2e-19	4.2e-20 / 8.8e-20	8.2e-21 / 1.9e-20	1.4e-18 / 2.3e-18	3.6e-19 / 6.5e-19	2.3e-17 / 3.3e-17
Traffic	MSE	6.5e-14 / 8.0e-14	9.2e-14 / 1.1e-13	1.5e-12 / 1.8e-12	2.6e-12 / 3.0e-12	1.7e-10 / 1.9e-10	9.1e-8 / 9.4e-8
	MAE	1.4e-15 / 1.9e-15	1.3e-14 / 1.6e-14	2.2e-12 / 2.6e-12	9.9e-11 / 1.1e-10	8.8e-12 / 1.0e-11	1.9e-5 / 1.9e-5
	CRPS	8.5e-9 / 8.9e-9	2.5e-7 / 2.6e-7	8.7e-10 / 9.3e-10	5.4e-7 / 5.5e-7†	3.6e-9 / 3.9e-9†	4.7e-10 / 5.1e-10†
Weather	MSE	7.5e-23 / 3.7e-22	1.1e-22 / 5.4e-22	1.4e-21 / 4.1e-21	3.7e-19 / 6.5e-19	7.8e-18 / 1.2e-17	7.5e-15 / 9.7e-15
	MAE	4.6e-22 / 1.8e-21	5.7e-22 / 2.0e-21	3.6e-21 / 9.6e-21	5.0e-18 / 8.1e-18	5.8e-19 / 1.0e-18	1.3e-10 / 1.5e-10†
	CRPS	7.0e-22 / 2.3e-21	7.1e-21 / 1.7e-20	5.0e-22 / 1.9e-21	3.0e-21 / 8.2e-21	5.9e-20 / 1.2e-19	1.1e-20 / 2.6e-20