# HYPERBOLIC TIME SERIES FORECASTING WITH LEARNABLE DECOMPOSITION AND MANIFOLD DYNAMICS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Multiscale structure is central to time series forecasting: temporal signals contain slowly varying trends, intermediate seasonal patterns, and high-frequency residuals. Traditional forecasters model this multiscale structure in Euclidean space, without representing the ordering it induces among components. We identify that cascaded decomposition orders components by their characteristic frequencies, forming a natural frequency hierarchy. This hierarchy aligns with the exponential volume growth of hyperbolic space, which provides increasing representational capacity with radius and naturally accommodates nested hierarchies. We propose **HyperForecast**, the first long-term time series forecasting (LTSF) framework operating in hyperbolic space. HyperForecast embeds decomposed temporal segments on the Poincaré ball where radial distance captures this nested structure, from low-frequency components near the origin to high-frequency components near the boundary. We model geodesic velocity through two manifold-dynamics variants: (a) autoregressive forecasting via Poincaré residual dynamics, and (b) non-autoregressive forecasting that predicts all future segments in parallel. HyperForecast consistently outperforms existing and state-of-the-art baselines on 9 of 11 long-term forecasting benchmarks, achieving up to 58% lower mean squared error at long horizons. Further analysis reveals that the learned radial embeddings recover the frequency hierarchy and that hyperbolic geometry contributes substantially to forecasting performance. [1]

## 1 INTRODUCTION

Time series forecasting is fundamental across domains, from energy management to financial markets. However, capturing multi-scale temporal structures remains a core challenge, since trends evolve slowly, seasonal patterns recur at multiple frequencies, and residuals capture high-frequency stochastic variation. Most existing works (Wu et al., 2021; Huang et al., 2025; Niu et al., 2026; Lin et al., 2024; Xu et al., 2024; Zhou et al., 2021; 2022; Wang et al., 2024; Zeng et al., 2023; Liu et al., 2024; Nie et al., 2023; Wu et al., 2023) operate in Euclidean space, lacking an explicit geometric inductive bias for the underlying frequency hierarchy. We argue that hyperbolic geometry provides a natural manifold for modeling the nested frequency structure induced by time series decomposition.

Many existing forecasting methods (Wang et al., 2024; Zeng et al., 2023; Bandara et al., 2025; Wu et al., 2021) decompose time series into separate components, but treat them as geometrically equivalent vectors in flat Euclidean space, thereby ignoring the intrinsic frequency-based ordering induced by the decomposition. This geometric agnosticism constitutes a structural limitation, as the cascaded decomposition naturally induces an ordering: trend $\prec$ seasonal$_{\text{coarse}}$ $\prec$ seasonal$_{\text{fine}}$ $\prec$ residual, where each component captures variation not explained by slower-varying components above it. This ordering defines a frequency-based hierarchy, corresponding to a tree-like containment structure, and hyperbolic geometry has been used to represent such tree-like/hierarchical structures at provably lower distortion than Euclidean space of equal or higher dimension (Nickel & Kiela, 2017; Ganea et al., 2018). Unlike language or graph data, where such structure is explicit, this ordering

[1] Code available at https://anonymous.4open.science/r/Hyperbolic_TimeSeries-3876/.



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.

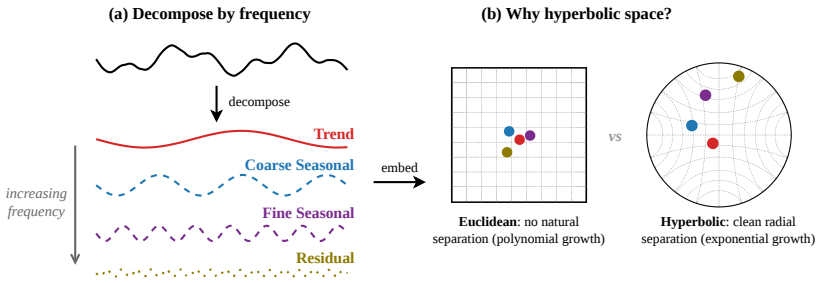



Figure 1: Multiscale temporal decomposition and geometric embedding in HyperForecast. (a) A time series is decomposed into trend, coarse seasonal, fine seasonal, and residual components with increasing frequency. (b) Euclidean space provides no natural separation of these components, whereas the Poincaré ball naturally separates them radially through its exponential volume growth.

emerges implicitly from the temporal dynamics, with slow-varying trends acting as parent components and high-frequency residuals as their descendants. Euclidean geometry, with its polynomial volume growth, provides no natural way to encode this ordering without explicit supervision. Hyperbolic space offers a principled alternative where its constant negative curvature induces exponential volume growth with radius, providing a geometric structure where radial distance naturally encodes frequency-based ordering, as illustrated in Figure 1. Hyperbolic geometry has demonstrated advantages across graph learning, computer vision, and natural language processing (Ganea et al., 2018; Nickel & Kiela, 2017; Chami et al., 2019; van Spengler et al., 2023; Yang et al., 2024; He et al., 2026; Zhang et al., 2026; Patil et al., 2026b; He et al., 2024). Despite this progress, hyperbolic geometry for time series forecasting remains largely unexplored.

HyperForecast fills this gap by combining (i) learnable decomposition with component-specific convolutional kernels, (ii) segment-level embeddings in Poincaré space where radial distance encodes frequency ordering, and (iii) two complementary forecasting paradigms operating via geodesic velocity on the manifold. The autoregressive variant estimates velocity from an exponentially weighted moving window and applies Poincaré residual updates for stable sequential rollout, while the non-autoregressive variant aggregates velocity over the full historical lookback and applies a velocity dampener for numerically stable, parallel prediction of all future segments. To the best of our knowledge, HyperForecast is the first long-term time series forecasting (LTSF) architecture to perform forecasting entirely in hyperbolic space, treating decomposed temporal segments as trajectories on a Riemannian manifold rather than points in a flat Euclidean space, a distinction that enables forecasting as geodesic evolution rather than vector extrapolation. We substantiate this argument empirically by computing the Gromov $\delta$-hyperbolicity of the embeddings across all 11 benchmark datasets, indicating the learned representations are broadly tree-like, and this is corroborated by spectral-centroid and radial-ordering analysis of the raw signal itself. The key contributions are:

- **First fully hyperbolic forecaster:** We propose, to our knowledge, the first long-term time series forecaster operating in hyperbolic space, applying hyperbolic operations such as exponential maps, Möbius addition, and Poincaré fully-connected (FC) layers.
- **Forecasting via geodesic velocity:** Rather than extrapolating future states as displacement vectors in a flat latent space, we cast forecasting as evolution along a geodesic on the curved manifold itself rather than in a Euclidean tangent space.
- **Dual hyperbolic forecasting paradigms:** We introduce autoregressive (AR) and non-autoregressive (NAR) variants, both forecasting via geodesic velocity on the Poincaré manifold but differing in velocity estimation and prediction scheme.

## 2 Related Works

Our work sits at the intersection of decomposition-based time series forecasting and hyperbolic representation learning. We review recent advances in both domains and position HyperForecast as the first to combine learnable decomposition with velocity-driven forecasting in hyperbolic space.

PatchTST (Nie et al., 2023) segments time series into subseries-level patches with channel independence, improving significantly over full-sequence Transformer models. TimeBase (Huang et al., 2025) similarly partitions histories into non-overlapping chunks and learns orthogonal basis components for forecasting, while PhaseFormer (Niu et al., 2026) departs from patch-based tokenization altogether, arguing that phase tokens aligned to a series' periodic structure are globally stationary, unlike patch tokens. These segment- or phase-level approaches preserve multi-scale dynamics while improving scalability; HyperForecast aligns with this trend but differs fundamentally, since its segment representations live in hyperbolic manifolds where radial distance encodes frequency hierarchy, rather than in Euclidean spaces.

Papaioannou et al. (2022) proposed an embed-forecast-lift framework using Diffusion Maps and LLE to reduce high-dimensional time series to low-dimensional manifolds before modeling dynamics; DeepEDM (Majeedi et al., 2025) similarly argues that existing methods ignore this underlying structure and instead proposes dynamical systems modeling via Takens' theorem. Both approaches discover manifold geometry empirically from data, whereas HyperForecast argues that the frequency-based hierarchy induced by cascaded decomposition theoretically warrants hyperbolic geometry.

Hyperbolic geometry has proven advantageous for data with nested or tree-like structure in graph learning (Chami et al., 2019), NLP (He et al., 2026; Yang et al., 2026), and sequential modeling (Zhang et al., 2026; Yang et al., 2024), with recent surveys (Peng et al., 2021; Patil et al., 2026a) taxonomizing the field, though none address long-horizon forecasting. HyperForecast fills this gap as, to our knowledge, the first complete LTSF framework operating entirely in hyperbolic space, combining frequency-based hierarchy encoding via radial distance, Poincaré FC layers for geometric consistency (Shimizu et al., 2021; van Spengler et al., 2023), and both AR and NAR forecasting paradigms.

## 3 PRELIMINARIES

**Hyperbolic manifold (Poincaré ball).** The $d$-dimensional Poincaré ball of constant negative curvature $\mathcal{K} = -c < 0$ is defined as $\mathbb{P}_c^d = \{\, \mathbf{x} \in \mathbb{R}^d : c\|\mathbf{x}\|^2 < 1 \,\}$. This is equipped with the Riemannian metric $g_{\mathbf{x}} = \lambda_{\mathbf{x}}^2\, g^{\mathrm{Euc}}$, where $\lambda_{\mathbf{x}} = \frac{2}{1-c\|\mathbf{x}\|^2}$ is the conformal factor. As $\mathbf{x}$ approaches the boundary $\|\mathbf{x}\| \to c^{-1/2}$, $\lambda_{\mathbf{x}}$ diverges, inducing an exponential expansion of geodesic distances that endows $\mathbb{P}_c^d$ with the capacity to embed hierarchical structure at provably lower distortion than any Euclidean space of equal or higher dimension (Nickel & Kiela, 2017; Ganea et al., 2018).

**Basic hyperbolic operations.** Let $\mathcal{M} = \mathbb{P}_c^d$ denote the hyperbolic manifold. The exponential map $\exp_{\mathbf{x}}^c : T_{\mathbf{x}}\mathcal{M} \to \mathcal{M}$ projects a tangent vector $\mathbf{v} \in T_{\mathbf{x}}\mathcal{M}$ onto the manifold by following the geodesic from $\mathbf{x}$ with initial velocity $\mathbf{v}$. The logarithmic map $\log_{\mathbf{x}}^c : \mathcal{M} \to T_{\mathbf{x}}\mathcal{M}$ is its inverse and recovers the corresponding tangent vector. Together, they induce a local diffeomorphism between $\mathbb{P}_c^d$ and $T_{\mathbf{x}}\mathcal{M}$, enabling Euclidean computations to lift and project back from the manifold. At origin $\mathbf{o} = \mathbf{0}$,

$$\exp_{\mathbf{o}}^c(\mathbf{v}) = \tanh\big(\sqrt{c}\,\|\mathbf{v}\|\big)\,\frac{\mathbf{v}}{\sqrt{c}\,\|\mathbf{v}\| + \varepsilon}, \quad \log_{\mathbf{o}}^c(\mathbf{x}) = \tanh^{-1}\big(\sqrt{c}\,\|\mathbf{x}\|\big)\,\frac{\mathbf{x}}{\sqrt{c}\,\|\mathbf{x}\| + \varepsilon}, \tag{1}$$

where $\varepsilon > 0$ is a small constant added throughout for numerical stability. The hyperbolic distance $d_c(\mathbf{o}, \mathbf{x})$ from the origin $\mathbf{o}$ determines the radial position of $\mathbf{x}$ on $\mathbb{P}_c^d$, inducing a natural frequency stratification, where points near the origin encode low-frequency components (trend, coarse seasonal) and points near the boundary encode high-frequency components (fine seasonal, residual). This geometric alignment between radial depth and temporal scale is the key structural property that distinguishes $\mathbb{P}_c^d$ from Euclidean spaces. Full definitions for Möbius addition $\oplus$ and scalar multiplication $\otimes_c$, the general exponential and logarithmic maps, hyperbolic distance, and parallel transport are provided in Appendix A.

**Poincaré FC layer.** A Poincaré fully-connected (FC) layer with weight $\mathbf{W} \in \mathbb{R}^{d\times d}$ and bias $\mathbf{b} \in \mathbb{R}^d$ extends the Euclidean affine map to $\mathcal{M}$ via the Möbius matrix-vector product plus bias (Shimizu et al., 2021; van Spengler et al., 2023),

$$\mathrm{PoincareFC}_{\mathbf{W},\mathbf{b}}(\mathbf{x}) \;=\; \exp_{\mathbf{o}}^c\big(\mathbf{W}\,\log_{\mathbf{o}}^c(\mathbf{x})\big) \oplus \mathbf{b}. \tag{2}$$

This realises a linear transformation on the manifold via the tangent space at the origin, with standard parameterization for $\mathbf{W}$ and $\mathbf{b}$, allowing optimization.

# 4 METHODOLOGY

## 4.1 ARCHITECTURE OVERVIEW

Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{L \times C}$ with lookback length $L$ and $C$ channels (features), our goal is to forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times C}$ across next $H$ timesteps. HyperForecast performs forecasting in hyperbolic space through four stages: (1) learnable multi-scale decomposition of $\mathbf{X}$ into trend, coarse seasonal, fine seasonal, and residual components, (2) channel-independent segment encoding with projection onto the Poincaré ball $\mathbb{P}_c^d$, (3) forecasting via geodesic velocity either autoregressively (moving-window velocity estimation, sequential rollout) or non-autoregressively (full-lookback velocity, parallel prediction), over horizon $H_{\text{pred}} = \lfloor H/S \rfloor$ segments of fixed length $S$, and (4) Möbius fusion of the components and Euclidean decoding. For clarity, we present the formulation for a single sample; all operations apply independently across batch samples during training. The complete pipeline is illustrated in Figure 2.

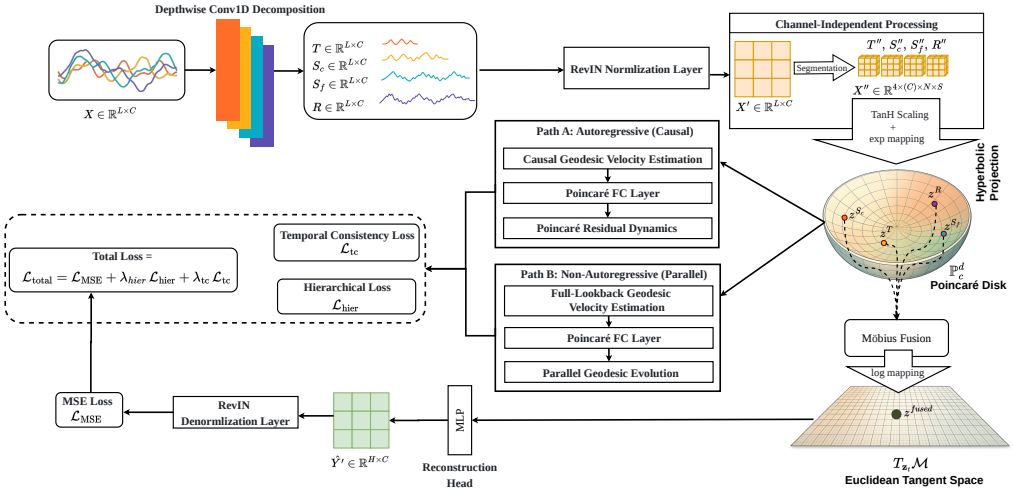



Figure 2: Overview of HyperForecast pipeline. Input $\mathbf{X} \in \mathbb{R}^{L \times C}$ is decomposed into trend $\mathbf{T}$, coarse seasonal $\mathbf{S}_c$, fine seasonal $\mathbf{S}_f$, and residual $\mathbf{R}$ components and normalized by a shared RevIN using statistics from the reconstructed signal $\mathbf{X} = \mathbf{T} + \mathbf{S}_c + \mathbf{S}_f + \mathbf{R}$. Each component is then segment-encoded, projected onto the Poincaré ball $\mathbb{P}_c^d$, forecast via geodesic dynamics on the manifold, and Möbius-fused before Euclidean decoding.

## 4.2 LEARNABLE DECOMPOSITION

Given a multivariate time series $\mathbf{X} \in \mathbb{R}^{L \times C}$, we first decompose the signal using an additive decomposition inspired by MSTL (Bandara et al., 2025),

$$\mathbf{X} = \mathbf{T} + \mathbf{S}_c + \mathbf{S}_f + \mathbf{R}, \tag{3}$$

where $\mathbf{T}$, $\mathbf{S}_c$,$\mathbf{S}_f$, and $\mathbf{R}$ denote the trend, coarse seasonal, fine seasonal, and residual components, respectively. Components are extracted via cascaded depthwise convolutions,

$$\begin{aligned} \mathbf{T} &= \text{Conv1D}_{\text{trend}}(\mathbf{X}), & \mathbf{S}_c &= \text{Conv1D}_{\text{coarse}}(\mathbf{X} - \mathbf{T}), \\ \mathbf{S}_f &= \text{Conv1D}_{\text{fine}}(\mathbf{X} - \mathbf{T} - \mathbf{S}_c), & \mathbf{R} &= \mathbf{X} - \mathbf{T} - \mathbf{S}_c - \mathbf{S}_f. \end{aligned} \tag{4}$$

This cascaded subtraction structure induces the ordering $\mathbf{T} \prec \mathbf{S}_c \prec \mathbf{S}_f \prec \mathbf{R}$, each component captures variation unexplained by the slower-varying components above it. Kernel sizes are initialized from the dataset sampling frequency with $k_c > k_f$, enabling coarse-to-fine modeling of periodic structures. A larger kernel averages over a wider temporal window, hence isolating slower-varying (coarser) periodicity, so that each successive component tends to capture higher-frequency content than the last, facilitating a nested structure that the radial loss subsequently encodes. All operations are differentiable and jointly optimized with the forecasting objective. Reversible instance normalization (RevIN) (Kim et al., 2022) is applied using statistics computed from the reconstructed signal, ensuring statistical consistency and exact denormalization during reconstruction.

### 4.3 SEGMENT ENCODING

Following normalization, each decomposed component $\mathbf{T}, \mathbf{S}_c, \mathbf{S}_f, \mathbf{R} \in \mathbb{R}^{L\times C}$ undergoes channel-independent processing. Channel-independent processing collapses the feature dimension into the batch axis, yielding $C$ univariate time series per component. Each component tensor is then partitioned along the temporal dimension into $N = L/S$ non-overlapping segments of fixed length $S$, i.e., $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$.

We define a segment encoding operator $\Phi_S(\cdot)$ that maps each segment to a $d$-dimensional latent vector. Four separate linear encoders are used, one for each decomposed component, mapping segments to $d$ Euclidean latent vectors:

$$\mathbf{h}_{\mathbf{k}} = \Phi_S^{\mathbf{k}}(\mathbf{k}), \quad \mathbf{k} \in \{\mathbf{T}, \mathbf{S}_c, \mathbf{S}_f, \mathbf{R}\}. \tag{5}$$

No parameters are shared across component encoders. Using separate encoders allows each component to learn representations tailored to its characteristic temporal scale and variability.

**Projection to Hyperbolic Space.** The Euclidean latent vectors are then mapped to the $d$-dimensional Poincaré ball $\mathbb{P}_c^d$ using the exponential map at the origin: $\mathbf{z} = \exp_{\mathbf{o}}(\gamma \cdot \mathbf{h})$, where $\gamma = \tanh(\theta)$ is a *learnable scaling factor*, with $\theta$ initialized to 1.0. The $\tanh$ activation constrains $\gamma \in (-1, 1)$ within the Poincaré ball radius, preventing numerical instability during the exponential map by bounding the tangent vector norm. This is important because the exponential map's closed form on the Poincaré ball involves hyperbolic tangent functions that saturate for large-norms. By adaptively scaling the Euclidean embeddings before projection, the model learns the appropriate "depth" in hyperbolic space for representing each component. This projection embeds segment-level representations into hyperbolic space, which is well-suited for modeling the hierarchical structure induced by multi-scale temporal decomposition. Once historical segments are encoded on $\mathbb{P}_c^d$, forecasting reduces to predicting the next $H_{\text{pred}} = \lfloor H/S \rfloor$ points along a manifold trajectory. We instantiate this in two variants that differ in their manifold step: an Autoregressive rollout (Sec. 4.4) and a Non-Autoregressive parallel evolution (Sec. 4.5).

### 4.4 HYPERBOLIC AUTOREGRESSIVE (AR) FORECASTING

Forecasting is performed independently for each component over horizon $H_{\text{pred}} = \lfloor H/S \rfloor$ corresponding to the number of future segments to be generated. At each prediction step $t$, a moving window of the previous $K$ hyperbolic embeddings is used to compute velocities via logarithmic maps between consecutive states. Unlike sliding windows over fixed input, the moving window dynamically incorporates newly predicted states during autoregressive generation to maintain temporal context for velocity estimation.

**Causal Geodesic Velocity Estimation.** At each step $t \in \{1, \ldots, H_{\text{pred}}\}$, we estimate a local geodesic velocity $\tilde{\mathbf{v}}_t$ using a causal moving window of the previous $K$ hyperbolic embeddings. Each displacement $\log_{\mathbf{z}_{t-i}}(\mathbf{z}_{t-i+1})$ lies in a different tangent space $T_{\mathbf{z}_{t-i}}\mathcal{M}$, so before aggregating we align every displacement into the shared tangent space $T_{\mathbf{z}_t}\mathcal{M}$ via parallel transport $\Gamma_{\mathbf{z}_{t-i}\to\mathbf{z}_t}$ (Eq. 19). Specifically, $\tilde{\mathbf{v}}_t$ is defined as a weighted aggregation of parallel-transported logarithmic-map displacements between consecutive states within the window,

$$\tilde{\mathbf{v}}_t = \sum_{i=0}^{K-2} w_i \, \Gamma_{\mathbf{z}_{t-1-i}\to\mathbf{z}_t}\left(\log_{\mathbf{z}_{t-1-i}}(\mathbf{z}_{t-i})\right), \tag{6}$$

where $\log_{\mathbf{z}_{t-1-i}}(\mathbf{z}_{t-i}) \in T_{\mathbf{z}_{t-1-i}}\mathcal{M}$ denotes the logarithmic map at base point $\mathbf{z}_{t-1-i}$, mapping the point $\mathbf{z}_{t-i}$ on the manifold to a tangent vector representing the geodesic displacement from $\mathbf{z}_{t-1-i}$ to $\mathbf{z}_{t-i}$, and $\Gamma_{\mathbf{z}_{t-1-i}\to\mathbf{z}_t}$ carries that displacement into the common reference frame $T_{\mathbf{z}_t}\mathcal{M}$ before summation. $w_i$ are normalized exponentially decaying weights that prioritize recent transitions. The resulting $\tilde{\mathbf{v}}_t$ provides a smoothed estimate of temporal evolution over a window of size $K$ on the manifold.

**Poincaré Residual Dynamics.** The aggregated velocity is transformed by a learnable linear map $\mathbf{W} \in \mathbb{R}^{d\times d}$ and scaled by a step-size parameter $\beta$ via sigmoid activation $\sigma$, yielding $\bar{\mathbf{v}}_t = \sigma(\beta)\cdot\mathbf{W}\tilde{\mathbf{v}}_t$, where $\sigma(\beta) \in (0, 1)$ adaptively controls the extrapolation magnitude. The next hyperbolic state is predicted using Poincaré residual dynamics,

$$\mathbf{z}_{t+1} = \big(\alpha \otimes \mathbf{z}_t\big) \oplus \big((1-\alpha) \otimes \exp_{\mathbf{z}_t}(\bar{\mathbf{v}}_t)\big), \tag{7}$$

where $\alpha \in (0, 1)$ is a learnable residual blending weight, $\otimes$ denotes Möbius scalar multiplication, $\oplus$ denotes Möbius addition, and $\exp_{\mathbf{z}_t}(\cdot)$ is the exponential map from $\mathbf{z}_t$. This formulation enables smooth extrapolation with this Poincaré residual connection to stabilize long-horizon forecasting along geodesic trajectories.

### 4.5 Hyperbolic Non-Autoregressive (NAR) Forecasting

The autoregressive branch in Section 4.4 predicts one segment per recurrent step. At every horizon $t$, it re-aggregates a window velocity for each component. It then pushes it through the Poincaré FC layer, and applies the Poincaré residual update. Although accurate, this approach is susceptible to error accumulation, since each step's prediction depends on the previous step, degrading performance at longer horizons. We therefore introduce a Non-Autoregressive (NAR) variant that produces all $H_{\text{pred}}$ future segments in parallel while reusing the same velocity mechanism.

**Full-Lookback Geodesic Velocity Estimation.** Let $(\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N)$ denote the $N = L/S$ historical embeddings on $\mathbb{P}_c^d$ in temporal order, where $\mathbf{z}_N$ is the most-recent encoded historical segment. As in Eq. 6, each displacement is computed in the tangent space at its own base point, so we parallel-transport every term into the shared frame $T_{\mathbf{z}_N}\mathcal{M}$ before aggregating over the *entire* lookback,

$$\tilde{\mathbf{v}} \;=\; \sum_{i=0}^{N-2} w_i \, \Gamma_{\mathbf{z}_{N-1-i} \to \mathbf{z}_N} \Big( \log_{\mathbf{z}_{N-1-i}}(\mathbf{z}_{N-i}) \Big), \tag{8}$$

which is structurally identical to Eq. 6 but uses the full $N$-segment historical lookback instead of AR's $K$-segment moving window. We then pass $\tilde{\mathbf{v}}$ through the Poincaré FC layer (Eq. 2) for each component. The two branches use the same velocity machinery (Eqs. 6-2) but on different supports. AR re-aggregates over a moving window of $K$ segments at every rollout step ($K$), whereas NAR aggregates once over the entire $N = L/S$ historical encodings.

**Parallel Geodesic Evolution.** In place of the autoregressive Poincaré residual update of Eq. 7, we extrapolate along the geodesic emitted from $\mathbf{z}_N$ with horizon-dependent step length, producing every future segment in a single batched call to $\exp_{\mathbf{z}_N}$,

$$\mathbf{z}_{1:H_{\text{pred}}} \;=\; \exp_{\mathbf{z}_N}\big(t \cdot \tilde{\mathbf{v}}\big), \tag{9}$$

where $t \in \{1, 2, \ldots, H_{\text{pred}}\}$ and the $t$-th output is $\mathbf{z}_t = \exp_{\mathbf{z}_N}(t \cdot \tilde{\mathbf{v}})$.

### 4.6 Hierarchical Constraints and Temporal Regularization

To align learned representations with the frequency hierarchy, we employ two auxiliary losses during training.

**Hierarchical Radial Loss.** Component-wise radial constraints enforce frequency hierarchy by encouraging fine-scale and residual components to occupy larger radii in hyperbolic space while coarse-scale and trend components remain closer to the origin,

$$\mathcal{L}_{\text{radial}} = \sum_{k=1}^{3} \text{ReLU}\big(r_k - r_{k+1} + m\big), \tag{10}$$

where $r_k = d(\mathbf{o}, \mathbf{z}^{(k)})$ denotes hyperbolic distance from the origin, with components ordered $k \in \{\mathbf{T}, \mathbf{S}_c, \mathbf{S}_f, \mathbf{R}\}$, and $m = 0.1$ is a fixed margin enforcing a minimum radial gap between consecutive components. A geodesic coherence term additionally enforces that adjacent components occupy neighboring manifold regions, reflecting the nested structure of the cascaded decomposition,

$$\mathcal{L}_{\text{hier}} = \mathcal{L}_{\text{radial}} + 0.5 \cdot \big[ d(\mathbf{z}^{\mathbf{T}}, \mathbf{z}^{\mathbf{S}_c}) + d(\mathbf{z}^{\mathbf{S}_c}, \mathbf{z}^{\mathbf{S}_f}) + d(\mathbf{z}^{\mathbf{S}_f}, \mathbf{z}^{\mathbf{R}}) \big]. \tag{11}$$

**Temporal Consistency Loss.** To promote smooth latent trajectories, we minimize abrupt velocity changes across prediction steps, promoting smooth and stable latent dynamics. Velocities are aligned via parallel transport $\Gamma$ before comparison,

$$\mathcal{L}_{\text{tc}} = \frac{1}{N-2} \sum_{t=1}^{N-2} \Big\| \log_{\mathbf{z}_{t+1}}(\mathbf{z}_{t+2}) - \Gamma_{\mathbf{z}_t \to \mathbf{z}_{t+1}} \big( \log_{\mathbf{z}_t}(\mathbf{z}_{t+1}) \big) \Big\|^2, \tag{12}$$

where $\Gamma_{\mathbf{z}_t \to \mathbf{z}_{t+1}}$ denotes parallel transport from tangent space $T_{\mathbf{z}_t}\mathcal{M}$ to $T_{\mathbf{z}_{t+1}}\mathcal{M}$. Trajectories with zero acceleration correspond to geodesics in hyperbolic space.

**Hyperbolic Fusion and Reconstruction** Predicted latent representations from all components are fused using Möbius addition, $\mathbf{z}^{\text{fused}} = \mathbf{z}^{\mathbf{T}} \oplus \mathbf{z}^{\mathbf{S}_c} \oplus \mathbf{z}^{\mathbf{S}_f} \oplus \mathbf{z}^{\mathbf{R}}$. This integrates information from multiple temporal scales into a single hyperbolic latent representation. This fused representation is mapped to Euclidean space via logarithmic mapping and decoded through a single-layer MLP with ReLU activation (Nair & Hinton, 2010) to produce the final forecast $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times C}$. RevIN denormalization restores the original scale. The total training loss combines forecasting MSE with the auxiliary losses,

$$\mathcal{L} = \mathcal{L}_{\text{MSE}} + \lambda_{\text{hier}} \mathcal{L}_{\text{hier}} + \lambda_{\text{tc}} \mathcal{L}_{\text{tc}}, \tag{13}$$

where $\lambda_{\text{hier}}$ and $\lambda_{\text{tc}}$ are tuned via grid search per dataset.

## 5 Experiments and Results

**Datasets.** We evaluate on 11 publicly available datasets spanning diverse domains, following standard LTSF evaluation practice with a fixed lookback length and multiple prediction horizons: **Weather** (Lai et al., 2018), **Solar-Energy** (Lai et al., 2018), **Exchange** (Lai et al., 2018), **AQShunyi, AQWan** (Zhang et al., 2017), **CzeLan, ZafNoo** (Poyatos et al., 2020), **PM2.5, Temp** (Wang et al., 2020), **Wind** (Li et al., 2022), **METR-LA** (Li et al., 2018). Dataset statistics are summarized alongside the geometric diagnostics in Table 2. We use a sequence length $L = 720$, and report results averaged over four standard horizons $H \in \{96, 192, 336, 720\}$; implementation details, including hyperparameters, optimization settings, and computational requirements, are in Appendix K.

**Results.** We report MSE and MAE between $\mathbf{Y}$ and $\hat{\mathbf{Y}} \in \mathbb{R}^{H \times C}$, averaged over all $H \times C$ (lower is better). The averaged forecasting results are in Table 1. HyperForecast NAR ranks first by MSE on 7 of 11 datasets and first-or-second on 10 of 11 (the exception is METR-LA, where PatchTST and iTransformer lead).NAR outperforms AR on most datasets, with the gap most pronounced at long horizons on Temp (NAR: 0.113, AR: 0.203 at $H = 720$), validating that parallel geodesic evolution avoids error accumulation in sequential rollout. Mean $\pm$ std MSE over 5 seeds on seven representative datasets (Appendix F) confirms low variance across runs; a full efficiency comparison, including parameter/MACs breakdown and an AR-vs-NAR cost analysis, is in Appendix C, and full per-horizon results are in Appendix J. We next examine the geometric structure of the learned embeddings underlying these gains.

Table 1: Average long-term forecasting results (MSE/MAE) across horizons $H \in \{96, 192, 336, 720\}$ on 11 datasets (input length $L = 720$ for all models). Best values are **red-bold**; second-best **blue-bold**.

| Dataset | HF NAR | | HF AR | | PhaseFormer | | TimeBase | | SparseTSF | | TimeMixer | | FITS | | iTransformer | | DLinear | | PatchTST | | TimesNet | | FEDformer | | Autoformer | | Informer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Weather | **0.218** | 0.273 | 0.221 | 0.270 | 0.223 | **0.261** | **0.219** | **0.263** | 0.244 | 0.285 | 0.238 | 0.281 | 0.244 | 0.286 | 0.233 | 0.273 | 0.242 | 0.299 | 0.223 | 0.264 | 0.254 | 0.293 | 0.355 | 0.391 | 0.446 | 0.457 | 0.567 | 0.513 |
| AQShunyi | **0.657** | **0.503** | **0.665** | **0.504** | 0.722 | 0.523 | 0.674 | 0.506 | 0.760 | 0.546 | 0.736 | 0.536 | 0.763 | 0.548 | 0.722 | 0.520 | 0.757 | 0.572 | 0.690 | **0.503** | 0.788 | 0.559 | 1.130 | 0.757 | 1.393 | 0.878 | 1.688 | 0.966 |
| AQWan | **0.751** | **0.486** | **0.762** | **0.495** | 0.837 | 0.517 | 0.779 | 0.499 | 0.826 | 0.526 | 0.822 | 0.521 | 0.813 | 0.519 | 0.843 | 0.538 | 0.883 | 0.560 | 0.810 | 0.513 | 0.857 | 0.543 | 0.931 | 0.565 | 1.001 | 0.654 | 1.123 | 0.744 |
| CzeLan | **0.221** | 0.286 | **0.224** | 0.286 | 0.225 | **0.273** | 0.225 | **0.262** | 0.240 | 0.292 | 0.234 | 0.282 | 0.250 | 0.304 | 0.245 | 0.296 | 0.241 | 0.290 | 0.229 | 0.275 | 0.258 | 0.312 | 0.266 | 0.314 | 0.289 | 0.368 | 0.347 | 0.426 |
| ZafNoo | **0.485** | **0.443** | **0.489** | 0.448 | 0.533 | 0.456 | 0.495 | **0.445** | 0.531 | 0.490 | 0.505 | 0.464 | 0.520 | 0.480 | 0.540 | 0.500 | 0.540 | 0.495 | 0.510 | 0.469 | 0.545 | 0.502 | 0.601 | 0.539 | 0.667 | 0.635 | 0.726 | 0.697 |
| Exchange | **0.289** | 0.408 | **0.274** | **0.379** | 0.504 | 0.452 | 0.309 | **0.392** | 0.316 | 0.410 | 0.330 | 0.426 | 0.302 | 0.400 | 0.301 | 0.401 | 0.311 | 0.410 | 0.308 | 0.400 | 0.327 | 0.425 | 0.380 | 0.472 | 0.447 | 0.589 | 0.432 | 0.581 |
| METR-LA | 1.266 | 0.741 | 1.296 | 0.754 | 1.348 | 0.762 | 1.254 | 0.754 | 1.302 | 0.764 | 1.270 | 0.741 | 1.296 | 0.760 | **1.253** | **0.734** | 1.300 | 0.759 | **1.210** | **0.706** | 1.390 | 0.812 | 1.550 | 0.875 | 1.639 | 0.974 | 1.893 | 1.146 |
| PM2.5 | **0.410** | 0.452 | **0.406** | **0.437** | 0.430 | **0.434** | 0.428 | 0.437 | 0.467 | 0.477 | 0.465 | 0.474 | 0.460 | 0.469 | 0.480 | 0.491 | 0.449 | 0.457 | 0.459 | 0.466 | 0.466 | 0.475 | 0.559 | 0.547 | 0.621 | 0.650 | 0.636 | 0.663 |
| Solar | **0.205** | 0.274 | 0.211 | 0.268 | **0.193** | **0.240** | 0.215 | **0.254** | 0.216 | 0.264 | 0.244 | 0.296 | 0.229 | 0.279 | 0.233 | 0.285 | 0.227 | 0.276 | 0.227 | 0.275 | 0.243 | 0.295 | 0.252 | 0.291 | 0.306 | 0.381 | 0.327 | 0.411 |
| Temp | **0.119** | **0.267** | **0.145** | **0.293** | 0.217 | 0.348 | 0.187 | 0.338 | 0.295 | 0.412 | 0.302 | 0.423 | 0.315 | 0.442 | 0.302 | 0.423 | 0.292 | 0.418 | 0.291 | 0.406 | 0.303 | 0.427 | 0.343 | 0.456 | 0.404 | 0.587 | 0.471 | 0.682 |
| Wind | **0.930** | **0.700** | 0.949 | 0.713 | 1.018 | 0.743 | **0.940** | 0.709 | 1.014 | 0.736 | 1.044 | 0.758 | 1.023 | 0.745 | 1.025 | 0.748 | 1.018 | 0.739 | 0.968 | **0.703** | 1.101 | 0.799 | 1.230 | 0.841 | 1.237 | 0.947 | 1.381 | 1.033 |

**Geometric Evidence for the Frequency Hierarchy.** To understand why HyperForecast's forecasting gains concentrate on certain datasets, we examine three complementary diagnostics of the learned frequency hierarchy: the Welch-PSD (Welch, 1967) spectral centroid $\Psi$ of each decomposed component, the paired norm ratio $\rho$ (whether embedding radii preserve the trend$\prec$coarse$\prec$fine$\prec$residual ordering), and Gromov $\delta$-hyperbolicity $\delta_D$ (overall tree-likeness of the embedding geometry). Formal definitions of $\Psi$, $\rho$, and $\delta_D$ are given in Appendix B. We first demonstrate whether the raw decomposed signal itself already exhibits the target frequency ordering, independent of any learned embedding (Figure 3 and Appendix D); we then examine whether the trained hyperbolic embeddings inherit this ordering radially ($\rho$) and whether the resulting geometry is broadly tree-like ($\delta_D$), both reported per dataset alongside basic dataset statistics in Table 2.

As per our illustration of decomposition on Solar at $H = 96$ in Figure 3, average centroids $\Psi_T = 0.0092 < \Psi_{S_c} = 0.0103 < \Psi_{S_f} = 0.0113 < \Psi_R = 0.0123$, with the full ordering holding for 69% of individual features. We also illustrate this trend for additional datasets spanning channel counts from 11 to 207 and different domains in Appendix D, all exhibiting the full ordering on the dataset average. The cascaded decomposition induces a genuine frequency separation on the raw signal,

prior to and independent of any hyperbolic embedding, establishing the geometric precondition that HyperForecast's radial design is built to leverage.

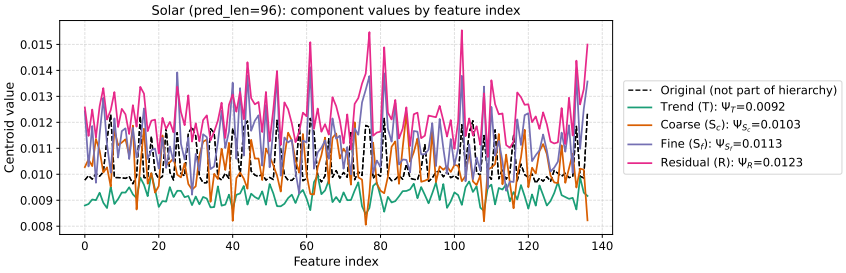



Figure 3: Per-feature Welch-PSD spectral centroid of each decomposed component on Solar (137 channels, $H = 96$). Black dashed: original (undecomposed) signal. Legend values are the per-component average across all 137 features. The ordering $T < S_c < S_f < R$ holds on the dataset average and for the large majority of individual features.

Table 2: Dataset statistics and frequency-hierarchy geometric diagnostics for the 11 LTSF benchmarks. **Var**: number of variables. **Length**: total series length. **Freq**: sampling frequency. $\rho$: paired norm ratio between consecutive components' embedding radii ($\rho<1$ indicates good hierarchical separation). $\delta_D$: Gromov $\delta$-hyperbolicity of the trained embeddings, both averaged across $H \in \{96, 192, 336, 720\}$.

| Dataset | Var | Length | Freq | $\rho$ | $\delta_D$ | Dataset | Var | Length | Freq | $\rho$ | $\delta_D$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Weather | 21 | 52,696 | 10m | 0.737 | 0.0139 | METR-LA | 207 | 34,272 | 5m | 1.023 | 0.0143 |
| AQShunyi | 11 | 35,064 | 1h | 1.127 | 0.0095 | PM2.5 | 184 | 11,688 | 3h | 0.997 | 0.0194 |
| AQWan | 11 | 35,064 | 1h | 0.848 | 0.0200 | Solar | 137 | 52,560 | 10m | 0.897 | 0.0150 |
| CzeLan | 11 | 19,934 | 30m | 0.749 | 0.0025 | Temp | 184 | 11,688 | 3h | 1.020 | 0.0227 |
| ZafNoo | 11 | 19,225 | 30m | 0.515 | 0.0021 | Wind | 7 | 48,673 | 15m | 0.935 | 0.0153 |
| Exchange | 8 | 7,588 | 1d | 0.886 | 0.0097 | | | | | | |

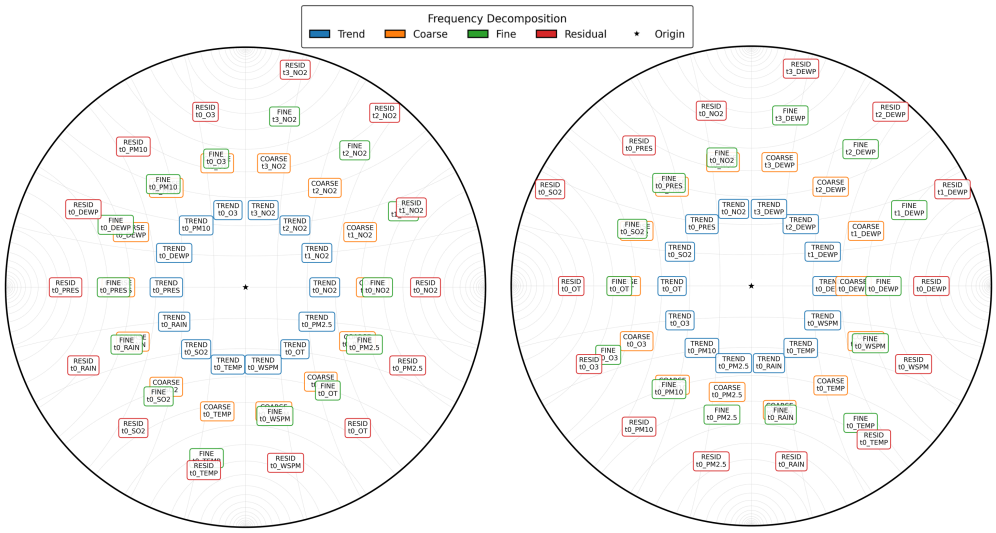
The **paired norm ratio** $\rho$ (Table 2) falls below 1 in 8 of 11 datasets, i.e., the trained embeddings preserve the specific component ordering (trend nearest the origin, residual farthest) in most cases. **Gromov $\delta$-hyperbolicity** $\delta_D$ is close to 0 for all 11 datasets (range 0.002–0.023), consistent with an overall tree-like embedding geometry favoring hyperbolic space; since $\delta_D$ pools all four components into a single point cloud, it alone does not test whether the ordering is preserved, which is what $\rho$ captures. Elevated $\rho$ does not uniformly translate into weaker forecasting, AQShunyi ($\rho = 1.127$) and Temp ($\rho = 1.020$) still rank first by MSE (Table 1). METR-LA is the only dataset where HyperForecast does not rank first or second, plausibly due to its high channel count (207) and length in addition to its radial geometry ($\rho = 1.023$).

To validate the consistency of learned frequency-based hierarchies across diverse datasets, we visualize embeddings for AQShunyi and AQWan on the Poincaré disk (Figure 4). Both datasets exhibits moderate radial separation, with trend components clustering near the origin and residual components distributed toward the boundary. The overall structure maintains a tree-like radial organization consistent with frequency hierarchy, although the strict ordering may not hold on average for all samples (AQShunyi's $\rho = 1.127$). To confirm this ordering is a geometric property of embedding rather than an effect of the radial loss, Appendix I reports the visualization with $\lambda_{\text{hier}} = 0$.

**Ablations.** We present two targeted ablations validating core design choices: the benefit of hyperbolic over Euclidean geometry, and sensitivity to lookback window length. Additional ablations (impact of learnable decomposition, channel-independent vs. channel-dependent processing, window size, loss weights) are provided in Appendix E.

*Poincaré vs. Euclidean Geometry:* We replace all manifold operations with standard Euclidean arithmetic, retaining identical decomposition and velocity dynamics. As shown in Figure 5a, Poincaré embeddings achieve consistently lower MSE than their Euclidean counterpart across all horizons on all four datasets tested. Temp shows the largest and most horizon-dependent gap, where the Poincaré variant's MSE actually *improves* with horizon (0.123→0.113) while the Euclidean variant steadily degrades (0.147→0.183), a 62% higher MSE at $H = 720$.

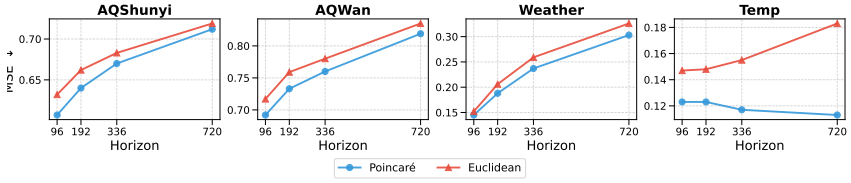
*Lookback Window Sensitivity:* The effect of input context length $L \in \{96, 192, 336, 720\}$ on NAR and AR variants across Temp and ZafNoo can be seen in Figure 5b. On Temp, longer lookback



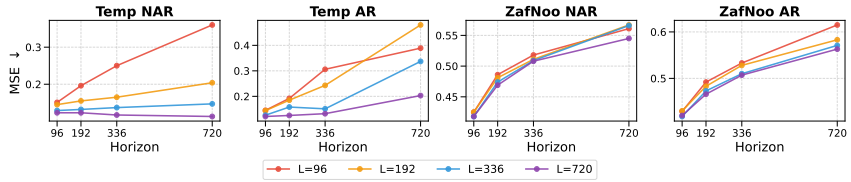

(a) AQShunyi Embeddings on Poincaré Disk

(b) AQWan Embeddings on Poincaré Disk

Figure 4: Each point is a (Scale, TimeSegment, Feature) tuple. The **radial hierarchy** shows our model learns frequency-based decomposition (Trend < Coarse < Fine < Residual). Labels follow `t{segment}_{feature}` (e.g., t0_Feature_1 = segment 0, feature 1).




(a) Poincaré vs. Euclidean geometry on AQShunyi, AQWan, Weather, and Temp (NAR).




(b) Lookback sensitivity ($L \in \{96, 192, 336, 720\}$) for Temp and ZafNoo, NAR and AR.

Figure 5: Ablation studies: (a) Hyperbolic vs. Euclidean geometry and (b) lookback window sensitivity, shown for both forecasting variants.

provides a decisive advantage, with $L = 720$ remaining consistently better while $L = 96$ degrades sharply at longer horizons, confirming that richer context improves geodesic velocity estimation.

## 6 Conclusion

This paper introduced HyperForecast, the first long-term time series forecasting framework operating in hyperbolic space. By treating decomposed segments as manifold trajectories that encode the frequency hierarchy the decomposition induces, HyperForecast achieves state-of-the-art performance on 9 of 11 benchmarks, with up to 58% MSE reduction (Temp, $H$=720) over the strongest Euclidean baseline, and yields interpretable hyperbolic radii that confirm the learned decomposition. Limitations include higher computational cost than ultra-lightweight models and the additive decomposition assumption, which may not hold for all time series. Future work could explore learnable curvature adaptation, hyperbolic probabilistic forecasting, and theoretical analysis of hyperbolic capacity for frequency hierarchies.

### AI USE STATEMENT

We did not use generative AI tools for any of the required-disclosure tasks under the ICLR 2027 AI Policy, such as generating synthetic data, developing theoretical models or conceptual frameworks, formulating or proving mathematical claims, proposing or refining hypotheses, designing or giving feedback on research methodology or experiments, implementing methods, or interpreting results. All modeling, experiments, and analysis in this work are our own.

We used generative AI tools only to aid or polish writing (improving clarity, grammar, and readability of the text). All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the final content of this work, including all text, claims, and artifacts.

### ETHICS STATEMENT

This work uses only publicly available, standard long-term forecasting benchmark datasets, none of which contain personally identifiable information, human-subjects data, or sensitive attributes. The work does not involve human subjects, and we are not aware of any conflicts of interest related to this research. We do not foresee direct negative societal impacts.

### REPRODUCIBILITY STATEMENT

Per-dataset statistics are given in Table 2. Full training, optimization, and hyperparameter-search details (hardware, architecture dimensions, loss weights, learning rates, and data splits) are given in Appendix K. Complete derivations of the hyperbolic operations underlying our method are given in Appendix A, and formal definitions of the $\Psi$, $\rho$, and $\delta_D$ diagnostics are given in Appendix B. Full per-horizon results for all 11 datasets are reported in Appendix J, multi-seed stability results (mean $\pm$ std over 5 seeds) in Appendix F, and ablations isolating each architectural and loss component in Appendix E. Anonymized source code is available as described in Appendix L.

## REFERENCES

Aaron B Adcock, Blair D Sullivan, and Michael W Mahoney. Tree-like structure in large social and information networks. In *2013 IEEE 13th international conference on data mining*, pp. 1–10. IEEE, 2013.

Kasun Bandara, Rob J Hyndman, and Christoph Bergmeir. MSTL: A seasonal-trend decomposition algorithm for time series with multiple seasonal patterns. *International Journal of Operational Research*, 52(1):79–98, 2025.

Ines Chami, Zhitao Ying, Christopher Ré, and Jure Leskovec. Hyperbolic graph convolutional neural networks. *Advances in neural information processing systems*, 32, 2019.

Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018.

Mikhael Gromov. Hyperbolic groups. In *Essays in group theory*, pp. 75–263. Springer, 1987.

Neil He, Rishabh Anand, Hiren Madhu, Ali Maatouk, Smita Krishnaswamy, Leandros Tassiulas, Menglin Yang, and Rex Ying. Helm: Hyperbolic large language models via mixture-of-curvature experts. *Advances in Neural Information Processing Systems*, 38:142604–142635, 2026.

Yuan He, Moy Yuan, Jiaoyan Chen, and Ian Horrocks. Language models as hierarchy encoders. *Advances in Neural Information Processing Systems*, 37:14690–14711, 2024.

Qihe Huang, Zhengyang Zhou, Kuo Yang, Zhongchao Yi, Xu Wang, and Yang Wang. Timebase: The power of minimalism in efficient long-term time series forecasting. In *ICML*, 2025.

Edmond Jonckheere, Poonsuk Lohsoonthorn, and Francis Bonahon. Scaled gromov hyperbolic graphs. *Journal of Graph Theory*, 57(2):157–180, 2008.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cGDAkQo1C0p.

Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch. *arXiv preprint arXiv:2005.02819*, 2020.

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.

Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=SJiHXGWAZ.

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.

Shengsheng Lin, Weiwei Lin, Wentai Wu, Haojun Chen, and Junjie Yang. SparseTSF: Modeling long-term time series forecasting with 1k parameters. In *Forty-first International Conference on Machine Learning*, 2024.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Abrar Majeedi, Viswanatha Reddy Gajjala, Satya Sai Srinath Namburi GNVV, Nada Magdi Elkordi, and Yin Li. LETS forecast: Learning embedology for time series forecasting. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=LLk1qYQatJ.

Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017.

Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=Jbdc0vTOcol.

Yiming Niu, Jinliang Deng, and Yongxin Tong. PhaseFormer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, volume 2026, pp. 8262–8285, 2026.

Panagiotis G Papaioannou, Ronen Talmon, Ioannis G Kevrekidis, and Constantinos Siettos. Time-series forecasting using manifold learning, radial basis function interpolation, and geometric harmonics. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 32(8), 2022.

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

Sarang Patil, Zeyong Zhang, Yiran Huang, Tengfei Ma, and Mengjia Xu. Hyperbolic large language models. *SIAM Review (to appear)*, 2026a.

Sarang Rajendra Patil, Ashish Parmanand Pandey, Ioannis Koutis, and Mengjia Xu. Hierarchical mamba meets hyperbolic geometry: A new paradigm for structured language embeddings. *Transactions on Machine Learning Research*, 2026b. ISSN 2835-8856. URL https://openreview.net/forum?id=a3g13FKzct.

Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on pattern analysis and machine intelligence*, 44(12):10023–10044, 2021.

Rafael Poyatos, Víctor Granda, Víctor Flo, Mark A Adams, Balázs Adorján, David Aguadé, Marcos PM Aidar, Scott Allen, M Susana Alvarado-Barrientos, Kristina J Anderson-Teixeira, et al. Global transpiration data from sap flow measurements: the SAPFLUXNET database. *Earth System Science Data Discussions*, 2020:1–57, 2020.

Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. In *International Conference on Learning Representations*, 2021.

Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pp. 369–386. SPIE, 2019.

Abraham Albert Ungar. *A Gyrovector Space Approach to Hyperbolic Geometry*. Synthesis Lectures on Mathematics & Statistics. Springer Cham, 2009. ISBN 978-3-031-01268-6. doi: 10.1007/978-3-031-02396-5. URL https://doi.org/10.1007/978-3-031-02396-5.

Max van Spengler, Erwin Berkhout, and Pascal Mettes. Poincare ResNet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5419–5428, October 2023.

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*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Shuo Wang, Yanran Li, Jiang Zhang, Qingye Meng, Lingwei Meng, and Fei Gao. PM2. 5-GNN: A domain knowledge enhanced graph neural network for PM2. 5 forecasting. In *Proceedings of the 28th international conference on advances in geographic information systems*, pp. 163–166, 2020.

Peter Welch. The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

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 *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=bWcnvZ3qMb.

Menglin Yang, Harshit Verma, Delvin Ce Zhang, Jiahong Liu, Irwin King, and Rex Ying. Hypformer: Exploring efficient transformer fully in hyperbolic space. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 3770–3781, 2024.

Menglin Yang, Aosong Feng, Bo Xiong, Jiahong Liu, Irwin King, Rex Ying, et al. Hyperbolic fine-tuning for large language models. *Advances in Neural Information Processing Systems*, 38: 29613–29648, 2026.

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.

Qianru Zhang, Honggang Wen, Wei Yuan, Crystal Chen, Menglin Yang, Siu-Ming Yiu, and Hongzhi Yin. HMamba: Hyperbolic mamba for sequential recommendation. *ACM Transactions on Information Systems*, 44(5):1–27, 2026.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in beijing. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 473(2205):20170457, 2017.

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 HYPERBOLIC GEOMETRY: ADDITIONAL DEFINITIONS

This appendix complements Section 3 with the remaining hyperbolic operations referenced in the methodology. All formulations are stated for the Poincaré ball $\mathbb{P}_c^d$ of curvature $-c < 0$, equipped with the conformal factor $\lambda_{\mathbf{x}} = \frac{2}{1-c\|\mathbf{x}\|^2}$ from Sec. 3.

**Möbius addition and scalar multiplication.** To perform neural network operations on $\mathbb{P}_c^d$, we use Möbius addition and scalar multiplication as foundational gyrovector arithmetic (Ungar, 2009; Ganea et al., 2018):

$$\mathbf{x} \oplus \mathbf{y} = \frac{\big(1 + 2c\langle \mathbf{x}, \mathbf{y}\rangle + c\|\mathbf{y}\|^2\big)\,\mathbf{x} + \big(1 - c\|\mathbf{x}\|^2\big)\,\mathbf{y}}{1 + 2c\langle \mathbf{x}, \mathbf{y}\rangle + c^2\|\mathbf{x}\|^2\|\mathbf{y}\|^2}, \tag{14}$$

$$\alpha \otimes_c \mathbf{x} = \tanh\big(\alpha\, \tanh^{-1}\big(\sqrt{c}\,\|\mathbf{x}\|\big)\big)\, \frac{\mathbf{x}}{\sqrt{c}\,\|\mathbf{x}\| + \varepsilon}. \tag{15}$$

**Exponential and logarithmic maps at an arbitrary base point.** For $\mathbf{x} \in \mathcal{M}$ and $\mathbf{v} \in T_{\mathbf{x}}\mathcal{M}$, the exponential map $\exp_{\mathbf{x}}^c : T_{\mathbf{x}}\mathcal{M} \to \mathcal{M}$ and its inverse $\log_{\mathbf{x}}^c : \mathcal{M} \to T_{\mathbf{x}}\mathcal{M}$ admit the closed-form expressions (Ganea et al., 2018),

$$\exp_{\mathbf{x}}^c(\mathbf{v}) = \mathbf{x} \oplus \left( \tanh\!\big(\tfrac{\sqrt{c}\,\lambda_{\mathbf{x}}\,\|\mathbf{v}\|}{2}\big)\, \frac{\mathbf{v}}{\sqrt{c}\,\|\mathbf{v}\| + \varepsilon} \right), \tag{16}$$

$$\log_{\mathbf{x}}^c(\mathbf{y}) = \frac{2}{\sqrt{c}\,\lambda_{\mathbf{x}}}\, \tanh^{-1}\!\big(\sqrt{c}\,\|-\mathbf{x} \oplus \mathbf{y}\|\big)\, \frac{-\mathbf{x} \oplus \mathbf{y}}{\|-\mathbf{x} \oplus \mathbf{y}\| + \varepsilon}, \tag{17}$$

where $\oplus$ is Möbius addition (Eq. 14). Setting $\mathbf{x} = \mathbf{o}$ recovers the origin-centered formulas already given in Eq. 1. Exponential mapping is invoked at every AR rollout step (Eq. 7) and once per horizon in the NAR branch (Eq. 9), whereas logarithmic mapping is used in the velocity aggregators.

**Hyperbolic distance.** The geodesic distance between two arbitrary points $\mathbf{x}, \mathbf{y} \in \mathcal{M}$ is,

$$d_c(\mathbf{x}, \mathbf{y}) \;=\; \frac{2}{\sqrt{c}}\, \tanh^{-1}\!\big(\sqrt{c}\,\|-\mathbf{x} \oplus \mathbf{y}\|\big). \tag{18}$$

At $\mathbf{x} = \mathbf{o}$, $d_c(\mathbf{o}, \mathbf{y}) = \frac{2}{\sqrt{c}} \tanh^{-1}(\sqrt{c}\|\mathbf{y}\|)$ recovers the radial position.

**Parallel transport.** The parallel transport $\Gamma_{\mathbf{x}\to\mathbf{y}} : T_{\mathbf{x}}\mathcal{M} \to T_{\mathbf{y}}\mathcal{M}$ moves a tangent vector along the unique geodesic from $\mathbf{x}$ to $\mathbf{y}$ while preserving the Riemannian inner product. On $\mathcal{M}$ it has the closed form (Ganea et al., 2018),

$$\Gamma_{\mathbf{x}\to\mathbf{y}}(\mathbf{v}) \;=\; \frac{\lambda_{\mathbf{x}}}{\lambda_{\mathbf{y}}}\, \mathrm{gyr}\big[\mathbf{y}, -\mathbf{x}\big]\, \mathbf{v}, \tag{19}$$

where $\mathrm{gyr}[\mathbf{y}, -\mathbf{x}]$ is the gyration that rotates the tangent vector along the geodesic from $\mathbf{x}$ to $\mathbf{y}$ (Ungar, 2009). Eq. 19 is invoked in the temporal-consistency loss $\mathcal{L}_{\mathrm{tc}}$ (Eq. 12) to align velocities sampled at different base points along the predicted trajectory before computing the acceleration penalty.

# B FREQUENCY-HIERARCHY METRICS: FORMAL DEFINITIONS

This section formally defines the three per-dataset diagnostics discussed in Sec. 5 that are only described informally in the main text. The spectral centroid $\Psi$ is reported among illustrations in Figure 3 and Appendix D. The paired norm ratio $\rho$, and Gromov $\delta$-hyperbolicity $\delta_D$ (both reported per dataset in Table 2).

**Spectral centroid.** For a decomposed component $k \in \{\mathbf{T}, \mathbf{S}_c, \mathbf{S}_f, \mathbf{R}\}$, let $P_k(f)$ denote its Welch power spectral density (Welch, 1967), estimated on the raw time-domain component (prior to any hyperbolic embedding), averaged across all samples, and computed per channel. The spectral centroid is the power-weighted mean frequency,

$$\Psi_k = \frac{\sum_f f \cdot P_k(f)}{\sum_f P_k(f)}, \tag{20}$$

where higher $C_k$ indicates higher-frequency content concentrated in component $k$.

**Paired norm ratio.** Let $r_k(s) = d(\mathbf{o}, \mathbf{z}_s^{(k)})$ denote the hyperbolic distance (Eq. 18) of the $s$-th segment's trained embedding for component $k$ from the origin $\mathbf{o}$. For each of the three consecutive component pairs $(k, k') \in \{(\mathbf{T}, \mathbf{S}_c), (\mathbf{S}_c, \mathbf{S}_f), (\mathbf{S}_f, \mathbf{R})\}$, we compute the per-segment ratio $r_k(s)/r_{k'}(s)$ and take its median over segments (median rather than mean, since segments embedded very close to the origin produce unstable individual ratios). The paired norm ratio $\rho$ is the mean of these three step-wise medians,

$$\rho = \frac{1}{3} \sum_{(k,k')} \text{median}_s \left[ \frac{r_k(s)}{r_{k'}(s)} \right], \tag{21}$$

$\rho < 1$ indicates each successive, higher-frequency component sits farther from the origin than its predecessor, consistent with hierarchical radial separation. $\rho \geq 1$ indicates the ordering is not preserved on average.

**Gromov $\delta$-hyperbolicity.** We measure each dataset's Gromov $\delta$-hyperbolicity (Gromov, 1987) on the trained historical encoder embeddings, pooling all four decomposed components into a single set of points (unlike $\rho$ above, this diagnostic does not use component identity, so it tests overall tree-likeness rather than the specific component ordering). For a metric space $(X, d)$ – here $d$ is hyperbolic distance (Eq. 18) between pooled segment embeddings – and any quadruple $x, y, z, w \in X$, let $S_{(1)} \geq S_{(2)} \geq S_{(3)}$ be the sorted pair-sums of $\{d(x,y)+d(z,w),\ d(x,z)+d(y,w),\ d(x,w)+d(y,z)\}$. We define

$$\delta(X) = \tfrac{1}{2} \sup_{x,y,z,w \in X} \big(S_{(1)} - S_{(2)}\big), \qquad \delta_D = \delta(X)/D, \tag{22}$$

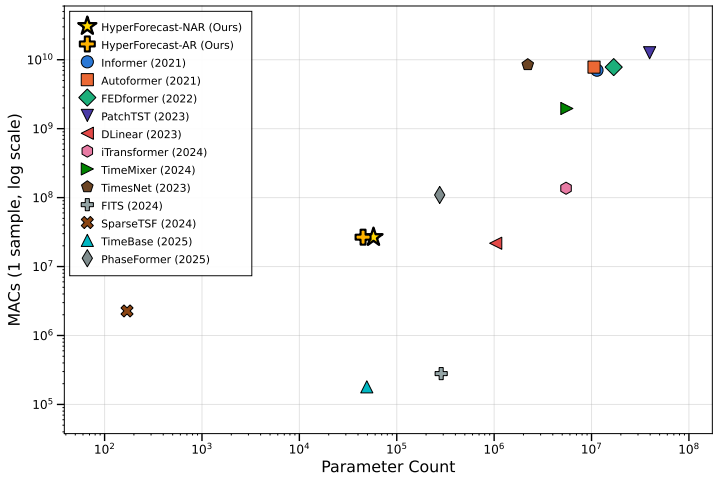
where $D$ is the diameter of $X$. We normalize $\delta$ by $D$ following the relative-hyperbolicity convention (Adcock et al., 2013; Jonckheere et al., 2008): $\delta_D \to 0$ indicates a tree-like structure that strongly benefits from hyperbolic geometry, while $\delta_D \to 0.5$ indicates a flat/Euclidean structure.

## C Model Efficiency Analysis

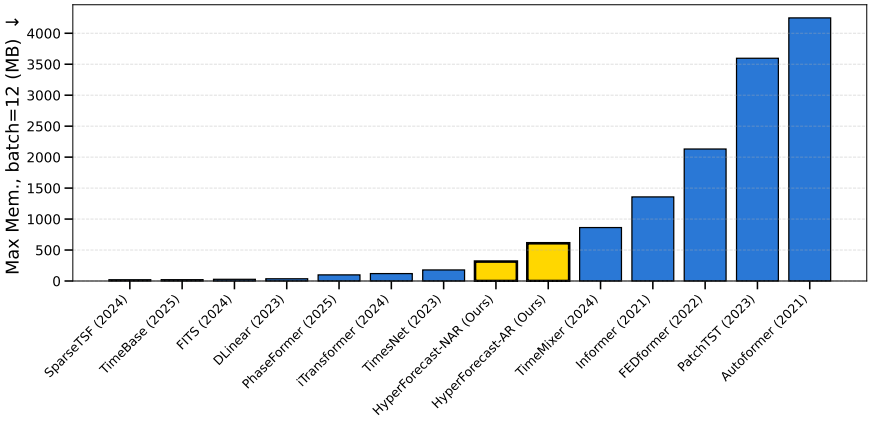
We benchmark HyperForecast against all baselines under a single, unified protocol. All fourteen models here are run on Weather with seq_len = pred_len = 720. The parameter count, MACs, and FLOPs are measured for a single forward pass over one sample. The peak memory (Max Mem.) is measured at batch size 12, and epoch time and CPU inference time are wall-clock measurements under this same protocol.

As shown in Figure 6a, HyperForecast occupies a distinct efficiency regime among the compared methods. Against the Transformer-based baselines (Informer, Autoformer, FEDformer, PatchTST), HyperForecast uses $264\times$ to $470\times$ fewer MACs (26.69M versus 7.05G to 12.62G) and up to $197\times$ fewer parameters (57.69K versus 11.38M for Informer), while remaining only slightly above DLinear (21.79M MACs) among the ultra-lightweight linear baselines. HyperForecast is therefore substantially lighter than any competitive Transformer-based method, at a small and bounded MAC cost relative to the linear family in exchange for its hierarchical hyperbolic representation. As Figure 6b shows, this advantage does not extend uniformly to measured peak memory. The segment-level hyperbolic operations used by HyperForecast, including logarithmic and exponential maps, parallel transport, and Möbius arithmetic, retain more intermediate activation state per sample than a single linear layer, placing HyperForecast in the middle of the memory ranking rather than at the low end.

In Figure 7, we isolate the cost of autoregressive decoding. As described in Sections 4.4 and 4.5, the two variants share the same per-step machinery, namely geodesic velocity estimation through parallel-transported logarithmic maps (Eq. 6) followed by a Poincaré residual update. They differ in how this velocity is used and how often the machinery is invoked. NAR aggregates the velocity once over the full lookback and produces every future segment with a single batched call to $\exp_{\mathbf{z}_N}$ (Eq. 9). AR instead re-estimates the velocity and re-applies the Poincaré residual update (Eq. 7) once per predicted segment, and each step's moving window depends on the previous step's own output. This dependency makes the $H_{\text{pred}}$ steps inherently sequential and prevents them from being batched across time, the same structural bottleneck that makes autoregressive decoding slower than non-autoregressive decoding in sequence generation more broadly, compounded here by each recurrent step performing a nontrivial manifold operation rather than a single matrix multiply. This yields a $12.4\times$ epoch-time gap (2688.16s versus 217.33s) and an $8.5\times$ CPU inference-time gap




(a) Parameter count vs. MACs (log–log).




(b) Peak memory at batch size 12, sorted ascending.

Figure 6: Efficiency comparison across all fourteen models on Weather (seq_len = pred_len = 720). **(a)** Parameter count vs. MACs for a single sample. **(b)** Peak memory at batch size 12, sorted ascending.

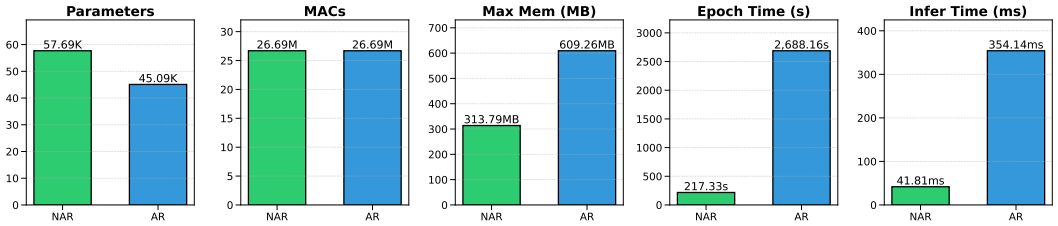



Figure 7: Efficiency of the two HyperForecast variants on Weather, seq_len/pred_len = 720/720. NAR is shown in green and AR in blue.

(354.14ms versus 41.81ms) between AR and NAR, along with a smaller 1.9× increase in peak memory (609.26MB versus 313.79MB) from retaining per-step intermediate state across the rollout. The two variants report identical MACs and FLOPs (26.69M and 53.38M) because our profiler traces a single forward call through the shared step function without multiplying by the number of sequential invocations. The MACs and FLOPs columns should therefore be read as per-step compute cost, while epoch time and inference time correctly reflect the $H_{\text{pred}}$-fold repetition that AR incurs at runtime. We report both variants for completeness and recommend NAR as the efficiency-preferred

configuration, whereas AR remains useful where the recurrent error-correction rollout discussed in Section 4.5 is preferred despite its cost.

## D ADDITIONAL SPECTRAL-CENTROID VISUALIZATIONS

As established in Figure 3 in Sec. 5, the raw decomposed signal exhibits the target frequency ordering $T < S_c < S_f < R$ on Solar at $H = 96$. Here we extend this evidence to four further representative benchmarks that include Weather, AQWan, METR-LA, and PM2.5, spanning channel counts from 11 to 207 and four distinct sensing domains (weather, air quality, traffic speed, and particulate matter).

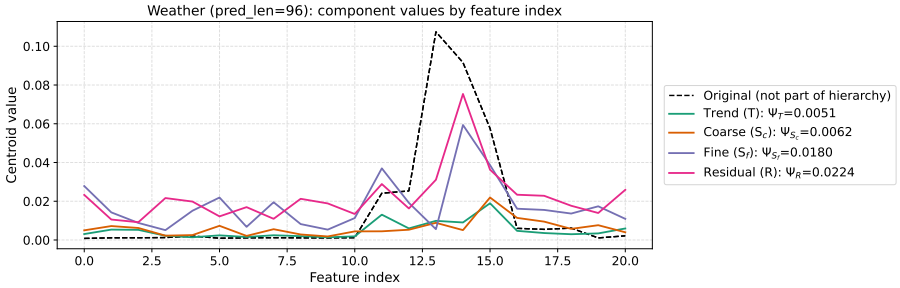



Figure 8: Weather (21 channels, $H = 96$). The full ordering holds on average centroids: $\Psi_T = 0.0051 < \Psi_{S_c} = 0.0062 < \Psi_{S_f} = 0.0180 < \Psi_R = 0.0224$.

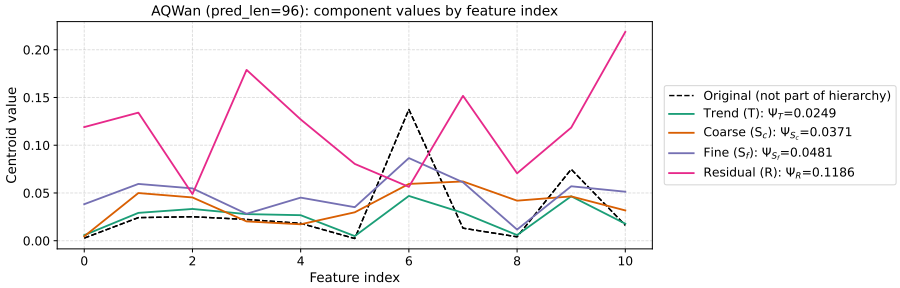



Figure 9: AQWan (11 channels, $H = 96$). The full ordering holds on average centroids: $\Psi_T = 0.0249 < \Psi_{S_c} = 0.0371 < \Psi_{S_f} = 0.0481 < \Psi_R = 0.1186$.

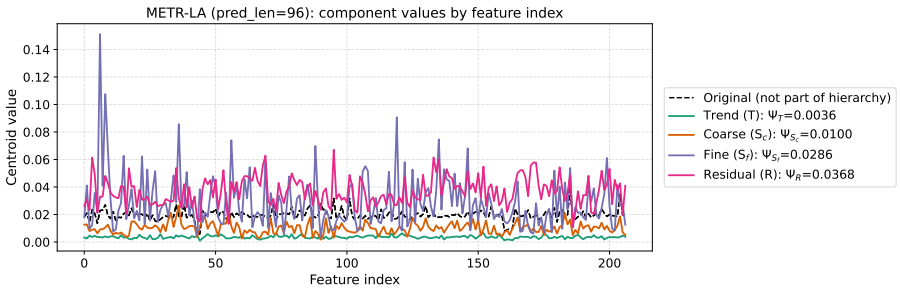



Figure 10: METR-LA (207 channels, $H = 96$). The full ordering holds on average centroids: $\Psi_T = 0.0036 < \Psi_{S_c} = 0.0100 < \Psi_{S_f} = 0.0286 < \Psi_R = 0.0368$.

Across these benchmarks, spanning different channel range and domains, the cascaded decomposition separates raw-signal frequency content in the same direction, $T < S_c < S_f < R$. This consistency, obtained without any hyperbolic embedding or hierarchy loss in the loop, supports treating the frequency hierarchy as a property of the decomposition itself rather than an artifact of training on any one domain. Moreover, Appendix I provides complementary evidence that the corresponding radial ordering in the trained embedding persists even when the hierarchy loss is removed.

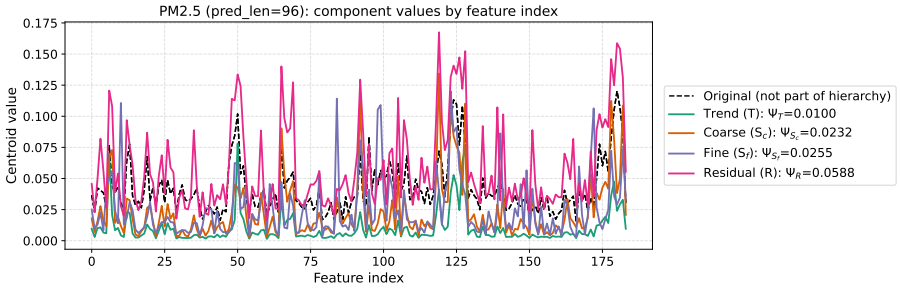



Figure 11: PM2.5 (184 channels, $H = 96$). The full ordering holds on average centroids: $\Psi_T = 0.0100 < \Psi_{S_c} = 0.0232 < \Psi_{S_f} = 0.0255 < \Psi_R = 0.0588$.

# E Ablation Studies

## E.1 Impact of Learnable Decomposition

Our learnable decomposition module employs cascaded depthwise convolutions with component-specific kernel sizes to extract trend, coarse seasonal, fine seasonal, and residual components from raw input sequences. To isolate the contribution of Learnable Decomposition, we compare our approach against a variant of our model that does not implement Learnable Decomposition; it merely processes raw time series directly through segment encoding and hyperbolic dynamics.

Table 3: Impact of learnable decomposition on forecasting accuracy, NAR and AR. Models with decomposition (trend, coarse/fine seasonal, residual) vs. without decomposition across prediction horizons $H \in \{96, 192, 336, 720\}$.

| Dataset | Variant | 96 | | 192 | | 336 | | 720 | |
|---|---|---|---|---|---|---|---|---|---|
| | | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| AQShunyi | NAR w/o Decomp. | 0.638 | 0.490 | 0.679 | 0.507 | 0.713 | 0.527 | 0.776 | 0.554 |
| | NAR w/ Decomp. | **0.607** | **0.475** | **0.640** | **0.498** | **0.670** | **0.510** | **0.712** | **0.528** |
| | AR w/o Decomp. | 0.631 | 0.483 | 0.690 | 0.512 | 0.713 | 0.524 | 0.773 | 0.551 |
| | AR w/ Decomp. | **0.606** | **0.473** | **0.642** | **0.493** | **0.669** | **0.508** | **0.742** | **0.542** |
| AQWan | NAR w/o Decomp. | 0.717 | 0.471 | 0.779 | 0.508 | 0.802 | 0.514 | 0.876 | 0.547 |
| | NAR w/ Decomp. | **0.692** | **0.458** | **0.733** | **0.479** | **0.760** | **0.492** | **0.819** | **0.514** |
| | AR w/o Decomp. | 0.713 | 0.470 | 0.776 | 0.501 | 0.805 | 0.514 | 0.875 | 0.541 |
| | AR w/ Decomp. | **0.696** | **0.464** | **0.742** | **0.486** | **0.766** | **0.499** | **0.846** | **0.532** |
| Weather | NAR w/o Decomp. | 0.188 | 0.243 | 0.218 | 0.271 | 0.268 | 0.309 | 0.333 | 0.355 |
| | NAR w/ Decomp. | **0.145** | **0.206** | **0.188** | **0.252** | **0.237** | **0.292** | **0.303** | **0.344** |
| | AR w/o Decomp. | 0.167 | 0.220 | 0.213 | 0.263 | 0.274 | 0.318 | 0.339 | 0.361 |
| | AR w/ Decomp. | **0.143** | **0.202** | **0.190** | **0.248** | **0.238** | **0.289** | **0.315** | **0.341** |

Results in Table 3 demonstrate that decomposition effectiveness is dataset- and horizon-dependent, and the pattern holds for both forecasting variants. On Weather, with its strong short-range multi-frequency structure (daily/weekly cycles across 21 channels), the NAR gain is largest at short horizons (22.9% MSE reduction at $H = 96$) and tapers to 9.0% at $H = 720$. AR shows the same shape (14.4% at $H = 96$ tapering to 7.1% at $H = 720$). Air quality datasets (AQShunyi, AQWan) show steady 3.5–4.9% NAR gains at $H = 96$, growing to 6.5–8.3% at $H = 720$; AR gains are similar in magnitude (2.4–4.0% at $H = 96$, 3.3–4.0% at $H = 720$), suggesting decomposition's value there comes more from stabilizing long-horizon rollout than from separating short-term seasonal content, for both variants.

## E.2 Channel-Independent vs. Channel-Dependent Processing

Our original methodology processes each channel independently, folding the feature dimension into the batch axis so that a shared encoder sees $C$ univariate sequences. We compare against a

channel-dependent variant that instead keeps all $C$ channels in a single joint tensor through the encoder, on Solar, AQShunyi, and AQWan (Poincaré), for both the NAR and AR variants.

Table 4: Channel-independent vs. channel-dependent processing (Poincaré, NAR; MSE).

| Dataset | Variant | 96 | 192 | 336 | 720 |
|---|---|---|---|---|---|
| Solar | Channel-Dependent | 0.207 | 0.234 | 0.239 | 0.223 |
| | Channel-Independent | **0.177** | **0.204** | **0.221** | **0.217** |
| AQShunyi | Channel-Dependent | 0.667 | 0.678 | 0.693 | **0.689** |
| | Channel-Independent | **0.607** | **0.646** | **0.678** | 0.717 |
| AQWan | Channel-Dependent | 0.774 | 0.754 | **0.770** | 0.829 |
| | Channel-Independent | **0.727** | **0.741** | 0.773 | **0.822** |

Channel-independent processing wins on 10 of 12 dataset/horizon cells, and wins all four horizons on Solar. The two exceptions, AQShunyi at $H = 720$ and AQWan at $H = 336$, differ in margin (4.1% and 0.4% relative, respectively) at different horizons in each dataset, so we don't read a systematic horizon-dependent crossover into them. Channel-independent processing is the consistently better choice across the three datasets tested. We use it as the default given its advantage at the horizons where HyperForecast shows its largest gains (Table 1).

Table 5: Channel-independent vs. channel-dependent processing (Poincaré, AR; MSE).

| Dataset | Variant | 96 | 192 | 336 | 720 |
|---|---|---|---|---|---|
| Solar | Channel-Dependent | 0.206 | 0.738 | 0.754 | 0.757 |
| | Channel-Independent | **0.170** | **0.206** | **0.227** | **0.243** |
| AQShunyi | Channel-Dependent | 0.658 | 0.684 | 0.692 | **0.713** |
| | Channel-Independent | **0.615** | **0.658** | **0.690** | 0.756 |
| AQWan | Channel-Dependent | 0.757 | 0.765 | 0.787 | **0.839** |
| | Channel-Independent | **0.704** | **0.754** | **0.781** | 0.867 |

The AR variant shows the same pattern as NAR, channel-independent processing wins 10 of 12 dataset/horizon cells, with both exceptions at $H = 720$ (AQShunyi and AQWan), where channel-dependent's larger joint parameterization appears to help stabilize the longest, error-accumulating AR rollout. The channel-dependent AR variant is also markedly less stable on Solar (137 channels) beyond $H = 96$, where its MSE roughly triples rather than growing gracefully with horizon, consistent with a joint 137-channel tensor being harder for autoregressive rollout to fit than the channel-independent formulation. Taken together, channel-independent processing is the more robust default across both forecasting paradigms.

### E.3 AR Window Size Sensitivity

The AR branch aggregates a moving window of $K$ historical segments into each rollout-step velocity estimate (Eq. 6). We sweep $K \in \{2, 4, 5, 6\}$ on AQShunyi, Temp, and ZafNoo.

Table 6: AR window size $K$ ablation (MSE). Best per dataset/horizon in **bold**.

| Dataset | $K$ | 96 | 192 | 336 | 720 |
|---|---|---|---|---|---|
| AQShunyi | 2 | 0.615 | **0.658** | 0.690 | 0.756 |
| | 4 | 0.617 | 0.662 | **0.689** | 0.758 |
| | 5 | **0.613** | 0.661 | 0.691 | 0.758 |
| | 6 | 0.616 | 0.662 | **0.689** | **0.737** |
| Temp | 2 | 0.126 | 0.125 | 0.136 | 0.637 |
| | 4 | 0.124 | 0.126 | 0.138 | 0.293 |
| | 5 | 0.124 | **0.122** | 0.150 | 0.407 |
| | 6 | **0.122** | 0.166 | **0.130** | **0.203** |
| ZafNoo | 2 | 0.422 | 0.481 | 0.526 | 0.579 |
| | 4 | **0.419** | 0.479 | 0.514 | 0.580 |
| | 5 | **0.419** | 0.472 | 0.519 | **0.566** |
| | 6 | 0.420 | **0.472** | **0.508** | 0.571 |

At $H \leq 336$, sensitivity to $K$ is modest on AQShunyi and ZafNoo (under 3.54% MSE spread across $K \in \{2, 4, 5, 6\}$), but Temp already shows substantial sensitivity at these horizons (up to 36% spread at $H = 192$). At $H = 720$, $K$ interacts strongly with dataset-specific dynamics: on Temp, $K = 2$ is markedly unstable (MSE 0.637, over $3\times$ worse than $K = 6$'s 0.203), while AQShunyi and ZafNoo show only mild sensitivity at the same horizon (under 2.85% spread). This indicates Temp's velocity field benefits from a wider historical window specifically at long rollout horizons, where a narrow 2-segment window is more exposed to compounding local noise across many rollout steps. $K \in \{4, 6\}$ is a robust default, and $K = 2$ should be avoided for long-horizon AR forecasting on datasets with Temp-like dynamics.

### E.4 Impact of Auxiliary Losses

To isolate the necessity of the two auxiliary losses (as opposed to their weight sensitivity, below), we retrain HyperForecast NAR on CzeLan, AQWan, and Exchange with the temporal-consistency loss removed, the hierarchy loss removed, and both removed, holding all other hyperparameters fixed.

Table 7: Effect of removing the temporal-consistency loss $\mathcal{L}_{\text{tc}}$ and hierarchy loss $\mathcal{L}_{\text{hier}}$ on HyperForecast (NAR), per horizon. Best per dataset/horizon in **bold**.

| Dataset | Variant | 96 | 192 | 336 | 720 |
|---|---|---|---|---|---|
| CzeLan | Main | **0.171** | 0.205 | **0.237** | **0.270** |
| | w/o TC | 0.183 | **0.203** | 0.239 | 0.331 |
| | w/o Hier | 0.190 | 0.223 | 0.240 | 0.279 |
| | w/o both | 0.200 | 0.211 | 0.238 | 0.285 |
| AQWan | Main | **0.692** | **0.733** | **0.760** | 0.819 |
| | w/o TC | 0.699 | 0.735 | 0.764 | **0.815** |
| | w/o Hier | 0.701 | 0.746 | 0.773 | 0.825 |
| | w/o both | 0.703 | 0.739 | 0.774 | 0.823 |
| Exchange | Main | **0.079** | 0.165 | **0.305** | **0.607** |
| | w/o TC | 0.080 | **0.163** | 0.745 | 0.752 |
| | w/o Hier | 0.120 | 0.210 | 0.558 | 1.910 |
| | w/o both | 0.124 | 0.209 | 0.528 | 1.767 |

Removing either loss degrades performance on all three datasets, and the effect is far larger on Exchange (main 0.289 $\rightarrow$ 0.700 MSE without $\mathcal{L}_{\text{hier}}$, +142%) than on CzeLan or AQWan (both under 6% degradation). Exchange is also the shortest and lowest-channel-count series of the three (Table 2, 8 channels, 7,588 timesteps versus CzeLan's 11 and 19,934 and AQWan's 11 and 35,064), giving the architecture less raw signal to passively exploit for frequency separation, so its embedding geometry likely depends more heavily on the explicit hierarchy constraint, consistent with the finding in Appendix I that the hierarchy loss sharpens, rather than creates, the radial ordering.

### E.5 Temporal Consistency and Hierarchy Losses

HyperForecast employs two auxiliary losses: temporal consistency loss $\mathcal{L}_{\text{tc}}$ penalizes abrupt hyperbolic velocity changes between prediction steps, while hierarchy loss $\mathcal{L}_{\text{hier}}$ enforces radial ordering (trend near origin, residual near boundary). We perform a grid search over both loss weights on ZafNoo, Temp, and AQShunyi to identify optimal regularization strengths (all other hyperparameters fixed, $H = 96$).

Table 8: Hierarchy-loss weight sensitivity at $H = 96$ (NAR), with $\lambda_{\text{tc}} = 0.1$ fixed. MSE reported; best per dataset in **bold**.

| | $\lambda_{\text{hier}}$ | | | | | | |
|---|---|---|---|---|---|---|---|
| Dataset | 0 | 0.0001 | 0.001 | 0.005 | 0.01 | 0.05 | 0.1 |
| ZafNoo | 0.439 | 0.429 | 0.429 | 0.425 | 0.423 | **0.419** | 0.422 |
| Temp | 0.128 | 0.128 | 0.135 | **0.124** | 0.124 | 0.128 | 0.138 |
| AQShunyi | 0.610 | **0.605** | 0.617 | 0.619 | 0.619 | 0.615 | 0.609 |

The optimal hierarchy weight is dataset-dependent: ZafNoo prefers the strongest setting tested ($\lambda_{\text{hier}} = 0.05$, 4.6% MSE reduction over $\lambda_{\text{hier}} = 0$), Temp prefers a moderate weight (0.005,

Table 9: Temporal-consistency loss weight sensitivity at $H = 96$ (NAR), with $\lambda_{\text{hier}} = 0.01$ fixed. MSE reported; best per dataset in **bold**.

| | $\lambda_{\text{tc}}$ | | | |
|---|---|---|---|---|
| Dataset | 0 | 0.05 | 0.1 | 0.2 |
| ZafNoo | 0.425 | 0.425 | **0.423** | 0.430 |
| Temp | 0.125 | **0.124** | 0.124 | 0.124 |
| AQShunyi | 0.619 | **0.618** | 0.619 | 0.618 |

3.5% reduction), and AQShunyi prefers the weakest nonzero weight (0.0001, 0.9% reduction). Removing the hierarchy loss entirely ($\lambda_{\text{hier}} = 0$) is worst or near-worst on all three, most clearly on ZafNoo, consistent with the necessity result above, where removing $\mathcal{L}_{\text{hier}}$ also hurt every dataset tested. Sensitivity to the temporal-consistency weight is an order of magnitude smaller across all three datasets (differences under 1.7% MSE between any two weights), indicating the model is comparatively robust to this hyperparameter and the hierarchy weight is the one worth tuning per dataset.

## F Run-to-Run Stability

To confirm the reported gains are not seed noise, we retrain HyperForecast NAR on seven representative datasets across 5 seeds. Standard deviations stay under 0.011 MSE on most datasets, with PM2.5 being a consistent exception, having MSE std between 0.010–0.027 across horizons (up to 6.5% of its mean at $H = 720$).

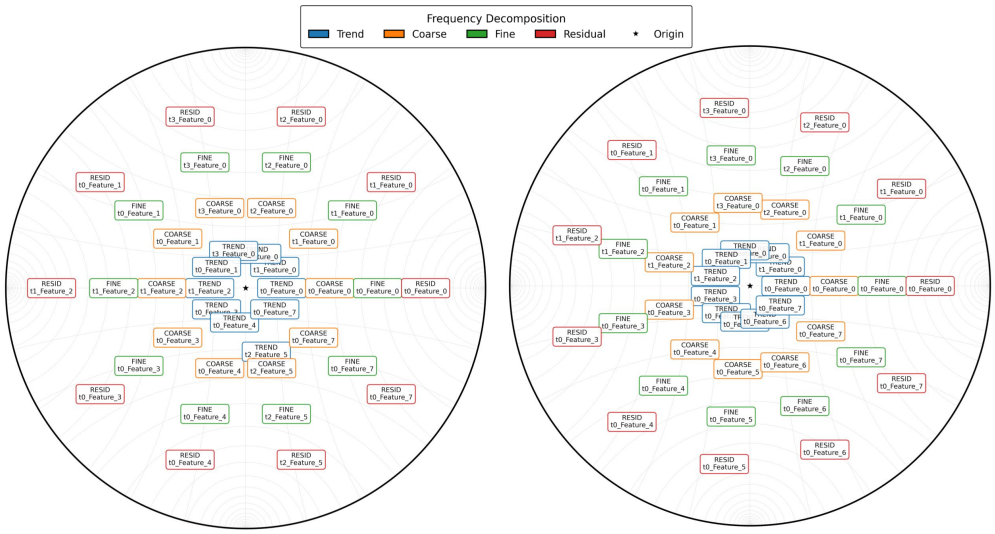
Table 10: Mean $\pm$ std MSE/MAE over 5 seeds for HyperForecast NAR.

| Dataset | $H$ | MSE (mean $\pm$ std) | MAE (mean $\pm$ std) |
|---|---|---|---|
| Weather | 96 | 0.146 $\pm$ 0.002 | 0.210 $\pm$ 0.003 |
| | 192 | 0.188 $\pm$ 0.001 | 0.253 $\pm$ 0.002 |
| | 336 | 0.245 $\pm$ 0.010 | 0.301 $\pm$ 0.012 |
| | 720 | 0.309 $\pm$ 0.005 | 0.349 $\pm$ 0.006 |
| AQShunyi | 96 | 0.609 $\pm$ 0.003 | 0.474 $\pm$ 0.002 |
| | 192 | 0.646 $\pm$ 0.002 | 0.498 $\pm$ 0.005 |
| | 336 | 0.680 $\pm$ 0.008 | 0.518 $\pm$ 0.006 |
| | 720 | 0.718 $\pm$ 0.003 | 0.533 $\pm$ 0.004 |
| AQWan | 96 | 0.698 $\pm$ 0.001 | 0.465 $\pm$ 0.004 |
| | 192 | 0.738 $\pm$ 0.008 | 0.482 $\pm$ 0.003 |
| | 336 | 0.765 $\pm$ 0.003 | 0.493 $\pm$ 0.003 |
| | 720 | 0.821 $\pm$ 0.008 | 0.520 $\pm$ 0.008 |
| ZafNoo | 96 | 0.420 $\pm$ 0.003 | 0.404 $\pm$ 0.004 |
| | 192 | 0.474 $\pm$ 0.006 | 0.441 $\pm$ 0.007 |
| | 336 | 0.517 $\pm$ 0.008 | 0.468 $\pm$ 0.006 |
| | 720 | 0.554 $\pm$ 0.011 | 0.481 $\pm$ 0.007 |
| PM2.5 | 96 | 0.428 $\pm$ 0.011 | 0.452 $\pm$ 0.014 |
| | 192 | 0.425 $\pm$ 0.010 | 0.453 $\pm$ 0.015 |
| | 336 | 0.428 $\pm$ 0.018 | 0.478 $\pm$ 0.021 |
| | 720 | 0.414 $\pm$ 0.027 | 0.487 $\pm$ 0.023 |
| Temp | 96 | 0.126 $\pm$ 0.002 | 0.273 $\pm$ 0.003 |
| | 192 | 0.123 $\pm$ 0.001 | 0.272 $\pm$ 0.001 |
| | 336 | 0.118 $\pm$ 0.001 | 0.266 $\pm$ 0.001 |
| | 720 | 0.119 $\pm$ 0.006 | 0.267 $\pm$ 0.007 |
| Wind | 96 | 0.774 $\pm$ 0.005 | 0.621 $\pm$ 0.004 |
| | 192 | 0.905 $\pm$ 0.009 | 0.689 $\pm$ 0.003 |
| | 336 | 1.007 $\pm$ 0.006 | 0.742 $\pm$ 0.001 |
| | 720 | 1.051 $\pm$ 0.006 | 0.773 $\pm$ 0.003 |

## G Additional Hyperbolic Embedding Visualizations

To further validate the consistency of learned frequency-based hierarchies across diverse datasets, we provide Poincaré disk visualizations for two additional benchmarks with markedly different

temporal characteristics, ZafNoo (an evapotranspiration dataset with strong multi-scale seasonality) and Exchange (a financial time series with weak periodic structure) (Figure 12). The autoregressive (AR) forecasting and non-autoregressive (NAR) variant produce embeddings with similar hierarchical structure on both datasets.




(a)ZafNoo Embeddings on Poincaré Disk

(b) Exchange Embeddings on Poincaré Disk

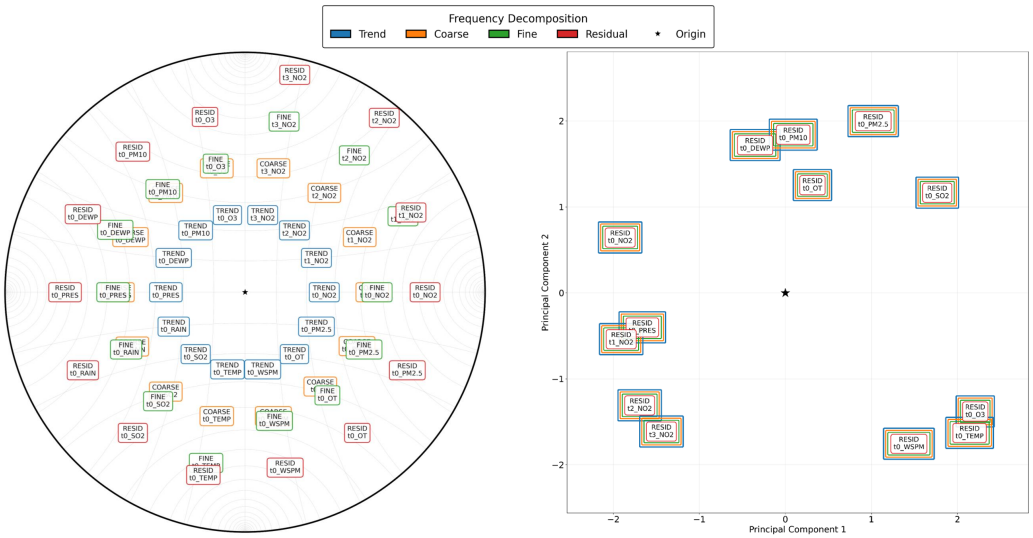
Figure 12: Each point represents a (Scale_TimeSegment_Feature) tuple at a specific frequency scale. Trend (innermost) captures low-frequency patterns closest to the origin, while Residual (outermost) encodes high-frequency noise farther from the origin. The **radial hierarchy** demonstrates that our model successfully learns frequency-based decomposition (Trend < Coarse < Fine < Residual). Node labels follow the format `t{segment}_{feature}` (e.g., t0_Feature_1 = time segment 0, feature 1).

The fact that both datasets, despite very different periodicity profiles, produce the same hierarchical structure validates that hyperbolic geometry adapts to varying degrees of temporal complexity and that the decomposition reliably induces frequency-based radial separation across diverse domains.

## H Qualitative Geometry Comparison: Poincaré vs. Euclidean Embeddings

To complement the quantitative geometry ablation in Sec. E, we visualize the learned segment embeddings produced by HyperForecast (Poincaré) and by a controlled Euclidean variant of the same architecture on two representative air-quality benchmarks: AQShunyi (Figure 13) and AQWan (Figure 14). The Euclidean variant retains an identical learnable decomposition, segment encoder, RevIN normalization, and forecasting head, replacing only the hyperbolic dynamics with their flat counterparts. The Euclidean embeddings are projected to two dimensions via PCA; the Poincaré embeddings are visualized directly on the unit disk via UMAP.

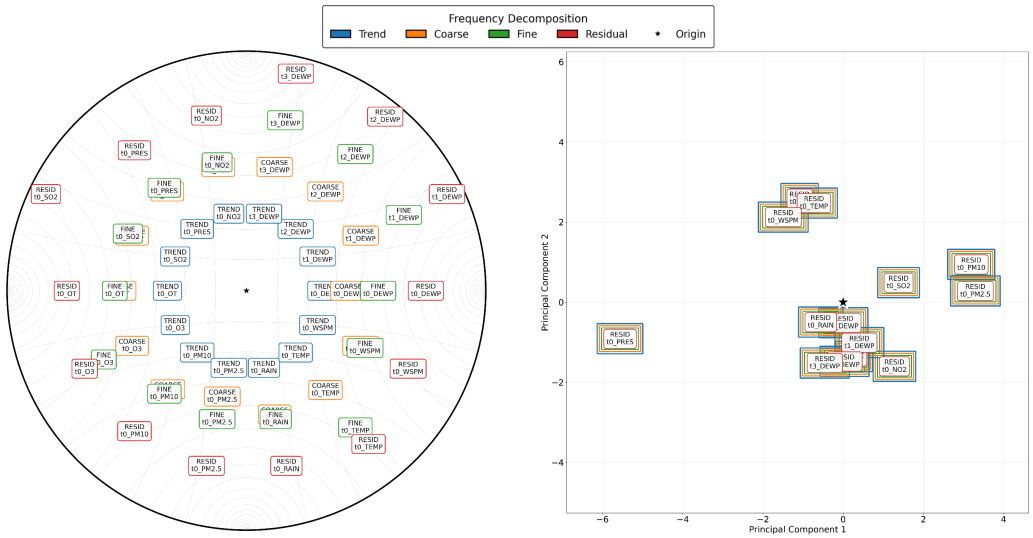
The Poincaré panels (left in each figure) exhibit an unambiguous radial separation by frequency band. Trend embeddings cluster tightly near the origin, while coarse seasonal, fine seasonal, and residual components occupy progressively larger radii, realizing the ordering $r_{\mathbf{T}} < r_{\mathbf{S}_c} < r_{\mathbf{S}_f} < r_{\mathbf{R}}$ induced by the cascaded decomposition. In the Euclidean PCA panels (right), this structure collapses. For every ⟨TIMESTEP, FEATURE⟩ pair, the four frequency bands project to nearly coincident 2D coordinates. The flat geometry, which allocates representational capacity uniformly across the Euclidean space, has no mechanism for encoding the frequency hierarchy, so decomposition is preserved as a *label* but not as a *geometric structure*. This visual collapse is consistent with the MSE degradation reported when the manifold is replaced (Sec. E).




(a) AQShunyi Embeddings on Poincaré Disk from HyperForecast

(a) AQShunyi Embeddings from baseline Euclidean model

Figure 13: Segment embeddings learned on AQShunyi. **Left:** Poincaré disk from HyperForecast. Trend embeddings cluster near the origin, while coarse, fine, and residual components sit at increasing radii. **Right:** Euclidean PCA from the same architecture. All four frequency components collapse to coincident 2D positions, eliminating the radial hierarchy.




(a) AQWan Embeddings on Poincaré Disk from HyperForecast

(a) AQWan Embeddings from baseline Euclidean model

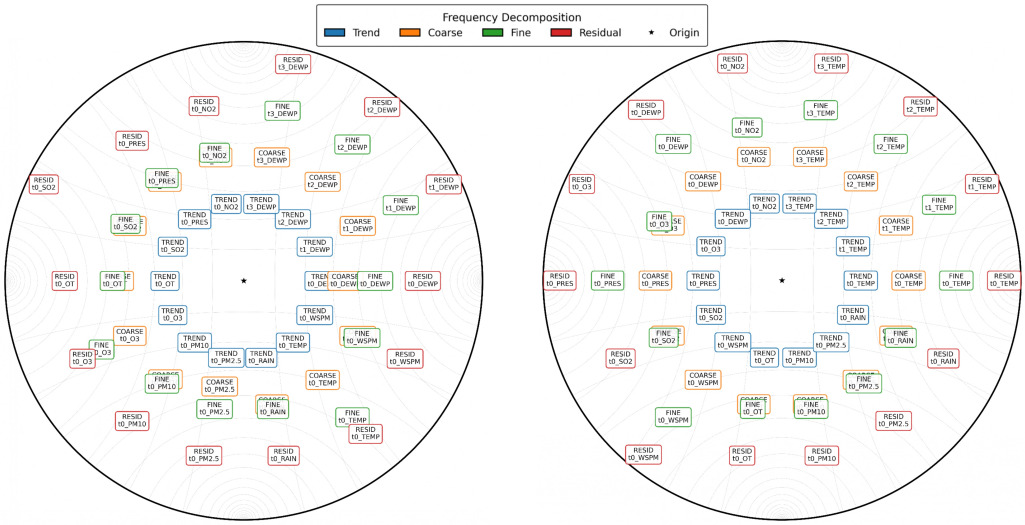
Figure 14: Segment embeddings learned on AQWan; conventions follow Figure 13. Similar radial ordering emerges in the Poincaré panel and is absent in the Euclidean PCA, where the four components stack at identical coordinates.

## I Radial Hierarchy Emerges Without Hierarchical Constraint

A natural concern is whether the radial frequency hierarchy reported is a genuine property of the embedding geometry or merely an artefact of the hierarchical radial loss $\mathcal{L}_{\text{hier}}$, which explicitly penalizes deviations from the target ordering. To rule out this confound, we retrain HyperForecast

on AQWan with $\lambda_{\text{hier}} = 0$ (no explicit hierarchy constraint) and compare the resulting Poincaré embeddings against the standard run with $\lambda_{\text{hier}} = 1e-4$.

Both embeddings on the Poincaré disk are illustrated in Figure 15. Both panels show the same monotonic ordering $r_{\mathbf{T}} < r_{\mathbf{S}_c} < r_{\mathbf{S}_f} < r_{\mathbf{R}}$. This is consistent with our design hypothesis, that the radial hierarchy is a geometric consequence of the interaction between the cascaded decomposition, which produces components of strictly decreasing temporal smoothness and strictly increasing variance, and the exponential volume growth of $\mathbb{P}_c^d$, which naturally allocates more capacity to higher-variance signals at larger radii. The hierarchical loss *sharpens* an ordering that the architecture already prefers. We therefore report results with $\lambda_{\text{hier}} > 0$ throughout the main paper for the cleanest visualizations and tightest radial bounds, but the underlying inductive bias is geometric, not enforced.




(a) AQWan Embeddings on Poincaré Disk (With Hierarchical Loss)

(b) AQWan Embeddings on Poincaré Disk (Without Hierarchical Loss)

Figure 15: Poincaré-disk embeddings learned on AQWan with and without the hierarchical radial loss $\mathcal{L}_{\text{hier}}$.

## J Full Results

The complete per-horizon forecasting results for all 11 benchmark datasets across prediction lengths $H \in \{96, 192, 336, 720\}$ are presented in Table 11. These detailed results supplement the averaged metrics reported in Table 1, which aggregates performance across all four horizons to provide a concise summary of model performance.

## K Implementation Details

All experiments were conducted on a single NVIDIA A100 GPU (40GB) with CUDA 12.6.0. We implemented HyperForecast in PyTorch (Paszke et al., 2019) with GeoOpt (Kochurov et al., 2020) for Riemannian optimization on hyperbolic manifolds. The model was trained for 30 epochs using the RiemannianAdam optimizer with OneCycleLR scheduling (Smith & Topin, 2019). We set the embedding dimension to 64 and hidden dimension to 256 across all experiments. The hyperbolic curvature was initialized to 1.0 with learnable scaling parameters for exponential map stability. Gradient clipping with a maximum norm of 5.0 was applied to prevent numerical instability in hyperbolic operations. We used a batch size of 32 for all datasets except METR-LA (207 channels, the dataset with the highest dimensionality in our benchmark suite). Dropout rates of 0.3 for encoding layers and 0.2 for reconstruction layers were employed for regularization. We performed hyperparameter tuning via grid search, exploring temporal consistency loss weights $\{0.1, 0.2\}$, hierarchy loss weights $\{1 \times 10^{-4}, 1 \times 10^{-3}, 0.1\}$, learning rates $\{1 \times 10^{-1}, 1 \times 10^{-2}, 1 \times 10^{-3}\}$,

Table 11: Long-term forecasting results across 11 datasets and 4 horizons ($H \in \{96, 192, 336, 720\}$). HF: HyperForecast (ours). The input length L is set to 720 across all models. Best results are marked in **red-bold**; second-best are **blue-bold**.

| Methods | | HF NAR | | HF AR | | PhaseFormer | | TimeBase | | SparseTSF | | TimeMixer | | FITS | | iTransformer | | DLinear | | PatchTST | | TimesNet | | FEDformer | | Autoformer | | Informer | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metrics | H | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE | MSE | MAE |
| Weather | 96 | **0.145** | 0.206 | **0.143** | 0.202 | 0.148 | **0.195** | 0.146 | **0.198** | 0.174 | 0.231 | 0.163 | 0.223 | 0.176 | 0.232 | 0.159 | 0.212 | 0.174 | 0.242 | 0.149 | 0.199 | 0.176 | 0.234 | 0.289 | 0.342 | 0.323 | 0.389 | 0.283 | 0.361 |
| | 192 | **0.188** | 0.252 | 0.190 | 0.248 | 0.193 | **0.237** | **0.185** | **0.241** | 0.216 | 0.267 | 0.201 | 0.254 | 0.216 | 0.268 | 0.203 | 0.252 | 0.215 | 0.277 | 0.193 | 0.243 | 0.219 | 0.270 | 0.340 | 0.394 | 0.389 | 0.423 | 0.445 | 0.461 |
| | 336 | **0.237** | 0.292 | 0.238 | 0.289 | 0.242 | **0.278** | **0.236** | **0.281** | 0.260 | 0.299 | 0.258 | 0.300 | 0.261 | 0.299 | 0.253 | 0.291 | 0.262 | 0.319 | 0.240 | 0.281 | 0.277 | 0.311 | 0.370 | 0.408 | 0.497 | 0.495 | 0.587 | 0.526 |
| | 720 | **0.303** | 0.344 | 0.315 | 0.341 | 0.309 | **0.332** | **0.309** | **0.331** | 0.325 | 0.345 | 0.329 | 0.348 | 0.325 | 0.346 | 0.317 | 0.337 | 0.319 | 0.359 | 0.312 | 0.334 | 0.344 | 0.356 | 0.420 | 0.421 | 0.573 | 0.520 | 0.953 | 0.703 |
| AQShunyi | 96 | **0.607** | **0.475** | **0.606** | **0.473** | 0.650 | 0.493 | 0.629 | 0.489 | 0.751 | 0.553 | 0.703 | 0.534 | 0.759 | 0.555 | 0.686 | 0.508 | 0.751 | 0.579 | 0.643 | 0.476 | 0.759 | 0.560 | 1.247 | 0.819 | 1.394 | 0.931 | 1.221 | 0.864 |
| | 192 | **0.640** | **0.498** | **0.642** | **0.493** | 0.729 | 0.519 | 0.661 | 0.500 | 0.774 | 0.551 | 0.720 | 0.524 | 0.774 | 0.553 | 0.728 | 0.520 | 0.771 | 0.571 | 0.692 | 0.501 | 0.785 | 0.557 | 1.218 | 0.813 | 1.394 | 0.872 | 1.595 | 0.951 |
| | 336 | **0.670** | **0.510** | **0.669** | **0.508** | 0.740 | 0.533 | 0.683 | 0.510 | 0.752 | 0.544 | 0.747 | 0.546 | 0.755 | 0.544 | 0.732 | 0.529 | 0.758 | 0.580 | 0.695 | 0.514 | 0.802 | 0.566 | 1.071 | 0.742 | 1.438 | 0.900 | 1.699 | 0.957 |
| | 720 | **0.712** | 0.528 | 0.742 | 0.542 | 0.770 | 0.546 | **0.725** | **0.523** | 0.763 | 0.537 | 0.772 | 0.541 | 0.763 | 0.538 | 0.744 | 0.524 | 0.748 | 0.559 | 0.732 | **0.520** | 0.807 | 0.554 | 0.985 | 0.655 | 1.344 | 0.809 | 2.236 | 1.094 |
| AQWan | 96 | **0.692** | **0.458** | **0.696** | **0.464** | 0.781 | 0.488 | 0.726 | 0.477 | 0.758 | 0.486 | 0.762 | 0.484 | 0.795 | 0.510 | 0.798 | 0.511 | 0.835 | 0.531 | 0.735 | 0.467 | 0.828 | 0.527 | 0.865 | 0.523 | 0.895 | 0.592 | 1.027 | 0.699 |
| | 192 | **0.733** | **0.479** | **0.742** | 0.486 | 0.823 | 0.507 | 0.761 | 0.492 | 0.851 | 0.542 | 0.779 | 0.492 | 0.765 | **0.485** | 0.785 | 0.501 | 0.884 | 0.558 | 0.792 | 0.500 | 0.807 | 0.510 | 1.001 | 0.603 | 0.935 | 0.600 | 1.293 | 0.834 |
| | 336 | **0.760** | **0.492** | **0.766** | **0.499** | 0.842 | 0.527 | 0.783 | 0.502 | 0.810 | 0.531 | 0.857 | 0.559 | 0.861 | 0.564 | 0.818 | 0.537 | 0.842 | 0.550 | 0.804 | 0.525 | 0.810 | 0.529 | 0.893 | 0.569 | 0.990 | 0.666 | 1.017 | 0.711 |
| | 720 | **0.819** | **0.514** | 0.846 | 0.532 | 0.900 | 0.547 | 0.847 | 0.524 | 0.883 | 0.546 | 0.891 | 0.549 | **0.831** | **0.517** | 0.971 | 0.604 | 0.972 | 0.600 | 0.909 | 0.561 | 0.982 | 0.606 | 0.967 | 0.566 | 1.185 | 0.759 | 1.156 | 0.731 |
| CzeLan | 96 | **0.171** | 0.236 | **0.172** | 0.235 | 0.175 | **0.229** | 0.177 | **0.224** | 0.204 | 0.263 | 0.182 | 0.233 | 0.208 | 0.268 | 0.202 | 0.259 | 0.187 | 0.239 | 0.184 | 0.236 | 0.204 | 0.262 | 0.208 | 0.260 | 0.251 | 0.334 | 0.293 | 0.384 |
| | 192 | **0.205** | 0.267 | 0.212 | 0.278 | 0.219 | 0.271 | **0.211** | **0.250** | 0.227 | 0.274 | 0.216 | 0.259 | 0.242 | 0.292 | 0.233 | 0.282 | 0.226 | 0.271 | 0.214 | **0.257** | 0.256 | 0.308 | 0.258 | 0.307 | 0.285 | 0.360 | 0.329 | 0.399 |
| | 336 | 0.237 | 0.305 | **0.234** | 0.298 | 0.237 | **0.280** | 0.243 | **0.276** | 0.239 | 0.305 | 0.246 | 0.312 | 0.247 | 0.316 | 0.239 | 0.305 | 0.238 | 0.301 | **0.229** | 0.290 | 0.274 | 0.347 | 0.299 | 0.365 | 0.263 | 0.354 | 0.366 | 0.476 |
| | 720 | **0.270** | 0.336 | 0.278 | 0.334 | 0.270 | **0.310** | **0.269** | **0.297** | 0.292 | 0.325 | 0.291 | 0.322 | 0.303 | 0.338 | 0.305 | 0.340 | 0.315 | 0.349 | 0.288 | 0.319 | 0.299 | 0.332 | 0.300 | 0.324 | 0.357 | 0.423 | 0.399 | 0.446 |
| ZafNoo | 96 | 0.418 | **0.399** | 0.420 | **0.398** | 0.457 | 0.410 | 0.435 | 0.407 | 0.435 | 0.439 | **0.410** | 0.411 | 0.458 | 0.461 | 0.442 | 0.447 | 0.423 | 0.423 | **0.412** | 0.413 | 0.443 | 0.444 | 0.484 | 0.473 | 0.548 | 0.566 | 0.651 | 0.672 |
| | 192 | **0.469** | 0.437 | **0.466** | **0.431** | 0.524 | 0.450 | 0.470 | **0.435** | 0.548 | 0.509 | 0.473 | 0.437 | 0.506 | 0.468 | 0.560 | 0.522 | 0.534 | 0.494 | 0.518 | 0.479 | 0.572 | 0.529 | 0.579 | 0.512 | 0.738 | 0.712 | 0.745 | 0.730 |
| | 336 | **0.508** | **0.462** | **0.507** | 0.462 | 0.556 | 0.473 | 0.511 | **0.453** | 0.525 | 0.474 | 0.576 | 0.520 | 0.528 | 0.477 | 0.536 | 0.486 | 0.596 | 0.537 | 0.528 | 0.477 | 0.579 | 0.523 | 0.566 | 0.499 | 0.624 | 0.564 | 0.662 | 0.601 |
| | 720 | **0.545** | **0.473** | 0.563 | 0.500 | 0.594 | 0.491 | 0.565 | **0.484** | 0.617 | 0.540 | **0.560** | 0.487 | 0.590 | 0.515 | 0.623 | 0.546 | 0.606 | 0.527 | 0.581 | 0.506 | 0.587 | 0.511 | 0.776 | 0.671 | 0.758 | 0.698 | 0.847 | 0.783 |
| Exchange | 96 | **0.079** | **0.207** | **0.082** | 0.216 | 0.107 | 0.236 | 0.087 | **0.207** | 0.083 | 0.211 | 0.090 | 0.229 | 0.094 | 0.241 | 0.096 | 0.244 | 0.096 | 0.243 | 0.083 | 0.211 | 0.088 | 0.224 | 0.092 | 0.224 | 0.119 | 0.316 | 0.106 | 0.276 |
| | 192 | **0.165** | 0.321 | **0.159** | **0.303** | 0.173 | **0.303** | 0.185 | 0.306 | 0.172 | 0.317 | 0.167 | 0.305 | 0.167 | 0.307 | 0.176 | 0.325 | 0.169 | 0.310 | 0.168 | 0.307 | 0.173 | 0.317 | 0.208 | 0.355 | 0.237 | 0.444 | 0.252 | 0.482 |
| | 336 | **0.305** | 0.457 | **0.263** | **0.395** | 0.313 | 0.405 | 0.355 | 0.436 | 0.308 | **0.395** | 0.342 | 0.437 | 0.319 | 0.409 | 0.312 | 0.399 | 0.344 | 0.438 | 0.324 | 0.413 | 0.362 | 0.462 | 0.405 | 0.512 | 0.437 | 0.561 | 0.536 | 0.700 |
| | 720 | **0.607** | 0.647 | **0.591** | **0.602** | 1.423 | 0.863 | 0.608 | **0.619** | 0.701 | 0.716 | 0.721 | 0.734 | 0.629 | 0.643 | 0.619 | 0.635 | 0.637 | 0.648 | 0.659 | 0.670 | 0.683 | 0.696 | 0.813 | 0.795 | 0.994 | 1.035 | 0.835 | 0.865 |
| METR-LA | 96 | 1.038 | 0.702 | 1.053 | **0.635** | 1.120 | 0.657 | **1.028** | 0.655 | 1.056 | 0.667 | 1.051 | 0.657 | 1.047 | 0.661 | **1.023** | **0.640** | 1.096 | 0.685 | 1.053 | 0.658 | 1.206 | 0.755 | 1.440 | 0.882 | 1.443 | 0.908 | 1.666 | 1.088 |
| | 192 | 1.239 | 0.706 | 1.248 | 0.759 | 1.264 | 0.732 | **1.112** | 0.711 | 1.232 | 0.730 | 1.163 | **0.689** | 1.295 | 0.772 | 1.196 | 0.713 | 1.330 | 0.788 | **1.134** | **0.672** | 1.359 | 0.806 | 1.478 | 0.839 | 1.366 | 0.830 | 1.452 | 0.904 |
| | 336 | 1.324 | 0.743 | 1.354 | 0.786 | 1.398 | 0.795 | 1.317 | 0.771 | 1.356 | 0.772 | **1.304** | **0.737** | 1.309 | 0.743 | 1.320 | 0.750 | 1.353 | 0.764 | **1.250** | **0.706** | 1.376 | 0.778 | 1.555 | 0.860 | 1.525 | 0.883 | 2.036 | 1.209 |
| | 720 | 1.462 | 0.812 | 1.528 | 0.837 | 1.611 | 0.864 | 1.558 | 0.877 | 1.562 | 0.885 | 1.564 | 0.879 | 1.534 | 0.865 | 1.474 | 0.834 | **1.423** | **0.800** | **1.404** | **0.789** | 1.619 | 0.910 | 1.725 | 0.921 | 2.221 | 1.277 | 2.420 | 1.384 |
| PM2.5 | 96 | **0.416** | **0.437** | **0.416** | 0.437 | 0.435 | 0.438 | 0.432 | **0.435** | 0.475 | 0.482 | 0.445 | 0.449 | 0.495 | 0.501 | 0.466 | 0.474 | 0.437 | 0.441 | 0.450 | 0.454 | 0.462 | 0.467 | 0.560 | 0.550 | 0.596 | 0.615 | 0.566 | 0.593 |
| | 192 | **0.418** | 0.444 | **0.418** | 0.441 | 0.440 | **0.435** | 0.429 | **0.432** | 0.458 | 0.465 | 0.479 | 0.483 | 0.455 | 0.460 | 0.505 | 0.511 | 0.454 | 0.458 | 0.461 | 0.465 | 0.452 | 0.456 | 0.480 | 0.474 | 0.567 | 0.614 | 0.670 | 0.678 |
| | 336 | **0.406** | 0.449 | **0.416** | 0.443 | 0.428 | **0.430** | 0.426 | **0.433** | 0.478 | 0.482 | 0.473 | 0.475 | 0.450 | 0.453 | 0.461 | 0.467 | 0.449 | 0.451 | 0.475 | 0.477 | 0.490 | 0.492 | 0.586 | 0.555 | 0.671 | 0.684 | 0.749 | 0.793 |
| | 720 | **0.400** | 0.477 | **0.374** | **0.427** | 0.418 | **0.433** | 0.425 | 0.448 | 0.456 | 0.479 | 0.464 | 0.487 | 0.440 | 0.461 | 0.489 | 0.514 | 0.457 | 0.479 | 0.448 | 0.470 | 0.461 | 0.484 | 0.612 | 0.609 | 0.652 | 0.686 | 0.561 | 0.589 |
| Solar | 96 | 0.177 | 0.246 | **0.170** | **0.227** | **0.176** | **0.226** | 0.192 | 0.239 | 0.205 | 0.241 | 0.232 | 0.271 | 0.218 | 0.257 | 0.217 | 0.255 | 0.200 | 0.234 | 0.205 | 0.239 | 0.243 | 0.284 | 0.231 | 0.254 | 0.279 | 0.339 | 0.308 | 0.377 |
| | 192 | **0.204** | 0.281 | 0.206 | 0.256 | **0.194** | **0.237** | 0.213 | **0.252** | 0.215 | 0.265 | 0.238 | 0.293 | 0.207 | 0.256 | 0.208 | 0.257 | 0.239 | 0.294 | 0.227 | 0.280 | 0.253 | 0.311 | 0.257 | 0.301 | 0.337 | 0.418 | 0.332 | 0.422 |
| | 336 | 0.221 | 0.289 | 0.227 | 0.282 | **0.199** | **0.247** | 0.222 | **0.261** | **0.213** | 0.276 | 0.234 | 0.301 | 0.229 | 0.295 | 0.238 | 0.309 | 0.231 | 0.298 | 0.225 | 0.290 | 0.222 | 0.287 | 0.260 | 0.319 | 0.263 | 0.352 | 0.314 | 0.421 |
| | 720 | **0.217** | 0.280 | 0.243 | 0.306 | **0.203** | **0.250** | 0.235 | **0.264** | 0.232 | 0.272 | 0.273 | 0.319 | 0.261 | 0.307 | 0.270 | 0.319 | 0.237 | 0.277 | 0.249 | 0.291 | 0.254 | 0.298 | 0.260 | 0.290 | 0.345 | 0.413 | 0.355 | 0.424 |
| Temp | 96 | **0.123** | **0.270** | **0.121** | **0.266** | 0.147 | 0.294 | 0.142 | 0.286 | 0.149 | 0.300 | 0.158 | 0.316 | 0.168 | 0.338 | 0.155 | 0.310 | 0.165 | 0.329 | 0.145 | 0.291 | 0.155 | 0.309 | 0.173 | 0.338 | 0.201 | 0.411 | 0.227 | 0.459 |
| | 192 | **0.123** | **0.272** | **0.125** | **0.272** | 0.161 | 0.311 | 0.155 | 0.305 | 0.168 | 0.323 | 0.163 | 0.312 | 0.181 | 0.351 | 0.168 | 0.324 | 0.174 | 0.335 | 0.166 | 0.319 | 0.187 | 0.358 | 0.179 | 0.326 | 0.246 | 0.491 | 0.281 | 0.564 |
| | 336 | **0.117** | **0.265** | **0.132** | **0.284** | 0.188 | 0.336 | 0.176 | 0.323 | 0.220 | 0.373 | 0.249 | 0.421 | 0.223 | 0.381 | 0.232 | 0.395 | 0.246 | 0.417 | 0.221 | 0.374 | 0.236 | 0.400 | 0.250 | 0.412 | 0.312 | 0.554 | 0.349 | 0.606 |
| | 720 | **0.113** | **0.260** | **0.203** | **0.352** | 0.371 | 0.451 | 0.274 | 0.437 | 0.642 | 0.654 | 0.637 | 0.643 | 0.689 | 0.699 | 0.652 | 0.663 | 0.584 | 0.590 | 0.632 | 0.639 | 0.636 | 0.642 | 0.769 | 0.747 | 0.859 | 0.893 | 1.026 | 1.101 |
| Wind | 96 | **0.771** | 0.618 | **0.765** | 0.616 | 0.824 | 0.653 | 0.785 | 0.626 | 0.796 | **0.589** | 0.908 | 0.669 | 0.787 | **0.581** | 0.824 | 0.609 | 0.824 | 0.607 | 0.802 | 0.591 | 0.892 | 0.657 | 1.008 | 0.718 | 1.110 | 0.847 | 1.132 | 0.836 |
| | 192 | **0.893** | **0.679** | **0.897** | 0.683 | 1.001 | 0.728 | 0.902 | 0.690 | 0.949 | 0.698 | 1.040 | 0.764 | 0.910 | **0.674** | 1.054 | 0.779 | 0.961 | 0.707 | 0.944 | 0.694 | 1.008 | 0.741 | 1.256 | 0.876 | 1.218 | 0.954 | 1.540 | 1.188 |
| | 336 | **1.011** | **0.734** | 1.015 | 0.748 | 1.090 | 0.774 | **0.997** | 0.737 | 1.182 | 0.851 | 1.039 | 0.747 | 1.100 | 0.795 | 1.025 | 0.742 | 1.022 | **0.734** | 1.043 | 0.750 | 1.185 | 0.853 | 1.375 | 0.923 | 1.209 | 0.935 | 1.399 | 1.015 |
| | 720 | **1.045** | **0.768** | 1.121 | 0.805 | 1.157 | 0.816 | **1.075** | 0.782 | 1.127 | 0.808 | 1.189 | 0.851 | 1.295 | 0.932 | 1.196 | 0.863 | 1.266 | 0.907 | 1.084 | **0.776** | 1.320 | 0.946 | 1.281 | 0.846 | 1.410 | 1.053 | 1.453 | 1.094 |

and moving window sizes $\{2, 4, 5, 6\}$. The best-performing configurations were selected based on validation MSE for each prediction horizon. RevIN normalization (Kim et al., 2022) was applied to all inputs. We used early stopping with a patience of 5 epochs, monitored on validation loss. All models used a lookback window of 720 timesteps, prediction horizons of $\{96, 192, 336, 720\}$, and contiguous 7:1:2 train/validation/test splits, following standard LTSF benchmarks (Wu et al., 2021; Huang et al., 2025).

## L Code Availability

The code is available at an anonymized GitHub repository: `https://anonymous.4open.science/r/Hyperbolic_TimeSeries-3876/`.