MATRYOSHKA: EVIDENCE-GATED NESTED CHANNEL MIXING FOR TIME SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, channel mixing has become a standard component of multivariate time series forecasting. Most existing methods either mix all channels or restrict mixing to a fixed or learned interaction structure, without testing whether the other channels carry predictive information. As a result, a module that helps on datasets with hundreds of channels can hurt on datasets with only a few. We propose **Matryoshka**, a plug-in module that wraps a forecasting backbone without modifying it. Inspired by the classical Granger causality test, Matryoshka formulates channel mixing as a selection problem, in which held-out evidence of predictive gain decides where and at what scale channels are mixed. The module arranges channels into nested neighborhoods built from a diffusion geometry of the training data, and keeps a neighborhood only if its shared history improves held-out prediction beyond each channel's own history. Each retained neighborhood receives a residual mixer, a "doll", whose capacity grows with the neighborhood's size. The dolls start at zero, so training begins exactly at the backbone. Our theoretical analysis shows that the optimal neighborhood balances omitted cross-channel information against an estimation cost that grows linearly with its size. On datasets with hundreds of channels, Matryoshka reduces the MSE of its backbone by 22% on average on the unified benchmark and by 45% on the CI/CD benchmark.

1 INTRODUCTION

Multivariate time series forecasting predicts the future values of several variables from a lookback window, in domains such as energy, traffic, and weather. Recent progress comes from transformer-based models (Zhou et al., 2021; Wu et al., 2021; Nie et al., 2023; Liu et al., 2024) and simple linear and MLP-based models (Zeng et al., 2023; Chen et al., 2023; Stitsyuk & Choi, 2025).

Real-world multivariate series consist of many channels, which raises a fundamental question: "How much should the channels of a series inform one another?" Channel Independent (CI) models such as DLinear and PatchTST forecast every channel from its own history and are remarkably robust. In contrast, Channel Dependent (CD) models like TSMixer and iTransformer mix all channels and can exploit cross-channel structure, but are less robust. Han et al. (2024) frame this contrast as a capacity-robustness trade-off and compare the two strategies under a common CI/CD protocol.

Between CI and CD, a recent line of work adds channel interaction to a fixed backbone through plug-in modules. In the taxonomy of Qiu et al. (2025a), these modules are Channel Partial (CP): each channel interacts only with a subset of related channels. LIFT (Zhao & Shen, 2024) refines the forecast of each channel with its leading channels, CCM (Chen et al., 2024) groups channels into learned clusters, and xCPD (Li et al., 2026) lets each channel set how much interaction it takes from each frequency band of a shared graph Fourier basis. Despite their differences, these methods decide *where* channels interact by a fixed rule or by an end-to-end learned structure. In both cases, the decision is never tested against held-out data.

Our unified benchmark of five backbones trained under one protocol shows why this matters. LIFT reduces the MSE of its backbones by 9% on average on datasets with hundreds of channels. However, on datasets with few channels (the ETT datasets and Weather, with at most 21 channels), the same plug-in increases the error in more than half of the backbone-dataset pairs, by up to 22%. At the population level, additional channels never increase the optimal forecasting risk. Therefore, when mixing fails, the other channels are not necessarily uninformative. Their information may

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.

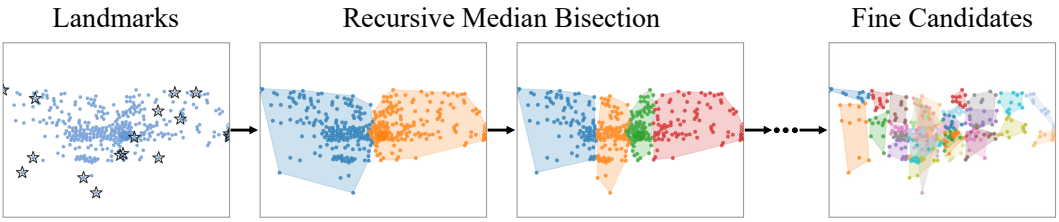


Figure 1: Candidate neighborhoods of Matryoshka on PEMS03 (358 channels), from the training split only. Each point is a channel in the plane of its diffusion embedding. From left to right: the landmarks (stars), two steps of the recursive median bisection, and the fine candidates at the leaves.

instead be lost to the finite sample, the model class, the optimization, or a shift between periods. Before any mixer is trained, a held-out test on the training data can reveal whether a gain survives the finite sample and, in part, whether it persists over time.

We therefore treat channel mixing as a selection problem, in which held-out evidence decides where and at what scale channels are mixed. We propose Matryoshka, a novel plug-in module that adds evidence-gated nested channel mixing to any forecasting backbone without modifying it. First, Matryoshka embeds the channels by how well their lags predict one another on the training data and recursively bisects this embedding into a tree of candidate neighborhoods (Figure 1). Second, the module tests candidates at nested scales (fine groups, coarse neighborhoods, and all channels). A candidate qualifies only if a linear pilot, in the spirit of a Granger test, reduces the error over each channel's own history on two consecutive blocks held out from the training split. Third, each qualified neighborhood receives a residual mixer, a *doll*, whose width grows with the neighborhood's size and is capped by the available data. The dolls start at zero, so training begins exactly at the backbone. We name the module after the matryoshka, a set of wooden dolls that nest one inside another: every neighborhood is contained in the next larger one, and within a scale, the doll of a smaller neighborhood is a prefix of the doll of a larger one.

We summarize our main contributions as follows:

- We formulate channel mixing as a selection problem with held-out evidence and propose Matryoshka, a novel plug-in module that qualifies channel neighborhoods at nested scales with a training-only predictive-gain test.
- We derive an exact expected-risk decomposition in a linear Gaussian reference, in which a larger neighborhood pays off only when the information it adds exceeds an estimation cost that grows linearly with its size, and prove a uniform accuracy bound for the held-out gain of the candidates at a cost logarithmic in the number of channels.
- We conduct extensive experiments on the unified benchmark and the CI/CD benchmark of Han et al. (2024). On datasets with hundreds of channels, Matryoshka reduces the MSE of its backbone by 22% and 45%, respectively.

2 Related Work

Channel strategies. The channel strategy of a forecasting model determines how the channels of a multivariate series interact. We adopt the taxonomy of Qiu et al. (2025a): Channel Independence (CI) forecasts each channel from its own history, Channel Dependence (CD) treats all channels as one entity, and Channel Partiality (CP) lets each channel interact with a subset of related channels.

Informer (Zhou et al., 2021), Autoformer (Wu et al., 2021), and FEDformer (Zhou et al., 2022) are early transformer-based models designed for long-term time series forecasting. These models embed all channels of a time step into one token and are therefore CD. TimesNet (Wu et al., 2023) is a CNN-based model that reshapes the series into 2D tensors by its dominant periods. It shares one embedding across all channels and is therefore also CD. Crossformer (Zhang & Yan, 2023) and CARD (Wang et al., 2024) are transformer-based models that attend along both time and channels. TSMixer (Chen et al., 2023) is an MLP-based model that mixes along the channel dimension, while iTransformer (Liu et al., 2024) embeds every channel as a token and attends across channels.

DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), and xPatch (Stitsyuk & Choi, 2025) are CI models that forecast every channel with a shared univariate network. Their success led Han et al. (2024) to characterize the choice of a channel strategy as a capacity-robustness trade-off and to propose PRReg, a regularized training recipe for CD models. The survey of Qiu et al. (2025a) agrees: CI wins on weakly correlated datasets, and CD or CP on strongly correlated ones.

CP methods and plug-ins. MTGNN (Wu et al., 2020) and CRAFT (Fang et al., 2026) are CP methods with a static partial structure. It comes from a learned graph in MTGNN and a learned channel reordering in CRAFT. MTGNN generalizes spatio-temporal graph models for sensor networks, such as DCRNN (Li et al., 2018) and Graph WaveNet (Wu et al., 2019), which propagate information along a road graph or a learned adjacency, and like them it is a complete forecasting architecture rather than a plug-in. In contrast, recent CP methods adapt their structure to the input, by a rule or end-to-end. LIFT (Zhao & Shen, 2024) refines follower channels from leading indicators selected by lookback correlation. CCM (Chen et al., 2024) groups channels into learned clusters, and HyperMixer (Tian et al., 2025) routes channels through learned hyperedges. PCD (Lee et al., 2024) and DUET (Qiu et al., 2025b) learn channel masks, while xCPD (Li et al., 2026) splits a shared graph Fourier basis into frequency bands and lets each channel set how much it takes from each band.

In all of these methods, the interaction structure is chosen by a fixed rule or by the training loss. Neither examines whether a set of channels carries predictive information about a channel beyond that channel's own history. Additionally, none of these methods ties the scale of interaction to such evidence. Matryoshka instead qualifies its neighborhoods by a held-out test at nested scales, spends capacity only on the qualified ones, and leaves a channel unmixed when no such evidence exists.

Predictive evidence, nesting, and landmark geometry. Granger causality (Granger, 1969) is a classical test of whether one series helps to predict another. The population gain behind our pilot is the Mean Squared Error (MSE) form of Granger causality, namely the reduction in forecasting risk from conditioning on other channels beyond a channel's own history. This gain has been used to build interaction graphs (Tank et al., 2022), but not as a gate on capacity. Our pilot itself compares fitted linear predictors on held-out data (Appendix B.1).

Matryoshka Representation Learning (MRL) (Kusupati et al., 2022) and MatFormer (Devvrit et al., 2024) nest the dimensions of one network so that its prefixes serve as smaller networks, which lets a single trained model be deployed at several sizes. We employ the same prefix mechanism for a different purpose, in which the width of a doll follows the size of the neighborhood it serves, so capacity is nested in the evidence rather than in a deployment budget.

Diffusion maps (Coifman & Lafon, 2006) embed points by a random walk on their affinity graph, and the Nyström method (Williams & Seeger, 2000) approximates it from a subset of landmarks. We use a landmark diffusion embedding, which keeps calibration linear in the number of channels.

3 METHOD

3.1 PROBLEM SETUP AND OVERVIEW

In multivariate time series forecasting, given a lookback window $\mathbf{X} \in \mathbb{R}^{L \times C}$ of C channels, the task is to predict the future values $\mathbf{Y} \in \mathbb{R}^{H \times C}$ over a horizon H. A backbone f_θ maps $\mathbf{X}$ to a forecast $\hat{\mathbf{Y}}$. We treat the backbone as given and leave it unchanged, regardless of whether it follows the Channel Independent (CI), Channel Dependent (CD), or Channel Partial (CP) strategy. Channels are indexed by $[C] = \{1, \ldots, C\}$. For a channel i and a subset of channels $S \subseteq [C]$, $\mathbf{X}_S \in \mathbb{R}^{L \times |S|}$ denotes the lookback of the channels in the subset S. The Bayes risk of forecasting channel i from $\mathbf{X}_S$ alone, at an origin drawn from a fixed distribution, is

$$R_i(S) = \min_f \mathbb{E}\|\mathbf{Y}_i - f(\mathbf{X}_S)\|^2, \tag{1}$$

where f is any function mapping the lookback of S to an H-step forecast.

Matryoshka consists of three stages, as illustrated in Figures 1 and 2. First, a training-only calibration embeds the channels in a landmark diffusion geometry and bisects them into a binary hierarchy

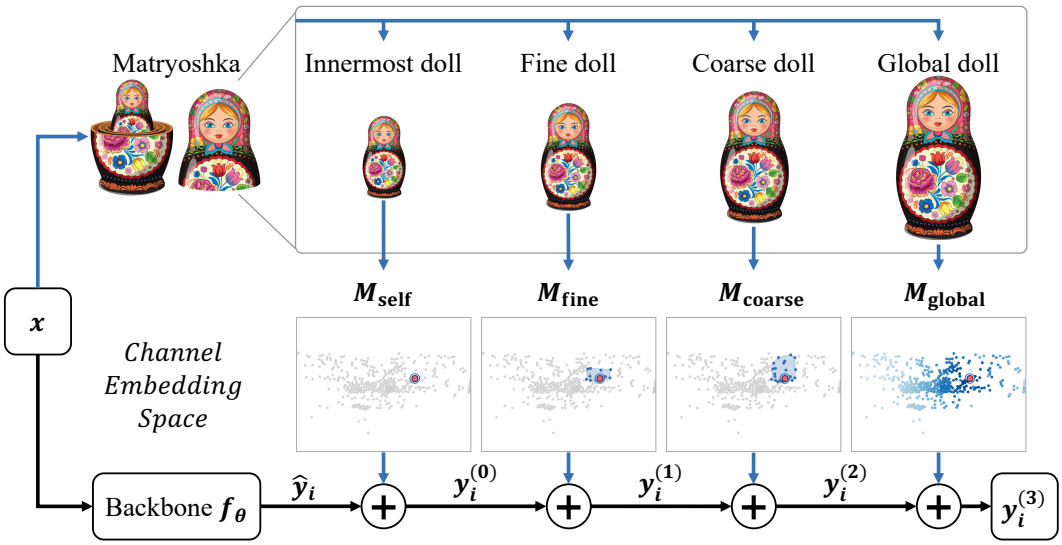


Figure 2: Architecture of Matryoshka. The backbone f_θ is not modified. Its forecast $\hat{\mathbf{y}}_i$ of channel i is refined in turn by the innermost, fine, coarse, and global dolls, which produce $\mathbf{y}_i^{(0)}$ to $\mathbf{y}_i^{(3)}$ (Equation 2). Each doll reads the lookback $\mathbf{x}$ and the current forecast, starts at zero, and has a width nested in the size of its scope and in the training data. The panels show the scope of each doll for one channel (red) in the channel embedding: the channel itself, its fine group, its coarse neighborhood, and all channels. A level that is not qualified for a channel contributes nothing to it.

of candidate neighborhoods (Section 3.2). Second, a linear pilot qualifies the candidates at three nested scales with a two-block held-out test (Section 3.3). These scales are fine groups of at most $g_{\max}$ channels, coarse neighborhoods of at most $2g_{\max}$ channels, and a global level. Finally, every qualified neighborhood receives a residual mixer, a *doll*, which is zero-initialized and whose width is nested in the size of its neighborhood and in the amount of training data (Section 3.4).

Matryoshka refines the backbone forecast $\hat{\mathbf{y}} = f_\theta(\mathbf{x})$ of a lookback window $\mathbf{x}$ once per scale,

$$\mathbf{y}_i^{(k)} = \mathbf{y}_i^{(k-1)} + \mathcal{M}_k\big(\mathbf{x}, \mathbf{y}^{(k-1)} \mid S_k(i)\big), \qquad k = 0, 1, 2, 3, \qquad \mathbf{y}^{(-1)} = \hat{\mathbf{y}}, \tag{2}$$

where $\mathcal{M}_0 = \mathcal{M}_{\text{self}}$, $\mathcal{M}_1 = \mathcal{M}_{\text{fine}}$, $\mathcal{M}_2 = \mathcal{M}_{\text{coarse}}$, and $\mathcal{M}_3 = \mathcal{M}_{\text{global}}$ are the innermost, fine, coarse, and global dolls. They read the channels of their scopes: $S_0(i) = \{i\}$, the fine group $S_1(i) = g(i)$ of channel i, its coarse neighborhood $S_2(i) = G(i)$, and all channels $S_3(i) = [C]$, and $\mathbf{y}_i^{(3)}$ is the final forecast. A level that is not qualified for channel i contributes nothing, so a channel for which no neighborhood qualifies stays a singleton and is mixed with no other channel. Every doll starts at zero, which ensures that training begins exactly at the backbone. The dolls are then trained jointly with the backbone under the backbone's loss, optimizer, and schedule.

3.2 Channel Hierarchy

The candidate neighborhoods are the nodes of a binary tree, which is built once on the training split, before training (Figure 1). Appendix A gives the full construction.

Calibration data. Let n calibration origins be spread over the training split. For every origin and channel, the predictors are the values at a set $\mathcal{L}$ of q lags, and the pilot target is the value at a pilot horizon $h \le H$, both expressed in the frame of the origin's lookback window. The earliest n_{fit} origins form the fitting segment, on which the geometry and all pilot predictors are fitted. The remaining origins form two consecutive held-out blocks B_1 and B_2, which are used only to evaluate gains.

Geometry. Every channel is described by its affinities to J landmark channels, which keeps the cost linear in C. The affinity of a channel to a landmark is the largest absolute correlation between

the two series, or between the lagged history of either and the pilot target of the other, minus a finite-sample noise floor. Two channels are therefore close if one helps to predict the other, and not only if they move together. The channel coordinates are the leading eigenvectors of the degree-normalized affinity matrix, a landmark form of diffusion maps computed from a $J \times J$ eigenproblem (Lemma 1).

Bisection. The hierarchy is the binary tree obtained by recursively bisecting the channel set at the median of the coordinate with the largest variance within the node. The median keeps the two halves of equal size, so the tree is as shallow as a binary tree can be. The nodes of this tree are the only candidate neighborhoods, which replaces the 2^C subsets of an unrestricted search by fewer than $2C$ of them. A channel therefore has at most $\lceil \log_2 C \rceil + 1$ candidates on its root-to-leaf chain. For example, this gives only eleven candidates on the 862 channels of Traffic.

3.3 Evidence-Gated Qualification

Evidence-gated qualification is a held-out test that accepts a candidate only if it carries predictive information about its channels beyond their own history. For $i \in S \subseteq T$, the tower property gives $R_i(S) - R_i(T) = \mathbb{E}\|\mathbb{E}[\mathbf{Y}_i \mid \mathbf{X}_T] - \mathbb{E}[\mathbf{Y}_i \mid \mathbf{X}_S]\|^2 \geq 0$ (Lemma 2), so at the population level additional channels never increase the optimal risk, and any failure of mixing is a failure to recover their information (Remark 1). For a channel $i \in S$, the gain of the neighborhood S over the singleton $\{i\}$ is $\Gamma_i(S) = R_i(\{i\}) - R_i(S)$. Unlike a correlation, the gain is zero for a noisy copy of channel i and can be positive for a channel that predicts i only with a lag. The correlation-based geometry therefore only proposes candidates, and the gain decides which of them are kept.

Pilot test. For a candidate S, the pilot fits two ridge predictors on the fitting segment. The first predictor forecasts the pilot target of each channel $i \in S$ from its own q lags. The second predictor forecasts the same target from the $q|S|$ lags of all channels in S. On each held-out block b, the test statistic is the mean relative reduction in squared error, which is calculated as:

$$\hat{\gamma}_b(S) = \frac{1}{|S|} \sum_{i \in S} \frac{\mathrm{MSE}_b(\{i\} \to i) - \mathrm{MSE}_b(S \to i)}{\mathrm{MSE}_b(\{i\} \to i)}, \tag{3}$$

where $\mathrm{MSE}_b(S \to i)$ is the error on block b of the predictor of i from the lags of S. The candidate S passes the test if $\min\big(\hat{\gamma}_1(S), \hat{\gamma}_2(S)\big) \geq \tau$ for a margin $\tau > 0$ (Appendix E). A gain must survive two consecutive periods of the training split, so a gain in one lucky interval cannot retain a neighborhood.

Nested scales. The tree is traversed top-down, and the three scales of Section 3.1 are qualified on the way. At the fine scale, a node with at most $g_{\max}$ channels that passes the test becomes the fine group $g(i)$ of its channels. Otherwise, the node is split and its children are visited, down to singletons. At the coarse scale, the first node on every root-to-leaf path with more than $g_{\max}$ and at most $2g_{\max}$ channels is tested with the same rule and, if it passes, becomes the coarse neighborhood $G(i)$ of its channels. At the global scale, the pilot asks whether each channel's own lags together with the lags of the channel-mean series predict better than its own lags alone, which is the linear form of the plain mean from which the global doll starts. The candidate family therefore has at most $3C$ members, which Theorem 2 uses for a bound that holds uniformly over the family.

Because the pilot horizon h can be much shorter than H, fine groups and the global level must also pass the test on a second target, the values at eight steps spread evenly over $1, \ldots, H$ (Appendix C.2).

3.4 Nested Dolls

A doll is a residual mixer that serves the qualified neighborhoods of one scale, which share its weights, each through a prefix sized to it. Each doll has its own parameters, and we reuse the letters **E**, **M**, **O**, and **F** for their common roles. The dolls read a common token, which concatenates the lookback of a channel with the current forecast, both centered and scaled by the mean m_i and the standard deviation s_i of that channel's lookback,

$$\mathbf{c}_i = \big[(\mathbf{x}_i - m_i)/s_i;\ (\mathbf{y}_i - m_i)/s_i\big] \in \mathbb{R}^{L+H}, \tag{4}$$

where $[\cdot\,;\cdot]$ denotes concatenation along time. Every doll that reads this token multiplies its output by s_i, which returns its residual to data units. The backbone's input is not normalized by the plug-in.

Innermost doll. The innermost doll $\mathcal{M}_{\text{self}}$ is the doll of the neighborhood of size one, applied to the backbone forecast first ($k = 0$ in Equation 2). It is a linear bottleneck, shared by all channels, that maps a channel's own lookback and forecast to a residual. On a backbone that does not normalize its input, it reads them in the raw frame instead of the token of Equation 4 and returns a raw residual (Appendix C.2). Since this doll mixes nothing, there is no cross-channel evidence to test.

Fine doll. The fine doll mixes the channels within each qualified fine group g in three steps. First, it encodes the token of every channel into a latent $\boldsymbol{\ell}_i = \mathrm{LayerNorm}(\mathbf{E}_{[w]}\mathbf{c}_i) \in \mathbb{R}^w$ (Ba et al., 2016). Second, every channel receives a weighted sum of the latents of its group, with signed message weights $\Pi_{g,ij}$. Third, a zero-initialized head $\mathbf{F}$ maps the result to the residual:

$$\mathbf{\Pi}_g = \mathbf{K}_g + \mathbf{\Delta}_g + \tanh\Big(\mathbf{Q}_{[w]}\boldsymbol{\ell}\,(\mathbf{U}_{[w]}\boldsymbol{\ell})^\top/\sqrt{w}\Big), \tag{5}$$

$$\mathbf{h}_i = \mathrm{GELU}\Big(\mathbf{M}_{[w]}\sum\nolimits_{j\in g}\Pi_{g,ij}\,\boldsymbol{\ell}_j + \mathbf{O}_{[w]}\boldsymbol{\ell}_i\Big), \qquad \mathcal{M}_{\text{fine}}(\cdot \mid g)_i = \mathbf{F}_{[w]}\,\mathbf{h}_i, \tag{6}$$

where $\boldsymbol{\ell}$ stacks the latents of the group, $\mathbf{M}$ reads the messages, and $\mathbf{O}$ reads the channel's own latent. The message weights have three parts. The kernel $\mathbf{K}_g$ is fixed at calibration and holds the row-normalized directed correlations at the pilot horizon between the target of channel i and the history of channel j, so the doll starts from the lead-lag structure of its group. The refinement $\mathbf{\Delta}_g$ is one free matrix per group, which training adjusts at the forecast horizon, so the kernel is a starting point rather than a constraint. The $\tanh$ term, whose maps $\mathbf{Q}$ and $\mathbf{U}$ are shared by all groups, adapts the weights to each window, whereas $\mathbf{K}_g$ and $\mathbf{\Delta}_g$ are the same for all windows. Both $\mathbf{\Delta}_g$ and the $\tanh$ term start at zero, as does the head $\mathbf{F}$, so the doll adds nothing at initialization.

The subscript $[w]$ denotes the first w rows and columns of a shared parameter. Groups whose sizes round up to the same power of two are batched together and use $w = \lceil W_{\text{fine}}\, n_g/g_{\max}\rceil$, where n_g is the size of the largest of them and W_{fine} is the full width of the fine doll. Smaller neighborhoods thus use smaller dolls made of the same weights (Kusupati et al., 2022). A singleton has no group and bypasses the fine doll. The width is also nested in the training data: it is capped in proportion to the number of training windows per forecast step of the channels that a doll serves, and a doll whose cap is too small is not built (Table 3). The same cap bounds the innermost and global dolls.

Coarse doll. The coarse doll gives every channel of a qualified coarse neighborhood G context from it at linear cost, which matters most for singletons and small groups. This doll has no pairwise kernel. Instead, every channel reads the mean latent of the neighborhood, its *core*, next to its own:

$$\mathbf{h}_i = \mathrm{GELU}\Big(\mathbf{M}_{[w]}\tfrac{1}{|G|}\sum\nolimits_{j\in G}\boldsymbol{\ell}_j + \mathbf{O}_{[w]}\boldsymbol{\ell}_i\Big), \qquad \mathcal{M}_{\text{coarse}}(\cdot \mid G)_i = \mathbf{F}_{[w]}\,\mathbf{h}_i, \tag{7}$$

where $\boldsymbol{\ell}_i$ is the latent of channel i, encoded from the token of the current forecast as in the fine doll, $\mathbf{M}$ reads the core, $\mathbf{O}$ reads the channel's own latent, and $\mathbf{F}$ is a zero-initialized head. The core is the same vector for every channel of the neighborhood, so the mixing costs $O(Cw)$ instead of the $O(Cg_{\max}w)$ of a pairwise kernel. The coarse doll follows the same prefix rule with its own parameters and full width W_{coarse}, and with $2g_{\max}$ in place of $g_{\max}$.

Global doll. At the global level, every channel reads a summary of all channels next to its own latent. The summary is attention-pooled with the channel's own weights $\alpha_{i\cdot} = \mathrm{softmax}_j\big((\mathbf{Q}\boldsymbol{\ell}_i)^\top(\mathbf{U}\boldsymbol{\ell}_j)/\sqrt{w}\big)$ over all channels j:

$$\mathbf{h}_i = \mathrm{GELU}\Big(\mathbf{M}\sum\nolimits_{j\in[C]}\alpha_{ij}\,\boldsymbol{\ell}_j + \mathbf{O}\,\boldsymbol{\ell}_i\Big), \qquad \mathcal{M}_{\text{global}}(\cdot)_i = \mathbf{F}\,\mathbf{h}_i, \tag{8}$$

where $\mathbf{Q}$ and $\mathbf{U}$, its own, map latents to queries and keys, and $\mathbf{M}$, $\mathbf{O}$, and $\mathbf{F}$ play the roles of Equation 7. The queries start at zero, so every channel starts by reading the plain mean of all channels and the doll starts in the form of a coarse doll over all channels. Training then lets each channel attend to its own subset of channels. This is the one component of Matryoshka that forms a $C \times C$ tensor, and it is built only when the global level qualifies.

3.5 Theoretical Analysis

Two results ground the design, proved in Appendix B. Let f^*_G be the best linear predictor of $\mathbf{Y}_i$ from the $q|G|$ lagged values of a neighborhood $G \ni i$, and let $\Lambda_i(G) = R_i(f^*_G) - R_i([C])$ be its linear omitted information, which is non-increasing along any chain $\{i\} = G_0 \subset \cdots \subset G_d$ of tree nodes.

Theorem 1 (Estimation cost of a neighborhood in a Gaussian reference)**.** *Fix a chain of the tree independently of n_{fit} i.i.d. fitting origins, let the lagged predictors of G_d and the target be jointly Gaussian with known zero means and a positive definite predictor covariance, and fit each node G by ordinary least squares without an intercept on its $p_G = q|G|$ predictors. If $n_{\mathrm{fit}} > p_{G_d} + 1$, then for every node of the chain,*

$$\mathbb{E}\big[R_i(\hat{f}_G)\big] - R_i([C]) = \Lambda_i(G) + v_i(G)\,\frac{p_G}{n_{\mathrm{fit}} - p_G - 1}, \tag{9}$$

where $v_i(G)$ is the residual variance of f^*_G summed over the H outputs, and the expectation is over the fitting origins.

Along the chain, the omitted information can only fall, whereas the factor $p_G/(n_{\mathrm{fit}} - p_G - 1)$ of the estimation term grows, so a larger neighborhood pays off only when the information it adds exceeds this cost. This motivates testing each neighborhood instead of mixing by default, and nesting the widths of the dolls in the data. The result concerns a linear reference and certifies neither the learned hierarchy, which is fitted on the same data, nor the neural dolls (Appendix B.2).

For the second result, let $\hat{r}(S)$ be the smaller block statistic of Equation 3, also minimized over the two targets where the horizon test applies, and $r(S)$ its population counterpart. A test is a candidate at one of its targets, $D \le 6C^2$ counts the channel-test pairs, $n_{\min}$ is the smallest held-out block, and a channel is active when its own-lag risk exceeds a numerical threshold ζ.

Theorem 2 (Uniform accuracy of the relative-gain pilot)**.** *Condition on the fitting segment and on everything computed from it, and let Assumption 1 hold: the origins of every held-out block are i.i.d. draws, the squared losses of the pilot have sub-gamma tails with constants (ν, β), every own-lag risk is either zero or at least s_0 away from ζ, every test has an active channel, and on the active channels the own-lag risk is at least m_0 and the candidate's risk at most Q times it. Let $\eta_\delta = \nu\sqrt{2t_\delta/n_{\min}} + 2\beta t_\delta/n_{\min}$ with $t_\delta = \log(8D/\delta)$ for $\delta \in (0,1)$, and suppose that $\eta_\delta < \min\{s_0, m_0\}$. Then with probability at least $1-\delta$, simultaneously for all candidates S,*

$$|\hat{r}(S) - r(S)| \le \varepsilon_\delta := \frac{(1+Q)\,\eta_\delta}{m_0 - \eta_\delta}. \tag{10}$$

The theorem is what makes the tree restriction pay off: the radius ε_δ depends on the channel count only through $\log D \le \log(6C^2)$, whereas a test over arbitrary subsets of channels would pay $\log 2^C$, linear in C, and a structure learned end to end admits no such bound. At our block sizes the bound is conservative, and the horizon test rests on a weaker form of its assumptions, so the theorem justifies the form of the gate rather than its margin (Appendix B.3).

4 Experiments

4.1 Experimental Setup

Datasets. We evaluate on twelve multivariate benchmarks whose channel counts span two orders of magnitude: the four ETT datasets with 7 channels (Zhou et al., 2021), Weather with 21, Solar with 137 (Lai et al., 2018), Electricity with 321, Traffic with 862 (Wu et al., 2021), PEMS03 with 358, PEMS04 with 307, PEMS07 with 883, and PEMS08 with 170 channels (Liu et al., 2022). The datasets thus range from at most 21 channels (the ETT datasets and Weather) to hundreds of channels (the other seven). All models use a lookback of $L = 96$ steps, with horizons $H \in \{96, 192, 336, 720\}$ ($H \in \{12, 24, 48, 96\}$ for PEMS) on the unified benchmark and $H = 48$ on the CI/CD benchmark. Appendix D provides the splits, preprocessing, protocols, and hyperparameters.

Backbones. Every plug-in is evaluated on the five backbones of the unified benchmark and on the Linear and Transformer backbones of the CI/CD benchmark (Section 4.3). DLinear (Zeng et al., 2023), PatchTST (Nie et al., 2023), and TSMixer (Chen et al., 2023) are the backbones on which LIFT (Zhao & Shen, 2024) and CCM (Chen et al., 2024) were evaluated. iTransformer (Liu et al., 2024) is the standard channel-mixing backbone, and DUET (Qiu et al., 2025b) is a recent state-of-the-art model with a channel strategy of its own. The five backbones cover the three channel strategies of Qiu et al. (2025a): DLinear and PatchTST are Channel Independent (CI), TSMixer and iTransformer are Channel Dependent (CD), and DUET is Channel Partial (CP) with a learned channel mask. Each backbone is a port of its official implementation with official per-dataset settings.

Baselines. We compare against the CP plug-ins, LIFT and CCM, using their published defaults without per-cell tuning (Appendix D). PRReg (Han et al., 2024) is a training recipe rather than a mixing module: it trains a CD model on last-value residuals with a weight decay selected on validation. We report it on the CI/CD benchmark, where it was proposed. The code of xCPD is not released, so we compare with its published numbers in the cells that its authors report.

4.2 Main Results

Table 1 reports MSE and MAE averaged over the four horizons, and Appendix F gives every horizon.

Table 1: Unified benchmark with lookback $L = 96$: MSE and MAE averaged over the four horizons, for every backbone alone and with each plug-in. ETT averages the four ETT datasets, and Average all twelve. **Bold** marks the best within each backbone. DUET and per-horizon results are in Appendix F.

Models	DLinear		+LIFT		+CCM		+Matryoshka		PatchTST		+LIFT		+CCM		+Matryoshka	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	0.417	0.427	0.460	0.443	**0.411**	**0.424**	0.415	0.428	0.374	**0.392**	0.380	0.397	0.380	0.395	**0.372**	**0.392**
Weather	0.268	0.322	0.250	0.299	0.267	0.318	**0.237**	**0.290**	0.255	0.278	**0.240**	0.275	0.256	0.280	0.246	**0.274**
Solar	0.326	0.398	0.268	0.302	0.327	0.397	**0.205**	**0.271**	0.231	**0.265**	0.225	0.271	0.232	**0.265**	**0.212**	0.276
Electricity	0.210	0.296	0.196	0.286	0.211	0.297	**0.180**	**0.280**	0.188	0.273	0.179	0.272	0.189	0.275	**0.173**	**0.268**
Traffic	0.625	0.384	0.554	0.345	0.625	0.385	**0.480**	**0.300**	**0.442**	**0.278**	0.524	0.310	**0.442**	0.279	0.447	0.289
PEMS03	0.264	0.358	0.158	0.265	0.264	0.358	**0.092**	**0.201**	0.178	0.272	0.122	0.228	0.181	0.276	**0.098**	**0.210**
PEMS04	0.264	0.354	0.169	0.281	0.267	0.358	**0.091**	**0.200**	0.236	0.312	0.132	0.242	0.238	0.316	**0.095**	**0.205**
PEMS07	0.311	0.373	0.175	0.274	0.313	0.375	**0.078**	**0.179**	0.192	0.278	0.127	0.223	0.187	0.272	**0.084**	**0.189**
PEMS08	0.332	0.371	0.201	0.287	0.336	0.384	**0.110**	**0.204**	0.217	0.288	0.179	0.263	0.218	0.289	**0.111**	**0.204**
Average	0.356	0.380	0.318	0.343	0.355	0.381	**0.261**	**0.303**	0.286	0.318	0.271	0.306	0.288	0.319	**0.246**	**0.290**

Models	TSMixer		+LIFT		+CCM		+Matryoshka		iTransformer		+LIFT		+CCM		+Matryoshka	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETT	**0.373**	**0.391**	0.382	0.397	0.377	0.392	0.376	0.392	0.387	0.402	0.400	0.410	0.389	0.405	**0.386**	**0.401**
Weather	0.247	0.276	**0.237**	0.274	0.249	0.280	0.244	**0.272**	0.260	0.281	**0.251**	0.280	0.264	0.284	0.253	**0.278**
Solar	0.242	0.279	0.245	0.284	0.256	0.288	**0.205**	**0.262**	0.243	0.266	0.228	**0.264**	0.245	0.266	**0.207**	**0.264**
Electricity	0.177	0.273	0.172	0.272	0.192	0.290	**0.171**	**0.265**	0.178	0.269	**0.171**	0.270	0.176	0.268	0.172	**0.265**
Traffic	0.512	0.338	0.547	0.342	0.507	0.341	**0.478**	**0.314**	0.409	0.274	0.527	0.340	**0.406**	**0.273**	0.450	0.293
PEMS03	0.138	0.246	0.112	0.226	0.132	0.241	**0.088**	**0.195**	0.135	0.241	0.105	0.211	0.136	0.241	**0.096**	**0.206**
PEMS04	0.121	0.236	0.103	0.218	0.127	0.242	**0.087**	**0.194**	0.109	0.215	0.097	0.203	0.111	0.220	**0.092**	**0.201**
PEMS07	0.113	0.217	0.097	0.198	0.113	0.217	**0.074**	**0.173**	0.097	0.197	**0.075**	**0.170**	0.104	0.205	0.082	0.181
PEMS08	0.201	0.291	0.160	0.265	0.194	0.289	**0.104**	**0.197**	0.165	0.235	0.133	0.216	0.164	0.234	**0.116**	**0.210**
Average	0.270	0.310	0.267	0.306	0.273	0.313	**0.246**	**0.287**	0.262	0.299	0.266	0.299	0.264	0.301	**0.251**	**0.292**

Comparison with the backbones. Matryoshka improves every backbone on average: over the twelve datasets, its MSE is 27%, 14%, 9%, 4%, and 4% lower than that of DLinear, PatchTST, TSMixer, iTransformer, and DUET, respectively (DUET in Appendix F). The gains concentrate on the datasets with hundreds of channels, 22% on average over the five backbones, while on the ETT datasets and Weather, where the gate accepts at most two groups and never the global level (Appendix E), Matryoshka stays within 2% of its backbone. On the PEMS archives, the CI backbones DLinear and PatchTST forecast every sensor from its own history and ignore the spatial structure of the road network, so LIFT and Matryoshka recover much of their error: LIFT lowers their MSE by 18–44% and Matryoshka by 45–75%. On TSMixer and iTransformer, which already mix all channels, the gains of Matryoshka are smaller, 15–48%. Over five seeds at the first horizon, Matryoshka beats DLinear for every seed on every dataset with hundreds of channels, and iTransformer on all but Traffic (Appendix F).

Comparison with the plug-ins. LIFT is the strongest competing plug-in. It lowers the MSE by 9% on average on the datasets with hundreds of channels, but it raises the error in more than half of the backbone-dataset pairs with few channels, by up to 22%, and by 29% on Traffic with iTransformer. LIFT is the better plug-in on Weather for three of the five backbones, and on Traffic no plug-in improves PatchTST, iTransformer, or DUET by more than 1%.

4.3 PLUG-INS ON THE CI/CD BENCHMARK

On average over the twelve datasets of Table 2, Matryoshka lowers the MSE of the CI form by 40% on the Linear backbone and by 33% on the Transformer, against 23% and 15% for PRReg, and it improves the CI form on every dataset. On the seven datasets with hundreds of channels, the reduction grows to 55% and 36% (45% on average) against 31% and 11% for PRReg. LIFT helps only the Linear backbone by 20%, and CCM stays within a few percent of the CI form. Against the published numbers of xCPD, Matryoshka is ahead in six of the eight cells that xCPD reports (Table 16).

Table 2: CI/CD benchmark of Han et al. (2024): MSE at horizon 48 of the Linear and Transformer backbones in their CD and CI forms and with each plug-in. Plug-ins wrap the CI form of each backbone, PRReg its CD form. **Bold** marks the best within each row and backbone, and the last row averages the twelve datasets. Every number is our run. MAE is in Appendix G.

Models	Linear						Transformer					
Dataset	CD	CI	+PRReg	+LIFT	+CCM	+Matryoshka	CD	CI	+PRReg	+LIFT	+CCM	+Matryoshka
ETTh1	0.400	0.342	0.341	0.347	0.353	**0.335**	0.802	0.661	0.441	0.632	0.672	**0.352**
ETTh2	0.662	0.231	0.240	0.259	0.241	**0.223**	1.380	0.275	0.255	0.282	0.300	**0.227**
ETTm1	0.393	0.345	0.306	0.361	0.347	**0.285**	0.417	0.359	0.307	0.370	0.376	**0.290**
ETTm2	0.160	0.150	**0.137**	0.150	0.149	0.143	0.273	0.148	0.137	0.164	0.178	**0.130**
Weather	0.141	0.168	0.132	0.169	0.169	**0.130**	0.230	0.164	0.148	0.180	0.163	**0.133**
Solar	0.192	0.323	0.216	0.258	0.329	**0.142**	**0.153**	0.295	0.160	0.313	0.315	0.190
Electricity	0.442	0.196	0.174	0.175	0.196	**0.126**	0.253	0.202	0.200	0.203	0.196	**0.139**
Traffic	0.777	0.703	0.815	0.552	0.705	**0.426**	0.656	0.589	0.940	0.592	0.601	**0.429**
PEMS03	0.244	0.318	0.145	0.184	0.322	**0.123**	0.150	0.203	**0.115**	0.205	0.205	0.125
PEMS04	0.148	0.323	0.124	0.195	0.327	**0.122**	**0.113**	0.213	0.122	0.212	0.217	0.134
PEMS07	0.215	0.376	0.126	0.199	0.379	**0.102**	0.176	0.236	**0.117**	0.233	0.250	0.135
PEMS08	0.389	0.384	0.217	0.263	0.391	**0.146**	0.267	0.280	0.152	0.277	0.294	**0.136**
Average	0.347	0.322	0.248	0.259	0.326	**0.192**	0.406	0.302	0.258	0.305	0.314	**0.202**

4.4 ANALYSIS

Ablations. Added to DLinear one level at a time, the fine, coarse, and global dolls lower the average MSE on the datasets with hundreds of channels from 0.333 to 0.199, 0.193, and 0.177, and each of them lowers the MSE on all seven datasets. We also ablate the evidence gate, diffusion hierarchy, and nested widths (Appendix E, Tables 6 and 7).

Cost and efficiency. LIFT is the slowest plug-in on every backbone on the five datasets with more than 300 channels. To select the leading indicators of each channel, it cross-correlates every pair of channels over all lags at every training step, so its cost grows with the square of the number of channels. From PEMS03 to PEMS07, which has 2.5 times as many channels, the running time of DLinear with LIFT grows 5.7 times, close to the square of 2.5, whereas with Matryoshka, which fixes its channel structure before training, it grows 2.1 times (Appendix E, Figure 4).

5 CONCLUSION

We revisited channel mixing as a selection problem with held-out evidence. At the population level, other channels never increase the optimal risk, so a failure of mixing is a failure to recover their information, and Matryoshka tests each neighborhood out of sample before spending nested, zero-initialized capacity on it. It is the only plug-in that improves every backbone on average. On datasets with hundreds of channels, it lowers the MSE by 22% on the unified benchmark and by 45% on the CI/CD benchmark. On the few-channel datasets of the unified benchmark, it stays within 2% of its backbone, whereas LIFT hurts more than half of the backbone-dataset pairs. Its main limitation points to future work. The pilot measures evidence against a channel's own history rather than against what a channel-dependent backbone already captures, so on iTransformer and DUET, which mix channels themselves, the dolls lower the MSE by only 4%, against 27% on DLinear.

Reproducibility Statement

Section 3 describes Matryoshka, Appendix A completes the construction of the channel hierarchy, and Appendix C gives the calibration details and the design rules. Table 3 lists every hyperparameter of the plug-in, fixed across backbones, datasets, and horizons. Appendix D describes the datasets, splits, and protocol of the unified benchmark, and Appendix G the setting of the CI/CD benchmark of Han et al. (2024). Appendix F reports the variation of the results over five seeds. All datasets are public (Zhou et al., 2021; Lai et al., 2018; Wu et al., 2021; Liu et al., 2022). The theoretical results are stated with their assumptions in Section 3.5 and proved in Appendix B. We provide the source code with all scripts and hyperparameters as supplementary material.

AI use statement

In this work, we used large language models (LLMs) to assist with writing, including polishing the text for grammar and clarity. We have not used generative AI tools to generate research ideas. All AI-assisted text was reviewed before being included in the paper. The problem formulation, the proposed method, and the theoretical results were developed by the authors.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Jialin Chen, Jan Eric Lenssen, Aosong Feng, Weihua Hu, Matthias Fey, Leandros Tassiulas, Jure Leskovec, and Rex Ying. From similarity to superiority: Channel clustering for time series forecasting. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://openreview.net/forum?id=MDgn9aazo0.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=wbpxTuXgm0.

Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, 2006.

Devvrit, Sneha Kudugunta, Aditya Kusupati, Tim Dettmers, Kaifeng Chen, Inderjit Dhillon, Yulia Tsvetkov, Hannaneh Hajishirzi, Sham Kakade, Ali Farhadi, and Prateek Jain. MatFormer: Nested transformer for elastic inference. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Yuchen Fang, Shiyu Wang, Yuxuan Liang, Zhou Ye, Yang Xiang, Yan Zhao, and Kai Zheng. Efficient high-dimensional time series forecasting with transformers: A channel reordering perspective. In *Proceedings of the ACM Web Conference 2026*, 2026. doi: 10.1145/3774904.3792392.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985.

Clive W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The capacity and robustness trade-off: Revisiting the channel independent strategy for multivariate time series forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7129–7142, 2024. doi: 10.1109/TKDE.2024.3400008.

Daniel Hsu, Sham M. Kakade, and Tong Zhang. Random design analysis of ridge regression. *Foundations of Computational Mathematics*, 14(3):569–600, 2014.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Aditya Kusupati, Gantavya Bhatt, Aniket Rege, Matthew Wallingford, Aditya Sinha, Vivek Ramanujan, William Howard-Snyder, Kaifeng Chen, Sham Kakade, Prateek Jain, and Ali Farhadi. Matryoshka representation learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30233–30249, 2022.

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.

Seunghan Lee, Taeyoung Park, and Kibok Lee. Partial channel dependence with channel masks for time series foundation models. In *NeurIPS Workshop on Time Series in the Age of Large Models*, 2024. URL https://openreview.net/forum?id=xVU6rY37X9.

Dongyuan Li, Shun Zheng, Chang Xu, Jiang Bian, and Renhe Jiang. Routing channel-patch dependencies in time series forecasting with graph spectral decomposition. In *International Conference on Learning Representations*, 2026.

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.

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

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

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, 2019.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards comprehensive and fair benchmarking of time series forecasting methods. *Proc. VLDB Endow.*, 17(9):2363–2377, 2024.

Xiangfei Qiu, Hanyin Cheng, Xingjian Wu, Junkai Lu, Jilin Hu, Chenjuan Guo, Christian S. Jensen, and Bin Yang. A comprehensive survey of deep learning for multivariate time series forecasting: A channel strategy perspective. arXiv preprint arXiv:2502.10721, 2025a.

Xiangfei Qiu, Xingjian Wu, Yan Lin, Chenjuan Guo, Jilin Hu, and Bin Yang. DUET: Dual clustering enhanced multivariate time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 1185–1196, 2025b. doi: 10.1145/3690624.3709325.

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

Alex Tank, Ian Covert, Nicholas Foti, Ali Shojaie, and Emily B. Fox. Neural Granger causality. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(8):4267–4279, 2022.

Changyuan Tian, Zhicong Lu, Zequn Zhang, Hao Yang, Wei Cao, Zhi Guo, Xian Sun, and Li Jin. HyperMixer: Specializable hypergraph channel mixing for long-term multivariate time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 20885–20893, 2025. doi: 10.1609/aaai.v39i19.34302.

Xue Wang, Tian Zhou, Qingsong Wen, Jinyang Gao, Bolin Ding, and Rong Jin. CARD: Channel aligned robust blend transformer for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MJksrOhurE.

Christopher K. I. Williams and Matthias Seeger. Using the Nyström method to speed up kernel machines. In *Advances in Neural Information Processing Systems*, volume 13, pp. 682–688, 2000.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in Neural Information Processing Systems*, 34:22419–22430, 2021.

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2D-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pp. 1907–1913, 2019.

Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 753–763, 2020. doi: 10.1145/3394486.3403118.

Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *The Annals of Probability*, 22(1):94–116, 1994.

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.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023.

Lifan Zhao and Yanyan Shen. Rethinking channel dependence for multivariate time series forecasting: Learning from leading indicators. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JiTVtCUOpS.

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 CHANNEL HIERARCHY

This appendix completes the construction of the channel hierarchy in Section 3.2.

Calibration data. Let $t_1 < \cdots < t_n$ be the n calibration origins and $\mathcal{L}$ the set of q lags. For every origin and channel, the predictors are the lagged values $\{x_{j,t-\lambda}\}_{\lambda\in\mathcal{L}}$, and the pilot target is $y_{j,t} = x_{j,t+h}$. The fitting segment consists of the earliest n_{fit} origins, and the two held-out blocks B_1 and B_2 of n_b origins each follow it. We purge the fitting origins whose target reaches the first held-out origin. The values of these constants are listed in Table 3, and Appendix E studies their sensitivity.

Landmarks. Comparing all channel pairs is quadratic in C. Therefore, the geometry describes every channel by its affinities to J landmark channels. The landmarks are chosen by farthest-point sampling (Gonzalez, 1985) on random sketches of the channels' histories and targets (Johnson & Lindenstrauss, 1984). They are a coordinate system rather than a set of cluster centers: landmarks taken from the densest region would give almost every channel the same affinities, whereas landmarks that cover the channel set separate them.

Affinities. The affinity between a channel j and a landmark k asks whether the two move together, whether the history of j predicts the future of k, and whether the history of k predicts the future of j, and keeps the largest absolute correlation of the three,

$$a_{jk} = \Big(\max\big\{ |\rho(x_j, x_k)|,\ \max_{\lambda\in\mathcal{L}} |\rho(x_{j,-\lambda}, y_k)|,\ \max_{\lambda\in\mathcal{L}} |\rho(y_j, x_{k,-\lambda})| \big\} - \xi \Big)_+^2, \tag{11}$$

where ρ denotes the correlation over the fitting segment and ξ is a finite-sample noise floor of order $1/\sqrt{n_{\mathrm{fit}}}$. The correlations of independent series scatter around zero at the scale of $1/\sqrt{n_{\mathrm{fit}}}$, so the floor discards them and leaves an affinity matrix through which the walk cannot diffuse along spurious links. The lagged terms make the geometry lead-lag aware: two channels are close if one helps to predict the other, and not only if they move together.

Diffusion coordinates. Let $\mathbf{A} \in \mathbb{R}^{C\times J}$ be the affinity matrix and $\mathbf{B} = \mathbf{D}_C^{-1/2}\mathbf{A}\mathbf{D}_J^{-1/2}$ be its degree-normalized form. The two degree matrices hold the row and the column sums of $\mathbf{A}$, so that a channel that attaches to every landmark does not attract the whole geometry and a landmark that many channels attach to does not dominate the coordinates. The channel coordinates are $\mathbf{\Psi} = \mathbf{B}\mathbf{V} \in \mathbb{R}^{C\times d_{\mathrm{e}}}$ with $d_{\mathrm{e}} \ll C$, where $\mathbf{V}$ contains the leading eigenvectors of the $J \times J$ matrix $\mathbf{B}^\top\mathbf{B}$ that follow the leading one, which carries only the degrees of the channels. These are the coordinates of the symmetrically normalized two-step walk channel $\to$ landmark $\to$ channel, which differ from the diffusion coordinates of Coifman & Lafon (2006) by a degree scaling. They are obtained from a landmark-sized eigenproblem (Williams & Seeger, 2000), without forming a $C \times C$ matrix (Lemma 1). The bisection of Section 3.2 then splits each node at the median of the coordinate of $\mathbf{\Psi}$ with the largest variance within the node. The geometry depends on H only through h, so all horizons with the same pilot horizon share one hierarchy and differ only in which nodes qualify.

Lemma 1 (Landmark factorization)**.** *Let $\mathbf{A} \in \mathbb{R}_{\geq 0}^{C\times J}$ have positive row and column sums, and let $\mathbf{B} = \mathbf{D}_C^{-1/2}\mathbf{A}\mathbf{D}_J^{-1/2}$ as above. If $\mathbf{B}^\top\mathbf{B}\mathbf{v} = \lambda\mathbf{v}$ with $\lambda > 0$, then $\mathbf{B}\mathbf{B}^\top(\mathbf{B}\mathbf{v}) = \lambda\mathbf{B}\mathbf{v}$ and $\|\mathbf{B}\mathbf{v}\|^2 = \lambda\|\mathbf{v}\|^2$, and $\mathbf{B}\mathbf{B}^\top$ and $\mathbf{B}^\top\mathbf{B}$ have the same nonzero eigenvalues. Thus the coordinates of the $C \times C$ operator $\mathbf{B}\mathbf{B}^\top$ are computed from a $J \times J$ eigenproblem without forming it.*

Proof. The eigenvector and norm identities follow by matrix multiplication, and the equality of the nonzero spectra follows from the singular value decomposition of $\mathbf{B}$. □

B THEORETICAL RESULTS AND PROOFS

This section gives the population identity behind predictive gain and proves the two results of Section 3.5. The Gaussian reference explains the trade-off between additional predictors and fitting error. The concentration result explains how a small candidate family supports held-out qualification. The results motivate the design without asserting optimality of the learned hierarchy or of the neural dolls.

B.1 Population Risk and Fitted Gain

At an origin drawn from a fixed distribution P, let $\mathbf{X}_S$ be the lookback of channels $S \subseteq [C]$ and $\mathbf{Y}_i \in \mathbb{R}^H$ the target of channel i, with finite second moments. For a square-integrable predictor f using $\mathbf{X}_S$, define

$$R_i(f) = \mathbb{E}_P\|\mathbf{Y}_i - f(\mathbf{X}_S)\|^2, \qquad m_i(S) = \mathbb{E}_P[\mathbf{Y}_i \mid \mathbf{X}_S], \qquad R_i(S) = R_i(m_i(S)). \tag{12}$$

For a fitted predictor, risk is evaluated on a fresh origin with the fitted function held fixed.

Lemma 2 (Monotone Bayes risk)**.** *For $i \in S \subseteq T \subseteq [C]$,*

$$R_i(S) - R_i(T) = \mathbb{E}_P\|m_i(T) - m_i(S)\|^2 \geq 0. \tag{13}$$

Proof. Expand $\mathbf{Y}_i - m_i(S) = (\mathbf{Y}_i - m_i(T)) + (m_i(T) - m_i(S))$. The second term is measurable with respect to $\mathbf{X}_T$, whereas the first has conditional mean zero given $\mathbf{X}_T$, so the expected cross term vanishes. □

Remark 1 (Predictive gain and correlation)**.** *Write $\Gamma_i(S) = R_i(\{i\}) - R_i(S)$. If $\mathbf{X}_{\{j\}} = \mathbf{X}_{\{i\}} + \boldsymbol{\epsilon}$ with $\boldsymbol{\epsilon}$ independent of $(\mathbf{X}_{\{i\}}, \mathbf{Y}_i)$, then $\Gamma_i(\{i,j\}) = 0$, even when the channels are highly correlated. Conversely, for independent standard Gaussian scalars X_i, X_j and $Y_i = X_j^2 - 1$, the correlation between X_j and Y_i is zero but $\Gamma_i(\{i,j\}) = 2$. The gain measures additional conditional-mean predictive value under P. It does not guarantee that a fitted model recovers that value or that it persists under distribution shift.*

What the linear pilot measures. Let $\boldsymbol{\varphi}_G$ collect the same q lags of each channel in $G \ni i$, let f_G^* be the population least-squares predictor from these features, and let $\hat{f}_G$ belong to the same linear predictor class. An intercept may be included in both classes, and the Gaussian reference below, with known zero means, omits it. Define $\Lambda_i(G) = R_i(f_G^*) - R_i([C])$ and $e_i(G) = R_i(\hat{f}_G) - R_i(f_G^*) \geq 0$. Then

$$\Lambda_i(G) = \underbrace{R_i(G) - R_i([C])}_{\text{omitted channels}} + \underbrace{R_i(f_G^*) - R_i(G)}_{\text{lag and model restrictions}}, \tag{14}$$

$$R_i(\hat{f}_G) - R_i([C]) = \Lambda_i(G) + e_i(G). \tag{15}$$

The linear classes are nested, so $\Lambda_i(G)$ is non-increasing as G grows, whereas $e_i(G)$, which includes regularization error for ridge fits, need not be monotone. Subtracting the second identity for $\{i\}$ and G gives the population difference between their fitted risks:

$$R_i(\hat{f}_{\{i\}}) - R_i(\hat{f}_G) = \Lambda_i(\{i\}) - \Lambda_i(G) + e_i(\{i\}) - e_i(G). \tag{16}$$

Thus a positive fitted gain need not imply $\Gamma_i(G) > 0$, and a linear pilot can miss nonlinear predictive gains.

B.2 Gaussian Reference: Proof of Theorem 1

Proof. Under the theorem's assumptions, fix a node G, and write $n = n_{\text{fit}}$, $p = p_G$, and $\boldsymbol{\Sigma} = \mathbb{E}_P[\boldsymbol{\varphi}_G \boldsymbol{\varphi}_G^\top]$. Joint Gaussianity gives $\mathbf{Y}_i = W_G^* \boldsymbol{\varphi}_G + \boldsymbol{\epsilon}_G$, where the residual is mean zero, independent of the features, and has covariance trace $v_i(G)$. For the fitting design matrix $\boldsymbol{\Phi} \in \mathbb{R}^{n \times p}$, ordinary least squares therefore satisfies

$$\mathbb{E}_{\text{fit}}[e_i(G) \mid \boldsymbol{\Phi}] = v_i(G)\,\mathrm{tr}\big(\boldsymbol{\Sigma}(\boldsymbol{\Phi}^\top \boldsymbol{\Phi})^{-1}\big)\,. \tag{17}$$

Write $\boldsymbol{\Phi} = \mathbf{Z}\boldsymbol{\Sigma}^{1/2}$ with independent standard Gaussian entries in $\mathbf{Z}$. The trace becomes $\mathrm{tr}((\mathbf{Z}^\top \mathbf{Z})^{-1})$. Each diagonal entry of this inverse is the reciprocal squared length of one column after projection off the other $p-1$ columns. That squared length is χ^2_{n-p+1}-distributed, with reciprocal expectation $1/(n-p-1)$ for $n > p+1$. Hence $\mathbb{E}_{\text{fit}}[e_i(G)] = v_i(G)p/(n-p-1)$, and Equation 15 gives Equation 9. □

When a larger neighborhood helps. In this reference, $v_i(G) = R_i(f_G^*)$, so $\mathbb{E}_{\text{fit}}[R_i(\hat{f}_G)] = (n-1)v_i(G)/(n-p_G-1)$. For two nodes $S \subset T$, the larger node has lower expected fitted risk exactly when

$$v_i(S) - v_i(T) > v_i(S)\frac{p_T - p_S}{n - p_S - 1}. \tag{18}$$

The dimension factor $p_G/(n-p_G-1)$ increases with size, but the estimation term itself need not increase because $v_i(G)$ can decrease. If $n \geq 2(p_{G_d}+1)$, it admits the linear upper bound

$$v_i(G)\frac{p_G}{n-p_G-1} \leq 2v_i(\{i\})\frac{q|G|}{n}. \tag{19}$$

These statements concern a fixed chain independent of the fitting sample and unregularized Gaussian regression. They motivate checking whether additional predictors help, but do not establish optimality of the top-down traversal or derive the neural width rule. The learned hierarchy and ridge pilots require additional analysis for a fitting-risk guarantee (see Hsu et al., 2014, for random-design ridge regression).

B.3 Held-Out Qualification: Proof of Theorem 2

The score. Let $\mathcal{F}$ contain the fitting data and randomness used to construct the hierarchy, choose target steps, fit preprocessing, and fit all candidate pilots. Conditionally on $\mathcal{F}$, the entire family of potential tests is fixed, including nodes that the traversal may not visit. A test $a = (S, u)$ pairs a candidate S (including its scale) with one of its required targets u. Write $D = \sum_a |S|$. At most $3C$ candidates and two targets give $D \leq 6C^2$. We consider a nonempty test family and two nonempty evaluation blocks $B_{a,1}, B_{a,2}$ for each test.

For channel $i \in S$, let $L_{i,a}^0$ and $L_{i,a}^1$ be the squared losses of the own-lag and candidate predictors, averaged over the steps of target u in the pilot's normalization frame. Let $\mu_{i,a}^v = \mathbb{E}_P[L_{i,a}^v \mid \mathcal{F}]$ denote risk on a fresh origin and $\hat{\mu}_{i,a,b}^v$ its empirical mean on block b, for $v \in \{0,1\}$. The active sets are $I_a = \{i \in S : \mu_{i,a}^0 > \zeta\}$ and $\hat{I}_{a,b} = \{i \in S : \hat{\mu}_{i,a,b}^0 > \zeta\}$, with $\zeta = 10^{-8}$. Define

$$\begin{aligned} r_a &= \frac{1}{|I_a|}\sum_{i \in I_a} \psi_a\left(1 - \frac{\mu_{i,a}^1}{\mu_{i,a}^0}\right), \\ \hat{r}_{a,b} &= \frac{1}{|\hat{I}_{a,b}|}\sum_{i \in \hat{I}_{a,b}} \psi_a\left(1 - \frac{\hat{\mu}_{i,a,b}^1}{\hat{\mu}_{i,a,b}^0}\right), \end{aligned} \tag{20}$$

where $\psi_a(z) = z$ for fine and coarse tests and $\psi_a(z) = \max\{z, -1\}$ for the global test. An empty empirical active set gives score $-\infty$. For the required targets $\mathcal{U}(S)$, set

$$r(S) = \min_{u \in \mathcal{U}(S)} r_{(S,u)}, \qquad \hat{r}(S) = \min_{u \in \mathcal{U}(S),\, b \in \{1,2\}} \hat{r}_{(S,u),b}. \tag{21}$$

A tested candidate qualifies when $\hat{r}(S) \geq \tau$. Equation 3 is the special case of the block score with all channels active and no clipping.

Assumption 1 (Evaluation data, loss tails, and denominators)**.** *Conditionally on $\mathcal{F}$, the following hold. (B1) Each evaluation block consists of i.i.d. fresh draws from P. (B2) There are $\nu, \beta > 0$ such that, for every i, a, v and $|\lambda| < 1/\beta$,*

$$\log \mathbb{E}_P\left[\exp\{\lambda(L_{i,a}^v - \mu_{i,a}^v)\} \mid \mathcal{F}\right] \leq \frac{\lambda^2\nu^2}{2(1-\beta|\lambda|)}. \tag{22}$$

(B3) There are $s_0, m_0 > 0$ and $Q \geq 0$ such that every own-lag risk is either zero or satisfies $|\mu_{i,a}^0 - \zeta| \geq s_0$. Each I_a is nonempty, and $\mu_{i,a}^0 \geq m_0$ and $\mu_{i,a}^1 \leq Q\mu_{i,a}^0$ for $i \in I_a$.

The tail condition is stronger than finite second moments. The denominator conditions prevent unstable ratios and changes in the active sets near the threshold. Independence between channels, tests, or the two blocks is not required.

Proof of Theorem 2. By (B1)–(B2) and the exponential Markov inequality, each loss mean satisfies

$$\Pr\big(|\hat{\mu}^{v}_{i,a,b} - \mu^{v}_{i,a}| \geq x \mid \mathcal{F}\big) \leq 2\exp\left(-\frac{|B_{a,b}|x^2}{2(\nu^2 + \beta x)}\right), \qquad x > 0. \tag{23}$$

With $n_{\min} = \min_{a,b} |B_{a,b}|$, the theorem's radius satisfies $n_{\min}\eta_\delta^2 \geq 2t_\delta(\nu^2 + \beta\eta_\delta)$. A union bound over two losses, two blocks, and D channel-test pairs bounds the probability of any error exceeding η_δ by $8De^{-t_\delta} = \delta$.

On the complementary event, $\eta_\delta < s_0$ keeps every nonzero baseline risk on the same side of ζ. A zero baseline risk has zero loss almost surely, so $\hat{I}_{a,b} = I_a$. For an active channel, suppressing indices, $\hat{\mu}^0 \geq m_0 - \eta_\delta > 0$ and

$$\left|\frac{\hat{\mu}^1}{\hat{\mu}^0} - \frac{\mu^1}{\mu^0}\right| \leq \frac{|\hat{\mu}^1 - \mu^1| + (\mu^1/\mu^0)|\hat{\mu}^0 - \mu^0|}{\hat{\mu}^0} \leq \frac{(1+Q)\eta_\delta}{m_0 - \eta_\delta} = \varepsilon_\delta. \tag{24}$$

Both choices of ψ_a are 1-Lipschitz. Averaging over the common active set and taking minima preserve the uniform error bound, proving Equation 10. □

On this event, every qualifying candidate satisfies $r(S) \geq \tau - \varepsilon_\delta$, and every candidate with $r(S) \geq \tau + \varepsilon_\delta$ qualifies if tested. In particular, if $\tau > \varepsilon_\delta$, no candidate with $r(S) \leq 0$ qualifies. The uniform bound covers the adaptive traversal, with candidate-count dependence through $\log D \leq \log(6C^2)$. The same argument applies to any family fixed before independent evaluation, and the tree supplies a small such family. The controlled quantity is the average, possibly clipped, fitted-pilot score, which does not ensure a gain for every channel or a reduction in aggregate MSE.

Scope for the implemented calibration. The theorem is an independent-evaluation reference, and its unknown constants do not certify the empirical margin $\tau = 0.01$. At our block sizes the bound is conservative: $n_{\min} \approx 410$ origins, $C = 862$, and $\delta = 0.05$ give $t_\delta \approx 20$, so for losses of unit scale η_δ alone is about 0.3, and ε_δ is larger still, far above τ. The default margin therefore filters out clearly unhelpful candidates rather than certifying marginal ones. Actual time-series origins are dependent. Purging overlapping targets does not establish (B1), and a dependent-data guarantee would require further assumptions, for example mixing conditions (Yu, 1994). In the current forecast-horizon test, evaluation also begins inside the prefix used to fit the geometry and preprocessing, so for the horizon test the theorem holds only under this weaker separation. A common fitting boundary before every target's evaluation blocks would remove this reuse, but temporal dependence would still require separate treatment. Accordingly, the bound explains the role of a restricted candidate family and a positive margin, without guaranteeing improvement of the trained dolls or persistence under distribution shift.

C Implementation Details

C.1 Matryoshka Hyperparameters

Table 3 lists the hyperparameters of the plug-in with the values used in all experiments. The values were fixed during development (Appendices C.2 and E) and then frozen for all backbones, datasets, and horizons. None is tuned per cell.

Calibration details. The n calibration origins are spread evenly over the training split, from the first origin with a full lookback to the last whose pilot target lies inside the split. At each origin, the predictors and targets are centered on the mean of the lookback window and then standardized with the statistics of the fitting segment. The horizon test repeats this construction with targets at eight evenly spaced steps up to the forecast horizon. It uses only the origins whose farthest target lies inside the training split, and it takes its own 60/40 split of them, purging the fitting origins whose farthest target reaches its held-out blocks. The landmark sketches of Appendix A are a fixed Gaussian projection of each channel's fitting-segment history and pilot target onto 32 dimensions, normalized before farthest-point sampling. The kernel $\mathbf{K}_g$ of a fine group is computed on the fitting segment, with the diagonal zeroed and each row normalized by the sum of its absolute values. The groups, the coarse neighborhoods, the global switch, and the kernels are stored in the checkpoint,

Table 3: Hyperparameters of Matryoshka, fixed across backbones, datasets, and horizons. Appendix C.2 describes the two design rules, for the frame of the innermost doll and the horizon test of the global level.

Hyperparameter	Symbol	Value
Calibration and hierarchy		
Calibration origins, spread over the training split	n	2048
Fitting segment and each of the two held-out blocks	n_{fit}, n_b	$0.6n$, $0.2n$
Pilot horizon	h	$\min(H, \lfloor L/4 \rfloor)$
Lags of the pilot predictors ($q = 4$)	$\mathcal{L}$	$\{0, 1, 2, 4\}$
Landmarks	J	16
Diffusion coordinates	d_{e}	8
Noise floor of the affinities	ξ	$\min(0.9, 2/\sqrt{n_{\mathrm{fit}}})$
Qualification		
Maximum fine-group size; coarse neighborhoods have size in $(g_{\max}, 2g_{\max}]$	$g_{\max}$	16
Required relative gain in both held-out blocks	τ	0.01
Ridge penalty of the pilots, relative to the mean diagonal of the Gram matrix	–	0.1
Dolls		
Width of the innermost doll, a linear bottleneck	W_{self}	256
Full width of the fine doll	W_{fine}	512
Full width of the coarse doll	W_{coarse}	128
Width of the global doll	W_{global}	512
Data ratio of the width cap $Nn_{\mathrm{srv}}/(\kappa H)$, for N training windows and n_{srv} served channels (2 on the CI/CD benchmark)	κ	8
Smallest width of a doll that is built	$w_{\min}$	8
Dropout, scaled by $\min(1, g_{\max}/n_{\mathrm{srv}})$	–	0.1
Upper bound on the learning rate of the dolls; the backbone keeps its own	–	10^{-3}

so that evaluation rebuilds the routing without the training data. The landmarks and the geometry cost $O(CJn_{\mathrm{fit}})$, the qualification fits ridge problems with at most $2qg_{\max}$ predictors for at most $3C$ candidates, and the global pilot fits C ridge problems with $2q$ predictors. Therefore, the calibration takes a fraction of a second on the ETT datasets and a few seconds on Traffic on one CPU.

C.2 Configuration and Design Rules

All experiments use one configuration of Matryoshka, fixed across backbones, datasets, and horizons, with the values of Table 3. A forward pass runs the backbone and then applies the innermost, fine, coarse, and global dolls of Equation 2 in turn, each reading tokens recomputed from the current forecast. Groups whose sizes round up to the same power of two are packed into one padded tensor, and the padding is masked out of messages and outputs. If no doll is built at any level, the plug-in returns the backbone forecast unchanged. Two rules cover what the pilot cannot test: the frame of the innermost doll and the horizon of the global level.

The innermost doll's frame follows the backbone's normalization. When the backbone has no input normalization, the innermost doll reads raw values, with bias terms. When the backbone normalizes each window itself, the innermost doll reads the lookback frame of the other dolls. On a normalizing backbone, a raw-frame residual can fit level structure that does not survive a shift between periods, whereas in the lookback frame it sees each window relative to its own level, as the backbone does.

The global level passes the forecast-horizon test. On datasets with few channels, the global pilot can pass at the pilot horizon but fail at every forecast horizon, and a global doll built on the pilot-horizon evidence alone then hurts the long horizons. Therefore, the global level passes the same horizon test as a fine group, which it does on the datasets with hundreds of channels. The

coarse level is tested at the pilot horizon alone, because the horizon test would also remove coarse neighborhoods that help at every horizon.

D Experimental Setup

Datasets. Table 4 lists the twelve public datasets. Their splits are chronological: 12/4/4 months for the ETT datasets, 60/20/20 for the PEMS archives as in iTransformer (Liu et al., 2024), and 70/10/20 for the others. On the PEMS archives, we use the traffic flow of every sensor. The scaler of every dataset is fitted on its training split.

Table 4: Datasets.

Dataset	Channels	Resolution	Time steps	Domain
ETTh1, ETTh2	7	1 hour	17,420	electricity transformer temperature
ETTm1, ETTm2	7	15 minutes	69,680	electricity transformer temperature
Weather	21	10 minutes	52,696	meteorological indicators
Solar	137	10 minutes	52,560	photovoltaic power production
Electricity	321	1 hour	26,304	electricity consumption
Traffic	862	1 hour	17,544	road occupancy
PEMS03	358	5 minutes	26,208	traffic flow
PEMS04	307	5 minutes	16,992	traffic flow
PEMS07	883	5 minutes	28,224	traffic flow
PEMS08	170	5 minutes	17,856	traffic flow

Backbones and baseline plug-ins. Every backbone and baseline plug-in is a port of its official implementation with its official hyperparameters. The plug-ins wrap the same backbone as the unwrapped run and are trained jointly with it.

Protocol details. All models use a lookback of 96 steps and the horizons 96, 192, 336, and 720 of TFB (Qiu et al., 2024), or 12, 24, 48, and 96 on the PEMS archives, as in iTransformer. Every run follows the optimizer, schedule, and loss of its backbone, with early stopping on the validation loss, and a plug-in and its backbone share the same seed. We report the MSE and MAE on the test split. The experiments are implemented in PyTorch (Paszke et al., 2019) and run on NVIDIA Quadro RTX 6000 GPUs.

E Analysis Details

Accepted structure. Table 5 lists the structure that the calibration accepts on every dataset. It follows the number of channels. On the ETT datasets, the gate accepts at most two pairs of channels, and on ETTh2 and ETTm2 it accepts nothing, so Matryoshka reduces to its innermost doll. On Weather, it accepts one fine group and one coarse neighborhood, and on none of these datasets does it accept the global level. On the datasets with hundreds of channels, the fine groups cover 84–100% of the channels, and the coarse neighborhoods and the global level are active.

Channel embeddings. Figure 3 shows the accepted structure in the channel embedding of Electricity, Traffic, and the PEMS archives. Even in this embedding, the PEMS archives stand out as the most structured. There, the gate accepts almost every candidate, and the fine groups and coarse neighborhoods cover 97–100% of the channels, so almost no channel is drawn in gray. On Electricity and Traffic, the gate accepts only about 40% of the fine candidates, more than a tenth of the channels stay in no fine group, and the coarse neighborhoods leave out a quarter of the channels of Electricity and almost half of those of Traffic. The global level confirms this difference. Its gain is 0.11 to 0.17 on the PEMS archives against 0.05 on Electricity and 0.06 on Traffic, and a tenth of the PEMS channels gain more than 0.29 from the channel mean, against 0.11 on Electricity and Traffic. Each PEMS archive records the traffic flow at the sensors of one freeway network in California every 5 minutes. Vehicles that pass one sensor reach the next ones within minutes, so the lags of a sensor predict its neighbors, and all sensors of a network follow the same rush hours, which the mean of

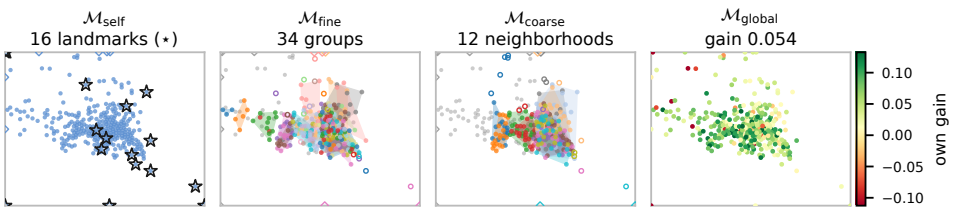
Table 5: Structure accepted by the calibration, which reads only the training split and is shared by all backbones: the fine groups of more than one channel, the coarse neighborhoods, the channels each level covers, and whether the global level is active. Each row is the horizon with the largest fine coverage; the structure changes by at most a few groups across horizons.

Dataset	C	Fine		Coarse		Global
		Groups	Covered	Neighborhoods	Covered	Active
ETTh1	7	2	4/7	0	0/7	no
ETTh2	7	0	0/7	0	0/7	no
ETTm1	7	1	2/7	0	0/7	no
ETTm2	7	0	0/7	0	0/7	no
Weather	21	1	11/21	1	21/21	no
Solar	137	16	137/137	8	137/137	yes
Electricity	321	34	271/321	12	241/321	yes
Traffic	862	95	753/862	17	458/862	yes
PEMS03	358	33	355/358	16	358/358	yes
PEMS04	307	32	307/307	16	307/307	yes
PEMS07	883	66	882/883	31	856/883	yes
PEMS08	170	16	170/170	8	170/170	yes

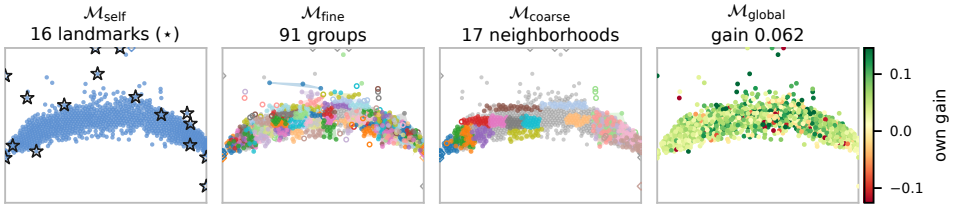
the channels captures. Traffic also records road sensors, but hourly, and at that resolution a lead of a few minutes falls within a single time step.

Roles of the dolls. Each doll adds information that no other doll provides. The innermost doll corrects each channel from its own lookback and forecast. It is the neural counterpart of the own-lag baseline against which the pilot tests every neighborhood, so the outer dolls only need to add what the pilot finds beyond a channel's own history. It is also the only doll for the channels that belong to no accepted neighborhood, such as every channel of ETTh2 and ETTm2 (Table 5). The fine doll is the only one whose messages are specific to each pair of channels, starting from the calibrated lead-lag structure of its group. Such messages cost $O(n^2)$ in a group of n channels, and the estimation cost of a neighborhood grows with its size (Theorem 1), so this level is limited to groups of at most $g_{\max}$ channels. The coarse and global dolls instead read a summary of their scope, so their capacity does not grow with its size. The coarse doll gives every channel of a neighborhood a summary of it at linear cost. On Weather, where the global level is rejected, it is the only cross-channel context of the ten channels outside the fine group. The global doll is the only one that reaches all channels, and it captures factors shared by the whole dataset. It is built only on the datasets with hundreds of channels, where it also serves the channels that no fine group covers, such as 109 of the 862 channels of Traffic.

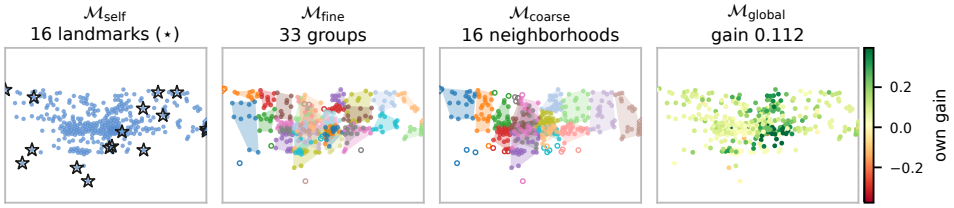
Ablations. Table 6 ablates Matryoshka on DLinear on the seven datasets with hundreds of channels, with all other settings as in the reported runs. The first block adds the levels of dolls one at a time. On DLinear, the innermost doll alone adds no capacity, since a linear doll on a linear backbone is still a linear map, so this row stays within 0.002 of DLinear. The fine dolls bring most of the gain and lower the average MSE from 0.333 to 0.199, and the coarse and global dolls lower it further to 0.193 and 0.177. Each of these three levels lowers the MSE on every one of the seven datasets. The second block replaces one component of Matryoshka at a time. Without the evidence gate, every candidate of the tree and the global level are built. The random tree applies the median bisection to random coordinates instead of the diffusion embedding, so the gate tests random groups of the same sizes. With full widths, every doll takes its full width, whatever the size of its neighborhood and the amount of training data. Each replacement raises the average MSE, to 0.181 without the gate, 0.184 with the random tree, and 0.181 with full widths, and none of them lowers the MSE on any of the seven datasets. The largest differences arise on Traffic and Electricity, where the gate rejects most fine candidates. On Traffic, the MSE rises from 0.480 to 0.503 without the gate and to 0.512 with the random tree, although more random groups pass the gate, covering 809 channels against 747 at horizon 96. On Electricity, it rises from 0.180 to 0.186 and 0.184. The PEMS archives separate these designs far less. Their sensors lie on connected roads and follow the same traffic, so their channels carry information about each other. The gate therefore groups almost every channel under either tree, and even random groups help, so the gate and the random tree raise the MSE there by at most



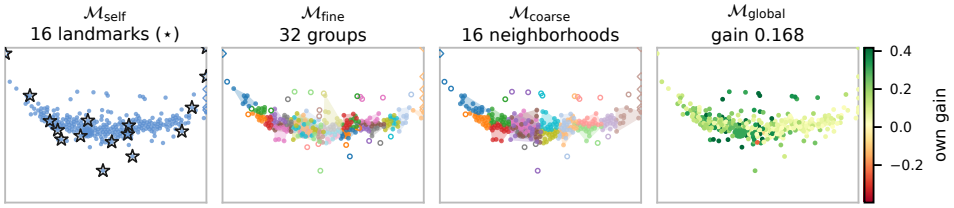
(a) Electricity (321 channels)



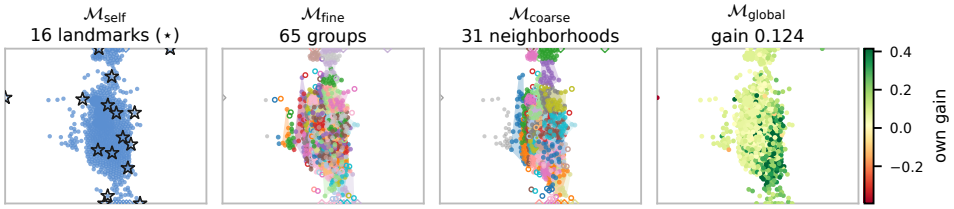
(b) Traffic (862 channels)



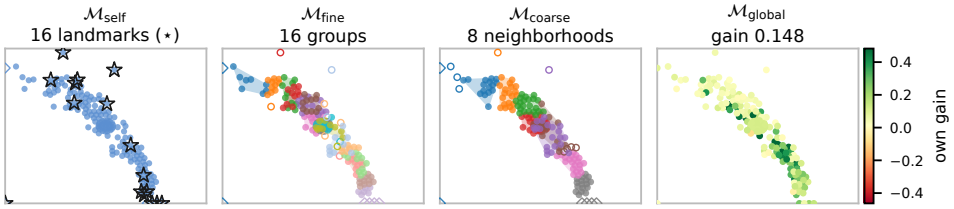
(c) PEMS03 (358 channels)



(d) PEMS04 (307 channels)



(e) PEMS07 (883 channels)



(f) PEMS08 (170 channels)

Figure 3: Structure accepted by the calibration at horizon 96, from the training split only. Each point is a channel in the plane of its diffusion embedding, and channels beyond the plotted range sit on its edge as diamonds. The first panel marks the landmarks as stars. The second and third show the fine groups and the coarse neighborhoods, each in its own color as the area of its core members, with its other members as open circles and the channels outside every group in gray. The last panel colors each channel by the gain of the channel-mean lags over its own lags, the test of the global level.

0.004. Traffic is thus the dataset on which the gate and the diffusion hierarchy matter most. Nested widths are also the cheaper design. Their fine and coarse dolls use 19–65% fewer parameters than at full width, 43% fewer on average, and process each window with 13–46% less computation. The extra capacity of full widths does not pay off. They raise the MSE on all seven datasets, most on Solar, from 0.205 to 0.216.

Table 6: Ablations of Matryoshka on DLinear: MSE and MAE averaged over the four horizons on the seven datasets with hundreds of channels, where the calibration builds every doll. The first block adds one level of dolls at a time. The second replaces one component of Matryoshka: every candidate is built without the evidence gate, the bisection splits random coordinates instead of the diffusion embedding (random tree), or every doll takes its full width whatever the size of its neighborhood and the training data (full widths). The last row is Matryoshka. The last column averages the seven datasets. **Bold** marks the best in each column.

Dataset	Solar		Electricity		Traffic		PEMS03		PEMS04		PEMS07		PEMS08		Average	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	0.326	0.398	0.210	0.296	0.625	0.384	0.264	0.358	0.264	0.354	0.311	0.373	0.332	0.371	0.333	0.362
+Innermost	0.327	0.398	0.210	0.296	0.625	0.384	0.264	0.358	0.264	0.356	0.311	0.374	0.331	0.373	0.333	0.363
+Innermost+Fine	0.214	0.268	0.193	0.288	0.526	0.327	0.113	0.226	0.111	0.225	0.106	0.211	0.130	0.225	0.199	0.253
+Innermost+Fine+Coarse	0.207	**0.267**	0.191	0.288	0.522	0.329	0.103	0.215	0.105	0.218	0.097	0.201	0.128	0.217	0.193	0.248
No evidence gate	**0.205**	0.271	0.186	0.287	0.503	0.305	0.094	0.203	**0.091**	**0.200**	0.080	0.182	0.111	0.206	0.181	0.236
Random tree	0.208	0.271	0.184	0.284	0.512	0.311	0.095	0.203	0.093	0.201	0.081	0.182	0.114	0.206	0.184	0.237
Full widths	0.216	0.274	0.183	0.284	0.485	**0.300**	0.093	0.203	0.095	0.205	0.082	0.184	0.116	0.210	0.181	0.237
+Matryoshka	**0.205**	0.271	**0.180**	**0.280**	**0.480**	**0.300**	**0.092**	**0.201**	**0.091**	**0.200**	**0.078**	**0.179**	**0.110**	**0.204**	**0.177**	**0.234**

Table 6 understates the role of the gate, because it covers only the datasets with hundreds of channels, where the fine groups already cover 84–100% of the channels. On the ETT datasets and Weather, the gate accepts at most two groups and never the global level (Table 5), whereas without it every channel is mixed and the global level is built. Table 7 shows that this harms the forecast. Without the gate, the average MSE across the five datasets rises from 0.379 to 0.421 on DLinear and from 0.349 to 0.370 on TSMixer. Both ungated models also have higher MSE and MAE than their backbone alone on all four ETT datasets and Weather.

Table 7: The evidence gate on the few-channel datasets: MSE and MAE averaged over the four horizons for DLinear and TSMixer, alone, with Matryoshka, and with Matryoshka without the evidence gate, which then mixes every channel and builds the global level. The last column averages the five datasets. **Bold** marks the best in each column for each backbone.

Dataset	ETTh1		ETTh2		ETTm1		ETTm2		Weather		Average	
Metric	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
DLinear	**0.453**	**0.448**	0.495	0.483	0.406	0.412	**0.313**	**0.365**	0.268	0.322	0.387	0.406
No evidence gate	0.546	0.520	0.522	0.494	0.414	0.427	0.350	0.393	0.273	0.329	0.421	0.433
+Matryoshka	0.455	0.452	**0.493**	**0.481**	**0.392**	**0.407**	0.319	0.371	**0.237**	**0.290**	**0.379**	**0.400**
TSMixer	0.439	**0.432**	**0.385**	**0.407**	**0.386**	**0.398**	**0.283**	**0.326**	0.247	0.276	**0.348**	**0.368**
No evidence gate	0.459	0.450	0.408	0.428	0.414	0.423	0.306	0.348	0.265	0.292	0.370	0.388
+Matryoshka	**0.438**	0.433	**0.385**	**0.407**	0.396	0.401	**0.283**	**0.326**	**0.244**	**0.272**	0.349	**0.368**

Number of levels. The bisection tree offers more scales than the four dolls use, one per depth between the fine cap and the root. In development runs, we gave every tree node above the fine cap its own core doll, which replaces the single coarse level by four to six levels on Solar, Electricity, and Traffic. The gate accepted almost all of these levels, yet the MSE averaged over DLinear and PatchTST at horizons 96 and 720 on Weather, Solar, Electricity, and Traffic rose from 0.281 to 0.286, and on Traffic from 0.485 to 0.497. The gate tests each level against the channel's own lags, not against the levels already built, and the summaries of nested nodes at adjacent depths carry nearly the same information. Each such level therefore passes, and together they add redundant residuals

on the same target. The global doll already covers the intermediate scales, since an attention pattern concentrated on a subset of channels reads a summary of that subset. Within a level, the nested prefixes of Section 3.4 already adapt the doll to the size of each neighborhood. We therefore use four dolls, the fewest that cover the roles above, and each of them adds information that no other doll provides.

Gate margin. Table 8 shows how many channels the fine groups cover at margins τ from 0 to 0.05, computed by the calibration alone. On five of the twelve datasets, the structure is the same at every margin, because every candidate there is either clearly useful or clearly not. Without a margin, the gate also accepts weak candidates, such as all seven channels of ETTh1, which made the forecasts worse in our development runs. A strict margin mainly shrinks Electricity and Traffic, to 148 and 450 covered channels at $\tau = 0.05$, which helped on Traffic but hurt on Electricity. We therefore keep $\tau = 0.01$ as a permissive filter.

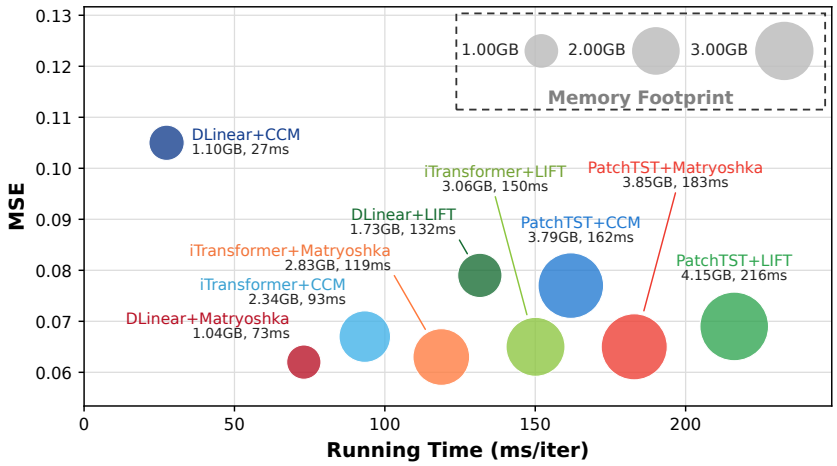
Table 8: The gate margin τ (default 0.01): channels covered by the fine groups of more than one channel, from the calibration alone, at the same horizon as in Table 5, whose values are the column $\tau = 0.01$. The structure is the same at every margin on ETTh2, ETTm2, Solar, PEMS04, and PEMS08.

Dataset	C	Channels covered			
		$\tau = 0$	$\tau = 0.01$	$\tau = 0.02$	$\tau = 0.05$
ETTh1	7	7/7	4/7	0/7	0/7
ETTh2	7	0/7	0/7	0/7	0/7
ETTm1	7	2/7	2/7	2/7	0/7
ETTm2	7	0/7	0/7	0/7	0/7
Weather	21	14/21	11/21	11/21	11/21
Solar	137	137/137	137/137	137/137	137/137
Electricity	321	288/321	271/321	266/321	148/321
Traffic	862	776/862	753/862	705/862	450/862
PEMS03	358	355/358	355/358	355/358	353/358
PEMS04	307	307/307	307/307	307/307	307/307
PEMS07	883	882/883	882/883	880/883	880/883
PEMS08	170	170/170	170/170	170/170	170/170

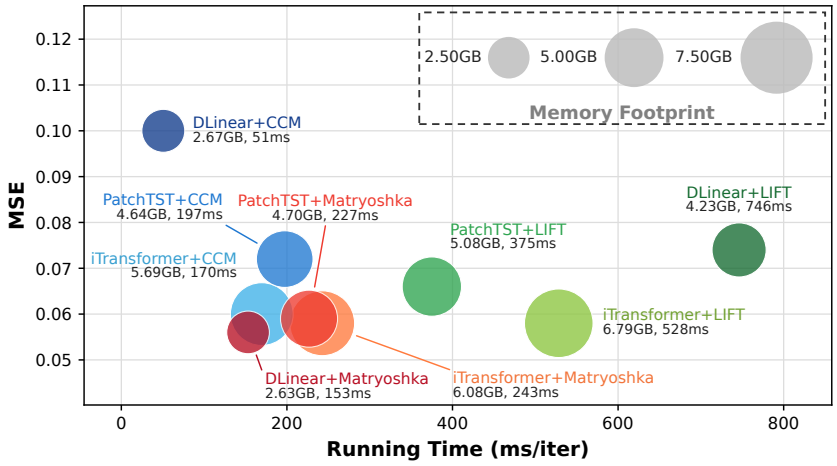
Efficiency. Figure 4 compares the plug-ins on PEMS03 and PEMS07, which have 358 and 883 channels. On both datasets, Matryoshka has the lowest MSE on every backbone, tied with LIFT on iTransformer on PEMS07, whereas CCM is the fastest but the least accurate. LIFT is the slowest plug-in and uses the most memory. To estimate leads, it cross-correlates every pair of channels over all lags at every training step, so its computation grows with the square of the number of channels. From PEMS03 to PEMS07, the number of channels grows 2.5 times, and the running time of DLinear with LIFT grows 5.7 times, from 132 to 746 ms per iteration, whereas with Matryoshka it grows 2.1 times, from 73 to 153 ms. Matryoshka finds its channel structure once, before training, and its dolls apply weights shared across channels to each channel. Only the attention of the global doll compares pairs of channels, so its running time grows far more slowly than that of LIFT.

F Full Results

Seed variation. Table 9 repeats DLinear and iTransformer, alone and with Matryoshka, with four more seeds (2027, 2028, 2029, 2023), at the first horizon of every dataset. The standard deviation of the MSE over the five seeds is at most 0.006 in 45 of the 48 cells and largest on Traffic, 0.015 for DLinear and 0.013 for iTransformer with Matryoshka. On every dataset with hundreds of channels, Matryoshka lowers the MSE of DLinear for every seed, and even its worst seed is better than the best seed of DLinear, for example on Solar, where the mean falls from 0.289 to 0.176. The same holds for iTransformer on all of these datasets except Traffic, the largest loss of Matryoshka at this horizon, which also holds for every seed: 0.424 ± 0.013 with Matryoshka against 0.387 ± 0.001 alone. On the ETT datasets, the two differ by at most 0.004 except on ETTm1, where Matryoshka is better on



(a) PEMS03 (358 channels)



(b) PEMS07 (883 channels)

Figure 4: Accuracy and cost of the plug-ins with lookback 96 and horizon 12: MSE against the running time per training iteration, with the bubble area proportional to the peak GPU memory during training (each panel has its own scale). Each plug-in has its own color, red for Matryoshka, blue for CCM, and green for LIFT, and each backbone its own shade of that color.

both backbones. Averaged over the twelve datasets, Matryoshka lowers the MSE of DLinear from 0.250 ± 0.001 to 0.201 ± 0.002 and that of iTransformer from 0.202 ± 0.000 to 0.199 ± 0.002.

Full Results. Tables 10–14 report MSE and MAE for every backbone, dataset, horizon, and method, with the best result in each row in bold. The results show two regularities that hold across plug-ins. First, the gains concentrate on the datasets with hundreds of channels. On average over the five backbones, Matryoshka changes the MSE of its backbone by -22% on these datasets but by -1% on the ETT datasets and Weather, and LIFT by -9% against $+4\%$. With few channels, the other channels carry little information beyond each channel's own history, so a plug-in has little to add and mainly risks overfitting. Second, the channel-independent backbones benefit most. On the datasets with hundreds of channels, Matryoshka reduces the MSE of DLinear and PatchTST by 47% and 28%, but that of TSMixer and iTransformer, which mix channels themselves, by 20% and 9%, and LIFT follows the same order (26%, 12%, 5%, and 0%). On DUET, which also mixes channels, LIFT reduces the MSE by 2% and Matryoshka by 9%. A backbone that already mixes channels captures part of the cross-channel information itself, which leaves less for a plug-in to add.

Table 9: Variation over seeds at the first horizon of every dataset (96, or 12 on the PEMS archives): MSE and MAE of DLinear and iTransformer, alone and with Matryoshka, as mean and standard deviation over five seeds (2026, 2027, 2028, 2029, and 2030). Seed 2026 is the run of the full tables. The last row averages the twelve datasets for each seed. **Bold** marks the lower mean within each backbone.

Models	DLinear		+Matryoshka		iTransformer		+Matryoshka	
Metrics	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	0.389±0.005	0.404±0.007	**0.386±0.003**	**0.403±0.004**	**0.388±0.001**	0.406±0.001	0.390±0.001	**0.405±0.001**
ETTh2	0.317±0.004	0.375±0.005	**0.313±0.003**	**0.371±0.003**	**0.300±0.003**	**0.350±0.002**	0.302±0.004	0.351±0.002
ETTm1	0.343±0.001	0.372±0.003	**0.328±0.001**	**0.370±0.001**	0.340±0.006	0.376±0.003	**0.334±0.004**	**0.370±0.003**
ETTm2	**0.186±0.005**	**0.277±0.011**	0.187±0.003	0.280±0.007	0.186±0.001	0.272±0.002	**0.183±0.001**	**0.268±0.002**
Weather	0.198±0.003	0.263±0.006	**0.161±0.005**	**0.223±0.005**	0.177±0.003	0.218±0.003	**0.169±0.001**	**0.213±0.002**
Solar	0.289±0.002	0.377±0.002	**0.176±0.006**	**0.250±0.005**	0.211±0.010	**0.236±0.002**	**0.182±0.006**	0.247±0.006
Electricity	0.195±0.000	0.277±0.000	**0.144±0.003**	**0.244±0.003**	0.148±0.001	0.240±0.001	**0.141±0.000**	**0.236±0.000**
Traffic	0.648±0.000	0.396±0.000	**0.458±0.015**	**0.285±0.001**	**0.387±0.001**	**0.262±0.000**	0.424±0.013	0.283±0.003
PEMS03	0.105±0.000	0.219±0.000	**0.062±0.001**	**0.165±0.002**	0.068±0.000	0.172±0.001	**0.062±0.000**	**0.165±0.000**
PEMS04	0.115±0.000	0.228±0.000	**0.070±0.000**	**0.173±0.000**	0.077±0.001	0.180±0.001	**0.071±0.001**	**0.173±0.000**
PEMS07	0.100±0.000	0.215±0.001	**0.056±0.001**	**0.151±0.001**	0.060±0.000	0.157±0.001	**0.057±0.001**	**0.153±0.002**
PEMS08	0.112±0.000	0.223±0.000	**0.072±0.001**	**0.168±0.001**	0.078±0.000	0.179±0.001	**0.075±0.003**	**0.176±0.004**
Average	0.250±0.001	0.302±0.001	**0.201±0.002**	**0.257±0.001**	0.202±0.000	0.254±0.000	**0.199±0.002**	**0.253±0.001**

G CI/CD BENCHMARK

Setup. The linear and Transformer backbones, in their CD and CI forms, are ports of the official implementation of Han et al. (2024), and we follow its published experimental setting, including the learning rates and batch sizes of its scripts. Tests load the official code, copy weights, and compare forecasts and parameter gradients. We reproduced all CD, CI, and PRReg experiments with this implementation. On the linear backbone, our results differ from the published numbers by 1.3% on average, and on the CI Transformer by 3%. The CD Transformer stops after one or two epochs. The plug-ins wrap the CI form of each backbone.

PRReg. PRReg (Han et al., 2024) is not a channel-mixing module but a training strategy for the CD form: it predicts the residual after last-value centering and regularizes the model with weight decay. As in the official implementation, it is trained with seven weight-decay values, and we report the one with the lowest validation loss.

Results. Table 15 reports MSE and MAE for both backbones. Matryoshka is the best method in all but five of the 24 cells of Table 2, and all five are lost to the CD form or to PRReg, which trains it. On Solar and PEMS04, the CD Transformer, which mixes all channels in every layer, is ahead of every plug-in on the CI form. On ETTm2, PRReg is ahead of Matryoshka on the linear backbone by 0.006. Table 16 compares with the published numbers of xCPD (Li et al., 2026) on the four datasets it reports, where xCPD is ahead of Matryoshka only on Weather with the linear backbone (0.128 against 0.130) and on ETTm1 with the Transformer (0.289 against 0.290).

Table 10: Full results on DLinear with lookback $L = 96$: MSE and MAE for every dataset and horizon, and their average over the four horizons (Avg). **Bold** marks the best in each row.

Models		DLinear		+LIFT		+CCM		+Matryoshka		Models		DLinear		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.388	**0.402**	0.394	0.408	0.398	0.414	**0.387**	0.407	Electricity	96	0.195	0.278	0.173	0.263	0.195	0.278	**0.146**	**0.245**
	192	**0.433**	**0.427**	0.442	0.436	0.438	0.435	0.440	0.437		192	0.194	0.280	0.180	0.270	0.195	0.282	**0.166**	**0.266**
	336	**0.480**	**0.457**	0.507	0.474	0.486	0.460	**0.480**	**0.457**		336	0.207	0.296	0.195	**0.287**	0.209	0.299	**0.189**	0.289
	720	**0.512**	**0.507**	0.531	0.510	0.526	0.519	**0.512**	0.508		720	0.242	0.329	0.236	0.322	0.243	0.329	**0.220**	**0.319**
	Avg	**0.453**	**0.448**	0.469	0.457	0.462	0.457	0.455	0.452		Avg	0.210	0.296	0.196	0.286	0.211	0.297	**0.180**	**0.280**
ETTh2	96	0.322	0.381	0.338	0.377	**0.303**	**0.355**	0.313	0.372	Traffic	96	0.649	0.396	0.555	0.347	0.648	0.398	**0.454**	**0.284**
	192	0.432	0.445	0.445	0.455	0.432	0.444	**0.424**	**0.439**		192	0.599	0.372	0.536	0.337	0.600	0.372	**0.451**	**0.295**
	336	0.516	0.499	0.532	0.499	**0.475**	**0.475**	0.525	0.506		336	0.605	0.373	0.543	0.338	0.605	0.374	**0.468**	**0.296**
	720	0.710	0.605	1.110	0.710	**0.664**	**0.576**	0.711	0.605		720	0.646	0.395	0.582	0.357	0.646	0.396	**0.547**	**0.325**
	Avg	0.495	0.483	0.606	0.510	**0.469**	**0.463**	0.493	0.481		Avg	0.625	0.384	0.554	0.345	0.625	0.385	**0.480**	**0.300**
ETTm1	96	0.344	0.371	0.348	0.381	0.346	0.377	**0.329**	**0.370**	PEMS03	12	0.105	0.219	0.079	0.188	0.105	0.220	**0.062**	**0.165**
	192	0.384	0.394	0.388	0.404	0.386	0.396	**0.367**	**0.388**		24	0.183	0.296	0.118	0.232	0.184	0.300	**0.076**	**0.184**
	336	0.417	0.420	0.421	0.420	0.415	0.416	**0.401**	**0.410**		48	0.319	0.409	0.185	0.292	0.317	0.407	**0.099**	**0.212**
	720	0.480	0.462	0.488	**0.459**	0.482	0.461	**0.472**	0.460		96	0.449	0.506	0.249	0.346	0.450	0.506	**0.130**	**0.244**
	Avg	0.406	0.412	0.411	0.416	0.407	0.413	**0.392**	**0.407**		Avg	0.264	0.358	0.158	0.265	0.264	0.358	**0.092**	**0.201**
ETTm2	96	**0.181**	**0.266**	0.188	0.282	0.182	0.271	0.183	0.273	PEMS04	12	0.115	0.228	0.091	0.202	0.115	0.229	**0.070**	**0.173**
	192	0.258	0.332	0.294	0.357	**0.251**	**0.322**	0.269	0.347		24	0.189	0.299	0.130	0.246	0.189	0.299	**0.080**	**0.187**
	336	0.322	0.379	0.362	0.408	0.324	0.379	**0.321**	**0.372**		48	0.322	0.405	0.200	0.316	0.322	0.405	**0.099**	**0.211**
	720	0.489	0.484	0.566	0.511	**0.470**	**0.472**	0.501	0.492		96	0.429	0.484	0.256	0.360	0.443	0.500	**0.116**	**0.229**
	Avg	0.313	0.365	0.353	0.390	**0.307**	**0.361**	0.319	0.371		Avg	0.264	0.354	0.169	0.281	0.267	0.358	**0.091**	**0.200**
Weather	96	0.201	0.269	0.170	0.230	0.200	0.261	**0.156**	**0.218**	PEMS07	12	0.100	0.214	0.074	0.181	0.100	0.214	**0.056**	**0.151**
	192	0.241	0.301	0.210	0.271	0.234	0.291	**0.209**	**0.270**		24	0.190	0.302	0.121	0.233	0.193	0.304	**0.068**	**0.169**
	336	0.283	0.334	0.268	0.319	0.285	0.336	**0.258**	**0.308**		48	0.376	0.434	0.216	0.311	0.380	0.440	**0.087**	**0.192**
	720	0.347	0.384	0.351	0.374	0.349	0.385	**0.326**	**0.363**		96	0.577	0.541	0.287	0.369	0.579	0.540	**0.100**	**0.203**
	Avg	0.268	0.322	0.250	0.299	0.267	0.318	**0.237**	**0.290**		Avg	0.311	0.373	0.175	0.274	0.313	0.375	**0.078**	**0.179**
Solar	96	0.287	0.376	0.240	0.289	0.283	0.369	**0.183**	**0.255**	PEMS08	12	0.111	0.222	0.095	0.201	0.113	0.228	**0.072**	**0.167**
	192	0.315	0.393	0.260	0.304	0.321	0.398	**0.205**	**0.283**		24	0.195	0.301	0.144	0.251	0.200	0.304	**0.092**	**0.185**
	336	0.349	0.412	0.277	0.310	0.349	0.412	**0.208**	**0.266**		48	0.382	0.424	0.224	0.315	0.400	0.453	**0.113**	**0.214**
	720	0.354	0.410	0.296	0.303	0.353	0.410	**0.222**	**0.279**		96	0.639	0.537	0.339	0.380	0.630	0.549	**0.163**	**0.250**
	Avg	0.326	0.398	0.268	0.302	0.327	0.397	**0.205**	**0.271**		Avg	0.332	0.371	0.201	0.287	0.336	0.384	**0.110**	**0.204**

Table 11: Full results on PatchTST with lookback $L = 96$: MSE and MAE for every dataset and horizon, and their average over the four horizons (Avg). **Bold** marks the best in each row.

Models		PatchTST		+LIFT		+CCM		+Matryoshka		Models		PatchTST		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.372**	**0.394**	0.379	0.400	0.383	0.397	0.374	0.395	Electricity	96	0.160	0.247	0.152	0.245	0.161	0.248	**0.138**	**0.236**
	192	0.431	0.425	0.436	0.436	**0.429**	0.432	**0.429**	**0.424**		192	0.170	0.257	0.164	0.257	0.174	0.261	**0.161**	**0.255**
	336	**0.472**	**0.450**	0.484	0.464	0.473	**0.450**	0.475	0.452		336	0.190	0.277	0.180	0.276	0.188	0.277	**0.176**	**0.271**
	720	**0.509**	**0.489**	0.578	0.521	0.521	0.500	**0.509**	0.490		720	0.230	0.312	0.220	**0.310**	0.231	0.314	**0.218**	**0.310**
	Avg	**0.446**	**0.440**	0.469	0.455	0.452	0.445	0.447	**0.440**		Avg	0.188	0.273	0.179	0.272	0.189	0.275	**0.173**	**0.268**
ETTh2	96	**0.289**	**0.340**	0.291	0.341	0.302	0.347	**0.289**	**0.340**	Traffic	96	0.416	**0.264**	0.524	0.323	**0.414**	0.265	0.419	0.276
	192	0.373	0.394	0.379	0.394	0.382	0.398	**0.370**	**0.390**		192	0.434	0.276	0.496	0.279	0.428	**0.270**	**0.426**	0.276
	336	0.419	0.428	0.418	0.429	0.424	0.432	**0.416**	**0.427**		336	**0.442**	**0.276**	0.529	0.332	0.449	0.283	0.448	0.292
	720	0.444	0.456	**0.426**	**0.444**	0.451	0.459	0.430	0.449		720	**0.476**	**0.295**	0.547	0.305	0.478	0.297	0.495	0.311
	Avg	0.381	0.405	0.379	**0.402**	0.390	0.409	**0.376**	**0.402**		Avg	**0.442**	**0.278**	0.524	0.310	**0.442**	0.279	0.447	0.289
ETTm1	96	**0.321**	**0.360**	0.325	0.362	0.347	0.372	0.322	0.362	PEMS03	12	0.075	0.182	0.069	0.174	0.077	0.185	**0.065**	**0.170**
	192	0.360	**0.381**	0.362	0.385	0.365	0.384	**0.359**	0.382		24	0.114	0.222	0.093	0.202	0.118	0.228	**0.083**	**0.195**
	336	0.400	**0.404**	0.399	0.411	**0.396**	**0.404**	0.400	0.407		48	0.201	0.297	0.137	0.244	0.201	0.297	**0.106**	**0.220**
	720	0.453	**0.436**	0.457	0.446	0.457	0.441	**0.448**	0.440		96	0.321	0.385	0.188	0.292	0.328	0.392	**0.139**	**0.255**
	Avg	0.384	**0.395**	0.386	0.401	0.391	0.400	**0.382**	0.398		Avg	0.178	0.272	0.122	0.228	0.181	0.276	**0.098**	**0.210**
ETTm2	96	**0.177**	**0.259**	0.183	0.263	**0.177**	**0.259**	0.178	0.260	PEMS04	12	0.095	0.200	0.081	0.188	0.099	0.205	**0.072**	**0.176**
	192	**0.247**	**0.305**	0.251	0.308	0.248	0.306	**0.247**	**0.305**		24	0.146	0.250	0.103	0.211	0.146	0.250	**0.084**	**0.194**
	336	0.303	0.341	0.303	0.341	**0.302**	**0.340**	0.303	0.341		48	0.259	0.339	0.149	0.262	0.260	0.342	**0.105**	**0.219**
	720	0.405	0.401	**0.403**	**0.398**	0.413	0.404	0.405	0.400		96	0.444	0.458	0.196	0.305	0.447	0.465	**0.119**	**0.231**
	Avg	**0.283**	**0.327**	0.285	0.328	0.285	**0.327**	**0.283**	**0.327**		Avg	0.236	0.312	0.132	0.242	0.238	0.316	**0.095**	**0.205**
Weather	96	0.173	0.215	**0.161**	0.209	0.173	0.215	**0.161**	**0.206**	PEMS07	12	0.073	0.177	0.066	0.165	0.072	0.174	**0.059**	**0.157**
	192	0.218	0.255	**0.204**	0.252	0.220	0.257	0.209	**0.249**		24	0.118	0.223	0.092	0.194	0.116	0.220	**0.073**	**0.176**
	336	0.276	0.297	**0.258**	**0.291**	0.276	0.298	0.268	0.294		48	0.219	0.308	0.139	0.242	0.215	0.303	**0.090**	**0.197**
	720	0.353	**0.346**	**0.337**	**0.346**	0.354	0.348	0.347	**0.346**		96	0.358	0.404	0.210	0.291	0.346	0.391	**0.115**	**0.226**
	Avg	0.255	0.278	**0.240**	0.275	0.256	0.280	0.246	**0.274**		Avg	0.192	0.278	0.127	0.223	0.187	0.272	**0.084**	**0.189**
Solar	96	0.198	**0.237**	0.206	0.264	0.206	0.247	**0.182**	0.248	PEMS08	12	0.083	0.183	0.080	0.181	0.085	0.187	**0.072**	**0.166**
	192	0.233	0.266	0.227	0.273	0.229	**0.261**	**0.207**	0.276		24	0.130	0.230	0.138	0.242	0.130	0.229	**0.095**	**0.198**
	336	0.244	0.279	**0.231**	**0.273**	0.244	0.274	0.240	0.302		48	0.242	0.320	0.166	0.266	0.249	0.331	**0.110**	**0.206**
	720	0.247	0.276	0.234	**0.274**	0.247	0.278	**0.219**	0.277		96	0.414	0.417	0.331	0.363	0.408	0.410	**0.165**	**0.244**
	Avg	0.231	**0.265**	0.225	0.271	0.232	**0.265**	**0.212**	0.276		Avg	0.217	0.288	0.179	0.263	0.218	0.289	**0.111**	**0.204**

Table 12: Full results on TSMixer with lookback $L = 96$: MSE and MAE for every dataset and horizon, and their average over the four horizons (Avg). **Bold** marks the best in each row.

Models		TSMixer		+LIFT		+CCM		+Matryoshka		Models		TSMixer		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.374**	**0.393**	0.387	0.399	0.378	**0.393**	0.377	0.394	Electricity	96	0.147	0.246	0.145	0.245	0.154	0.254	**0.141**	**0.237**
	192	0.429	**0.423**	0.437	0.428	0.432	0.424	**0.424**	**0.423**		192	0.168	0.264	0.163	0.263	0.174	0.273	**0.158**	**0.253**
	336	**0.474**	**0.445**	0.482	0.452	0.479	0.449	**0.474**	0.446		336	0.184	0.280	0.176	0.278	0.195	0.293	**0.174**	**0.269**
	720	**0.478**	**0.468**	0.515	0.491	0.484	0.471	**0.478**	0.469		720	0.208	0.300	**0.202**	0.300	0.245	0.338	0.209	**0.299**
	Avg	0.439	**0.432**	0.455	0.443	0.443	0.434	**0.438**	0.433		Avg	0.177	0.273	0.172	0.272	0.192	0.290	**0.171**	**0.265**
ETTh2	96	0.300	0.345	0.299	0.346	0.301	0.346	**0.298**	**0.344**	Traffic	96	0.505	0.333	0.522	0.337	0.487	0.330	**0.442**	**0.292**
	192	**0.377**	0.396	0.380	**0.395**	0.381	0.396	0.378	0.396		192	0.500	0.337	0.534	0.334	0.488	0.332	**0.458**	**0.309**
	336	0.424	0.435	**0.420**	**0.431**	0.424	0.434	0.424	0.435		336	0.504	0.331	0.551	0.345	0.509	0.344	**0.482**	**0.318**
	720	0.440	0.453	**0.422**	**0.441**	0.441	0.453	0.441	0.454		720	0.539	0.349	0.579	0.350	0.545	0.359	**0.529**	**0.335**
	Avg	0.385	0.407	**0.380**	**0.403**	0.387	0.407	0.385	0.407		Avg	0.512	0.338	0.547	0.342	0.507	0.341	**0.478**	**0.314**
ETTm1	96	**0.318**	**0.357**	0.339	0.371	0.322	0.360	0.323	0.360	PEMS03	12	0.066	0.171	0.063	0.167	0.067	0.172	**0.060**	**0.163**
	192	**0.365**	**0.384**	0.367	0.388	0.368	0.385	0.374	0.386		24	0.092	0.203	0.083	0.195	0.093	0.205	**0.072**	**0.177**
	336	**0.400**	**0.407**	0.402	0.408	0.413	0.415	0.415	0.411		48	0.156	0.271	0.121	0.240	0.152	0.267	**0.095**	**0.204**
	720	**0.461**	**0.443**	0.471	0.447	0.465	0.445	0.471	0.447		96	0.239	0.340	0.180	0.303	0.217	0.320	**0.123**	**0.235**
	Avg	**0.386**	**0.398**	0.395	0.404	0.392	0.401	0.396	0.401		Avg	0.138	0.246	0.112	0.226	0.132	0.241	**0.088**	**0.195**
ETTm2	96	**0.177**	0.260	0.189	0.270	0.180	0.264	**0.177**	**0.259**	PEMS04	12	0.074	0.180	0.072	0.177	0.075	0.183	**0.067**	**0.168**
	192	0.243	0.302	0.250	0.310	**0.239**	**0.300**	0.242	0.302		24	0.091	0.203	0.084	0.195	0.092	0.203	**0.076**	**0.180**
	336	**0.304**	**0.342**	0.332	0.363	0.305	**0.342**	0.305	0.343		48	0.130	0.250	0.111	0.229	0.156	0.278	**0.093**	**0.202**
	720	**0.406**	**0.398**	0.425	0.414	0.411	0.403	0.408	0.399		96	0.190	0.312	0.145	0.270	0.183	0.304	**0.112**	**0.225**
	Avg	**0.283**	**0.326**	0.299	0.339	0.284	0.327	**0.283**	**0.326**		Avg	0.121	0.236	0.103	0.218	0.127	0.242	**0.087**	**0.194**
Weather	96	0.157	0.205	0.156	0.207	0.160	0.210	**0.154**	**0.200**	PEMS07	12	0.059	0.157	0.057	0.154	0.060	0.158	**0.053**	**0.147**
	192	0.206	**0.250**	**0.202**	0.252	0.213	0.259	0.208	0.252		24	0.081	0.185	0.073	0.172	0.084	0.188	**0.066**	**0.163**
	336	0.271	0.300	**0.260**	0.295	0.271	0.302	0.265	**0.292**		48	0.121	0.229	0.104	0.208	0.128	0.239	**0.082**	**0.185**
	720	0.354	0.350	**0.329**	**0.340**	0.351	0.349	0.348	0.345		96	0.190	0.295	0.155	0.256	0.181	0.283	**0.096**	**0.196**
	Avg	0.247	0.276	**0.237**	0.274	0.249	0.280	0.244	**0.272**		Avg	0.113	0.217	0.097	0.198	0.113	0.217	**0.074**	**0.173**
Solar	96	0.195	**0.242**	0.209	0.255	0.218	0.262	**0.189**	0.250	PEMS08	12	0.080	0.186	0.077	0.183	0.080	0.186	**0.067**	**0.166**
	192	0.246	0.282	0.255	0.289	0.250	0.280	**0.201**	**0.245**		24	0.122	0.235	0.103	0.213	0.127	0.243	**0.086**	**0.184**
	336	0.262	0.294	0.263	0.299	0.276	0.304	**0.212**	**0.275**		48	0.219	0.323	0.172	0.289	0.205	0.310	**0.114**	**0.205**
	720	0.266	0.297	0.254	0.292	0.281	0.307	**0.217**	**0.278**		96	0.384	0.418	0.286	0.374	0.365	0.417	**0.148**	**0.231**
	Avg	0.242	0.279	0.245	0.284	0.256	0.288	**0.205**	**0.262**		Avg	0.201	0.291	0.160	0.265	0.194	0.289	**0.104**	**0.197**

Table 13: Full results on iTransformer with lookback $L = 96$: MSE and MAE for every dataset and horizon, and their average over the four horizons (Avg). **Bold** marks the best in each row.

Models		iTransformer		+LIFT		+CCM		+Matryoshka		Models		iTransformer		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	**0.389**	0.407	0.391	0.407	0.393	0.412	0.391	**0.406**	Electricity	96	0.148	0.239	**0.140**	0.238	0.148	0.240	0.141	**0.236**
	192	0.446	0.439	**0.441**	**0.436**	0.443	0.442	0.445	0.437		192	0.167	0.258	**0.160**	0.259	0.166	0.256	0.162	**0.254**
	336	**0.478**	0.455	0.528	0.484	0.491	0.466	**0.478**	**0.454**		336	0.180	0.273	**0.171**	0.274	0.179	0.273	0.174	**0.269**
	720	0.508	0.494	0.685	0.583	0.508	0.494	**0.496**	**0.488**		720	0.218	0.307	0.213	0.310	**0.212**	0.304	**0.212**	**0.302**
	Avg	0.455	0.449	0.511	0.478	0.459	0.454	**0.453**	**0.446**		Avg	0.178	0.269	**0.171**	0.270	0.176	0.268	0.172	**0.265**
ETTh2	96	0.303	0.352	0.304	0.353	**0.301**	**0.351**	0.305	0.353	Traffic	96	0.387	0.261	0.490	0.335	**0.385**	**0.260**	0.435	0.286
	192	**0.379**	**0.398**	0.386	0.403	0.387	0.404	**0.379**	**0.398**		192	0.412	0.273	0.498	0.337	**0.407**	**0.271**	0.438	0.289
	336	0.426	0.433	0.428	0.435	0.431	0.436	**0.424**	**0.432**		336	0.410	0.276	0.521	0.342	**0.406**	**0.275**	0.459	0.292
	720	0.431	0.448	**0.429**	**0.447**	0.460	0.462	0.432	0.448		720	**0.425**	0.287	0.598	0.346	0.426	**0.286**	0.469	0.306
	Avg	**0.385**	**0.408**	0.387	0.410	0.395	0.413	**0.385**	**0.408**		Avg	0.409	0.274	0.527	0.340	**0.406**	**0.273**	0.450	0.293
ETTm1	96	0.347	0.379	**0.334**	**0.371**	0.339	0.376	0.339	0.373	PEMS03	12	0.068	0.173	0.065	0.168	0.067	0.172	**0.063**	**0.165**
	192	0.388	0.399	0.385	0.399	**0.379**	**0.394**	0.388	0.397		24	0.094	0.204	0.082	0.189	0.094	0.203	**0.077**	**0.184**
	336	0.428	0.424	**0.426**	0.426	0.428	0.424	**0.426**	**0.420**		48	0.150	0.259	0.116	0.222	0.152	0.261	**0.105**	**0.218**
	720	0.499	0.463	**0.494**	0.467	**0.494**	**0.462**	0.505	0.464		96	0.229	0.328	0.157	0.263	0.231	0.328	**0.140**	**0.255**
	Avg	0.416	0.416	**0.410**	0.416	**0.410**	**0.414**	0.415	**0.414**		Avg	0.135	0.241	0.105	0.211	0.136	0.241	**0.096**	**0.206**
ETTm2	96	0.187	0.273	0.187	0.273	0.190	0.276	**0.183**	**0.270**	PEMS04	12	0.076	0.181	0.075	0.179	0.077	0.181	**0.071**	**0.173**
	192	0.251	0.312	0.252	0.313	0.255	0.315	**0.249**	**0.309**		24	0.092	0.199	0.088	0.194	0.094	0.203	**0.082**	**0.187**
	336	**0.312**	0.349	0.314	0.351	0.315	0.352	0.313	**0.348**		48	0.118	0.226	0.105	0.212	0.122	0.233	**0.098**	**0.209**
	720	0.411	**0.404**	**0.409**	**0.404**	0.413	0.406	0.416	0.409		96	0.148	0.254	0.119	**0.228**	0.152	0.262	**0.118**	0.234
	Avg	**0.290**	0.335	0.291	0.335	0.293	0.337	**0.290**	**0.334**		Avg	0.109	0.215	0.097	0.203	0.111	0.220	**0.092**	**0.201**
Weather	96	0.174	0.215	0.176	0.214	0.179	0.219	**0.169**	**0.213**	PEMS07	12	0.060	0.156	**0.058**	**0.153**	0.060	0.156	**0.058**	**0.153**
	192	0.224	0.257	0.221	0.257	0.225	0.260	**0.217**	**0.255**		24	0.080	0.182	0.068	0.167	0.078	0.178	**0.067**	**0.164**
	336	0.283	0.301	**0.266**	**0.297**	0.286	0.303	0.274	**0.297**		48	0.100	0.199	**0.078**	**0.173**	0.100	0.200	0.089	0.190
	720	0.358	0.349	**0.340**	0.350	0.365	0.355	0.352	**0.348**		96	0.146	0.252	**0.094**	**0.188**	0.178	0.285	0.112	0.215
	Avg	0.260	0.281	**0.251**	0.280	0.264	0.284	0.253	**0.278**		Avg	0.097	0.197	**0.075**	**0.170**	0.104	0.205	0.082	0.181
Solar	96	0.200	0.237	0.201	**0.231**	0.217	0.243	**0.187**	0.241	PEMS08	12	0.079	0.179	**0.072**	**0.172**	0.079	0.179	0.075	0.177
	192	0.253	0.267	0.240	0.279	0.246	**0.258**	**0.197**	0.262		24	0.113	0.211	0.097	0.192	0.115	0.214	**0.088**	**0.187**
	336	0.253	0.277	0.246	0.278	0.256	0.280	**0.223**	**0.273**		48	0.210	0.259	0.150	0.229	0.206	0.251	**0.130**	**0.221**
	720	0.264	0.281	0.226	**0.267**	0.261	0.284	**0.221**	0.280		96	0.257	0.292	0.213	0.269	0.255	0.292	**0.172**	**0.256**
	Avg	0.243	0.266	0.228	**0.264**	0.245	0.266	**0.207**	**0.264**		Avg	0.165	0.235	0.133	0.216	0.164	0.234	**0.116**	**0.210**

Table 14: Full results on DUET with lookback $L = 96$: MSE and MAE for every dataset and horizon, and their average over the four horizons (Avg). **Bold** marks the best in each row.

Models		DUET		+LIFT		+CCM		+Matryoshka		Models		DUET		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	96	0.386	**0.398**	0.405	0.404	**0.384**	**0.398**	0.395	0.400	Electricity	96	0.147	0.234	**0.142**	**0.230**	0.146	0.233	**0.142**	0.231
	192	**0.426**	**0.423**	0.455	0.435	0.431	0.426	0.440	0.429		192	0.161	0.246	**0.156**	**0.243**	0.161	0.247	0.163	0.250
	336	**0.472**	**0.446**	0.512	0.465	0.479	0.451	0.484	0.449		336	0.174	0.261	**0.170**	**0.259**	0.175	0.261	0.178	0.265
	720	0.504	0.489	0.579	0.520	**0.462**	**0.462**	0.476	0.463		720	0.203	**0.288**	0.211	0.295	**0.202**	0.290	0.206	0.294
	Avg	0.447	0.439	0.488	0.456	**0.439**	**0.434**	0.449	0.435		Avg	0.171	**0.257**	**0.170**	**0.257**	0.171	0.258	0.172	0.260
ETTh2	96	0.290	0.340	0.295	0.342	**0.289**	0.340	0.292	**0.338**	Traffic	96	0.405	**0.248**	0.422	**0.248**	**0.403**	0.249	0.422	0.267
	192	**0.364**	0.387	0.377	0.390	0.369	0.389	0.366	**0.385**		192	**0.418**	**0.257**	0.454	0.259	0.425	0.260	0.459	0.283
	336	0.406	0.420	0.455	0.438	0.413	0.423	**0.395**	**0.413**		336	**0.436**	**0.266**	0.483	**0.266**	0.443	**0.266**	0.460	0.283
	720	0.421	0.438	0.434	0.442	**0.418**	**0.435**	0.447	0.451		720	0.477	0.288	0.571	0.289	**0.469**	**0.287**	0.489	0.299
	Avg	**0.370**	**0.396**	0.390	0.403	0.372	0.397	0.375	0.397		Avg	**0.434**	**0.265**	0.483	0.266	0.435	0.266	0.458	0.283
ETTm1	96	0.329	0.360	**0.320**	0.351	0.333	0.362	0.325	**0.348**	PEMS03	12	0.072	0.178	0.067	0.171	0.071	0.177	**0.059**	**0.158**
	192	0.376	0.384	0.371	0.378	**0.369**	0.382	**0.369**	**0.371**		24	0.098	0.207	0.091	0.203	0.098	0.207	**0.076**	**0.182**
	336	**0.408**	0.406	0.414	0.405	0.409	0.407	0.411	**0.398**		48	0.150	0.266	0.132	0.244	0.144	0.264	**0.124**	**0.233**
	720	0.479	0.444	0.507	0.457	**0.476**	**0.442**	0.503	0.453		96	0.193	0.298	0.168	0.278	0.203	0.312	**0.135**	**0.245**
	Avg	0.398	0.399	0.403	0.398	**0.397**	0.398	0.402	**0.393**		Avg	0.128	0.237	0.115	0.224	0.129	0.240	**0.099**	**0.205**
ETTm2	96	0.174	0.256	0.178	0.256	**0.171**	**0.252**	0.177	0.255	PEMS04	12	0.080	0.183	0.076	0.177	0.080	0.183	**0.067**	**0.164**
	192	0.240	0.299	0.253	0.308	**0.239**	**0.298**	0.249	0.307		24	0.104	0.212	0.094	0.201	0.099	0.208	**0.078**	**0.177**
	336	**0.307**	**0.344**	0.361	0.373	0.318	0.351	0.310	0.345		48	0.133	0.249	0.105	0.217	0.127	0.243	**0.093**	**0.199**
	720	**0.414**	**0.408**	0.432	0.414	0.434	0.420	0.418	0.410		96	0.152	0.267	0.131	0.245	0.154	0.274	**0.114**	**0.223**
	Avg	**0.284**	**0.327**	0.306	0.338	0.291	0.330	0.289	0.329		Avg	0.117	0.228	0.102	0.210	0.115	0.227	**0.088**	**0.191**
Weather	96	0.159	0.199	0.159	0.199	0.159	0.200	**0.149**	**0.192**	PEMS07	12	0.065	0.165	0.061	0.154	0.064	0.164	**0.052**	**0.140**
	192	0.207	0.244	0.206	0.242	0.212	0.248	**0.199**	**0.241**		24	0.089	0.196	0.076	0.177	0.089	0.195	**0.065**	**0.158**
	336	0.258	**0.283**	0.260	0.284	0.261	0.286	**0.250**	**0.283**		48	0.120	0.232	0.105	0.207	0.115	0.226	**0.090**	**0.187**
	720	0.330	**0.332**	0.332	0.335	0.333	0.336	**0.316**	0.338		96	0.154	0.263	0.141	0.246	0.165	0.276	**0.111**	**0.209**
	Avg	0.239	0.265	0.239	0.265	0.241	0.268	**0.229**	**0.264**		Avg	0.107	0.214	0.096	0.196	0.108	0.215	**0.080**	**0.174**
Solar	96	0.194	**0.203**	0.195	0.214	0.192	0.211	**0.175**	0.214	PEMS08	12	0.075	0.176	0.076	0.169	0.075	0.176	**0.066**	**0.157**
	192	0.254	0.231	0.227	0.235	0.247	0.243	**0.191**	**0.218**		24	0.106	0.201	0.098	0.193	0.102	0.201	**0.096**	**0.177**
	336	0.254	0.251	0.232	0.235	0.250	0.249	**0.217**	**0.232**		48	0.146	0.241	0.118	0.215	0.137	0.235	**0.115**	**0.199**
	720	**0.244**	0.244	**0.244**	**0.243**	0.247	0.245	**0.244**	0.251		96	0.211	0.273	0.177	0.260	0.207	0.287	**0.149**	**0.229**
	Avg	0.237	0.232	0.225	0.232	0.234	0.237	**0.207**	**0.229**		Avg	0.135	0.223	0.117	0.209	0.130	0.225	**0.107**	**0.191**

Table 15: CI/CD benchmark of Han et al. (2024): MSE and MAE at horizon 48 of the Linear and Transformer backbones in their CD and CI forms and with each plug-in. Plug-ins wrap the CI form, PRReg the CD form, with its weight decay chosen on validation. **Bold** marks the best in each row, and the last two rows average the twelve datasets.

Models		CD		CI		+PRReg		+LIFT		+CCM		+Matryoshka	
Metrics		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh1	Linear	0.400	0.424	0.342	0.372	0.341	0.377	0.347	0.385	0.353	0.384	**0.335**	**0.371**
	Transformer	0.802	0.706	0.661	0.557	0.441	0.459	0.632	0.543	0.672	0.564	**0.352**	**0.389**
ETTh2	Linear	0.662	0.610	0.231	0.306	0.240	0.319	0.259	0.330	0.241	0.319	**0.223**	**0.295**
	Transformer	1.380	0.956	0.275	0.346	0.255	0.334	0.282	0.350	0.300	0.361	**0.227**	**0.300**
ETTm1	Linear	0.393	0.417	0.345	0.376	0.306	0.357	0.361	0.386	0.347	0.377	**0.285**	**0.340**
	Transformer	0.417	0.423	0.359	0.397	0.307	0.356	0.370	0.403	0.376	0.408	**0.290**	**0.342**
ETTm2	Linear	0.160	0.279	0.150	0.254	**0.137**	**0.238**	0.150	0.255	0.149	0.254	0.143	0.239
	Transformer	0.273	0.388	0.148	0.256	0.137	0.242	0.164	0.264	0.178	0.280	**0.130**	**0.228**
Weather	Linear	0.141	0.218	0.168	0.229	0.132	0.190	0.169	0.227	0.169	0.233	**0.130**	**0.184**
	Transformer	0.230	0.315	0.164	0.231	0.148	0.220	0.180	0.231	0.163	0.229	**0.133**	**0.181**
Solar	Linear	0.192	0.311	0.323	0.437	0.216	0.312	0.258	0.319	0.329	0.443	**0.142**	**0.208**
	Transformer	**0.153**	**0.193**	0.295	0.338	0.160	0.209	0.313	0.356	0.315	0.355	0.190	0.258
Electricity	Linear	0.442	0.487	0.196	0.276	0.174	0.293	0.175	0.261	0.196	0.277	**0.126**	**0.225**
	Transformer	0.253	0.355	0.202	0.298	0.200	0.319	0.203	0.301	0.196	0.290	**0.139**	**0.239**
Traffic	Linear	0.777	0.513	0.703	0.427	0.815	0.441	0.552	0.342	0.705	0.429	**0.426**	**0.281**
	Transformer	0.656	0.366	0.589	0.330	0.940	0.449	0.592	0.344	0.601	0.332	**0.429**	**0.288**
PEMS03	Linear	0.244	0.358	0.318	0.409	0.145	0.268	0.184	0.292	0.322	0.416	**0.123**	**0.237**
	Transformer	0.150	0.257	0.203	0.306	**0.115**	**0.232**	0.205	0.309	0.205	0.308	0.125	0.242
PEMS04	Linear	0.148	0.275	0.323	0.405	0.124	0.250	0.195	0.307	0.327	0.407	**0.122**	**0.241**
	Transformer	**0.113**	**0.221**	0.213	0.320	0.122	0.239	0.212	0.320	0.217	0.325	0.134	0.251
PEMS07	Linear	0.215	0.311	0.376	0.436	0.126	0.250	0.199	0.301	0.379	0.439	**0.102**	**0.212**
	Transformer	0.176	0.249	0.236	0.326	**0.117**	**0.228**	0.233	0.325	0.250	0.334	0.135	0.244
PEMS08	Linear	0.389	0.404	0.384	0.431	0.217	0.306	0.263	0.347	0.391	0.429	**0.146**	**0.244**
	Transformer	0.267	0.291	0.280	0.326	0.152	0.250	0.277	0.325	0.294	0.329	**0.136**	**0.236**
Average	Linear	0.347	0.384	0.322	0.363	0.248	0.300	0.259	0.313	0.326	0.367	**0.192**	**0.256**
	Transformer	0.406	0.393	0.302	0.336	0.258	0.295	0.305	0.339	0.314	0.343	**0.202**	**0.267**

Table 16: Comparison with the published numbers of xCPD (Li et al., 2026) on the CI/CD benchmark of Han et al. (2024): MSE at horizon 48 on the four datasets reported by xCPD, which wraps the CI form of each backbone like the other plug-ins. The xCPD values are taken from its paper. The last two rows average the four datasets. **Bold** marks the best in each row.

Models		CD	CI	+PRReg	+LIFT	+CCM	+xCPD	+Matryoshka
ETTh1	Linear	0.400	0.342	0.341	0.347	0.353	0.338	**0.335**
	Transformer	0.802	0.661	0.441	0.632	0.672	0.512	**0.352**
ETTm1	Linear	0.393	0.345	0.306	0.361	0.347	0.305	**0.285**
	Transformer	0.417	0.359	0.307	0.370	0.376	**0.289**	0.290
Weather	Linear	0.141	0.168	0.132	0.169	0.169	**0.128**	0.130
	Transformer	0.230	0.164	0.148	0.180	0.163	0.161	**0.133**
Electricity	Linear	0.442	0.196	0.174	0.175	0.196	0.192	**0.126**
	Transformer	0.253	0.202	0.200	0.203	0.196	0.181	**0.139**
Average	Linear	0.344	0.263	0.238	0.263	0.266	0.241	**0.219**
	Transformer	0.426	0.347	0.274	0.346	0.352	0.286	**0.229**