BEYOND MODEL LIBRARIES: LEARNING TO COMPOSE FORECASTING ARCHITECTURES FOR HETEROGENEOUS TIME SERIES

Anonymous authors
Paper under double-blind review

ABSTRACT

Time-series forecasting models often exhibit heterogeneous performance across series with diverse temporal patterns, making it challenging to identify a universally optimal architecture. Existing approaches either select from predefined model libraries or rely on large-scale pretrained models, but they are limited by restricted structural coverage, high computational cost, or insufficient adaptation to individual series. In this paper, we propose **RankTS**, a component-level model selection framework that learns architectural preferences from historical forecasting tasks and transfers them to unseen series. RankTS constructs a diverse compositional search space from forecasting operators, explores effective architectures through task-local evolution, and learns a series-conditioned ranker from sparse architecture evaluations. By jointly modeling time-series characteristics and architecture configurations, RankTS efficiently generates and selects suitable forecasting architectures without exhaustive search on each target series. Extensive experiments on five time-series datasets and four forecasting horizons demonstrate that RankTS consistently outperforms existing model selection strategies and achieves competitive performance against strong forecasting model families with limited target-side training.

1 INTRODUCTION

Time-series forecasting plays a crucial role in many real-world applications, including power systems, energy management, and weather forecasting (Lu et al., 2025; Li et al., 2026). As these systems continue to scale, the corresponding time series exhibit increasingly heterogeneous patterns in trend, periodicity, and local dynamics, making it challenging for a single forecasting model to achieve consistently strong performance across diverse scenarios. As illustrated in Figure 1, existing studies mainly address such heterogeneity from two perspectives. Model selection and ensemble methods select or combine predictors from a predefined model library according to time-series characteristics (Montero-Manso et al., 2020), but their performance is inherently constrained by the structural coverage of the candidate library. Time-series foundation models instead learn transferable forecasting knowledge through large-scale pretraining (Ansari et al., 2024; Das et al., 2023; Woo et al., 2024). However, they usually require substantial computational resources, and a universally strong architecture may still fail to achieve optimal performance for individual time series.

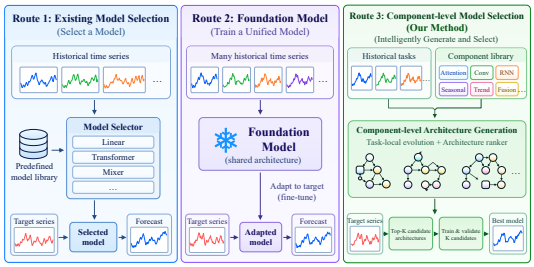


Figure 1: The three approaches address heterogeneous time-series forecasting.

Recently, neural architecture search (NAS) has emerged as a promising paradigm for automatically discovering effective neural structures across various domains (Wu et al., 2021b; 2023; 2024). Inspired by this idea, we explore a new direction of **component-level model selection**, where forecasting models are intelligently generated and selected by composing diverse forecasting opera-

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.

tors according to the characteristics of individual time series. Compared with selecting from a fixed model library, component-level model selection provides a larger and more flexible structural space, enabling models to better adapt to heterogeneous temporal patterns. However, realizing this paradigm for time-series forecasting introduces three fundamental challenges. First, **structural diversity is limited**: existing search spaces often rely on complete models or simple combinations of a few predefined modules, making it difficult to explore diverse component-level architectures. Second, **candidate effectiveness is difficult to guarantee**: expanding the component space inevitably introduces redundant or ineffective structures and substantially increases the cost of architecture evaluation. Third, **target-side adaptation remains expensive**: directly searching and evaluating numerous candidate architectures for each unseen series requires repeated model training, limiting practical deployment. These challenges call for a component-level model selection framework that can intelligently generate effective architectures while efficiently transferring architectural knowledge from historical forecasting tasks.

To address these challenges, we propose **RankTS**, a component-level model selection framework that transfers architectural preferences learned from historical searches to unseen forecasting tasks. Instead of repeatedly searching from scratch, RankTS learns which architectural components and configurations are suitable for different types of time series, enabling efficient architecture generation and selection. To improve **structural diversity**, RankTS constructs a compositional search space based on directed acyclic graphs (DAGs) consisting of 54 forecasting operators from 11 operator families and introduces *coverage- and diversity-aware initialization* to ensure broad exploration of component configurations. To improve **candidate effectiveness**, RankTS performs *task-local evolution* through an *evaluate–retain–propose* process, where task-specific feedback guides the retention and generation of promising architectures. To reduce **target-side adaptation cost**, RankTS learns a series-conditioned architecture ranker from sparse historical evaluations through pairwise preference supervision. The ranker jointly captures time-series characteristics and component-level architecture information, including topology, operators, and operator hyperparameters. For an unseen series, RankTS ranks candidate architectures from the historical archive and only trains a small number of highly ranked candidates, among which the final predictor is selected based on target validation performance. Our contributions are summarized as follows:

- We propose **RankTS**, a component-level model selection framework that transfers historical architecture-search experience into series-conditioned architecture rankings, enabling efficient generation and selection of forecasting models for unseen time series.
- We introduce **coverage- and diversity-aware initialization** and **task-local evolution** to jointly achieve broad operator–hyperparameter coverage, structural diversity, and task-specific optimization within a compositional architecture space.
- We develop a **parameter-aware graph encoder** and a **balanced within-task ranking formulation** that jointly model time-series characteristics and component-level architecture information, enabling effective preference learning for intelligent architecture selection.
- Extensive experiments on five time-series datasets and four forecasting horizons demonstrate that RankTS achieves the best performance in 11 out of 20 settings and ranks among the top two in 16 settings across the expanded comparison, while requiring only three target-side candidate trainings.

2 Related Work

Time-series foundation models. Time-series foundation models transfer predictive knowledge through large-scale pretraining on heterogeneous datasets. Representative approaches such as Chronos, TimesFM, Moirai, and MOMENT learn transferable forecasting or representation capabilities across tasks (Ansari et al., 2024; Das et al., 2023; Woo et al., 2024; Goswami et al., 2024). They primarily transfer shared predictive representations or model parameters rather than explicitly adapting architectures to individual series.

Model selection and meta-learning. Feature-based selection and combination transfer historical performance information to new forecasting tasks (Montero-Manso et al., 2020; Vanschoren, 2018). Joint task–configuration encoders extend this idea to architecture and hyperparameter recommendation (Moeini et al., 2026). RankTS learns from sparse evaluations of compositional DAGs, with

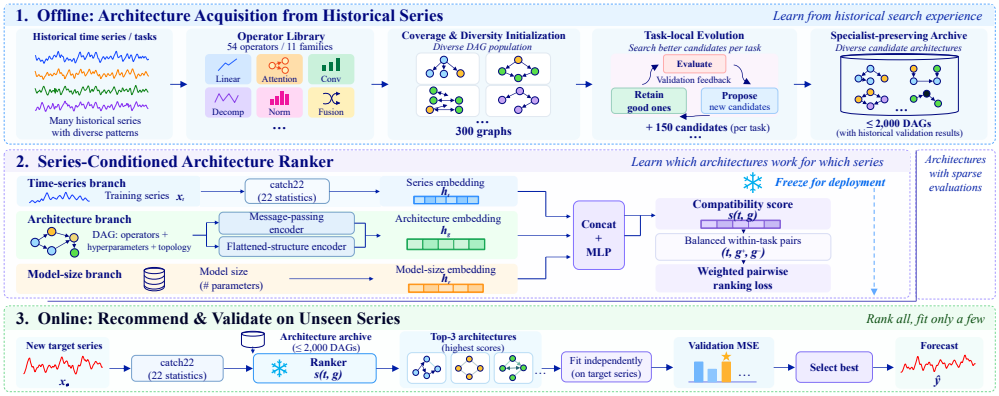


Figure 2: Overview of RankTS.

parameter-aware graph representations and within-task preferences supporting a small target-side fitting budget.

3 RankTS

As illustrated in Figure 2, RankTS consists of two stages: offline preference learning and online architecture recommendation. In the offline stage, a series-conditioned architecture ranker is trained from historical series–architecture evaluations. In the online stage, the frozen ranker identifies the Top-K candidate architectures for an unseen series, and only these candidates are trained and validated to determine the final forecasting architecture.

3.1 Offline Architecture and Preference Learning

3.1.1 Operator Library and Executable DAG Space

Computational DAGs are widely used to represent model architectures (Zhang et al., 2019); accordingly, we also adopt DAGs to represent time-series forecasting architectures. To enhance the structural diversity of candidate architectures, as shown in Table 5, we construct an operator library consisting of 54 forecasting operators from 11 operator families, including Linear, Local, and Trend operators. These heterogeneous operators provide complementary modeling capabilities, enabling the resulting architectures to capture diverse temporal characteristics such as trends, local patterns, and other dynamic behaviors.

To avoid excessive architectural complexity, each graph contains at most ten nodes and is equipped with a linear forecasting head. For compatible multi-input nodes, incoming representations are merged by mean, summation, gating, or subtraction before being processed by the node operator, with an optional residual connection. Acyclicity constraints and typed fast Fourier transform (FFT) and inverse fast Fourier transform (IFFT) transitions ensure valid and executable paths across the time and frequency domains. In addition, preprocessing operations such as scaling, differencing, or detrending can be applied before the input enters the graph.

3.1.2 Coverage- and Diversity-Aware Initialization

Candidate Pool Generation. We first construct 282 base architectures using heuristic rules and expand them through a hierarchical *skeleton–operator-family–operator–hyperparameter* strategy. We sample chain, parallel, decomposition, or frequency-path skeletons, instantiate compatible operators and hyperparameters, and optionally add normalization, activation, residual connections, and preprocessing. After deduplication and validity checks, this yields a candidate pool $\mathcal{G}$ containing 1,965 architectures.

Seed Selection and Population Initialization. We construct a seed architecture set S using two complementary criteria. *Marginal coverage gain*, $|C(g) \cap U|$, counts newly covered operator–

hyperparameter configurations, where $C(g)$ denotes the configurations present in architecture g and U contains those not yet covered by S.

A *structural heuristic score*, $Q(g)$, rewards operator-family coverage and selected structural features within each architecture. We prioritize marginal coverage gain and use $Q(g)$ to break ties, obtaining 118 seed architectures after retaining valid representative recipes. To further improve structural diversity, we bucket the remaining candidates by attributes such as operator-family composition and node count, and select across buckets in round-robin order to form a 300-architecture initial population P_0. All historical tasks share this ordered population, with its first 90 architectures designated as the persistent anchor set A for task-local evolution. Detailed rules and resource constraints are provided in the appendix.

Algorithm 1 Coverage- and Diversity-Aware Initialization

Require: Hyperparameter catalogs $\{\Lambda_o\}$, recipes $\mathcal{R}$, resource constraints $\mathcal{E}$; $M = 2000$, $N = 300$, $a = 90$
Ensure: Initial population P_0 and anchors A
$\mathcal{G} \leftarrow \mathrm{Filter}\big(\mathrm{ExpandUnique}(\mathrm{Base}(), M)\big)$
$U \leftarrow \{(o, \lambda) : \lambda \in \Lambda_o\}$; $S \leftarrow [\,]$
while $U \neq \varnothing$ **do**
  $g^\star \leftarrow \arg\max_{g \in \mathcal{G} \setminus S} \big(|C(g) \cap U|,\ Q(g),\ \mathrm{ID}(g)\big)$
  Append $g^\star$ to S; $U \leftarrow U \setminus C(g^\star)$
Append unselected recipes from $\mathcal{G} \cap \mathcal{R}$ to S; verify $\mathcal{E}$
Bucket $\mathcal{G} \setminus S$ by family composition, node count, preprocessing, and multi-input connectivity
Sort each bucket by descending $Q(g)$; initialize a round-robin bucket queue $\mathcal{B}$
while $|S| < N$ **do**
  $b \leftarrow \mathrm{Dequeue}(\mathcal{B})$; $g \leftarrow \mathrm{PopFirst}(b)$
  if g satisfies $\mathcal{E}$ **then** append g to S
  if $b \neq \varnothing$ **then** $\mathrm{Enqueue}(\mathcal{B}, b)$
return $P_0 = S$, $A = \{S^{(1)}, \ldots, S^{(a)}\}$

3.1.3 Task-local evolution and archive construction

Task Evaluation and Survivor Selection. Each historical task t independently evolves the shared population P_0, recording validation mean squared error (MSE), resource costs, and execution status in Ω_t while reusing cached evaluations. Before each population update, we retain 150 valid architectures: successful anchors from A, validation elites occupying half the remaining slots, and performance–resource tradeoffs occupying up to a quarter. The remaining slots are filled by maximizing the minimum structural-feature distance to selected architectures, balancing performance, efficiency, and diversity.

Algorithm 2 Task-Local Evolution

Require: Historical tasks $\mathcal{H}_d$, initial population P_0, anchors A, rounds $R = 2$, archive budget $B = 2000$
Ensure: Historical observations Ω_d and deployment archive $\mathcal{A}_d$
$\Omega_d \leftarrow \varnothing$
for each task $t \in \mathcal{H}_d$ **do**
  $P \leftarrow P_0$; $\Omega_t \leftarrow \varnothing$
  for $r = 1, \ldots, R$ **do**
    $\Omega_t \leftarrow \Omega_t \cup \mathrm{Eval}_t\big(P \setminus \mathrm{Graphs}(\Omega_t)\big)$
    if $r = R$ or $\mathrm{Stop}(\Omega_t)$ **then break**
    $S \leftarrow \mathrm{Retain}_{150}(P, \Omega_t, A)$
    $\mathcal{C} \leftarrow \mathrm{Propose}_{75,30,30,15}(S, \Omega_t)$
    $P \leftarrow S \cup \mathrm{FilterFill}_{150}(\mathcal{C}, \Omega_t)$
  $\Omega_d \leftarrow \Omega_d \cup \Omega_t$
$\mathcal{U}_d \leftarrow \mathrm{Graphs}(\Omega_d)$
$\mathcal{A}_d \leftarrow A \cup \bigcup_{t \in \mathcal{H}_d} \mathrm{Top}_6^{\mathrm{val}}(\Omega_t)$
Fill $\mathcal{A}_d$ to budget B from $\mathcal{U}_d$ by median loss percentile, mean percentile, and decreasing task coverage
Fill any remaining slots by structural diversity
return $\Omega_d, \mathcal{A}_d$

Candidate Generation and Population Update. From the retained set S, we generate 150 new architectures, targeting 75 mutations, 30 crossovers, 30 surrogate-guided candidates, and 15 random candidates. Mutation changes operators, hyperparameters, connections, or preprocessing; crossover transfers domain-compatible operators and hyperparameters between parents. A task-local random forest predicts log validation loss and prioritizes candidates minimizing $\mu_t(g) - 0.5\sigma_t(g)$, where $\mu_t(g)$ and $\sigma_t(g)$ denote the mean and standard deviation (SD) of tree predictions. Proposals undergo duplicate, structural, and resource checks, with random generation filling unmet quotas. Retained and new architectures form the next 300-member population. Two rounds yield 450 distinct evaluation attempts per task, with independent initialization and at most five training epochs per predictor.

Specialist-Preserving Archive Construction. We consolidate the explored union $\mathcal{U}_d$ into a deployment archive $\mathcal{A}_d$ of at most 2,000 architectures. Anchors and each task's six best observed architectures are retained first. After deduplication, remaining candidates are ordered by median within-task validation-loss percentile, mean percentile, and decreasing task coverage, with structural-diversity selection filling any remaining capacity. The archive bounds deployment scoring, while

ranker supervision uses successful observations with finite positive validation losses from the larger explored union. Detailed rules and stopping criteria appear in Appendix D.

3.1.4 JOINT REPRESENTATION AND COMPATIBILITY SCORING

Joint Series–Architecture Representation. We represent each series t using a 22-dimensional catch22 descriptor (Lubba et al., 2019) extracted from its training segment. After imputation and normalization fitted on historical training tasks, a multilayer perceptron (MLP) with two layers maps it to $h_t \in \mathbb{R}^{96}$. To encode candidate architectures, we develop a parameter-aware graph encoder that captures both operator configurations and their directed dependencies. Initial node states combine operator and position embeddings, operator hyperparameter values and applicability masks, and connection attributes. Since the role of an operator depends on the transformations preceding it, three directed message-passing layers with gated recurrent unit (GRU) updates incorporate upstream information into each node representation:

$$m_v^{(\ell)} = \frac{\sum_{u \in \mathrm{pred}(v)} z_u^{(\ell)}}{\max(1, |\,\mathrm{pred}(v)|)}, \qquad z_v^{(\ell+1)} = \mathrm{GRU}_\ell\Big(m_v^{(\ell)}, z_v^{(\ell)}\Big). \tag{1}$$

A graph readout combines mean- and max-pooled node states with the last active node state to form h_{MP}. In parallel, a projection encodes flattened adjacency and node attributes as h_{flat}, preserving explicit structural information that complements message passing. The two branches are concatenated and projected, with Gaussian error linear unit (GELU) activation and layer normalization (LayerNorm), into the final architecture representation:

$$h_g = \mathrm{LayerNorm}(\mathrm{GELU}(W_f[h_{\mathrm{MP}}, h_{\mathrm{flat}}] + b_f)) \in \mathbb{R}^{96}. \tag{2}$$

Series–Architecture Compatibility Scoring. To account for model size, we project the standardized log-parameter count $\log(1 + p_{t,g})$ into a model-size embedding $h_r \in \mathbb{R}^{96}$, where $p_{t,g}$ is the number of parameters in architecture g instantiated for task t. An MLP combines the series, architecture, and model-size embeddings to predict a series-conditioned compatibility score:

$$s_\theta(t, g) = \mathrm{MLP}([h_t, h_g, h_r]). \tag{3}$$

3.1.5 WITHIN-TASK RANKING OBJECTIVE AND TRAINING

Within-Task Pair Construction. For valid historical evaluations, we define utility as $u_{t,g} = -\log L_{t,g}^{\mathrm{val}}$, making within-task utility differences invariant to positive rescaling of losses. We sample up to 1,024 pairs per task, targeting 30% head, 30% cross-stratum, 20% semi-hard, and 20% random comparisons to balance discrimination among competitive candidates with broader performance ordering. Local gap thresholds exclude near ties, while structural weights favor diverse comparisons. Sampling proceeds without replacement, with uniform backfilling; details appear in Appendix E.4.

Ranking Objective and Training. Let $\Delta u_{tij} = u_{t,g_i} - u_{t,g_j}$, $y_{tij} = \mathrm{sign}(\Delta u_{tij})$, and $\Delta s_{tij} = s_\theta(t, g_i) - s_\theta(t, g_j)$. We jointly train the encoders and scorer using

$$\mathcal{L}(\theta) = \frac{1}{|\mathcal{P}|} \sum_{(t,i,j) \in \mathcal{P}} w_{tij} \log(1 + \exp(-y_{tij} \Delta s_{tij})), \tag{4}$$

where $w_{tij} = \mathrm{clip}\big(|\Delta u_{tij}| / \max(m, 10^{-4}),\, 0.25,\, 4\big)$ and m is the median absolute training-pair utility gap. This objective emphasizes correct ordering with bounded performance-gap weights. We select training duration by normalized discounted cumulative gain (NDCG) at rank 20 on held-out historical tasks, then reinitialize and refit on all historical tasks with recomputed normalization and gap scaling.

3.2 ONLINE TOP-K RECOMMENDATION AND FORECASTING

For an unseen series t, we extract its catch22 descriptor from the training segment and apply the frozen historical imputation and normalization. The ranker scores all architectures in $\mathcal{A}_d$, and the K highest-scoring candidates form the shortlist P_t^K. Each candidate is independently fitted on the target training segment, and validation MSE determines the final predictor:

$$\hat{g}_t^K = \underset{g \in P_t^K :\, L_{t,g}^{\mathrm{val}} \text{ finite}}{\arg\min} L_{t,g}^{\mathrm{val}}. \tag{5}$$

Table 1: Test MSE (↓) of RankTS and baselines over three seeds. "+V" denotes target-validation selection; Hist/Select denote Historical best/Meta-select. CM3+V applies "+V" to Select's top three complete-model configurations. Bold indicates best values; **red bold** and blue underline mark first and second by unrounded means. See Appendix B for the evaluation protocol. The corresponding means and sample SDs across seeds are reported in Table 9.

		RankTS		Complete-model baselines											Library selection		
Data	H	**Top1**	**Top3+V**	**DLinear**	**Patch**	**iTrans**	**Times**	**Mixer**	**FED**	**Auto**	**Inform**	**Trans**	**Nonstat**	**Light**	**Hist**	**Select**	**CM3+V**
Weather	96	1.186	**1.182**	1.195	1.682	1.779	1.741	1.695	1.836	2.114	1.189	1.208	1.286	1.283	1.208	1.208	1.183
	192	1.050	**1.035**	1.049	1.876	1.851	1.920	1.788	2.021	2.444	1.062	1.043	1.155	1.131	1.043	1.043	1.045
	336	0.993	0.981	0.989	1.540	1.524	1.726	1.500	1.779	1.975	1.016	1.000	1.048	1.058	1.000	1.000	**0.981**
	720	1.302	1.293	1.299	1.897	1.825	1.956	1.908	2.229	2.210	1.336	1.314	1.837	1.373	1.314	1.314	**1.286**
	Mean	*1.133*	*1.123*	*1.133*	*1.749*	*1.744*	*1.836*	*1.723*	*1.966*	*2.185*	*1.151*	*1.141*	*1.332*	*1.211*	*1.141*	*1.141*	*1.124*
London	96	**1.174**	1.184	1.176	1.215	1.192	1.191	1.182	1.230	1.283	1.308	1.238	1.267	1.227	1.182	1.189	1.174
	192	**1.200**	1.205	1.205	1.246	1.226	1.227	1.226	1.287	1.381	1.363	1.291	1.315	1.254	1.226	1.228	1.227
	336	1.209	**1.204**	1.214	1.258	1.244	1.245	1.245	1.320	1.438	1.398	1.311	1.323	1.261	1.245	1.243	1.239
	720	1.260	**1.250**	1.273	1.313	1.306	1.306	1.309	1.413	1.528	1.459	1.392	1.377	1.315	1.309	1.303	1.303
	Mean	*1.211*	*1.211*	*1.217*	*1.258*	*1.242*	*1.242*	*1.241*	*1.313*	*1.408*	*1.382*	*1.308*	*1.320*	*1.264*	*1.241*	*1.241*	*1.236*
Wind	96	0.337	**0.329**	0.345	0.367	0.389	0.396	0.376	0.462	0.542	0.487	0.451	0.361	0.364	0.376	0.362	0.356
	192	0.503	0.502	**0.501**	0.592	0.621	0.639	0.605	0.672	0.770	0.601	0.576	0.583	0.512	0.605	0.581	0.624
	336	0.634	**0.632**	0.640	0.804	0.850	0.860	0.812	0.918	1.028	0.714	0.683	0.775	0.647	0.812	0.758	0.724
	720	0.834	0.835	0.816	1.156	1.195	1.205	1.162	1.372	1.387	0.824	0.824	1.094	**0.815**	1.162	1.046	0.873
	Mean	*0.577*	*0.575*	*0.575*	*0.730*	*0.764*	*0.775*	*0.739*	*0.856*	*0.932*	*0.657*	*0.634*	*0.703*	*0.585*	*0.739*	*0.687*	*0.644*
KDD	96	0.491	0.480	**0.480**	0.553	0.555	0.564	0.551	0.550	0.589	0.518	0.511	0.526	0.498	0.551	0.491	0.483
	192	0.503	**0.489**	0.500	0.575	0.583	0.588	0.577	0.569	0.604	0.530	0.516	0.527	0.514	0.577	0.507	0.496
	336	0.507	**0.498**	0.513	0.592	0.601	0.604	0.595	0.590	0.620	0.534	0.520	0.540	0.524	0.595	0.516	0.520
	720	0.501	**0.496**	0.513	0.621	0.637	0.641	0.633	0.637	0.638	0.516	0.513	0.529	0.521	0.633	0.525	0.513
	Mean	*0.500*	*0.491*	*0.502*	*0.585*	*0.594*	*0.599*	*0.589*	*0.587*	*0.613*	*0.525*	*0.515*	*0.531*	*0.514*	*0.589*	*0.510*	*0.503*
Electricity	96	0.190	**0.181**	0.211	0.182	0.187	0.187	0.196	0.248	0.299	0.598	0.341	0.284	0.233	0.187	0.187	0.182
	192	0.201	0.192	0.209	0.190	0.198	0.197	0.208	0.282	0.469	0.668	0.535	0.327	0.232	0.197	0.198	**0.189**
	336	0.222	0.210	0.223	**0.210**	0.220	0.224	0.228	0.320	0.620	0.743	0.689	0.451	0.248	0.224	0.224	0.211
	720	0.270	**0.247**	0.255	0.255	0.270	0.273	0.275	0.442	0.746	0.779	0.776	0.676	0.278	0.273	0.274	0.256
	Mean	*0.221*	*0.208*	*0.224*	*0.209*	*0.219*	*0.220*	*0.227*	*0.323*	*0.533*	*0.697*	*0.585*	*0.435*	*0.248*	*0.220*	*0.221*	*0.210*
1st / 2nd		**2** / 6	**11** / 5	**2** / 3	**1** / 2	**0** / 0	**0** / 0	**0** / 0	**0** / 0	**0** / 0	**0** / 0	**0** / 1	**0** / 0	**1** / 0	**0** / 1	**0** / 1	**3** / 3

This procedure combines historical architectural preferences with target-specific validation, requiring at most three candidate fits under the default $K = 3$.

4 Experiments

We evaluate RankTS on five Monash datasets (Godahewa et al., 2021). Each dataset provides 150 historical series and a disjoint query pool of 17 series, with each series chronologically split into training, validation, and test segments in a 60%/20%/20% ratio. We fix the input length at 96 and consider forecasting horizons $H \in \{96, 192, 336, 720\}$, denoted H96, H192, H336, and H720, respectively. RankTS is compared with 11 complete-model families under a shared budget of at most five training epochs per predictor. Detailed protocols are provided in Appendices B–G.

We address the following six research questions (RQs).

4.1 Forecasting Accuracy on Unseen Series (RQ1)

Table 1 shows that RankTS achieves state-of-the-art results across multiple datasets under the shared evaluation protocol. Top3+V ranks first in 11 of the 20 settings and among the top two in 16, demonstrating that a compact shortlist can capture architectures suited to heterogeneous temporal patterns. Even with a single candidate fit, Top1 reaches the top two in eight settings, indicating that the learned ranking transfers useful architectural preferences to unseen series. Target validation further refines these recommendations by selecting among complementary candidates, yielding strong forecasting performance with only three candidate fits.

RankTS also outperforms the historical model-selection policies. Hist chooses a fixed model based on historical performance, whereas Select adapts model identity to query-series features. Their remaining performance gap suggests that adaptive selection alone may be insufficient when the available choices are restricted to a predefined model library. By combining a compositional architecture space with series-conditioned ranking and target validation, RankTS provides greater flexibility to accommodate diverse forecasting tasks.

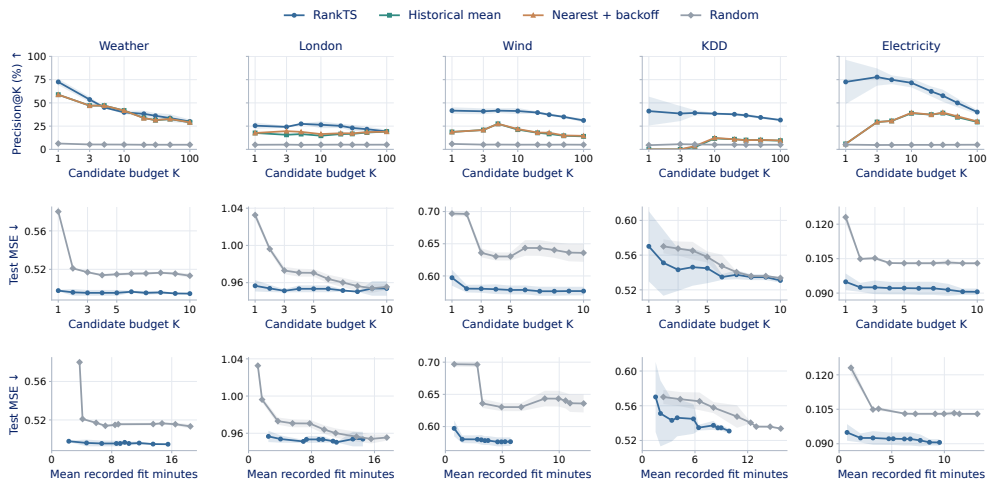


Figure 3: Precision@K and validation-selected test MSE under different candidate budgets. Evaluation protocols appear in Appendices J and G.

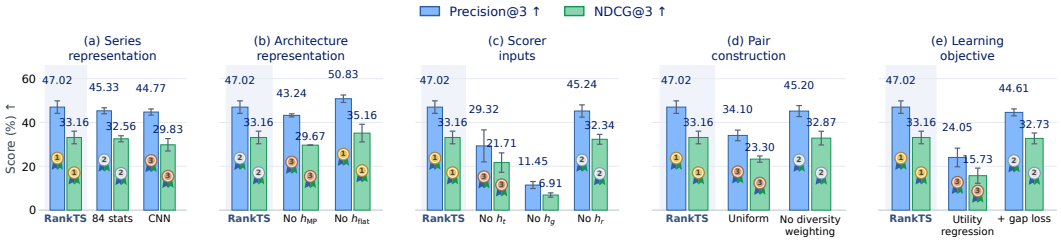


Figure 4: Retrieval performance of ranker ablations at H96..

4.2 Architecture Recommendation under Limited Budgets (RQ2)

RankTS achieves strong retrieval performance across datasets and budgets, outperforming historical-mean, nearest-history, and random ranking in most settings (Figure 3, top). At $K = 3$, it leads all three controls on every dataset, reaching 53.6% precision on Weather and 77.8% on Electricity, compared with 47.1% and 29.4% for historical-mean ranking. These results indicate that series-conditioned scoring concentrates competitive architectures within compact shortlists. Precision generally declines as larger budgets include more lower-ranked candidates, while historical-mean ranking remains competitive on Weather at intermediate budgets.

The target-side cost study further examines whether this ranking quality translates into economical forecasting (Figure 3, middle and bottom). On the 15-query subset, RankTS requires only 1–10 candidate fits to match or improve the dataset-level mean MSE achieved by random generation with ten fits, reducing the required fitting budget by 0%–90%. Additional fits beyond three generally provide modest gains while increasing cost, supporting compact shortlists for deployment. Occasional test-error increases, as observed on KDD, arise when validation selects a different candidate whose advantage does not carry over to testing. Together, these results support effective retrieval and limited target-side fitting, with recorded runtimes reflecting execution on shared graphics processing units (GPUs).

4.3 Ablation Study (RQ3)

Figure 4 compares series representations, architecture representations, scorer inputs, pair construction, and learning objectives using Precision@3 and NDCG@3 on fixed H96 labels.

Series and Architecture Representations. catch22 achieves the strongest mean head retrieval among the tested series encoders, suggesting that compact statistical summaries effectively charac-

terize architectural preferences. The window-based convolutional neural network (CNN) provides no additional advantage, indicating that detailed local encoding is not necessarily required for recommendation.

Scorer Inputs. Removing series or architecture information reduces Precision@3 from 47.0% to 29.3% and 11.5%, respectively, supporting their complementary roles in matching temporal characteristics with candidate structures. Removing model-size information yields a smaller decline to 45.2%, consistent with different series favoring different model capacities.

Pair Construction and Learning Objective. Balanced sampling improves Precision@3 from 34.1% to 47.0% and Recall@100 from 20.8% to 30.4%, strengthening head recommendation while preserving broader retrieval quality. Diversity weighting provides a smaller additional gain. Pairwise training also outperforms utility regression, while the auxiliary gap loss offers no mean head-retrieval improvement, supporting the simpler weighted pairwise objective.

Initialization Diversity. Diverse initialization depends on both candidate-pool construction and strategic seed selection. Table 2 compares the final 90-anchor sets under process-level interventions. Uniform selection from the proposed pool reduces configuration coverage from 92.95% to 53.22%, showing that a diverse candidate pool does not automatically yield representative seed architectures. Applying the proposed selection procedure to a random pool achieves full coverage, but produces larger architectures on average (6.56 versus 5.61 nodes). The complete pipeline achieves the highest coverage per 100 nodes (41.80), supporting the joint design of candidate generation and seed selection for compact configuration coverage.

Table 2: Coverage and complexity of 90 anchors selected from a pool of 2,000 candidates.

Candidate pool	Seed selection	Coverage (%) ↑	Nodes/arch.	Configs/100 nodes ↑
Proposed	Proposed	92.95±0.83	5.61±0.07	**41.80**±0.46
Random	Proposed	**100.00**±0.00	6.56±0.15	38.46±0.88
Proposed	Random	53.22±2.53	4.29±0.13	31.30±1.83
Random	Random	74.93±1.48	5.55±0.26	34.14±1.60

Task-Local Evolution. We compare the quality of 150 new candidates per task following 300 shared initial evaluations (Figure 5). A candidate is considered high-quality if its validation MSE does not exceed the 20th percentile of the initial population's validation losses. Task-local evolution produces a higher proportion of such candidates than random generation on all five historical tasks, increasing the average from 21.5% to 32.5%. These results suggest that combining task-specific feedback with guided proposals concentrates exploration on promising architectures under a fixed evaluation budget.

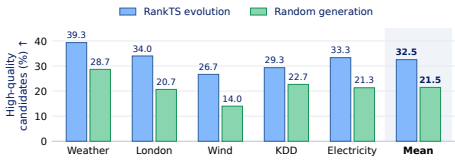


Figure 5: Comparison of evolution and random strategy.

4.4 Series-Conditioned Architecture Choices (RQ4)

Figure 6 illustrates dataset-dependent patterns in architecture recommendation, with different datasets emphasizing distinct subsets of their respective archives. Within each embedding space, the Top10 recommendations tend to concentrate in several regions, while repeated selections across query series indicate shared structural preferences. These patterns suggest that heterogeneous temporal characteristics favor different architectural choices, yet related series can benefit from common structures. RankTS accommodates this combination of variation and concentration through series-conditioned selection and architecture reuse.

4.5 Cross-Dataset Ranker Transfer (RQ5)

Figure 7 suggests that shared temporal characteristics support the transfer of architectural preferences across datasets. For example, the ranker trained on Electricity achieves 35.9% Precision@3

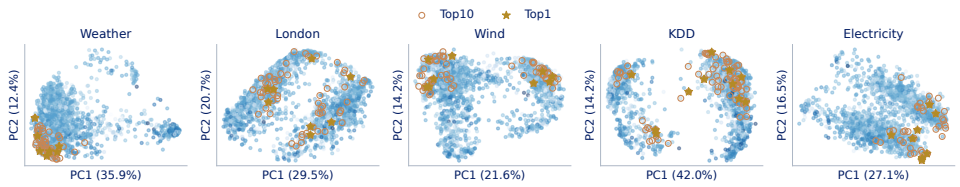


Figure 6: Principal component analysis (PCA) projection of the 2,000 archive embeddings. Blue intensity indicates node count; circles and stars denote the Top10 and Top1 recommendation unions, respectively. Axis labels indicate the variance explained by each principal component.

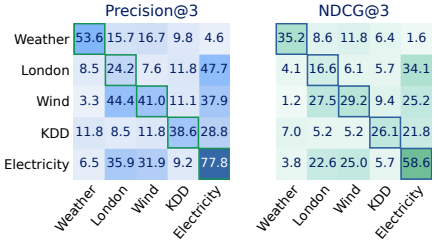


Figure 7: Cross-dataset results. Rows and columns indicate source and target datasets.

Table 3: Results of training-budget sensitivity. Mean Spearman correlations and validation-selection agreement compare the 5- and 30-epoch protocols.

Dataset	$\bar{\rho}_{\text{val}} \uparrow$	$\bar{\rho}_{\text{test}} \uparrow$	Agreement (%)	RankTS MSE 5 / 30 epochs	DLinear MSE 5 / 30 epochs
Weather	0.588	0.676	64.7	**1.1894** / **1.1851**	1.1954 / 1.1949
London	0.531	0.469	64.7	**1.1628** / **1.1671**	1.1760 / 1.1808
Wind	0.625	0.625	68.8	**0.3267** / **0.3190**	0.3443 / 0.3379
KDD	0.875	0.719	88.2	**0.4779** / **0.4783**	0.4800 / 0.4787
Electricity	0.618	0.647	76.5	**0.1812** / **0.1685**	0.2100 / 0.1946

on London, exceeding the within-dataset result of 24.2%, while the London-trained ranker achieves 47.7% on Electricity.

These results are consistent with related consumption patterns enabling useful preference transfer. Nevertheless, London-to-Electricity transfer remains below the within-dataset precision of 77.8%, indicating that transferability is asymmetric and dataset-specific adaptation remains valuable. All source rankers remain frozen and operate on the target dataset's existing archive.

4.6 Sensitivity to Predictor Training Budget (RQ6)

Table 3 supports five predictor-training epochs as a practical evaluation budget in the main experiments. Extending training to 30 epochs yields modest test MSE changes on most datasets, although Electricity benefits more from additional optimization. Architectural preferences remain broadly consistent within the fixed Top3 shortlists. Mean validation-rank correlations range from 0.531 to 0.875, and validation selects the same architecture on 72.6% of queries. These results suggest that five epochs provide informative performance estimates for architecture comparison, balancing evaluation cost and ranking reliability without training every candidate to convergence.

5 Discussion and Limitations

RankTS enables architecture recommendation with few target-side fits, but our evaluation uses fixed archives, one series split, and limited training budgets. Cross-dataset transfer still depends on target archives, while the amortization of offline search costs remains unmeasured. Future work will examine broader task distributions, causal preprocessing, and end-to-end efficiency.

6 Conclusion

We introduced RankTS, a component-level model selection framework that transfers historical architectural preferences to unseen time series. By combining compositional DAGs, diversity-aware initialization, task-local evolution, and series-conditioned pairwise ranking, RankTS recommends compact architecture shortlists for target fitting and validation selection. Across five datasets and four horizons, Top3+V ranks first in 11 of 20 settings and among the top two in 16, using three candidate fits. Future work will examine broader generalization and offline cost amortization.

AI USE STATEMENT

The main arguments, logical organization, and outline of each section were developed by the authors. OpenAI Codex assisted in expanding these outlines into prose and refining the language for clarity and fluency. It also assisted with code revisions.

ETHICS STATEMENT

This study evaluates architecture recommendation on existing time-series forecasting datasets. Applications involving household energy use, environmental observations, and operational infrastructure may raise privacy and reliability concerns. Individual-series forecasts can reveal sensitive patterns, and prediction errors can affect downstream decisions. Responsible deployment requires appropriate data access controls and assessment in the intended application setting. Dataset-specific access and redistribution conditions apply to any release of the underlying data.

REPRODUCIBILITY STATEMENT

Appendix A formalizes the learning problem, and Sections 3 and 4 describe the method and experimental protocol. Appendices B–F specify data preprocessing, operator configurations, initialization, evolution, ranker training, and forecasting baselines. Appendix H documents evaluation units, random seeds, and aggregation procedures. Extended results, recommendation analysis, retrieval protocols, and ablations are provided in Appendices G and I–K. The implementation and reproduction instructions are available at `https://anonymous.4open.science/r/RankTS-7E51`.

REFERENCES

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

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. *arXiv preprint arXiv:2310.10688*, 2023.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I Webb, Rob J Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. *arXiv preprint arXiv:2105.06643*, 2021.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Xinyu Li, Yuchen Luo, Hao Wang, Haoxuan Li, Liuhua Peng, Feng Liu, Yandong Guo, Kun Zhang, and Mingming Gong. Towards accurate time series forecasting via implicit decoding. *Advances in Neural Information Processing Systems*, 38:10003–10031, 2026.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 11116–11140, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Jiecheng Lu, Yan Sun, and Shihao Yang. In-context time series predictor. In *International Conference on Learning Representations*, volume 2025, pp. 79882–79908, 2025.

Carl H Lubba, Sarab S Sethi, Philip Knaute, Simon R Schultz, Ben D Fulcher, and Nick S Jones. catch22: Canonical time-series characteristics: Ch lubba et al. *Data mining and knowledge discovery*, 33(6):1821–1852, 2019.

Erfan Moeini, Christopher Vox, Marie Anastacio, Wadie Skaf, Mitra Baratchi, and Holger H Hoos. Neural architecture and hyperparameter selection through meta-learning on time series. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 24405–24413, 2026.

Pablo Montero-Manso, George Athanasopoulos, Rob J Hyndman, and Thiyanga S Talagala. Fforma: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92, 2020.

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

Joaquin Vanschoren. Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Zhang, and Jun Zhou. Timemixer: Decomposable multiscale mixing for time series forecasting. In *International conference on learning representations*, volume 2024, pp. 38626–38652, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. *arXiv preprint arXiv:2402.02592*, 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021a.

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

Xinle Wu, Dalin Zhang, Chenjuan Guo, Chaoyang He, Bin Yang, and Christian S Jensen. Autocts: Automated correlated time series forecasting. *Proc. VLDB Endow.*, 15(4):971–983, 2021b.

Xinle Wu, Dalin Zhang, Miao Zhang, Chenjuan Guo, Bin Yang, and Christian S Jensen. Autocts+: Joint neural architecture and hyperparameter search for correlated time series forecasting. *Proceedings of the ACM on Management of Data*, 1(1):1–26, 2023.

Xinle Wu, Xingjian Wu, Bin Yang, Lekui Zhou, Chenjuan Guo, Xiangfei Qiu, Jilin Hu, Zhenli Sheng, and Christian S Jensen. Autocts++: zero-shot joint neural architecture and hyperparameter search for correlated time series forecasting. *The VLDB Journal*, 33(5):1743–1770, 2024.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Muhan Zhang, Shali Jiang, Zhicheng Cui, Roman Garnett, and Yixin Chen. D-vae: A variational autoencoder for directed acyclic graphs. *Advances in neural information processing systems*, 32, 2019.

Tianping Zhang, Yizhuo Zhang, Wei Cao, Jiang Bian, Xiaohan Yi, Shun Zheng, and Jian Li. Less is more: Fast multivariate time series forecasting with light sampling-oriented mlp structures. *arXiv preprint arXiv:2207.01186*, 2022.

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.

Table 4: Dataset statistics and historical/query splits. Dense-label coverage counts candidates with finite validation and test metrics at H96.

Dataset	Frequency	History	Initial queries	Dense labels / query
Weather	Daily	150	17	1,997–2,000
London Smart Meters	Half-hourly	150	17	1,992–2,000
Wind Farms Minutely	Minutely	150	17	1,969–2,000
KDD Cup 2018	Hourly	150	17	1,997–2,000
Electricity Hourly	Hourly	150	17	2,000

A Problem Formulation

For each dataset d, let $\mathcal{H}_d$ and $\mathcal{Q}_d$ denote disjoint sets of historical and query series, respectively. Each univariate series is chronologically partitioned into training, validation, and test segments. During offline acquisition, only a subset of series–architecture pairs is evaluated, yielding sparse observations $\Omega_d = \{(t, g, L^{\text{val}}_{t,g}, p_{t,g})\}$, where $L^{\text{val}}_{t,g}$ is the validation MSE of architecture g on series t, and $p_{t,g}$ denotes the instantiated parameter count. Let $\mathcal{U}_d$ be the union of explored architectures and $\mathcal{A}_d \subseteq \mathcal{U}_d$ the candidate archive used for deployment. A ranker s_θ is trained from Ω_d to model series-conditioned preferences over candidate architectures.

Given an unseen series $t \in \mathcal{Q}_d$ with descriptor $\phi(t)$ and a target-side fitting budget K, the ranker recommends

$$P^K_t = \text{Top}_K\{s_\theta(\phi(t), g) : g \in \mathcal{A}_d\}. \tag{6}$$

Only architectures in P^K_t are fitted on the target series, and the final architecture is selected by validation error:

$$\hat{g}^K_t = \underset{g \in P^K_t:\, L^{\text{val}}_{t,g} \text{ finite}}{\arg\min} L^{\text{val}}_{t,g}. \tag{7}$$

Each shortlisted predictor is initialized and fitted independently on the target training segment. Historical validation supervises the ranker, target validation chooses among the fitted recommendations, and target test data are reserved for evaluation.

B Datasets and Evaluation Splits

B.1 Datasets and Series Selection

The experiments use individual univariate series from five Monash datasets (Godahewa et al., 2021): Weather, London Smart Meters, Wind Farms Minutely, KDD Cup 2018, and Electricity Hourly. Table 4 gives their sampling frequencies and series counts. Weather uses the Monash daily dataset. Each dataset contributes 150 historical series and a disjoint query pool of 17 series, for 750 historical and 85 query series in total. Series identities, rather than windows, define the outer split.

For a series of length T, the chronological training and validation segments contain $\lfloor 0.6T \rfloor$ and $\lfloor 0.2T \rfloor$ observations; the remainder forms the test segment. Eligibility requires at least 1,024 training windows of length $96 + 96$ and at least 128 validation and test targets at horizon 96. Eligibility is checked before the series permutation. All selected queries support $H \in \{96, 192, 336, 720\}$. Validation and test inputs may include preceding observations, while each target remains within its assigned segment.

B.2 Preprocessing and Window Sampling

Predictor inputs use training-segment means and standard deviations. Missing values are interpolated over each series before chronological partitioning. All compared predictors share this preprocessing. Each fit samples at most 4,096 training windows and 1,024 validation and test windows, using sampling seed 0 independently of predictor initialization. Input length is fixed at 96. Changing the horizon changes the available windows; comparisons pair methods within each horizon.

Series descriptors use the chronological training segment. Ranker imputation and normalization are estimated from historical fitting tasks during epoch selection and from all historical tasks during

Table 5: Representative operators in the 54-operator DAG space. Table 6 lists the complete vocabulary.

Operator family	Representative operators	Properties
Linear and local	Temporal/channel linear, convolution, pooling	Bias, kernel, dilation
Attention and recurrence	Temporal/patch attention, GRU, LSTM	Width, heads, patch size
Trend and frequency	Moving average, difference, FFT, wavelets	Kernel, order, modes
Mixing and dynamics	Multiscale mixers, state space, Koopman	Scales, state/latent size
Supporting operations	Normalization, activation, gates, dropout	Type, gate, dropout rate

final refitting. Query series do not contribute to these estimates. The historical split contains 135 fitting and 15 validation tasks per dataset, with permutations determined by ranker seeds 0, 1, and 2.

B.3 Primary Query Set and Diagnostic Subsets

The primary comparisons share a post-hoc distribution-shift-screened cohort of 84 queries across methods and horizons: 16 from Wind and 17 from each other dataset.

Retrieval evaluates common candidate sets with finite cached validation and test metrics at H96. The Electricity cache covers all 2,000 archive candidates per query; the other datasets have the finite coverage shown in Table 4. These predictor labels remain fixed across ranker seeds and are distinct from the four-horizon forecasting fits.

The cost study uses three queries per dataset. Within each retained query pool, series are sorted by length and task identity; three equally spaced interior positions select the diagnostic queries without using model outcomes. Training-budget sensitivity uses all retained H96 queries. The evolution study uses one prespecified historical task per dataset, and initialization uses generation seeds 1000–1009.

C Architecture Space and Feasibility

C.1 Operators and Hyperparameter Sets

A forecasting architecture is a typed DAG with at most ten nodes and a linear prediction head. Each node specifies its operator, discrete hyperparameters, predecessor identities, merge rule, and residual flag. Graph-level attributes specify the output node and preprocessing. Table 5 summarizes the building blocks; Table 6 lists all 54 operators in 11 operator families.

Recurrent neural network (RNN) operators include GRU and long short-term memory (LSTM) units. ReLU, SiLU, and RMS denote rectified linear unit, sigmoid linear unit, and root mean square, respectively.

Table 6: Operator vocabulary by computational role and configurable properties. Table 7 lists the discrete hyperparameter values.

Operator family	Operators	Hyperparameter examples
Linear	identity, channel linear, temporal linear	bias
Local	conv1d, causal conv1d, depthwise conv1d, pointwise conv1d, average pooling, max pooling, temporal convolution, inception convolution, isometric convolution	kernel, dilation, scale
Attention	temporal attention, variable attention, patch attention, two-stage attention, pyramidal attention	hidden size, heads, patch size, scales
Recurrence	GRU, LSTM, segment recurrent	hidden size, segment length
Trend	moving average, decomposition, difference, exponential smoothing	kernel, order, smoothing coefficient
Frequency	FFT, frequency mask, complex linear, IFFT, spectral filter, wavelet filter, frequency mixer, Legendre projection, autocorrelation	modes, levels, order, top-k
Mixing	channel mixer, temporal mixer, patch mixer, multiscale mixer, sparse mixture of experts	expansion, patch size, scales, experts
Dynamics	selective state space, Koopman operator	state size, latent size
Variable graphs	adaptive graph convolution, dynamic graph filter, channel cluster	top-k, patch size, clusters
Support	layer/RMS/batch normalization, scalar gate, gated residual, dropout	gate initialization, expansion, dropout
Activation / position	ReLU, GELU, SiLU, sigmoid, positional encoding	position base

Table 7 specifies the 227 operator–hyperparameter configurations used for coverage-based selection. Multiple hyperparameter sets within a row form Cartesian products. A multiscale setting denotes one ordered set of scales. These operator hyperparameters are separate from predictor-training hyperparameters.

Table 7: Discrete hyperparameter sets for the 54 operators.

Operators	Hyperparameter values
identity, pointwise convolution, layer norm, RMS normalization, batch norm, ReLU, GELU, SiLU, sigmoid, FFT, complex linear, IFFT	No discrete operator hyperparameters
channel linear, temporal linear	bias: $\{0, 1\}$
1D convolution, depthwise convolution, average pooling, max pooling	kernel: $\{3, 5, 7\}$
causal convolution	dilation: $\{1, 2, 4\}$; kernel: $\{3, 5\}$
variable attention, temporal attention, two stage attention	heads: $\{1, 2, 4\}$; width: $\{16, 32, 64\}$
GRU, LSTM	width: $\{8, 16, 32\}$
dropout	dropout (%): $\{5, 10, 20, 30\}$
moving average, decomposition	kernel: $\{3, 7, 13, 25\}$
difference	order: $\{1, 2\}$
frequency mask	modes: $\{4, 8, 12, 16\}$
positional encoding	base: $\{100, 1000, 10000\}$
scalar gate	gate initialization (%): $\{25, 50, 75\}$
spectral filter, frequency mixer	modes: $\{4, 8, 16, 32\}$
wavelet filter	levels: $\{1, 2, 3\}$
temporal conv	dilation: $\{1, 2, 4\}$; kernel: $\{3, 5, 7\}$
channel mixer, temporal mixer, gated residual	expansion: $\{1, 2, 4\}$
multiscale mixer	scales: $\{\{2, 4\}, \{2, 4, 8\}\}$

Continued on next page

Table 7 (continued)

Operators	Hyperparameter values
exponential smoothing	smoothing (%): $\{10, 30, 50, 70, 90\}$
legendre projection	order: $\{8, 16, 32\}$
inception conv	maximum kernel: $\{3, 5, 7\}$
isometric conv	kernel: $\{3, 5\}$; scale: $\{2, 4\}$
patch mixer	expansion: $\{1, 2, 4\}$; patch length: $\{4, 8, 16\}$
patch attention	heads: $\{1, 2, 4\}$; width: $\{16, 32, 64\}$; patch length: $\{4, 8, 16\}$
segment recurrent	width: $\{8, 16, 32\}$; segment length: $\{4, 8, 12\}$
autocorrelation, adaptive graph conv	top-k: $\{1, 3, 5\}$
pyramidal attention	heads: $\{1, 2, 4\}$; width: $\{16, 32\}$; scales: $\{\{2, 4\}, \{2, 4, 8\}\}$
selective state space	state size: $\{8, 16, 32, 64\}$
koopman operator	latent size: $\{8, 16, 32\}$
dynamic graph filter	patch length: $\{4, 8, 16\}$; top-k: $\{1, 3, 5\}$
channel cluster	clusters: $\{2, 4, 8\}$
sparse mixture of experts	expansion: $\{1, 2\}$; experts: $\{2, 4\}$; top-k: $\{1, 2\}$

C.2 Execution and Structural Validity

Nodes are ordered topologically. A node first merges compatible predecessor tensors by mean, sum, learned gating, or ordered subtraction, applies its operator, and optionally adds a residual connection. Operators preserve or restore the sequence/channel interface; the final linear head maps the time-domain representation to the forecast horizon. Every retained node lies on a path to the designated output.

FFT and IFFT nodes explicitly change the representation domain. Frequency masks and complex linear transformations operate between these transitions; other spectral operators encapsulate their transforms within a time-domain interface. Structural filtering removes duplicate inputs, disconnected nodes, redundant consecutive support operations, empty FFT–IFFT round trips, and incompatible domains. Graph identity is defined by the ordered architecture specification, including hyperparameters and preprocessing.

Preprocessing comprises z-score normalization, robust scaling, differencing, and detrending. These transformations precede the DAG. A graph identity specifies an architecture; its forecasting weights are fitted separately for every series, horizon, and predictor seed. The reported experiments use one channel per series, including when an operator admits a multivariate formulation.

C.3 Resource Screening and Fit Eligibility

The initialization screen allows at most 50 million parameters per graph and uses a 20 GiB planning envelope with input length 96, horizon 96, 862 channels, and requested batch size 128. This common conservative planning shape is used by all initialization controls; downstream forecasting uses the actual univariate task shape. Complete-model baselines allow up to 250 million parameters.

Predictor training uses micro-batch eight and effective batch 128 through gradient accumulation. Candidates that fail structural, resource, or execution checks remain part of the attempted-fit budget. Only finite positive historical validation losses provide pairwise supervision. At deployment, validation eligibility is assessed within the fixed Top-K prefix; a failed candidate does not expand the prefix, and an entirely invalid prefix yields no selected model. Runtime and memory protocols are detailed in Appendix F.

D Initialization, Evolution, and Archive Construction

Algorithm 2 describes task-local acquisition. This appendix specifies how the shared population, proposal mechanisms, and deployment archive are constructed.

Table 8: Four architecture skeletons for hierarchical candidate generation.

Skeleton	Probability	Construction
Chain	0.42	Sample one to three sequential blocks, subject to the node budget. Each block samples an operator family and then an operator and hyperparameter tuple.
Parallel	0.30	Sample two different families, apply one undecorated block from each to the input, and merge by mean, sum, or gating. Add a further block with probability 0.75 when space permits.
Decomposition	0.18	Extract a moving-average trend with kernel in $\{3, 7, 13, 25\}$ and form a residual by subtraction. Process both branches using linear, local, mixing, frequency, or recurrent blocks, then merge by sum or gating. Add a final block with probability 0.55 when feasible.
Frequency path	0.10	Apply FFT, an optional frequency mask (probability 0.65), complex linear transformation, and IFFT. Add a time-domain block with probability 0.8 when feasible.

D.1 Hierarchical Candidate Generation

The base catalog contains heuristic recipes, single-operator hyperparameter variants, ordered operator pairs, and merge, residual, and preprocessing variants. The main initialization starts with 282 distinct base graphs and adds hierarchical proposals until the serialization-deduplicated pool contains 2,000 graphs. The generator first selects a skeleton, then compatible operator families, operators within those families, and a complete discrete hyperparameter tuple from the operator catalog. Family membership during generation expresses computational roles; it is not a separate learned classifier.

An eligible decorated block adds normalization with probability 0.45, activation with probability 0.42, and dropout with probability 0.22, subject to remaining node capacity. Residual connections are sampled with probability 0.55 for compatible operators. Preprocessing chooses z-score scaling with probability 0.28, robust scaling with probability 0.14, or neither otherwise. Independently, a stationarity transform is added with probability 0.18: differencing with conditional probability 0.6, otherwise detrending. Differencing orders are sampled from $\{1, 2\}$.

Structural filtering requires a linear head, one to ten nodes, and a path from every node to the designated output. It rejects duplicate inputs, redundant consecutive support operators, empty FFT–IFFT round trips, and graphs lacking an informative transformation. The main initialization retains 1,965 valid graphs. Deduplication uses the ordered graph specification.

D.2 Coverage and Structural Scoring

Let $C(g)$ contain the distinct pairs of an operator and its complete discrete hyperparameter tuple present in graph g. Let $\mathcal{U}_{\text{cfg}}$ be the catalog of 227 such configurations. For the selected set S, marginal coverage gain is

$$G(g \mid S) = \left| C(g) \cap \left(\mathcal{U}_{\text{cfg}} \setminus \bigcup_{g' \in S} C(g') \right) \right|. \tag{8}$$

Repeated occurrences of an identical configuration count once; changing a hyperparameter tuple creates a different coverage item. For example, if S covers A_1 and B_1, then $A_1 \to B_2 \to C_1$ adds two configurations. This score concerns coverage across the population rather than complexity within one graph.

The complementary structural heuristic is

$$\begin{aligned} Q(g) = {} & 2|F(g)| + I_{\text{multi}} + 0.8 I_{\text{residual}} + 0.6 I_{\text{pre}} \\ & + 1.5 I_{\text{trend-linear}} + 1.5 I_{\text{variable-attention}} + I_{\text{frequency-path}} - 0.08\big||V(g)| - 6\big|, \end{aligned} \tag{9}$$

where $F(g)$ is the set of represented generator families. The indicators denote a multi-input node, any residual, any preprocessing, joint presence of moving average and temporal linear operators,

variable attention, and joint presence of FFT, complex linear, and IFFT, respectively. The final term mildly favors six-node graphs. $Q(g)$ encodes structural preferences without forecasting labels.

Starting with $S = \varnothing$, greedy selection maximizes $(G(g \mid S), Q(g), \mathrm{ID}(g))$ lexicographically and updates uncovered configurations after each choice. Coverage selection requires all catalog options to be represented. The main initialization uses 107 greedy selections to cover the catalog; adding valid representative recipes and deduplicating gives 118 seed architectures. Thus, coverage is the primary criterion and the heuristic breaks equal-gain ties.

D.3 Bucket Completion and Persistent Anchors

For each unselected graph, the bucket key contains four attributes: its single generator family, or a shared hybrid category for multiple families; the node-count bin $\min(3, \lfloor(|V|-1)/3\rfloor)$; preprocessing category; and whether any node has multiple inputs. Preprocessing categories are none, scaling only, stationarity only, and combined. In particular, hybrid graphs do not receive a separate bucket for every exact family combination.

Within each bucket, graphs are ordered by decreasing $Q(g)$ with graph-ID tie breaking. Round-robin traversal takes one feasible graph from each active bucket and requeues nonempty buckets until the population reaches 300. This step controls the distribution of structures even after all operator–hyperparameter configurations have been covered. A graph rejected by resource profiling is skipped during completion; an infeasible mandatory coverage seed instead causes initialization to fail. No forecasting labels enter these decisions.

The resulting ordered population is shared across historical tasks, and its first 90 entries are persistent anchors. The counts $282 \to 2{,}000 \to 1{,}965 \to 107 \to 118 \to 300$ refer to successive generation and selection stages; the anchor count 90 is a separate retention decision. Counts beyond the recorded seed can vary with generation, while population and anchor budgets remain fixed. Appendix K.4 evaluates these design choices under matched initialization and exploration budgets.

D.4 Survivor selection

Eligible survivors have successful status and finite positive validation MSE. Let a_t be the number of eligible anchors and $s = 150 - a_t$ the local capacity. Selection preserves these anchors, admits up to $\max(1, \lfloor s/2 \rfloor)$ additional candidates in validation-loss order, and then admits up to $\max(1, \lfloor s/4 \rfloor)$ additional performance–resource candidates. The latter are obtained by sorting by loss, parameter count, runtime, and graph ID, and retaining entries that improve either the smallest parameter count or the smallest runtime observed in that scan. This scan supplies survivors with complementary resource costs.

Structural diversity fills the remaining slots. Each step selects the candidate with the greatest minimum Euclidean distance to already selected graphs in the structural-feature representation. Validation order provides a final fallback. The procedure requires 150 valid survivors; insufficient valid evaluations cause the task-local search to fail explicitly. With all anchors valid, the allocation begins with 90 anchors and 30 elites, adds at most 15 resource tradeoffs, and fills the rest through diversity.

D.5 Proposal generation and evaluation budget

Each proposal batch targets 75 mutations, 30 crossovers, 30 surrogate-guided candidates, and 15 random candidates. Mutation can replace an operator, alter hyperparameters, add or delete a terminal node, swap node configurations, modify skip inputs or residual flags, and change preprocessing. Grammar checks reject invalid mutations. Crossover replaces one node's operator and hyperparameters using a donor with matching input/output domains, while preserving the recipient connections and enforcing residual compatibility. Preprocessing can be inherited from either parent.

The task-local surrogate is a random forest with 96 trees, minimum leaf size three, and feature fraction 0.5. It regresses log validation MSE on graph features from that task's successful observations. A hierarchical pool of $\max(120, 6n_s)$ proposals, with $n_s = 30$, is ordered by predicted mean minus 0.5 times the standard deviation across trees. With fewer than 30 valid training observations, the first pool entries are used instead. This surrogate is distinct from the cross-task neural ranker.

Every source passes duplicate, grammar, structural, and resource checks. Candidates already evaluated on the task are excluded. Mutation and crossover have finite attempt limits; random proposals fill any remaining quota. Consequently, realized source counts can differ from the nominal allocation. Each predictor is initialized independently, with at most five training epochs. The two-round experiment evaluates 300 initial graphs and 150 new graphs while reusing survivor measurements, giving 450 distinct attempts per historical task. With 150 historical tasks per dataset, the nominal offline budget is 67,500 attempts per dataset, or 337,500 across the five datasets. The resulting archive and preference observations are reused for subsequent queries. Invalid attempts do not supply ranking preferences.

This offline acquisition budget is separate from the K predictor fits used for a new query. Its average contribution per query depends on how many queries share the archive; the target-fitting curves measure the online fitting component.

Acquisition is capped at two rounds. The stopping rule uses patience two for relative validation improvement below 0.005 or Top20 Jaccard overlap of at least 0.9; the two-round cap bounds the main acquisition budget.

D.6 Specialist-preserving archive selection

Let $B = 2{,}000$, a be the number of retained anchors, and T the number of historical tasks with valid observations. The per-task specialist quota is

$$q = \max\left(1, \min\left(25, \left\lfloor \frac{\max(0, B - a)}{2T} \right\rfloor\right)\right). \tag{10}$$

Anchors enter in deterministic graph-ID order, followed by the q lowest-validation-loss graphs per task, with identities deduplicated. The configured $a = 90$ and $T = 150$ give $q = 6$. Remaining graphs are ordered lexicographically by median within-task loss percentile, mean percentile, decreasing task coverage, and graph ID. Percentiles are ascending validation-loss ranks with average ties, divided by the number of valid observations for that task. If these observations do not fill the available catalog budget, structural-diversity selection adds remaining graph identities. The result is truncated to B.

Ranker supervision uses successful observations from the larger historical graph union, while online scoring uses the bounded archive. Unobserved task–graph pairs remain missing. The acquisition comparisons in Appendix K.4 evaluate the initialization process and proposal quality under matched budgets.

E Ranker Representation and Training

E.1 Series Descriptors

The main ranker uses the 22 canonical catch22 descriptors (Lubba et al., 2019), computed from the complete chronological training segment at its original sampling interval. Nonfinite values are imputed by per-feature medians from the historical fitting tasks; an entirely missing feature is imputed with zero. Historical means and standard deviations standardize the features, with unit scale for dimensions whose standard deviation is below 10^{-3}. The main descriptor consists of these 22 features alone.

A $22 \to 96 \to 96$ MLP with GELU and final LayerNorm produces h_t. Preprocessing statistics are estimated on 135 tasks during historical holdout training and on all 150 tasks during final refitting. Deployment and cross-dataset transfer retain the fitted source statistics.

E.2 Graph Branches and Compatibility Scorer

Graphs are padded to ten nodes, and padding is masked from all readouts. Each initial 96-dimensional node state combines operator, position, merge, residual, and direct-input embeddings with a projection of numerical operator hyperparameter values and applicability masks. The masks distinguish absent hyperparameters from valid zero-valued settings; structural values are scaled by their hyperparameter catalog.

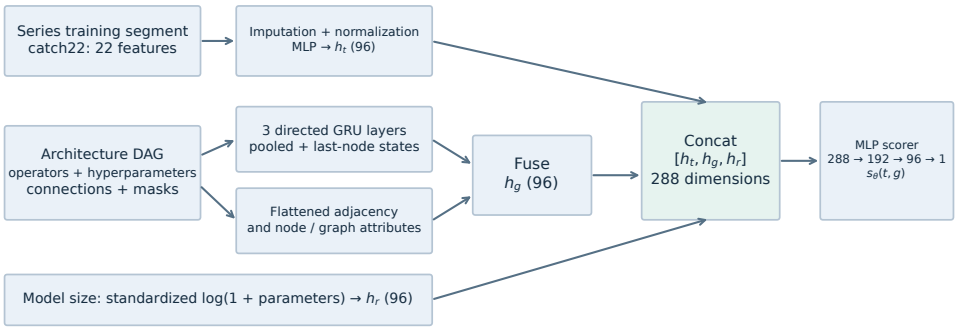


Figure 8: Architecture of the series–architecture compatibility scorer. Series, architecture, and model-size embeddings jointly determine the compatibility score.

Three directed message-passing layers apply the predecessor aggregation and GRU update in Eq. (1). Concatenating the mean and maximum node states, the last active node, and graph-level attributes gives a 384-dimensional readout. A linear projection, GELU, and LayerNorm produce $h_{\mathrm{MP}} \in \mathbb{R}^{96}$. The graph-level term combines the prediction head, preprocessing, node count, and connection statistics.

A second branch flattens ordered adjacency, operator indicators, node hyperparameters and masks, merge and connection attributes, and graph-level attributes. A linear projection, GELU, and LayerNorm produce $h_{\mathrm{flat}} \in \mathbb{R}^{96}$. The architecture embedding is

$$h_g = \mathrm{LayerNorm}(\mathrm{GELU}(W_f[h_{\mathrm{MP}}, h_{\mathrm{flat}}] + b_f)), \qquad h_g \in \mathbb{R}^{96}. \tag{11}$$

Both branches represent the supplied topological ordering.

Model size is represented separately. The scalar $\log(1 + p_{t,g})$ is standardized using historical instantiated parameter counts, with a standard-deviation floor of 10^{-4}. A linear projection and GELU give $h_r \in \mathbb{R}^{96}$. The scorer concatenates $[h_t, h_g, h_r]$ and applies widths $288 \to 192 \to 96 \to 1$, with GELU activations and dropout 0.1 after the first hidden layer. All learned branches are optimized jointly.

E.3 Alternative Series Encoders

The 84-statistic alternative summarizes distributional statistics, trends and regime shifts, first- and second-order differences, autocorrelations, spectral bands and shape, fixed-filter responses, multi-scale energy, and channel dependence. Training length, window count, channel count, horizon ratio, and frequency indicators complete the descriptor. Continuous features use historical normalization; frequency indicators retain their categorical encoding. The series projection and downstream scorer match the main model apart from input dimensionality.

The CNN alternative takes eight uniformly spaced 256-point windows from the training segment, standardized by its mean and standard deviation. Its three Conv1D–GELU stages use channels 16/32/64, kernels 7/5/3, stride two, and padding 3/2/1. Temporal mean/max pooling gives 128 features per window, averaged across windows. The training mean and $\log(1 + \mathrm{std})$, standardized across historical tasks, add two scalar inputs. A $130 \to 96$ projection and LayerNorm produce the series embedding. Architecture encoding, scoring, and pairwise supervision are shared across the three series representations.

E.4 Balanced Pair Construction

For a historical task with N finite positive validation losses, define $h = \max(20, \mathrm{round}(0.1N))$. Boundaries $3, 20, h, \lceil 0.3N \rceil, \lceil 0.6N \rceil$ partition validation ranks into up to six bands. Each unordered pair of bands forms a stratum. From its absolute log-loss gaps greater than 10^{-3}, compute the 10th percentile $q_{.1}$ and median $q_{.5}$ when at least 16 gaps are available. Otherwise, use the fixed lower floor and the median available gap. If no such gaps exist, both cutoffs equal the floor. Eligible pairs satisfy $|\Delta u| > \max(10^{-3}, q_{.1})$.

Head comparisons have an endpoint in the best 20 and both endpoints within the best h. Cross-stratum comparisons span different bands and have a worse-ranked endpoint beyond h. Semi-hard comparisons have an endpoint within the best 30% and a gap at most the stratum's upper cutoff. These eligible sets can overlap, but each sampled pair appears only once. These task-local thresholds retain comparisons with informative validation-loss separation.

Let d_{ij} be binary structural-token Jaccard distance, using operators, directed connections, depth, merge rules, residuals, and preprocessing. Architecture skeleton signatures preserve indexed operators and connections while ignoring numerical hyperparameters. If f_i is the frequency of candidate i's skeleton within its performance band, the structural sampling weight is

$$a_{ij} = \frac{(0.1 + 0.9d_{ij})^2}{\sqrt{f_i f_j}}. \tag{12}$$

Weights are further divided by the number of currently available pairs in the corresponding band-pair stratum. Head pairs crossing the historical Top3 boundary receive a factor of two, as do semi-hard pairs touching the best 20. These factors encourage structural variation and balanced performance comparisons without changing the preference label.

The sampler targets 30% head, 30% cross-stratum, 20% semi-hard, and 20% random pairs, with at most 1,024 pairs per task. It reserves semi-hard pairs first, then selects head and cross-stratum pairs without replacement. Uniform draws from the remaining eligible pairs fill the random allocation and any unmet quota. Pair orientation is randomized, and the resulting static pairs are shuffled across epochs. The Uniform ablation retains the same eligibility thresholds but samples the entire budget uniformly. No structural weighting sets $a_{ij} = 1$ while retaining band-pair balancing, boundary boosts, and allocation quotas.

E.5 Optimization and Historical Epoch Selection

For utility $u_{t,g} = -\log L^{\mathrm{val}}_{t,g}$, the main objective is the weighted pairwise logistic loss in Eq. (4). A pair's loss weight is its absolute utility gap divided by the median gap of the training pairs, clipped to $[0.25, 4]$. These loss weights are separate from the structural weights used during pair sampling.

Training uses AdamW (Loshchilov & Hutter, 2017) with learning rate 2×10^{-3}, weight decay 10^{-3}, pair batch size 512, gradient clipping at five, and at most 60 epochs. Each seed partitions historical tasks into 135 fitting and 15 validation tasks. Mean historical validation NDCG@20 selects the epoch count. For a task with N observed candidates and zero-based ascending validation-loss rank r, selection relevance is $\max(0, 1 - r/\max(1, \mathrm{round}(0.1N)))$. Checkpoints are eligible from epoch three; early stopping is enabled after epoch 15 with patience ten and minimum improvement 10^{-4}.

After epoch selection, the model is reinitialized with the same seed and fitted on all 150 historical tasks for the selected number of epochs. Imputation, feature scaling, sampled pairs, and utility-gap normalization are recomputed for this refit. Historical observations and the deployment archive remain fixed across configurations. The resulting ranker is frozen before query recommendation.

F Forecasting Baselines and Compute Protocol

F.1 Complete-Model Configurations

The comparison includes DLinear, iTransformer, PatchTST, TimesNet, FEDformer, TimeMixer, Autoformer, Informer, Transformer, Nonstationary Transformer, and LightTS (Zeng et al., 2023; Liu et al., 2024; Nie et al., 2022; Wu et al., 2022; Zhou et al., 2022; Wang et al., 2024; Wu et al., 2021a; Zhou et al., 2021; Vaswani et al., 2017; Liu et al., 2022; Zhang et al., 2022). Table 1 abbreviates PatchTST, iTransformer, TimesNet, TimeMixer, FEDformer, Autoformer, Informer, Transformer, Nonstationary Transformer, and LightTS as Patch, iTrans, Times, Mixer, FED, Auto, Inform, Trans, Nonstat, and Light, respectively. The eleven families comprise thirteen configurations: iTransformer small/base/large use $(d_{\mathrm{model}}, n_{\mathrm{heads}}, e_{\mathrm{layers}}, d_{\mathrm{ff}}) = (32, 4, 1, 64), (64, 4, 2, 128), (128, 8, 2, 256)$, and target validation selects among these three configurations. Other configurable deep baselines use $(64, 4, 2, 128)$ where applicable. PatchTST uses patch length 16 and stride 8.

All predictors are fitted independently on individual univariate series. Task splits and the training budget are shared across methods within a comparison, while each model's forecasting weights are

specific to its target series. The complete-model comparison evaluates these configurations under the common bounded-training protocol.

F.2 Predictor Training and Validation Selection

Predictors use Adam with learning rate 0.001, gradient clipping at five, micro-batch eight, and effective batch 128 through accumulation. The final partial accumulation group is normalized by its actual size. The five-epoch protocol uses a minimum of three epochs and stopping patience two. Window limits and sampling are specified in Appendix B.

Forecasting uses paired ranker/predictor seeds 0, 1, and 2. Top1 is the first member of the frozen list used by Top3+V; shared candidates use identical fitted models within a query, horizon, and seed. Target validation selects among eligible candidates in the prescribed prefix. The selected identity is fixed before test evaluation. A failed fit remains an attempted candidate, and the prefix is not extended. The complete comparison contains 16,128 method–query–horizon–seed outcomes and 320 aggregate cells.

F.3 Historical Model-Library Selection

Historical best (Hist) and Meta-select (Select) use the same thirteen complete-model configurations. Each configuration is fitted to all 150 historical series per dataset, for 9,750 historical fits. Within-task median-centered log validation losses provide the selection targets; a failed fit receives a loss two log units above the worst finite value for that task.

Hist selects the configuration with the lowest mean centered historical loss. Select uses a multi-output random forest on 84 training-segment statistics to predict the centered losses. The forest has 256 trees, feature fraction 0.7, and seed 0. A 135/15 historical split selects minimum leaf size from $\{1, 3, 5\}$; the forest is then refitted on all history. H96 model identities are retained across horizons. Each policy fits one target predictor; RankTS Top3+V fits three and selects among them by target validation.

F.4 Training-Budget Sensitivity

The H96 seed-0 comparison fixes each query's Top3 identities. Each candidate and DLinear are evaluated under the five-epoch protocol and a 30-epoch protocol with minimum ten epochs and stopping patience eight. Validation selection is repeated within the same shortlist. This intervention changes both the epoch cap and stopping schedule.

Spearman correlations compare the aligned candidate MSEs between the two budgets, separately on validation and test. Correlations require three finite paired evaluations with nonconstant losses. One query in each of London and KDD has two finite paired candidates and is excluded from correlation aggregation; all retained queries contribute to selected MSE and selection agreement. The numbers of correlation-eligible queries are 17, 16, 16, 16, and 17 for Weather, London, Wind, KDD, and Electricity. Agreement is the percentage selecting the same validation winner at both budgets.

F.5 Resources and Timing

Graph planning and parameter limits are given in Appendix C. Predictor runs retain micro-batch eight and effective batch 128. The forecasting records include runs under different process-memory limits; additional fits use a nominal 10^9-byte process limit, a 7×10^8-byte allocator budget, and a 9.5×10^8-byte sampled process threshold. The matched evolution study uses a common 3 GB protocol.

A fitted model is reused only when data, architecture, predictor seed, sampled windows, optimization schedule, implementation, and actual batch settings agree. Reused measurements retain their original resource conditions. Cost curves sum recorded wall times for every attempted candidate in a logical prefix, including compatible reused fits. Runs share GPUs, so reported times describe this execution setting. The curves measure target-side predictor fitting. Average cost per query additionally includes an amortized share of offline acquisition and ranker training, as well as descriptor extraction and archive scoring. The amortized share depends on the number of queries served.

Table 9: Test MSE (↓; mean ± SD over three seeds) under the protocol of Table 1.

Dataset	H	Top1	Top3+V	Hist	Select	DLinear	iTrans	Patch	Times	FED	Mixer	Auto	Informer	Trans	Nonstat	Light	CM3+V
Weather	96	1.186 ± 0.022	1.182 ± 0.021	1.208 ± 0.053	1.208 ± 0.053	1.195 ± 0.002	1.779 ± 0.016	1.682 ± 0.012	1.741 ± 0.029	1.836 ± 0.436	1.695 ± 0.014	2.114 ± 0.051	1.189 ± 0.006	1.208 ± 0.053	1.286 ± 0.053	1.283 ± 0.012	1.183 ± 0.012
	192	1.050 ± 0.024	1.035 ± 0.023	1.043 ± 0.019	1.043 ± 0.019	1.049 ± 0.006	1.851 ± 0.010	1.876 ± 0.021	1.920 ± 0.034	2.021 ± 0.415	1.788 ± 0.044	2.444 ± 0.041	1.062 ± 0.017	1.043 ± 0.019	1.155 ± 0.054	1.131 ± 0.003	1.045 ± 0.018
	336	0.993 ± 0.024	0.981 ± 0.016	1.000 ± 0.005	1.000 ± 0.005	0.989 ± 0.000	1.524 ± 0.023	1.540 ± 0.016	1.726 ± 0.025	1.779 ± 0.308	1.500 ± 0.048	1.975 ± 0.017	1.016 ± 0.007	1.000 ± 0.005	1.048 ± 0.017	1.058 ± 0.004	0.981 ± 0.009
	720	1.302 ± 0.010	1.293 ± 0.013	1.314 ± 0.006	1.314 ± 0.006	1.299 ± 0.002	1.825 ± 0.014	1.897 ± 0.011	1.956 ± 0.024	2.229 ± 0.005	1.908 ± 0.027	2.210 ± 0.007	1.336 ± 0.008	1.314 ± 0.006	1.837 ± 0.095	1.373 ± 0.003	1.286 ± 0.005
London	96	1.174 ± 0.008	1.184 ± 0.021	1.182 ± 0.003	1.189 ± 0.003	1.176 ± 0.001	1.192 ± 0.003	1.215 ± 0.002	1.191 ± 0.003	1.230 ± 0.009	1.182 ± 0.003	1.283 ± 0.022	1.308 ± 0.003	1.238 ± 0.004	1.267 ± 0.011	1.227 ± 0.002	1.174 ± 0.001
	192	1.200 ± 0.006	1.205 ± 0.015	1.226 ± 0.004	1.228 ± 0.001	1.205 ± 0.002	1.226 ± 0.001	1.246 ± 0.002	1.227 ± 0.002	1.287 ± 0.028	1.226 ± 0.004	1.381 ± 0.010	1.363 ± 0.007	1.291 ± 0.011	1.315 ± 0.013	1.254 ± 0.001	1.227 ± 0.008
	336	1.209 ± 0.009	1.204 ± 0.005	1.245 ± 0.002	1.243 ± 0.008	1.214 ± 0.001	1.244 ± 0.001	1.258 ± 0.002	1.245 ± 0.008	1.320 ± 0.057	1.245 ± 0.002	1.438 ± 0.005	1.398 ± 0.002	1.311 ± 0.011	1.323 ± 0.004	1.261 ± 0.000	1.239 ± 0.004
	720	1.260 ± 0.007	1.250 ± 0.009	1.309 ± 0.002	1.303 ± 0.002	1.273 ± 0.001	1.306 ± 0.001	1.313 ± 0.002	1.306 ± 0.004	1.413 ± 0.048	1.309 ± 0.002	1.528 ± 0.002	1.459 ± 0.003	1.392 ± 0.030	1.377 ± 0.007	1.315 ± 0.001	1.303 ± 0.002
Wind	96	0.337 ± 0.001	0.329 ± 0.002	0.376 ± 0.002	0.362 ± 0.002	0.345 ± 0.002	0.389 ± 0.006	0.367 ± 0.004	0.396 ± 0.009	0.462 ± 0.042	0.376 ± 0.002	0.542 ± 0.009	0.487 ± 0.003	0.451 ± 0.006	0.361 ± 0.003	0.364 ± 0.003	0.356 ± 0.005
	192	0.503 ± 0.012	0.502 ± 0.009	0.605 ± 0.008	0.581 ± 0.008	0.501 ± 0.004	0.621 ± 0.004	0.592 ± 0.005	0.639 ± 0.003	0.672 ± 0.055	0.605 ± 0.008	0.770 ± 0.002	0.601 ± 0.005	0.576 ± 0.011	0.583 ± 0.009	0.512 ± 0.002	0.624 ± 0.013
	336	0.634 ± 0.006	0.632 ± 0.005	0.812 ± 0.009	0.758 ± 0.021	0.640 ± 0.001	0.850 ± 0.003	0.804 ± 0.011	0.860 ± 0.011	0.918 ± 0.094	0.812 ± 0.009	1.028 ± 0.001	0.714 ± 0.004	0.683 ± 0.004	0.775 ± 0.018	0.647 ± 0.003	0.724 ± 0.001
	720	0.834 ± 0.006	0.835 ± 0.007	1.162 ± 0.014	1.046 ± 0.010	0.816 ± 0.001	1.195 ± 0.011	1.156 ± 0.004	1.205 ± 0.010	1.372 ± 0.030	1.162 ± 0.014	1.387 ± 0.001	0.824 ± 0.007	0.824 ± 0.008	1.094 ± 0.015	0.815 ± 0.003	0.873 ± 0.000
KDD	96	0.491 ± 0.002	0.480 ± 0.003	0.551 ± 0.008	0.491 ± 0.000	0.480 ± 0.001	0.555 ± 0.001	0.553 ± 0.002	0.564 ± 0.004	0.550 ± 0.029	0.551 ± 0.008	0.589 ± 0.005	0.518 ± 0.003	0.511 ± 0.005	0.526 ± 0.004	0.498 ± 0.001	0.483 ± 0.002
	192	0.503 ± 0.001	0.489 ± 0.003	0.577 ± 0.001	0.507 ± 0.002	0.500 ± 0.001	0.583 ± 0.004	0.575 ± 0.005	0.588 ± 0.002	0.569 ± 0.030	0.577 ± 0.001	0.604 ± 0.002	0.530 ± 0.001	0.516 ± 0.001	0.527 ± 0.001	0.514 ± 0.000	0.496 ± 0.002
	336	0.507 ± 0.002	0.498 ± 0.001	0.595 ± 0.005	0.516 ± 0.001	0.513 ± 0.001	0.601 ± 0.002	0.592 ± 0.002	0.604 ± 0.007	0.590 ± 0.037	0.595 ± 0.005	0.620 ± 0.001	0.534 ± 0.008	0.520 ± 0.003	0.540 ± 0.008	0.524 ± 0.001	0.520 ± 0.003
	720	0.501 ± 0.003	0.496 ± 0.001	0.633 ± 0.001	0.525 ± 0.001	0.513 ± 0.001	0.637 ± 0.002	0.621 ± 0.002	0.641 ± 0.005	0.637 ± 0.004	0.633 ± 0.001	0.638 ± 0.001	0.516 ± 0.002	0.513 ± 0.001	0.529 ± 0.000	0.521 ± 0.002	0.513 ± 0.001
Electricity	96	0.190 ± 0.011	0.181 ± 0.001	0.187 ± 0.005	0.187 ± 0.005	0.211 ± 0.001	0.187 ± 0.001	0.182 ± 0.000	0.187 ± 0.005	0.248 ± 0.004	0.196 ± 0.002	0.299 ± 0.019	0.598 ± 0.021	0.341 ± 0.005	0.284 ± 0.011	0.233 ± 0.001	0.182 ± 0.001
	192	0.201 ± 0.011	0.192 ± 0.002	0.197 ± 0.003	0.198 ± 0.003	0.209 ± 0.000	0.198 ± 0.000	0.190 ± 0.000	0.197 ± 0.003	0.282 ± 0.014	0.208 ± 0.002	0.469 ± 0.015	0.668 ± 0.007	0.535 ± 0.035	0.327 ± 0.013	0.232 ± 0.001	0.189 ± 0.001
	336	0.222 ± 0.013	0.210 ± 0.003	0.224 ± 0.003	0.224 ± 0.003	0.223 ± 0.001	0.220 ± 0.001	0.210 ± 0.001	0.224 ± 0.003	0.320 ± 0.026	0.228 ± 0.002	0.620 ± 0.014	0.743 ± 0.002	0.689 ± 0.022	0.451 ± 0.052	0.248 ± 0.001	0.211 ± 0.001
	720	0.270 ± 0.031	0.247 ± 0.010	0.273 ± 0.002	0.274 ± 0.001	0.255 ± 0.001	0.270 ± 0.001	0.255 ± 0.001	0.273 ± 0.002	0.442 ± 0.004	0.275 ± 0.002	0.746 ± 0.007	0.779 ± 0.008	0.776 ± 0.001	0.676 ± 0.026	0.278 ± 0.001	0.256 ± 0.001

Table 10: Test MAE (↓) of RankTS and baselines. Evaluation protocol and method notation follow Table 1.

Dataset	H	Top1	Top3+V	Hist	Select	DLinear	iTrans	Patch	Times	FED	Mixer	Auto	Informer	Trans	Nonstat	Light
Weather	96	0.647	0.644	**0.644**	**0.644**	0.655	0.826	0.810	0.828	0.858	0.810	0.985	0.647	**0.644**	0.673	0.690
	192	0.638	**0.632**	0.632	0.632	0.646	0.910	0.897	0.920	0.972	0.890	1.144	0.644	0.632	0.670	0.682
	336	0.629	**0.626**	0.631	0.631	0.636	0.813	0.810	0.842	0.934	0.804	1.012	0.640	0.631	0.652	0.668
	720	0.670	**0.667**	0.672	0.672	0.676	0.848	0.855	0.870	1.047	0.853	1.037	0.688	0.672	0.751	0.708
	Mean	0.646	0.642	0.645	0.645	0.653	0.849	0.843	0.865	0.953	0.839	1.044	0.655	0.645	0.686	0.687
London	96	0.548	**0.541**	0.547	0.547	0.553	0.549	0.557	0.543	0.587	0.547	0.618	0.615	0.578	0.586	0.579
	192	0.556	**0.551**	0.558	0.555	0.562	0.560	0.565	0.553	0.616	0.558	0.660	0.647	0.600	0.600	0.589
	336	0.558	**0.551**	0.561	0.563	0.566	0.565	0.568	0.561	0.625	0.561	0.685	0.670	0.613	0.602	0.593
	720	0.566	**0.558**	0.570	0.573	0.576	0.575	0.574	0.571	0.648	0.570	0.705	0.685	0.637	0.614	0.603
	Mean	0.557	0.550	0.559	0.560	0.564	0.562	0.566	0.557	0.619	0.559	0.667	0.654	0.607	0.600	0.591
Wind	96	0.243	0.239	0.226	**0.225**	0.254	0.231	0.226	0.235	0.289	0.226	0.315	0.299	0.273	0.230	0.271
	192	0.324	0.323	0.309	**0.308**	0.338	0.313	0.309	0.321	0.383	0.309	0.384	0.368	0.347	0.309	0.349
	336	0.401	0.401	0.386	0.386	0.411	0.398	0.388	0.402	0.465	0.386	0.468	0.437	0.413	**0.385**	0.419
	720	0.488	0.487	0.494	**0.478**	0.495	0.504	0.497	0.510	0.571	0.494	0.569	0.506	0.489	0.489	0.499
	Mean	0.364	0.362	0.354	0.349	0.374	0.362	0.355	0.367	0.427	0.354	0.434	0.402	0.381	0.353	0.385
KDD	96	0.467	0.457	0.466	**0.450**	0.471	0.467	0.466	0.472	0.488	0.466	0.504	0.485	0.475	0.471	0.477
	192	0.476	0.465	0.476	**0.459**	0.491	0.479	0.478	0.481	0.495	0.476	0.505	0.492	0.481	0.475	0.494
	336	0.477	0.469	0.487	**0.464**	0.502	0.489	0.486	0.491	0.504	0.487	0.515	0.497	0.483	0.478	0.502
	720	0.477	0.471	0.492	**0.466**	0.507	0.495	0.492	0.497	0.516	0.492	0.512	0.487	0.484	0.476	0.508
	Mean	0.474	0.466	0.480	0.460	0.493	0.482	0.481	0.485	0.501	0.480	0.509	0.490	0.481	0.475	0.495
Electricity	96	0.301	**0.291**	0.293	0.293	0.317	0.295	0.294	0.293	0.350	0.307	0.388	0.616	0.431	0.380	0.345
	192	0.314	0.304	0.305	0.306	0.318	0.307	**0.301**	0.305	0.383	0.320	0.508	0.659	0.571	0.417	0.346
	336	0.334	0.322	0.328	0.329	0.332	0.327	**0.318**	0.328	0.411	0.336	0.609	0.705	0.668	0.505	0.360
	720	0.372	0.355	0.365	0.365	0.362	0.364	**0.353**	0.365	0.498	0.370	0.687	0.726	0.721	0.656	0.386
	Mean	0.330	0.318	0.323	0.323	0.332	0.323	0.316	0.323	0.410	0.333	0.548	0.677	0.598	0.490	0.359

Table 11: Test MAE (↓; mean ± SD over three seeds) under the protocol of Table 1.

Dataset	H	Top1	Top3+V	Hist	Select	DLinear	iTrans	Patch	Times	FED	Mixer	Auto	Informer	Trans	Nonstat	Light
Weather	96	0.647 ± 0.007	0.644 ± 0.006	0.644 ± 0.008	0.644 ± 0.008	0.655 ± 0.000	0.826 ± 0.002	0.810 ± 0.001	0.828 ± 0.004	0.858 ± 0.151	0.810 ± 0.001	0.985 ± 0.025	0.647 ± 0.003	0.644 ± 0.008	0.673 ± 0.007	0.690 ± 0.003
	192	0.638 ± 0.009	0.632 ± 0.006	0.632 ± 0.002	0.632 ± 0.002	0.646 ± 0.001	0.910 ± 0.003	0.897 ± 0.005	0.920 ± 0.005	0.972 ± 0.172	0.890 ± 0.007	1.144 ± 0.018	0.644 ± 0.006	0.632 ± 0.002	0.670 ± 0.015	0.682 ± 0.001
	336	0.629 ± 0.007	0.626 ± 0.005	0.631 ± 0.002	0.631 ± 0.002	0.636 ± 0.000	0.813 ± 0.002	0.810 ± 0.003	0.842 ± 0.002	0.934 ± 0.125	0.804 ± 0.006	1.012 ± 0.008	0.640 ± 0.001	0.631 ± 0.002	0.652 ± 0.007	0.668 ± 0.001
	720	0.670 ± 0.004	0.667 ± 0.004	0.672 ± 0.001	0.672 ± 0.001	0.676 ± 0.001	0.848 ± 0.003	0.855 ± 0.001	0.870 ± 0.002	1.047 ± 0.002	0.853 ± 0.004	1.037 ± 0.003	0.688 ± 0.004	0.672 ± 0.001	0.751 ± 0.013	0.708 ± 0.001
London	96	0.548 ± 0.012	0.541 ± 0.002	0.547 ± 0.003	0.547 ± 0.002	0.553 ± 0.000	0.549 ± 0.001	0.557 ± 0.000	0.543 ± 0.004	0.587 ± 0.003	0.547 ± 0.003	0.618 ± 0.011	0.615 ± 0.009	0.578 ± 0.004	0.586 ± 0.009	0.579 ± 0.001
	192	0.556 ± 0.011	0.551 ± 0.001	0.558 ± 0.002	0.555 ± 0.002	0.562 ± 0.000	0.560 ± 0.001	0.565 ± 0.001	0.553 ± 0.003	0.616 ± 0.016	0.558 ± 0.002	0.660 ± 0.005	0.647 ± 0.011	0.600 ± 0.003	0.600 ± 0.007	0.589 ± 0.000
	336	0.558 ± 0.011	0.551 ± 0.002	0.561 ± 0.001	0.563 ± 0.003	0.566 ± 0.000	0.565 ± 0.000	0.568 ± 0.001	0.561 ± 0.002	0.625 ± 0.029	0.561 ± 0.001	0.685 ± 0.003	0.670 ± 0.002	0.613 ± 0.008	0.602 ± 0.006	0.593 ± 0.001
	720	0.568 ± 0.009	0.558 ± 0.002	0.570 ± 0.002	0.573 ± 0.002	0.576 ± 0.000	0.575 ± 0.001	0.574 ± 0.001	0.571 ± 0.003	0.648 ± 0.022	0.570 ± 0.001	0.705 ± 0.001	0.685 ± 0.004	0.637 ± 0.012	0.614 ± 0.004	0.603 ± 0.000
Wind	96	0.243 ± 0.005	0.239 ± 0.007	0.226 ± 0.001	0.225 ± 0.001	0.254 ± 0.002	0.231 ± 0.002	0.226 ± 0.001	0.235 ± 0.004	0.289 ± 0.012	0.226 ± 0.001	0.315 ± 0.002	0.299 ± 0.003	0.273 ± 0.004	0.230 ± 0.001	0.271 ± 0.005
	192	0.324 ± 0.005	0.323 ± 0.002	0.309 ± 0.002	0.308 ± 0.001	0.338 ± 0.003	0.313 ± 0.001	0.309 ± 0.002	0.321 ± 0.001	0.383 ± 0.018	0.309 ± 0.002	0.384 ± 0.002	0.368 ± 0.007	0.347 ± 0.002	0.309 ± 0.002	0.349 ± 0.001
	336	0.401 ± 0.002	0.401 ± 0.003	0.386 ± 0.002	0.386 ± 0.005	0.411 ± 0.001	0.398 ± 0.002	0.388 ± 0.001	0.402 ± 0.002	0.465 ± 0.007	0.386 ± 0.002	0.468 ± 0.000	0.437 ± 0.006	0.413 ± 0.004	0.385 ± 0.002	0.419 ± 0.003
	720	0.488 ± 0.001	0.487 ± 0.001	0.494 ± 0.002	0.478 ± 0.003	0.495 ± 0.000	0.504 ± 0.002	0.497 ± 0.000	0.510 ± 0.001	0.571 ± 0.004	0.494 ± 0.002	0.569 ± 0.001	0.506 ± 0.002	0.489 ± 0.006	0.489 ± 0.004	0.499 ± 0.002
KDD	96	0.467 ± 0.004	0.457 ± 0.004	0.466 ± 0.006	0.450 ± 0.000	0.471 ± 0.002	0.467 ± 0.001	0.466 ± 0.000	0.472 ± 0.002	0.488 ± 0.013	0.466 ± 0.006	0.504 ± 0.006	0.485 ± 0.004	0.475 ± 0.003	0.471 ± 0.002	0.477 ± 0.001
	192	0.476 ± 0.001	0.465 ± 0.002	0.476 ± 0.002	0.459 ± 0.001	0.491 ± 0.001	0.479 ± 0.001	0.478 ± 0.003	0.481 ± 0.003	0.495 ± 0.011	0.476 ± 0.002	0.505 ± 0.003	0.492 ± 0.002	0.481 ± 0.001	0.475 ± 0.001	0.494 ± 0.001
	336	0.477 ± 0.003	0.469 ± 0.002	0.487 ± 0.002	0.464 ± 0.000	0.502 ± 0.001	0.489 ± 0.001	0.486 ± 0.001	0.491 ± 0.001	0.504 ± 0.015	0.487 ± 0.002	0.515 ± 0.001	0.497 ± 0.004	0.483 ± 0.005	0.478 ± 0.004	0.502 ± 0.003
	720	0.477 ± 0.005	0.471 ± 0.001	0.492 ± 0.001	0.466 ± 0.000	0.507 ± 0.001	0.495 ± 0.001	0.492 ± 0.001	0.497 ± 0.002	0.516 ± 0.003	0.492 ± 0.001	0.512 ± 0.001	0.487 ± 0.005	0.484 ± 0.001	0.476 ± 0.001	0.508 ± 0.003
Electricity	96	0.301 ± 0.010	0.291 ± 0.001	0.293 ± 0.006	0.293 ± 0.006	0.317 ± 0.001	0.295 ± 0.001	0.294 ± 0.001	0.293 ± 0.006	0.350 ± 0.001	0.307 ± 0.002	0.388 ± 0.014	0.616 ± 0.016	0.431 ± 0.003	0.380 ± 0.007	0.345 ± 0.002
	192	0.314 ± 0.008	0.304 ± 0.001	0.305 ± 0.004	0.306 ± 0.004	0.318 ± 0.000	0.307 ± 0.001	0.301 ± 0.001	0.305 ± 0.004	0.383 ± 0.013	0.320 ± 0.001	0.508 ± 0.011	0.659 ± 0.006	0.571 ± 0.027	0.417 ± 0.008	0.346 ± 0.001
	336	0.334 ± 0.009	0.322 ± 0.001	0.328 ± 0.002	0.329 ± 0.002	0.332 ± 0.001	0.327 ± 0.001	0.318 ± 0.001	0.328 ± 0.002	0.411 ± 0.021	0.336 ± 0.002	0.609 ± 0.008	0.705 ± 0.002	0.668 ± 0.015	0.505 ± 0.034	0.360 ± 0.001
	720	0.372 ± 0.016	0.355 ± 0.003	0.365 ± 0.001	0.365 ± 0.001	0.362 ± 0.001	0.364 ± 0.000	0.353 ± 0.001	0.365 ± 0.001	0.498 ± 0.005	0.370 ± 0.002	0.687 ± 0.004	0.726 ± 0.007	0.721 ± 0.002	0.656 ± 0.014	0.386 ± 0.001

G Extended Forecasting Results and Sensitivity

G.1 Forecasting Metrics and Aggregation

For each method, dataset, horizon, and seed, we first average test errors over the complete retained query set. We then report the mean and sample standard deviation of the three seed-level averages. Dataset-level Mean rows weight the four horizons equally; cross-dataset summaries weight datasets equally. Correlated windows are not treated as independent observations. The forecasting evaluation contains 16,128 records covering 320 method–dataset–horizon cells, with no invalid selected outcomes.

Top1 and Top3+V share fitted candidate records within a seed and horizon. Only the validation-selected test error is reported; the best test error in a shortlist is not a deployable selection rule. Rankings originate at H96 and remain frozen for H192, H336, and H720, while each predictor is independently fitted for its target horizon.

Table 12: Fitting cost and validation-selected test MSE at different H96 candidate budgets, averaged over seeds 0, 1, and 2. Time sums all candidate fits within each budget.

Dataset	Selector	MSE K1	K3	K5	K10	Minutes K1	K3	K5	K10
Weather	RankTS	0.4978	0.4955	0.4955	0.4947	2.28	6.66	9.13	15.50
Weather	Random	0.5800	0.5170	0.5148	0.5133	3.71	5.92	8.44	18.48
London	RankTS	0.9566	0.9512	0.9534	0.9538	2.52	6.96	8.93	14.53
London	Random	1.0327	0.9731	0.9705	0.9554	1.18	3.73	7.77	17.57
Wind	RankTS	0.5972	0.5801	0.5782	0.5765	0.77	2.81	3.47	5.72
Wind	Random	0.6968	0.6359	0.6302	0.6358	0.78	3.26	6.66	12.15
KDD	RankTS	0.5701	0.5432	0.5449	0.5309	1.52	3.37	5.83	9.89
KDD	Random	—	0.5676	0.5579	0.5337	—	4.32	8.11	15.76
Electricity	RankTS	0.0949	0.0925	0.0922	0.0906	0.75	3.22	5.14	9.54
Electricity	Random	0.1230	0.1052	0.1030	0.1030	1.12	3.75	7.25	13.13

G.2 Target-Cost Protocol

The cost study uses three input-length-selected queries per dataset at H96 and paired ranker/predictor seeds 0, 1, and 2. Random proposals sample without replacement from the full archive, before checking target losses. Three fixed random lists per query are evaluated at nested prefixes $K = 1, \ldots, 10$. Each draw selects its own validation winner; test error is averaged over all draws within a query and then over queries and seeds. The procedure neither chooses the best random draw nor pools its candidates with other draws. The same fitted candidates define every prefix, and ranker weights remain frozen throughout the cost study.

For a fixed prefix, let E_K denote the test error of the validation-selected model and B_K the retrospective minimum test error among its finite candidates. Although B_K cannot increase as the prefix grows, E_K can increase when validation changes the selected identity. This distinction explains why additional candidate evaluations need not monotonically improve reported test error. Recorded fitting times include every attempted candidate in the logical prefix, even when its measurement is reused from a compatible cache.

G.3 Longer Predictor Training

The H96 seed-0 budget comparison uses all retained queries and a fixed Top3 per query. RankTS retains lower mean MSE than DLinear on every dataset at both training budgets, but its own test error increases slightly on London and KDD. Of the 84 query shortlists, 23 select a different graph at 30 epochs. The comparison measures sensitivity to the joint epoch-cap and stopping-patience protocol described in Appendix F.4.

H Deployment and Statistical Reporting

H.1 Frozen-Ranker Deployment

Algorithm 3 specifies deployment. Query descriptors use training observations and frozen historical preprocessing. The archive contains architecture specifications, and every shortlisted predictor is fitted independently on the target series. H96 rankings are reused for longer horizons, with predictor weights fitted separately for each horizon.

Algorithm 3 Online Recommendation with a Frozen Archive and Ranker

Require: Target training/validation data, archive $\mathcal{A}_d$, ranker θ, budget K
1: Extract catch22 features using frozen historical imputation and normalization
2: Score $g \in \mathcal{A}_d$ and retain the Top-K graph identities
3: Fit these predictors independently on target training data
4: Select the eligible predictor with the lowest validation MSE
5: **return** its architecture and fitted weights, or no model if all fits are invalid

Table 13: Evaluation units and replication settings for each study.

Study	Evaluation unit	Replication
Main forecasting	Retained query, dataset, horizon	Three paired ranker/predictor seeds
Retrieval and ranker transfer	H96 query with fixed predictor labels	Three ranker seeds
Four-horizon ablations	Query and horizon	Ranker/predictor seed 0
Target fitting cost	Three queries per dataset, H96	Seeds 0/1/2; three random lists per query
Training-budget sensitivity	Fixed Top3 per H96 query	Predictor seed 0
Initialization	Final set of 90 anchors	Ten generation seeds
Evolution	One historical task per dataset	Predictor seed 0

H.2 Aggregation and Random Seeds

Within each dataset and seed, metrics are averaged over the retained query series. Cross-dataset means give the five datasets equal weight. Multi-horizon summaries first average the four horizons within a dataset. Unless indicated otherwise, reported uncertainty is the sample standard deviation of seed-level means.

The historical/query identities and acquired archives are fixed across ranker seeds. Retrieval uncertainty therefore describes ranker variation conditional on the same predictor labels. Forecasting seeds also vary target-predictor initialization. Initialization variability concerns architecture generation and selection. Configuration comparisons and final reporting use the same query cohort and acquired archives.

H.3 Measurement Definitions

Forecasting reports test errors after validation-based selection. Retrospective retrieval uses test-loss ranks to define relevance, as specified in Appendix J. Historical validation losses supply ranker supervision. Graph identities, series identities, horizons, seeds, candidate prefixes, and fit eligibility determine the paired evaluation units. Failed candidates remain within attempted-fit budgets; missing historical observations are not assigned surrogate performance labels.

I Architecture Embeddings and Recommendation Diversity

I.1 Embedding Projection

Figure 6 projects the 96-dimensional h_g embeddings of each dataset's 2,000-architecture archive using ranker seed 0. PCA centers the embeddings and uses a full singular value decomposition without coordinate-wise scaling. Each dataset has its own fitted graph encoder and PCA basis. PC1 and PC2 denote the first and second principal components. Axis percentages give explained variance within that dataset.

The projection uses the archive alone; query rankings supply the Top1 and Top10 overlays. Blue intensity indicates node count, circles mark the Top10 union, and stars mark the Top1 union. Distances describe the corresponding dataset-specific embedding space. Explicit structural coverage is measured separately by the initialization metrics in Table 2.

I.2 Reuse Across Query Series

Table 14 counts distinct architecture identities recommended across retained queries before fitting or validation selection. Top1 diversity ranges from five to thirteen identities per dataset and seed; Top10 unions contain 34–96 identities. The results quantify both query-dependent recommendations and reuse of structures across series.

These counts concern architecture specifications. When two queries select the same identity, each predictor is fitted on its own training segment. Likewise, a fixed recommendation list can yield different validation-selected identities across horizons. Forecasting accuracy, recommendation diversity,

Table 14: Number of distinct architectures in Top1 and Top10 recommendations across query series, before validation selection, for each ranker seed.

Dataset	Top1 s0	s1	s2	Top10 s0	s1	s2
Weather	7	7	7	52	48	34
London	8	10	13	66	80	69
Wind	9	7	11	71	66	69
KDD	11	13	13	87	96	86
Electricity	8	5	6	42	38	54

and two-dimensional embedding geometry thus summarize different aspects of the recommendation process.

J Multi-Budget Retrieval and Ranker Transfer

J.1 Shared Candidates and Metric Definitions

All selectors use the same candidates with finite cached validation and test metrics within each H96 query. The query set matches the primary forecasting comparisons. Let $r_{t,g}$ be the ascending test-MSE rank of graph g in this shared candidate set, with ties resolved by the fixed candidate order. Define $R_t = \{g : r_{t,g} \leq 100\}$. For a recommended prefix P_t^K,

$$\text{Precision}\,@K = \frac{|P_t^K \cap R_t|}{K}, \qquad \text{Recall}\,@K = \frac{|P_t^K \cap R_t|}{100}. \tag{13}$$

At fixed K, recall is therefore $K/100$ times precision. To also evaluate graded relevance and ordering, define $a_{t,g} = \max(0, 1 - (r_{t,g} - 1)/100)$ and

$$\text{NDCG}\,@K = \frac{\sum_{j=1}^{K} a_{t,g_j} / \log_2(j+1)}{\sum_{j=1}^{K} a_{t,g_j^*} / \log_2(j+1)}, \tag{14}$$

where g_j follows the recommendation order and g_j^* follows the ideal test-MSE order. Test outcomes define retrospective relevance only; they do not supervise the ranker or choose a deployed predictor. NDCG uses linear graded relevance directly, without an exponential gain transformation.

J.2 Historical and Random Controls

Historical mean orders architectures by mean within-task historical validation percentile. Nearest-history selects the historical task closest to the query in the main ranker's normalized catch22 space, uses its observed validation percentiles, and backs off to historical means for unobserved graphs. Both controls use historical validation performance and query training descriptors, never target forecast outcomes. Random ranking averages 100 independently generated permutations per query.

We evaluate $K \in \{1, 3, 5, 10, 20, 30, 50, 100\}$. RankTS first averages query metrics within each dataset and seed; reported means then average the three ranker seeds. Random repeats are averaged within query before aggregation. Figure 9 and Table 15 complement the head-precision curves with broader recall and graded ordering. The dense-label retrieval study is evaluated at H96; the separate forecasting study evaluates the frozen recommendation prefixes at all four horizons.

J.3 Cross-Dataset Ranker Transfer

Each source dataset contributes three frozen rankers, yielding all 25 source–target combinations. Source imputation, normalization, graph encoder, and scorer weights remain unchanged. Each source ranker ranks the target's complete archive; retrospective evaluation then uses the same finite candidate subset for every source within a target query. This yields 1,260 source–target–seed–query evaluations, each assessed at $K = 3$ and 100.

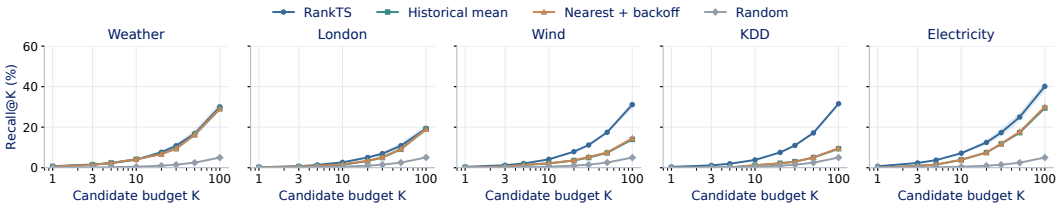


Figure 9: Recall of the empirical Top-100 candidates under the protocol of Figure 3. Shaded regions indicate SD across ranker seeds.

Table 15: Retrieval performance at H96 (%, ↑) under the protocol of Figure 3.

Dataset	Selector	P@3	NDCG@3	R@100	NDCG@100
Weather	RankTS	53.59	35.22	30.00	31.90
Weather	Historical mean	47.06	33.17	29.12	32.06
Weather	Nearest + backoff	47.06	33.17	29.12	32.09
Weather	Random	5.29	2.78	4.99	4.27
London	RankTS	24.18	16.63	19.35	20.05
London	Historical mean	15.69	10.81	19.12	18.09
London	Nearest + backoff	19.61	12.44	19.00	18.31
London	Random	5.10	2.51	5.06	4.29
Wind	RankTS	40.97	29.21	31.12	33.09
Wind	Historical mean	20.83	20.42	14.00	17.84
Wind	Nearest + backoff	20.83	20.42	14.56	18.31
Wind	Random	5.12	2.69	5.01	4.25
KDD	RankTS	38.56	26.15	31.57	31.58
KDD	Historical mean	0.00	0.00	9.35	10.18
KDD	Nearest + backoff	0.00	0.00	9.59	10.31
KDD	Random	5.57	2.86	5.02	4.27
Electricity	RankTS	77.78	58.58	40.12	48.81
Electricity	Historical mean	29.41	16.11	29.47	32.51
Electricity	Nearest + backoff	29.41	16.11	30.06	33.00
Electricity	Random	4.71	2.50	5.01	4.24

Table 16: Cross-dataset Precision@3 (%, ↑) over three ranker seeds. Rows and columns indicate source and target datasets.

Source / target	Weather	London	Wind	KDD	Electricity
Weather	53.59 ± 3.00	15.69 ± 1.96	16.67 ± 7.51	9.80 ± 5.19	4.58 ± 7.92
London	8.50 ± 4.08	24.18 ± 1.13	7.64 ± 3.18	11.76 ± 7.07	47.71 ± 8.16
Wind	3.27 ± 2.26	44.44 ± 7.42	40.97 ± 3.18	11.11 ± 3.00	37.91 ± 14.18
KDD	11.76 ± 7.07	8.50 ± 1.13	11.81 ± 4.81	38.56 ± 8.16	28.76 ± 23.72
Electricity	6.54 ± 8.16	35.95 ± 1.13	31.94 ± 6.36	9.15 ± 4.08	77.78 ± 9.27

Diagonal entries use the corresponding within-dataset scores and their fixed tie ordering. Target labels neither update the transferred model nor select a preferred source. The target archive still depends on target historical searches, so the experiment evaluates compatibility-function transfer conditional on that archive.

Within-dataset training leads both head metrics on Weather, Wind, KDD, and Electricity. London instead benefits from Wind- and Electricity-trained scorers. The asymmetric directions indicate that dataset identity alone does not determine transferability. All 25 source–target combinations are reported without selecting a source using target-query labels.

Table 17: Cross-dataset NDCG@3 (%, ↑) over three ranker seeds. Rows and columns indicate source and target datasets.

Source / target	Weather	London	Wind	KDD	Electricity
Weather	35.22 ± 3.07	8.62 ± 1.66	11.76 ± 6.20	6.39 ± 5.06	1.57 ± 2.73
London	4.12 ± 3.14	16.63 ± 1.31	6.06 ± 2.14	5.72 ± 3.26	34.12 ± 7.18
Wind	1.17 ± 0.73	27.47 ± 7.78	29.21 ± 4.49	9.43 ± 1.38	25.24 ± 12.09
KDD	6.95 ± 4.18	5.17 ± 1.72	5.23 ± 3.19	26.15 ± 4.62	21.75 ± 18.67
Electricity	3.77 ± 4.89	22.60 ± 0.66	24.99 ± 6.21	5.70 ± 2.41	58.58 ± 12.26

Table 18: Ranker ablation configurations. Each intervention modifies the catch22–Concat reference under the same evaluation protocol.

Configuration	Intervention
84 statistics	The 84-statistic descriptor replaces catch22 (Appendix E.3).
CNN	Window-based series encoding (Appendix E.3).
No h_{MP}	$h_g = h_{\mathrm{flat}}$; message-branch output and fusion are bypassed.
No h_t	Normalized series descriptors are zeroed before encoding; h_t is a learned constant.
No h_g	$h_g = 0$; series and model-size inputs are retained.
No h_r	$h_r = 0$; the complete graph encoder, including operator hyperparameter values and masks, is retained.
Uniform pairs	Uniform sampling from all eligible pairs under the same total pair budget.
No diversity weighting	Structural-distance and skeleton-frequency weights are removed; band balancing and quotas are retained.
Utility regression	Squared-error regression of negative log validation MSE on the same sampled endpoints.
Auxiliary gap loss	Pairwise logistic loss plus $0.1\,\mathrm{SmoothL1}(s_i - s_j, u_i - u_j)$.
No m_v	Predecessor messages are zeroed at all three GRU layers; node attributes, readout, flat branch, and fusion are retained.

K Ranker and Acquisition Ablations

K.1 Configurations and Matched Comparisons

Figure 4 compares twelve distinct configurations, including the catch22–Concat RankTS reference. Tables 19 and 20 report twelve configurations, including an isolated predecessor-message intervention, No m_v, across five datasets and three ranker seeds, giving 180 dataset–configuration–seed settings. Historical observations, archives, inner task splits, and H96 predictor labels are shared. Each configuration selects its epoch count by historical NDCG@20 and then refits on all 150 historical tasks.

Table 18 defines the interventions. Representation, scorer-input, and objective ablations use the same sampled pair identities as the reference within each seed. Pair-construction controls change the sampler while retaining the eligibility criteria. Each seed's metrics first average queries within a dataset and then weight the five datasets equally; SD is computed across the three seed-level means. Gold, silver, and bronze icons in Figure 4 mark the top three unrounded means within each panel and metric.

The listed architecture variants retain a 96-dimensional graph output and the 288-dimensional concatenation input. Active computation and parameter counts can differ across these variants. No m_v preserves access to connections through the flattened branch, whereas No h_{MP} removes the message-branch representation entirely.

K.2 Head and Broader Retrieval

Tables 19 and 20 provide per-dataset results for the listed interventions. Among series descriptors, catch22 gives the highest aggregate Precision@3 and NDCG@3. Removing the full architecture embedding gives 11.45% precision, while removing only model size gives 45.24%. These comparisons distinguish the effects of architectural information and the additional size input.

Table 19: Precision@3 (%, ↑) of ranker ablations at H96 over three ranker seeds.

Configuration	Weather	London	Wind	KDD	Electricity	Mean
RankTS	53.59 ± 3.00	24.18 ± 1.13	40.97 ± 3.18	38.56 ± 8.16	77.78 ± 9.27	47.02 ± 2.86
84 statistics	42.48 ± 9.27	34.64 ± 2.26	42.36 ± 4.81	37.25 ± 6.79	69.93 ± 4.93	45.33 ± 1.37
CNN	40.52 ± 12.61	33.99 ± 2.26	38.89 ± 1.20	36.60 ± 8.16	73.86 ± 13.35	44.77 ± 1.40
No h_{MP}	45.10 ± 8.55	23.53 ± 8.99	42.36 ± 2.41	39.87 ± 1.13	65.36 ± 3.00	43.24 ± 0.71
No h_t	30.72 ± 21.15	21.57 ± 18.70	3.47 ± 4.34	22.22 ± 13.91	68.63 ± 18.70	29.32 ± 7.31
No h_g	16.34 ± 5.66	13.07 ± 1.13	6.94 ± 2.41	9.80 ± 3.40	11.11 ± 7.92	11.45 ± 1.59
No h_r	50.33 ± 1.13	16.34 ± 3.00	34.03 ± 5.24	48.37 ± 12.61	77.12 ± 5.99	45.24 ± 2.80
Uniform pairs	20.92 ± 13.06	19.61 ± 5.19	31.94 ± 4.81	35.29 ± 3.92	62.75 ± 5.19	34.10 ± 2.41
No diversity weighting	54.25 ± 5.99	21.57 ± 3.40	42.36 ± 5.24	38.56 ± 3.00	69.28 ± 3.00	45.20 ± 2.51
Utility regression	33.33 ± 1.96	8.50 ± 4.08	11.11 ± 8.42	19.61 ± 5.19	47.71 ± 5.99	24.05 ± 4.25
Auxiliary gap loss	56.86 ± 7.84	16.34 ± 6.30	36.11 ± 7.89	40.52 ± 6.30	73.20 ± 4.93	44.61 ± 1.58
No m_v	50.33 ± 9.06	27.45 ± 1.96	38.89 ± 8.42	45.10 ± 11.76	76.47 ± 7.07	47.65 ± 6.59

Table 20: NDCG@3 (%, ↑) of ranker ablations at H96 over three ranker seeds.

Configuration	Weather	London	Wind	KDD	Electricity	Mean
RankTS	35.22 ± 3.07	16.63 ± 1.31	29.21 ± 4.49	26.15 ± 4.62	58.58 ± 12.26	33.16 ± 2.88
84 statistics	27.10 ± 1.37	24.28 ± 1.85	31.94 ± 7.48	26.89 ± 8.55	52.59 ± 6.86	32.56 ± 1.38
CNN	27.05 ± 8.05	19.12 ± 2.11	27.86 ± 5.55	20.41 ± 5.71	54.71 ± 11.68	29.83 ± 2.83
No h_{MP}	34.32 ± 12.20	15.11 ± 7.37	24.23 ± 0.49	27.10 ± 1.88	47.58 ± 4.63	29.67 ± 0.10
No h_t	17.85 ± 10.30	14.05 ± 16.32	2.11 ± 2.79	14.64 ± 15.14	59.90 ± 18.41	21.71 ± 4.43
No h_g	12.36 ± 5.71	7.17 ± 0.29	3.19 ± 1.11	6.41 ± 2.95	5.41 ± 3.84	6.91 ± 0.99
No h_r	35.61 ± 4.00	9.89 ± 4.71	22.57 ± 0.82	34.28 ± 10.58	59.32 ± 1.61	32.34 ± 2.25
Uniform pairs	14.20 ± 10.07	12.10 ± 4.82	22.07 ± 4.23	17.25 ± 4.11	50.86 ± 4.06	23.30 ± 1.43
No diversity weighting	38.56 ± 5.18	15.80 ± 2.81	30.71 ± 6.56	26.89 ± 5.29	52.39 ± 5.85	32.87 ± 3.14
Utility regression	18.86 ± 3.39	4.17 ± 1.54	6.20 ± 3.61	13.61 ± 4.94	35.79 ± 7.59	15.73 ± 3.46
Auxiliary gap loss	39.27 ± 1.87	11.50 ± 5.33	27.75 ± 7.62	26.69 ± 4.16	58.45 ± 2.04	32.73 ± 2.47
No m_v	34.34 ± 6.28	20.83 ± 0.39	29.16 ± 5.47	33.11 ± 11.37	60.11 ± 2.20	35.51 ± 4.88

Table 21: Recall@100 and NDCG@100 at H96 (%, ↑), averaged equally over five datasets.

Configuration	Recall@100	NDCG@100
RankTS	30.43 ± 1.11	33.09 ± 1.21
84 statistics	31.27 ± 1.08	33.99 ± 1.01
CNN	30.92 ± 0.70	32.89 ± 0.22
No h_{MP}	27.41 ± 0.35	29.49 ± 0.66
No h_t	21.15 ± 2.19	23.32 ± 2.16
No h_g	10.55 ± 0.52	9.21 ± 0.22
No h_r	29.86 ± 1.12	32.66 ± 0.60
Uniform pairs	20.82 ± 0.39	21.20 ± 0.91
No diversity weighting	30.61 ± 0.57	33.52 ± 1.07
Utility regression	17.01 ± 2.93	17.21 ± 2.81
Auxiliary gap loss	30.37 ± 0.13	32.81 ± 0.41
No m_v	30.98 ± 0.57	34.19 ± 1.06

Table 21 extends the comparison to $K = 100$. Balanced sampling improves Recall@100 from 20.82% to 30.43% relative to uniform pairs. The 84-statistic descriptor reaches 31.27%. Head retrieval and broader coverage therefore need not favor identical representations. The isolated No m_v comparison retains competitive means with larger seed variation in head precision; its retained flattened branch continues to encode structure.

K.3 Four-Horizon Forecasting

Table 22 evaluates the nine configurations for which target predictors were fitted at all four horizons, using ranker/predictor seed 0. Each frozen H96 ranking supplies three candidates, fitted independently at the target horizon and selected by target validation. These forecasting results complement the fixed-label retrieval comparisons with the deployed selection procedure.

Table 22: Four-horizon mean Top3+V test MSE (↓) for nine configurations with completed target fits at seed 0.

Configuration	Weather	London	Wind	KDD	Electricity	Mean
RankTS	1.1174	1.1984	0.5780	0.4905	0.2113	0.7191
84 statistics	1.1340	1.2052	0.5531	0.4906	0.2108	0.7187
CNN	1.1204	1.1986	0.5780	0.4954	0.2105	0.7206
No h_{MP}	1.1386	1.2062	0.5691	0.4887	0.2250	0.7255
No h_t	1.1414	1.2102	0.7634	0.4938	0.2089	0.7635
Uniform pairs	1.1266	1.2072	0.5732	0.4891	0.2101	0.7212
No diversity weighting	1.1250	1.2060	0.5766	0.4925	0.2080	0.7216
Utility regression	1.1622	1.2199	0.6282	0.4971	0.2325	0.7480
Auxiliary gap loss	1.1367	1.2153	0.5693	0.4925	0.2121	0.7252

Table 23: Structural diversity of the 90-anchor sets in Table 2, with mean ± sample SD over ten generation seeds.

Candidate pool	Seed selection	Families/arch.	Eff. topologies	Eff. skeletons	NN distance	Pool distance
Proposed	Proposed	3.15 ± 0.05	6.72 ± 0.61	90.00 ± 0.00	0.643 ± 0.016	0.764 ± 0.002
Random	Proposed	4.07 ± 0.13	56.53 ± 3.14	90.00 ± 0.00	0.816 ± 0.011	0.807 ± 0.003
Proposed	Random	2.15 ± 0.07	13.03 ± 0.63	87.44 ± 1.79	0.657 ± 0.025	0.720 ± 0.004
Random	Random	3.32 ± 0.12	52.32 ± 7.46	90.00 ± 0.00	0.829 ± 0.010	0.794 ± 0.007

The four-horizon mean MSE is 0.7191 for RankTS, 0.7187 for the 84-statistic descriptor, and 0.7206 for the CNN. Removing series descriptors increases it to 0.7635, and utility regression gives 0.7480. These seed-0 results measure validation-selected forecasting performance, while the three-seed retrieval tables measure candidate relevance and ordering at H96.

K.4 Initialization at the 90-Anchor Endpoint

Table 2 compares a 2×2 combination of candidate-pool construction and seed selection. Every condition starts with 2,000 unique raw candidates and is evaluated on 90 selected architectures, across generation seeds 1000–1009. Structural filters, operator hyperparameter sets, preprocessing probabilities, node limits, and resource checks are shared.

The proposed pool starts with 282 base architectures and expands through hierarchical skeleton–family–operator–hyperparameter sampling. The random pool omits the base architectures and samples node count uniformly from one to ten, followed by grammar-compatible operators and catalog hyperparameter tuples, a sequential backbone, and optional compatible skip inputs with probability 0.4. Both pools use the same marginal preprocessing policy.

Proposed selection applies coverage, structural scoring, representative-recipe retention, and population completion, then takes the first 90 entries as anchors. Random selection draws 90 feasible architectures uniformly without replacement from the corresponding accepted pool, bypassing coverage and structural-score ordering. The main pipeline reaches more than 90 coverage seed architectures before bucket completion, so the anchor comparison concerns the earlier selection stages; buckets fill the subsequent 300-member population.

Let $C(S)$ be the set of distinct operator–hyperparameter configurations covered by an anchor set S. The catalog contains 227 configurations. The two coverage measures are

$$\mathrm{Coverage}(S) = 100\frac{|C(S)|}{227}, \qquad \mathrm{ConfigsPer100Nodes}(S) = 100\frac{|C(S)|}{\sum_{g\in S}|V(g)|}. \tag{15}$$

Thus, coverage counts distinct configurations once, while the denominator of the second metric counts every node occurrence. The proposed pipeline yields 92.95% coverage and 41.80 configurations per 100 nodes. Applying proposed selection to the random pool reaches full coverage with larger mean architectures (6.56 versus 5.61 nodes).

Table 23 supplements these measures with structural diversity. For category frequencies p_c, the effective count is $\exp(-\sum_c p_c \log p_c)$. Topology signatures use indexed predecessor lists and output identity; architecture skeleton signatures additionally include operators, merge rules, residuals, and

Table 24: High-quality proposal counts among 150 attempts per task.

Dataset	Series	Initial $q_{.2}$	RankTS	Random	No surrogate
Weather	T1999	0.5329	59	43	45
London	T792	0.3376	51	31	42
Wind	T122	0.2747	40	21	36
KDD	T190	0.1810	44	34	42
Electricity	T255	0.1518	50	32	32

Table 25: Mean proposal-quality rates (%) across five tasks at different quantile thresholds.

Quantile	RankTS	Random	No surrogate
0.1	18.93	10.67	14.93
0.2	32.53	21.47	26.27
0.3	45.07	33.47	37.07

Table 26: Validation gain and validation-selected test MSE in the matched evolution study at H96 (seed 0).

Dataset	RankTS gain (%)	Random	No surrogate	RankTS MSE	Random	No surrogate
Weather	1.183	0.567	2.587	0.4468	0.4486	0.4394
London	0.000	0.000	0.000	0.4160	0.4160	0.4160
Wind	0.000	0.000	0.000	0.8573	0.8573	0.8573
KDD	0.503	0.000	0.503	0.2362	0.2238	0.2362
Electricity	4.418	11.040	2.343	0.0974	0.0915	0.0993

preprocessing. Families/architecture averages the number of represented operator families. Nearest-neighbor distance averages each selected graph's minimum structural-token Jaccard distance to another selected graph. Pool distance averages the minimum distance to the selected set over the shared union of the two accepted pools within a seed. These are structural diagnostics obtained without fitting forecasting models.

K.5 Task-Local Proposal Quality

The matched evolution comparison uses one prespecified historical task per dataset, H96, and predictor seed 0. Each policy receives the same 300 initial evaluations and 150 additional attempts. RankTS proposes 75 mutations, 30 crossovers, 30 surrogate-guided candidates, and 15 random candidates. Random uses 150 random proposals; No surrogate replaces the 30 surrogate-guided proposals with random proposals while retaining the other mechanisms.

For task t, let $q_{t,\alpha}$ be the α quantile of finite positive validation MSEs in the initial population. The proposal-quality rate is

$$Q_t(\alpha) = \frac{1}{150} \sum_{g \in \mathcal{C}_t} \mathbf{1}\left[0 < L_{t,g}^{\mathrm{val}} \le q_{t,\alpha},\ L_{t,g}^{\mathrm{val}} < \infty\right]. \tag{16}$$

The main diagnostic uses $\alpha = 0.2$. Failed attempts contribute zero to the numerator and remain in the denominator. Table 24 gives the task identities, thresholds, and counts behind Figure 5.

The mean high-quality rate across five tasks is 32.53% for RankTS, 21.47% for Random, and 26.27% for No surrogate. Table 25 shows the same aggregate ordering at the 10th and 30th initial-loss percentiles. The quantile analysis is a post-hoc diagnostic of the completed proposal sets on these five tasks.

K.6 Best-Candidate Validation and Test Outcomes

Proposal quality and the best selected predictor capture different aspects of acquisition. Table 26 reports validation improvement relative to the best initial candidate and test MSE after validation selection over all evaluated candidates. RankTS improves validation more than Random on Weather and KDD; Random improves more on Electricity, while London and Wind tie. On KDD, the validation-selected RankTS candidate has higher test MSE despite its lower validation loss. The study holds the acquired main archives and rankers fixed and evaluates the proposal policies on these five historical tasks.