ADAPTIVE PATCHING IS HARDER THAN IT LOOKS FOR TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Adaptive patching has been proposed for time-series forecasting: allocate finer patches where the sequence looks locally informative. We compare three adaptive patching methods with tuned uniform counterparts that use one fixed patch size throughout the input, while preserving each method's backbone family and shared training protocol. Across the evaluated long-horizon benchmarks, we observe mostly small accuracy differences and no consistent advantage for adaptive patching; uniform variants generally train faster and avoid adaptive routing at inference. Motivated by these observations, we develop a theoretical framework for budgeted patch allocation through a common resolution-density representation of uniform, nonuniform, and overlapping layouts, independently of the patching rule. Within a separable shared-loss model, we derive an exact criterion for when nonuniform allocation improves on uniform allocation at the same budget. We characterize modeled gains locally through a quadratic surrogate and bound them globally under strong convexity and a bounded loss derivative. The analysis distinguishes where resolution is placed from how much it is redistributed: even an allocation that correctly ranks regions by their weights can redistribute too much resolution and increase modeled loss. Without coupling across time, positive regional weights preserve the minimizers of the shared loss. When the shared loss has a common minimizing rate, the remaining absolute headroom for improvement tends to zero as the uniform rate approaches it. These results characterize the roles of regional weighting, placement, and redistribution strength under the stated model. The empirical findings support evaluating adaptive patching against a tuned uniform baseline and weighing any accuracy gain against the additional training and inference cost of adaptive routing.

1 INTRODUCTION

Patching is widely used to tokenize time series for Transformer-based forecasting. As in vision Transformers (Dosovitskiy et al., 2021), it groups neighboring observations into tokens, providing the representation used by PatchTST (Nie et al., 2023) and subsequent forecasting foundation models such as TimesFM and Moirai (Das et al., 2024; Woo et al., 2024). But a fixed patch length assigns the same temporal granularity throughout the input. In vision, a line of work has explored alternatives that vary input resolution, patch granularity, or token computation to use resources more efficiently (Dehghani et al., 2023; Ronen et al., 2023; Yin et al., 2022; Bolya et al., 2023; Yin et al., 2025). Carrying this idea over to time series sounds reasonable: change points, regime shifts, and bursts of high-frequency structure look meaningful in the input, making them natural candidates for finer patches. Recent dynamic patching methods pursue this intuition by placing boundaries or choosing patch sizes using predictive entropy, temporal heterogeneity, learned infor-

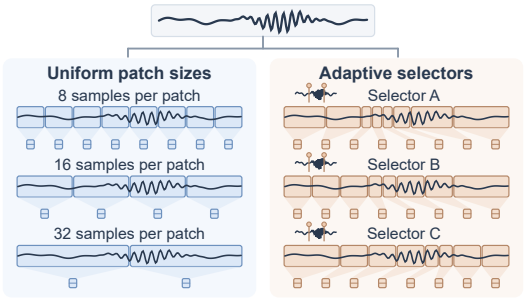


Figure 1: **Patching choices.** Uniform lengths versus input-adaptive boundaries.

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

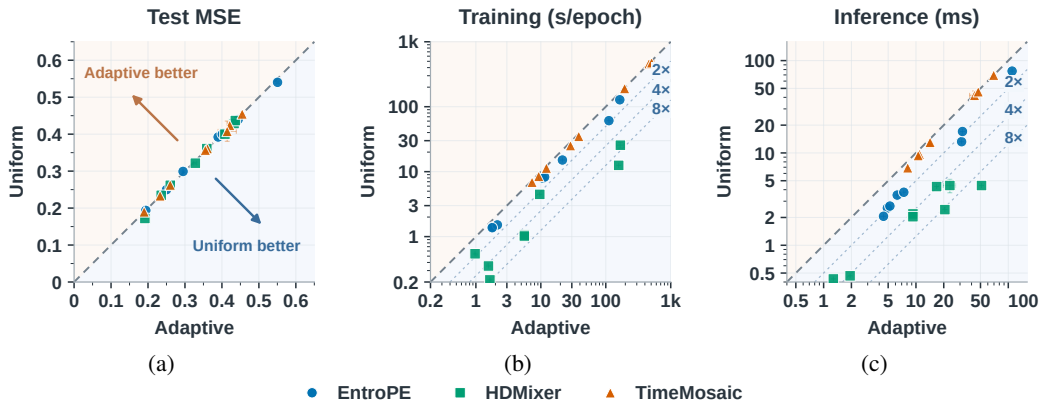


Figure 2: **Validation-selected uniform versus adaptive patching.** Each point represents one method and dataset, averaging over seeds and forecast horizons. Colors and markers identify methods. Points below equality favor uniform; $2\times$, $4\times$, and $8\times$ guides show uniform speedups. Training cost is steady-state time in seconds per epoch and inference cost is batch latency; runtime axes are logarithmic.

mation density, or mixtures of patch sizes (Abeywickrama et al., 2025; Ding et al., 2026; Huang et al., 2024; Ankireddy et al., 2025; Feng et al., 2025). Figure 1 illustrates this allocation choice: uniform patching uses a fixed patch length throughout the input, whereas adaptive patching varies boundaries to assign finer resolution to selected regions.

The premise is harder to defend than it looks, because "looks informative" and "finer patching helps the loss" are two different statements. A measure of local structure can guide a patching rule, but its usefulness depends on whether the resulting allocation improves forecasting. Pointwise losses such as MSE do not by themselves specify where extra resolution should be spent: a high-variance region need not be one whose error finer patches can reduce. Under a fixed resolution budget, refinement also requires coarsening elsewhere. The question is therefore both where to place finer patches and how much resolution to redistribute.

These considerations raise two questions. Empirically, does adaptive patching improve forecasting enough to justify its additional training and inference cost relative to a carefully tuned uniform alternative? Mathematically, how can different patch layouts be represented and compared under a common resolution budget? Addressing the first requires comparisons within the same model family and training protocol. Addressing the second requires a formulation that separates the patch layout from the rule that generates it and the objective used to assess it.

This paper. We compare independently validation-selected adaptive and uniform variants of EntroPE (Abeywickrama et al., 2025), HDMixer (Huang et al., 2024), and TimeMosaic (Ding et al., 2026) within each method's backbone family and training protocol. Across the evaluated settings, 79.2% of observed MSE differences lie within $\pm 2\%$; uniform has lower mean test MSE in 56.3%, 71.9%, and 37.5% of settings, respectively, while reducing measured training and inference time (Figure 2 and table 1). The larger accuracy differences are method- and dataset-specific. DLinear (Zeng et al., 2023) and TSMixer (Chen et al., 2023) established the importance of simple forecasting baselines. Our comparison examines the corresponding choice at the tokenization stage.

Motivated by these observations, we develop, to our knowledge, the first general theoretical framework for time-series patch allocation through a common resolution-density representation. The representation accommodates uniform, nonuniform, and overlapping patch layouts under a fixed nominal budget. Its generality lies in separating how patches distribute resolution across the input from both the patching rule and the objective used to evaluate the allocation.

As one instantiation, we analyze separable objectives in which regional weights K_t scale a shared loss response $D(r_t)$ to resolution. An exact improvement criterion separates an allocation-dependent alignment term from a dispersion term that is nonnegative under convexity. A local quadratic sur-

rogate characterizes the roles of alignment and redistribution strength, and a global bound controls gains under strong-convexity assumptions. The analysis admits both improving reallocations and uniform optimality: when the mean resolution equals the shared loss-minimizing rate, uniform allocation is optimal even with heterogeneous regional weights (Section 4). These results distinguish regional importance from preferred resolution and show why identifying where to refine leaves the amount of refinement to be determined.

Contributions.

- A *validation-selected comparison* of adaptive and uniform patching in EntroPE, TimeMosaic, and HDMixer across eight datasets and four forecast horizons per method, with independent selection of both variants and evaluation on fresh seeds. Most MSE differences are small, while uniform variants reduce measured training and inference costs; the largest accuracy effects remain method- and dataset-specific (Figure 2 and table 1).
- A *general theoretical framework for time-series patch allocation* that represents uniform, nonuniform, and overlapping layouts through a common resolution density, independently of the patching rule. Within a separable shared-loss model, we derive an exact criterion for improvement over uniform allocation, characterize gains locally, obtain a global gain bound under strong-convexity assumptions, and establish uniform optimality when the mean resolution equals the common loss-minimizing rate (Section 4).

2 Related work

Uniform patching for time-series Transformers. PatchTST (Nie et al., 2023) established that partitioning a length-T series into $N = \lceil T/b \rceil$ patches of fixed length b reduces attention cost from $\mathcal{O}(T^2)$ to $\mathcal{O}(N^2)$ while providing implicit local smoothing; iTransformer (Liu et al., 2024) kept a fixed-token convention while reorganising what each token represents. Subsequent foundation-model work (TimesFM (Das et al., 2024), Chronos (Ansari et al., 2024), Moirai (Woo et al., 2024)) mostly retained static patching or tokenization choices as the default. Our empirical comparison uses a validation-selected uniform patch size for each method, dataset, and horizon, following the emphasis on simple baselines in DLinear and TSMixer (Zeng et al., 2023; Chen et al., 2023).

Dynamic patching and adaptive tokenization in time series. Recent methods place boundaries adaptively or vary granularity using a range of proxies: EntroPE (Abeywickrama et al., 2025) uses predictive entropy; TimeSqueeze (Ankireddy et al., 2025) uses an SSM encoder to measure information density; TimeMosaic (Ding et al., 2026) uses temporal heterogeneity; PathFormer (Chen et al., 2024), DualSG (Ding et al., 2025), PatchMLP (Tang & Zhang, 2025), and Kairos (Feng et al., 2025) route or mix multiple granularities; and HDMixer (Huang et al., 2024) mixes hierarchical, length-variable patches. ReinPatch (Wu et al., 2026) instead learns patch boundary placement as a reinforcement-learning policy jointly with the downstream sequence backbone. A related but distinct line reshapes the training objective or patch-specific processing rather than the boundaries themselves (Kudrat et al., 2025; Hu et al., 2025).

Adaptive tokenization and computation. Vision transformers provide the empirical precedent for adaptive tokenization. NaViT (Dehghani et al., 2023) processes images at their native resolution by packing variable-length token sequences; adaptive patch-size ViTs (Ronen et al., 2023) learn spatially-varying granularity; A-ViT (Yin et al., 2022) halts computation per token according to an informativeness criterion; and Token Merging (Bolya et al., 2023) compresses redundant tokens post-hoc. The same idea has a longer history in *adaptive computation*: Adaptive Computation Time lets recurrent networks learn input-dependent computation steps (Graves, 2016); PonderNet learns a probabilistic halting distribution (Banino et al., 2021); sparse Mixture-of-Experts routes examples or tokens to a subset of experts (Shazeer et al., 2017; Fedus et al., 2022); and Mixture-of-Depths routes tokens through a subset of Transformer blocks (Raposo et al., 2024). These works motivate examining both the task benefit and the execution cost of an adaptive routing or halting policy.

3 EMPIRICAL STUDY: ADAPTIVE VERSUS TUNED UNIFORM PATCHING

3.1 COMPARISON AND SELECTION PROTOCOL

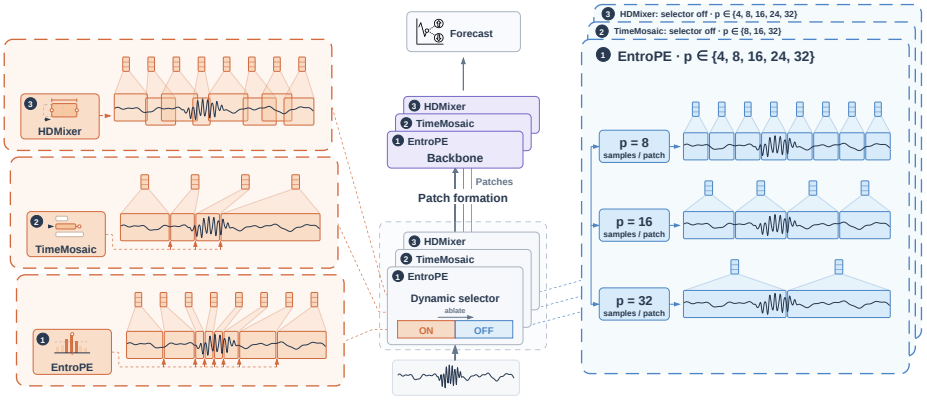


Figure 3: **Ablation on the dynamic methods.** For each method, the adaptive component is replaced by a fixed patch-length sweep.

We compare each adaptive method with a counterpart using one fixed patch length, select both configurations independently on validation data, and evaluate their test accuracy and training and inference costs on fresh runs (Figure 3). We evaluate three methods: **EntroPE** (Abeywickrama et al., 2025), which places boundaries using predictive entropy; **TimeMosaic** (Ding et al., 2026), which routes patch sizes using temporal heterogeneity; and **HDMixer** (Huang et al., 2024), which uses length-extendable patching. The backbone, prediction-head construction, optimizer, training budget, and preprocessing follow the matched implementation (Section D). This comparison evaluates complete selected systems: token counts, parameter counts, and nominal resolution budgets can differ between arms. Horizon grids, patch-size grids, seeds, and selection rules are reported in Section E; Section C gives the complete selected results. We evaluate eight standard real-world multivariate forecasting benchmarks: ETT (ETTh1, ETTh2, ETTm1, ETTm2) and Weather (Zhou et al., 2021), and Electricity, Traffic, and Exchange (Lai et al., 2018).

Metrics and aggregation. Let D_{dh} and U_{dh} denote mean test MSE over seeds for dataset d and horizon h. We report $G_{dh} = 100(D_{dh} - U_{dh})/D_{dh}$; positive gains favor uniform. Dataset-level gains first average each arm across horizons and then apply the same formula. Training cost is measured from timed optimization epochs after warm-up and reported in seconds per epoch. Inference cost is batch latency; speedup is adaptive/uniform mean time, so values above one favor uniform. Measurement details are given in Section E.

3.2 ACCURACY AND COMPUTATIONAL COST

Uniform has lower mean MSE in 56.3%, 71.9%, and 37.5% of settings for EntroPE, HDMixer, and TimeMosaic, respectively. Across all setting-level gains, 79.2% lie within $\pm 2\%$ and 92.7% within $\pm 5\%$. The dataset averages in Figure 2a cluster near equal test MSE, with the larger deviations depending on method and dataset. The full range is -5.02% (EntroPE, ETTm2, horizon 336) to $+15.15\%$ (HDMixer, Electricity, horizon 720). Dataset-level gains average $+0.20\%$, $+1.81\%$, and -0.10% in the same method order (Table 1). HDMixer on Electricity is the clearest outlier: uniform reduces horizon-averaged MSE by 10.26%. TimeMosaic favors adaptive on 75% of dataset averages, although those losses for uniform remain below 1%. Thus, the largest uniform advantage is concentrated in HDMixer–Electricity, while TimeMosaic more often favors adaptive patching. Figure 2b shows that most training-time points lie below equality, indicating faster epochs for the uniform variants. Figure 2c shows the same pattern for batch latency. Table 1 shows that the magnitude depends strongly on the method. EntroPE averages $1.41\times$ for training and $1.94\times$ for inference. The corresponding HDMixer ratios are $5.74\times$ and $5.63\times$; TimeMosaic remains close to parity at $1.07\times$ and $1.08\times$.

Table 1: **Dataset-level accuracy and efficiency after validation selection.** Relative test-MSE gain $G = 100(D - U)/D$ (%; positive favors uniform) and adaptive/uniform training and inference speedups. Table entries average seeds and horizons before forming ratios. Bold speedups exceed $1\times$; colored gains show direction, not significance. Details are in Section E.

	EntroPE			HDMixer			TimeMosaic		
Dataset	G	S_{tr}	S_{inf}	G	S_{tr}	S_{inf}	G	S_{tr}	S_{inf}
ETTh1	+1.00	**1.29**	**1.79**	+1.97	**4.43**	**3.87**	−0.66	**1.07**	**1.09**
ETTh2	+0.34	**1.42**	**1.97**	+2.02	**7.62**	**8.45**	−0.14	**1.09**	**1.12**
ETTm1	−0.67	**1.36**	**1.94**	−0.53	**5.28**	**4.24**	−0.44	**1.14**	**1.18**
ETTm2	−1.45	**1.38**	**1.96**	−0.54	**5.47**	**4.55**	−0.91	**1.10**	**1.13**
Electricity	+0.15	**1.28**	**1.43**	+10.26	**12.63**	**11.57**	−0.12	**1.00**	**1.00**
Exchange	+0.28	**1.30**	**2.16**	+0.94	**1.79**	**2.95**	+1.59	**1.07**	**1.09**
Traffic	+1.85	**1.83**	**2.36**	−0.13	**6.57**	**5.25**	+0.03	**1.06**	**1.02**
Weather	+0.08	**1.42**	**1.87**	+0.48	**2.15**	**4.15**	−0.17	**1.02**	**1.03**
Mean	+0.20	1.41	1.94	+1.81	5.74	5.63	−0.10	1.07	1.08

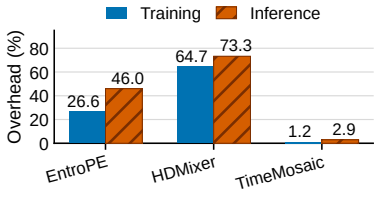


Figure 4: Mean router overhead.

To estimate routing overhead, we first run the adaptive model and save the patch tokens it produces. We then time a second run with the same checkpoint and inputs, bypassing adaptive token generation and feeding the saved tokens to the forecasting backbone. The measured time reduction (Figure 4) accounts for 26.6% of EntroPE training time and 46.0% of its inference time, compared with 64.7% and 73.3% for HDMixer. TimeMosaic's routing contribution is small at 1.2% and 2.9%. HDMixer's auxiliary objective contributes a further 12.2% of native adaptive training-step time on average. Uniform speedups are substantial for HDMixer and small for TimeMosaic, whose adaptive variant has small accuracy advantages on most datasets. The practical benefit of removing adaptive patching therefore depends on the method.

4 A GENERAL FRAMEWORK FOR BUDGETED PATCH ALLOCATION

The empirical comparison motivates the study of patch allocation as a mathematical problem. Inspired by the allocation perspective of rate–distortion theory (Cover & Thomas, 2006), we represent uniform, nonuniform, and overlapping patch layouts through their resolution densities under a fixed nominal budget. This representation is independent of the boundary-generation rule and can be paired with different models of how resolution affects loss. Here we study separable objectives with regional weights and a shared loss response to characterize the roles of placement, dispersion, and budget in reallocation.

Figure 5 previews the exact improvement criterion, its local quadratic surrogate, and the common-minimum regime.

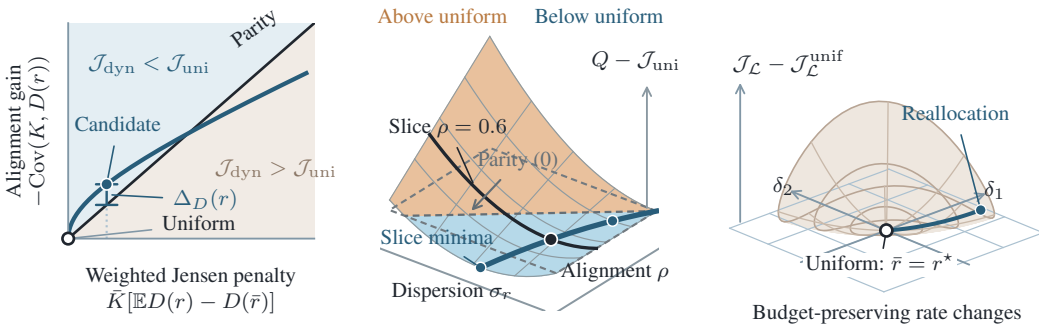
4.1 SETUP

Definition 4.1 (Nominal resolution density)**.** Fix an integer sequence length $T \geq 1$ and a nominal per-token budget $C > 0$. A patching scheme produces N nonempty patches $P_1, \ldots, P_N$ covering (possibly overlapping) intervals $S_i \subseteq \{0, \ldots, T-1\}$ of length $L_i = |S_i| > 0$. The *nominal resolution density* at step t is

$$r_t := \sum_{i=1}^{N} \frac{C}{L_i} \mathbf{1}\{t \in S_i\} \tag{1}$$

Each patch contributes C/L_i units uniformly across its span. The constant C defines a nominal accounting unit for comparing resolution across patch layouts.

Definition 4.2 (Budget constraint)**.** With common C and fixed N, T, every such patching scheme spends the same total nominal budget: $\sum_t r_t = NC$. The mean rate $\bar{r} := \mathbb{E}_t[r_t] = NC/T$ is the same for uniform and nonuniform schemes with those fixed quantities.



(a) Exact improvement criterion. (b) Local quadratic surrogate. (c) Common-loss minimum.

Figure 5: **Budgeted allocation in the shared-loss model.** All panels use illustrative objectives. **(a) Exact improvement criterion, Eq. (2).** Alignment gain $-\text{Cov}(K, D(r))$ must exceed the Jensen penalty $\bar{K}[\mathbb{E}D(r) - D(\bar{r})]$. The vertical gap above the diagonal is the gain $\Delta_D(r)$. **(b) Local quadratic surrogate.** For the illustrative surrogate $Q - \mathcal{J}_{\text{uni}} = \sigma_r^2/2 - \rho\sigma_r$ corresponding to Eq. (3), blue lies below uniform and orange above. The black curve fixes alignment $\rho = 0.6$; the dark blue curve joins minima at positive alignment. At positive alignment, the optimal gain in this surrogate scales with ρ^2 by Eq. (4). **(c) Common-loss minimum, Eq. (6).** When the mean rate equals the common minimizing rate $r^\star$, uniform allocation is globally optimal: $\Delta_{\mathcal{L}}(r) \le 0$.

Definition 4.3 (Feasible allocations and objective)**.** For analysis, sample t uniformly from $\{0, \dots, T-1\}$ and write $K = K_t$, $r = r_t$, and $\mathbb{E}$ for the empirical average over time. Let $K_t \ge 0$ be regional loss weights with $\bar{K} := \mathbb{E}[K] > 0$, and let $I \subset (0, \infty)$ be the admissible rate interval. The continuous relaxation is $\mathcal{F}(\bar{r}, I) := \{r \in I^T : \mathbb{E}[r] = \bar{r}\}$, of which realizable patching policies form a subset. For a shared convex loss $D : I \to \mathbb{R}$, define the normalized dynamic loss $\mathcal{J}_{\text{dyn}}(r) := \mathbb{E}[K\, D(r)]$, uniform loss $\mathcal{J}_{\text{uni}} := \bar{K}\, D(\bar{r})$, and gain $\Delta_D(r) := \mathcal{J}_{\text{uni}} - \mathcal{J}_{\text{dyn}}(r)$. Positive Δ_D denotes improvement over uniform allocation. For the decreasing-loss results, we additionally assume $D'(r) < 0$ and $D''(r) > 0$ on the relevant interval. We set $\sigma_X^2 = \mathbb{E}[(X - \mathbb{E}X)^2]$ and define $\text{Corr}(X, Y) = \text{Cov}(X, Y)/(\sigma_X \sigma_Y)$ when both standard deviations are positive, and zero otherwise. Derivatives are ordinary ambient derivatives; assumptions at included endpoints refer to an extension to a neighborhood of the endpoint.

The regional loss weight K_t has a direct interpretation within the assumed objective: at a uniform rate $\bar{r}$ with $D'(\bar{r}) < 0$, the marginal reduction in the regional loss term from additional resolution is $K_t[-D'(\bar{r})]$. Larger weights therefore identify regions with greater marginal benefit at this operating point.

The comparison holds the weights, loss function, and mean rate fixed. We ask which feasible allocations satisfy $\Delta_D(r) > 0$.

4.2 An exact criterion for improvement

The covariance decomposition separates the effect of placing resolution from the effect of varying it across regions.

Theorem 4.4 (Exact improvement criterion)**.** *Regard t as a uniform random index, and let $K = K_t$ and $r = r_t$ with $\mathbb{E}[r] = \bar{r}$. Then $\mathcal{J}_{dyn}(r) < \mathcal{J}_{uni}$ if and only if*

$$\underbrace{-\text{Cov}\big(K, D(r)\big)}_{\textit{alignment gain}} \;>\; \mathbb{E}[K]\; \underbrace{\big(\mathbb{E}[D(r)] - D(\bar{r})\big)}_{\textit{Jensen penalty}}. \tag{2}$$

Proof. See Section A.3. □

Convexity makes the right-hand side nonnegative. A candidate allocation therefore improves the objective only when its alignment term exceeds this dispersion penalty. The criterion applies to any feasible allocation and uses no approximation or assumption about the procedure that produced it.

Figure 5a plots this comparison: the vertical gap between alignment gain and the parity line is $\Delta_D(r)$. Allocations above the parity line improve on uniform; those below it increase the objective. The illustrated curve uses $D(r) = 1/r$, $K = (1/2, 3/2)$, and $r = (1-u, 1+u)$, $0 \le u \le 0.63$, at fixed mean rate $\bar{r} = 1$.

Allocations independent of the weights. If the rate and weight are independent under the uniform time index, the alignment term vanishes and only the dispersion penalty remains.

Proposition 4.5 (Nonimprovement under independence)**.** *If r_t is statistically independent of K_t, $\mathbb{E}[r] = \bar{r}$, and D is convex, then $\mathcal{J}_{dyn}(r) \ge \mathcal{J}_{uni}$. If D is strictly convex on the interval I, equality holds iff $r_t \equiv \bar{r}$ almost surely.*

Proof. See Section A.2. □

4.3 Improving allocations for decreasing loss

For a decreasing loss, heterogeneous weights permit a small improving reallocation when the uniform rate is interior to the feasible interval.

Proposition 4.6 (Local existence of an improving aligned allocation)**.** *Assume D is differentiable at $\bar{r}$, $D'(\bar{r}) < 0$, $\mathrm{Var}(K) > 0$, and $\bar{r} \in \mathrm{int}(I)$ (the uniform mean rate lies in the interior of the admissible interval). For sufficiently small $\varepsilon > 0$, set $r_t^{(\varepsilon)} := \bar{r} + \varepsilon(K_t - \bar{K})$. Then $\mathbb{E}[r^{(\varepsilon)}] = \bar{r}$ and $\mathcal{J}_{dyn}(r^{(\varepsilon)}) < \mathcal{J}_{uni}$.*

Proof. See Section A.6. □

The construction gives an improving direction in the continuous relaxation. Applying it to integer patches involves checking which rate changes can be realized by admissible boundaries. For a strictly convex differentiable loss with $D'(r) < 0$, positive weights, and an interior constrained optimum, Theorem A.3 gives $K_t D'(r_t^\star) + \lambda = 0$ with $\lambda > 0$. Optimal rates then increase with the weights. The fixed budget couples the regions: increasing one rate requires reducing another. Without coupling, positive scalar weights preserve the minimizer of a shared pointwise loss (Theorem A.1).

4.4 Local quadratic analysis

A Taylor expansion describes the gain near the uniform allocation. The exact expansion retains a weighted second-order term and a remainder; a scalar quadratic surrogate requires an additional approximation.

The exact second-order expansion with remainder is given in Proposition A.4. Neglecting its controlled remainder gives the second-order criterion $-D'(\bar{r})\mathrm{Cov}(K, r) > \frac{1}{2} D''(\bar{r}) \mathbb{E}[K(r-\bar{r})^2]$. A further joint-moment approximation $\mathrm{Cov}(K, \delta^2) \approx 0$ gives the scalar surrogate below. Low skewness of K alone does not justify this step: for the special perturbation $\delta = \varepsilon(K - \bar{K})$, the omitted covariance equals $\varepsilon^2 \mathbb{E}[(K-\bar{K})^3]$, but a general allocation need not have this form. Under the additional approximation,

$$\underbrace{-D'(\bar{r})\,\mathrm{Cov}(K, r)}_{\text{alignment gain (linear)}} \;>\; \underbrace{\tfrac{1}{2}\,\bar{K}\,D''(\bar{r})\,\mathrm{Var}(r)}_{\text{dispersion penalty (quadratic)}}\,. \tag{3}$$

The right-hand side combines the local curvature with the variance of the allocation. Equation (3) describes the surrogate; the exact improvement criterion remains Equation (2).

Theorem 4.7 (Maximum of the quadratic surrogate)**.** *Let $a = -D'(\bar{r}) > 0$, $b = D''(\bar{r}) > 0$, and $\rho = \mathrm{Corr}(K, r)$, treated as fixed. The local surrogate $a\rho\sigma_K\sigma_r - \frac{1}{2}\bar{K}b\sigma_r^2$ has maximum over $\sigma_r \ge 0$*

$$\Delta_{\max}^{quad} \;=\; \begin{cases} \dfrac{\left(-D'(\bar{r})\right)^2 \rho^2 \sigma_K^2}{2\bar{K} D''(\bar{r})}, & \rho > 0, \\ 0, & \rho \le 0, \end{cases} \tag{4}$$

with maximiser $\sigma_r^\star = (-D'(\bar{r}))\rho\sigma_K/(\bar{K}D''(\bar{r}))$ *when* $\rho > 0$.

Proof. The surrogate is a concave quadratic in σ_r at fixed ρ, σ_K; read off its maximiser. See Section A.4. □

Expression (4) maximizes the scalar surrogate over $\sigma_r \geq 0$ while holding its other coefficients fixed. This step also relaxes the constraints on realizable allocations, so the value need not be achievable. At fixed remaining coefficients, the surrogate optimum scales with $\max\{\rho, 0\}^2\sigma_K^2$ and inversely with $D''(\bar{r})$.

Figure 5b plots $Q - \mathcal{J}_{\text{uni}} = \sigma_r^2/2 - \rho\sigma_r$. The dark blue curve joins minima across positive-alignment slices; the black curve fixes $\rho = 0.6$. Zero dispersion recovers uniform allocation.

4.5 A GLOBAL GAIN BOUND

Strong convexity and a bounded derivative give a bound that holds across the feasible set, without a local approximation.

Theorem 4.8 (Global gain bound under strong convexity)**.** *Assume D is twice differentiable at every point of the interval I, with ordinary ambient derivatives at any included endpoints. Set $m_D := \inf_{u\in I} D''(u) > 0$, $L_D := \sup_{u\in I} |D'(u)| < \infty$, and $\alpha_+(r) := \max\{\mathrm{Corr}(K, -D(r)), 0\}$. For any feasible allocation,*

$$\Delta_D(r) \leq \alpha_+(r)L_D\sigma_K\sigma_r - \tfrac{1}{2}\bar{K}m_D\sigma_r^2 \leq \frac{\alpha_+(r)^2L_D^2\sigma_K^2}{2\bar{K}m_D} \leq \frac{L_D^2\sigma_K^2}{2\bar{K}m_D}. \tag{5}$$

Proof. See Section A.8. □

The allocation-dependent bound uses $\alpha_+(r) = \max\{\mathrm{Corr}(K, -D(r)), 0\}$; the quadratic surrogate instead fixes $\mathrm{Corr}(K, r)$. The final inequality removes the correlation term and bounds the gain uniformly over the feasible set.

4.6 UNIFORM OPTIMALITY AT A COMMON LOSS MINIMUM

Consider a shared loss $\mathcal{L}$ with a common global minimum at $r^\star$. Positive regional weights preserve this minimizing rate. The following assumption ensures uniqueness; the gain bound itself requires only a common global minimum and nonnegative weights.

Assumption 4.9 (Unique common loss minimum)**.** $\mathcal{L} \in C^2(I)$, $\mathcal{L}''(r) > 0$ on I, and there exists $r^\star \in I$ with $\mathcal{L}'(r^\star) = 0$. Hence $r^\star$ is the unique global minimiser of $\mathcal{L}$.

Write $\mathcal{J}_{\mathcal{L}}(r) := \mathbb{E}[K\mathcal{L}(r)]$ and $\mathcal{J}_{\mathcal{L}}^{\text{unif}} := \bar{K}\mathcal{L}(\bar{r})$. The exact decomposition and improvement criterion apply with D replaced by $\mathcal{L}$; convexity again makes the dispersion penalty nonnegative.

Theorem 4.10 (Uniform optimality and remaining absolute gain)**.** *Assume $K_t \geq 0$ and that $\mathcal{L}$ has a common global minimiser $r^\star \in I$. For every feasible allocation,*

$$\Delta_{\mathcal{L}}(r) \leq \bar{K}\,[\mathcal{L}(\bar{r}) - \mathcal{L}(r^\star)], \qquad \Delta_{\mathcal{L}} := \mathcal{J}_{\mathcal{L}}^{\text{unif}} - \mathcal{J}_{\mathcal{L}}(r). \tag{6}$$

In particular, if $\bar{r} = r^\star$, uniform allocation is globally optimal. Under Assumption 4.9 with $K_t > 0$, equality of dynamic and uniform loss at that budget holds only for $r_t \equiv r^\star$.

Proof. The pointwise inequality $\mathcal{L}(r_t) \geq \mathcal{L}(r^\star)$ gives $\mathcal{J}_{\mathcal{L}}(r) \geq \bar{K}\mathcal{L}(r^\star)$. Subtract it from $\bar{K}\mathcal{L}(\bar{r})$. See Section A.9 for equality conditions.

□

For fixed K and continuous $\mathcal{L}$, the absolute-gain upper bound tends to zero as $\bar{r} \to r^\star$.

The local expansion and its remainder bound are given in Section A.10. An optional decomposition of $\mathcal{L}$ into a base loss and an auxiliary scalar cost is examined in Section A.11.

Figure 5c illustrates this result on a surface of budget-preserving reallocations: uniform lies at the minimum when $\bar{r} = r^\star$. The example uses $\mathcal{L}(r) = 1 + (r-1)^2$, $K = (1, 2, 3)$, and $\bar{r} = r^\star = 1$. The two free rate changes satisfy $\delta_3 = -\delta_1 - \delta_2$, preserving the budget throughout the surface.

4.7 Allocation regimes

The exact criterion and gain analyses distinguish two regimes in the shared-loss model. Heterogeneous weights permit local improvement for decreasing loss under the interior conditions of Proposition 4.6, whereas uniform allocation is optimal at a common minimizing rate (Theorem 4.10). The local surrogate and global bound quantify gain under their respective assumptions.

5 Using the allocation framework

Interpreting regional weights. Regional weights let the model represent differences in sensitivity to resolution. At a uniform rate with $D'(\bar{r}) < 0$, the marginal benefit of added resolution is $K_t[-D'(\bar{r})]$, so σ_K measures heterogeneity in this benefit up to a common factor. Treating K_t as a specified weight allows that heterogeneity to be studied separately from the allocation.

Boundary placement and resolution dispersion. Boundaries enter through the induced rates r_t (Definition 4.1). Two nonoverlapping partitions with the same multiset of patch lengths have the same rate distribution, budget, and Jensen penalty, but can have different alignment gains $-\mathrm{Cov}(K, D(r))$. The formulation thus distinguishes where resolution is placed from how unevenly it is distributed. Candidate boundary rules can be studied through their induced allocations, accounting for both refined regions and those coarsened to preserve the budget.

Alignment and redistribution strength. In the two-region example of Figure 5a, $\mathrm{Corr}(K, r) = 1$ along the nonuniform part of the curve, yet excessive redistribution makes the objective exceed its uniform value. Correctly ordering the rates by the weights therefore leaves the amount of redistribution undetermined. The maximizing dispersion of the local surrogate depends on the loss slope and curvature, regional weights, and the specified alignment (Theorem 4.7).

Modeling regional resolution needs. For a forecasting application, the framework offers a way to distinguish regional importance from preferred patch resolution. Positive regional weights represent differences in the scale of a shared loss curve while preserving its minimizing rate. Under this modeling choice, uniform allocation is optimal when the mean resolution budget matches that common preferred rate (Theorem 4.10).

6 Conclusion and future work

Future work. Future work could extend the analysis to objectives with interactions between regions, retaining the resolution-density representation and fixed budget. In such models, refining one region could increase or decrease the benefit of refining another. A central question is how the structure and strength of these interactions change the conditions for improving reallocations and uniform optimality.

The empirical comparison finds mostly small accuracy differences between adaptive and tuned uniform systems, with generally lower measured execution costs for uniform variants. These observations support careful uniform-baseline selection in the evaluated forecasting settings and motivate a formal study of patch allocation. Our theoretical contribution is a general framework that represents uniform, nonuniform, and overlapping patch layouts through nominal resolution under a common budget. The empirical results support a practical starting point: tune the uniform patch length carefully and assess any accuracy gain from adaptive patching alongside its additional routing cost. The shared-loss analysis adds a design lesson: even an allocation that correctly ranks regions by importance can redistribute too much resolution, so placement and refinement strength should be considered together.

AI USE STATEMENT

The authors developed the initial research idea from their own experiments and proposed the initial mathematical formulation. Generative AI tools assisted with identifying relevant literature, coding, outlining and drafting portions of the paper, and polishing its language. They were also used at intermediate stages of proof development to explore possible directions and to assist in refining mathematical statements and parts of their proofs. The authors take responsibility for the final text, mathematical claims, proofs, and reported results, including material developed with AI assistance.

REPRODUCIBILITY STATEMENT

The assumptions and proofs are provided in Sections A and 4. Sections B and C report the statistical procedures and full selected comparisons. Sections D to F document the uniform adaptations, search spaces, training and timing protocols, and dataset splits. We also provide the modified code for the uniform-patch variants of EntroPE, HDMixer, and TimeMosaic as supplementary material, together with a Lean formalization of the paper's theoretical results.

REFERENCES

Sachith Abeywickrama, Emadeldeen Eldele, Min Wu, Xiaoli Li, and Chau Yuen. Entrope: Entropy-guided dynamic patch encoder for time series forecasting. *arXiv preprint arXiv:2509.26157*, 2025.

Sravan Kumar Ankireddy, Nikita Seleznev, Nam H Nguyen, Yulun Wu, Senthil Kumar, Furong Huang, and C. Bayan Bruss. Timesqueeze: Dynamic patching for efficient long-context time series forecasting. In *Recent Advances in Time Series Foundation Models Have We Reached the 'BERT Moment'?*, 2025.

Abdul Fatir Ansari, Lorenzo Stella, Ali Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, Jasper Zschiegner, Danielle C. Maddix, Hao Wang, Michael W. Mahoney, Kari Torkkola, Andrew Gordon Wilson, Michael Bohlke-Schneider, and Bernie Wang. Chronos: Learning the language of time series. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Andrea Banino, Jan Balaguer, and Charles Blundell. Pondernet: Learning to ponder. *ArXiv*, 2021.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *The Eleventh International Conference on Learning Representations*, 2023.

Peng Chen, Yingying ZHANG, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

Si-An Chen, Chun-Liang Li, Sercan O Arik, Nathanael Christian Yoder, and Tomas Pfister. Tsmixer: An all-mlp architecture for time series forecast-ing. *Transactions on Machine Learning Research*, 2023.

T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 2nd edition, 2006.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 10148–10167, 2024.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, Avital Oliver, Piotr Padlewski, Alexey A. Gritsenko, Mario Lucic, and Neil Houlsby. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274, 2023.

Kuiye Ding, Fanda Fan, Yao Wang, Ruijie Jian, Xiaorui Wang, Luqi Gong, Yishan Jiang, Chunjie Luo, and Jianfeng Zhan. Dualsg: A dual-stream explicit semantic-guided multivariate time series forecasting framework. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 508–517, 2025.

Kuiye Ding, Fanda Fan, Chunyi Hou, Zheya Wang, Lei Wang, Zhengxin Yang, and Jianfeng Zhan. Timemosaic: Temporal heterogeneity guided time series forecasting via adaptive granularity patch and segment-wise decoding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20790–20798, 2026.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

Kun Feng, Shaocheng Lan, Yuchen Fang, Wenchao He, Sihan Lu, Shuqi Gu, Lintao Ma, Xingyu Lu, and Kan Ren. Kairos: Toward adaptive and parameter-efficient time series foundation models. *arXiv preprint arXiv:2509.25826*, 2025.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Yifan Hu, Guibin Zhang, Peiyuan Liu, Disen Lan, Naiqi Li, Dawei Cheng, Tao Dai, Shu-Tao Xia, and Shirui Pan. Timefilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *International Conference on Machine Learning*, pp. 24893–24911. PMLR, 2025.

Qihe Huang, Lei Shen, Ruixin Zhang, Jiahuan Cheng, Shouhong Ding, Zhengyang Zhou, and Yang Wang. Hdmixer: Hierarchical dependency with extendable patch for multivariate time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 12608–12616, 2024.

Dilfira Kudrat, Zongxia Xie, Yanru Sun, Tianyu Jia, and Qinghua Hu. Patch-wise structural loss for time series forecasting. In *International Conference on Machine Learning*, pp. 31841–31859. PMLR, 2025.

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

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. In *The Twelfth International Conference on Learning Representations*, 2024.

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

David Raposo, Sam Ritter, Blake Richards, Timothy Lillicrap, Peter Conway Humphreys, and Adam Santoro. Mixture-of-depths: Dynamically allocating compute in transformer-based language models. *arXiv preprint arXiv:2404.02258*, 2024.

Tomer Ronen, Omer Levy, and Avram Golbert. Vision transformers with mixed-resolution tokenization. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 4613–4622. IEEE, 2023.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *ArXiv*, 2017.

Peiwang Tang and Weitai Zhang. Unlocking the power of patch: Patch-based mlp for long-term time series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 12640–12648, 2025.

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

Yulun Wu, Sravan Kumar Ankireddy, Samuel Sharpe, Nikita Seleznev, Dehao Yuan, Hyeji Kim, and Nam H. Nguyen. Dynamic tokenization via reinforcement patching: End-to-end training and zero-shot transfer. *arXiv preprint arXiv:2603.26097*, 2026.

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10809–10818, 2022.

Shicheng Yin, Kaixuan Yin, Yang Liu, Weixing Chen, and Liang Lin. Dart: Differentiable dynamic adaptive region tokenizer for vision foundation models. *arXiv preprint arXiv:2506.10390*, 2025.

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

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

APPENDIX

Structure. Section A collects the proofs and technical formal statements supporting the main theory. Section B defines the aggregation, paired seed tests, dataset-level uncertainty, and multiplicity corrections for the selected comparison. Section C reports all 96 selected accuracy and efficiency comparisons, including configuration choices. Section D describes the uniform adaptations. Section E specifies candidate grids, validation selection, seed counts, fixed hyperparameters, and measurement protocols; Section F gives the chronological dataset splits.

A PROOFS

The proofs below also include formal statements moved out of the main theory section for space.

A.1 PROOF OF THEOREM A.1 (SCALAR INVARIANCE)

Without any constraint linking the time steps, a common per-step rate loss g gives the decoupled problem

$$\min_{\{r_t \in I\}} \sum_{t=0}^{T-1} K_t \, g(r_t) \;=\; \sum_{t=0}^{T-1} \min_{r_t \in I} K_t \, g(r_t). \tag{7}$$

For $K_t > 0$, each weight rescales the same pointwise loss and preserves its minimizer.

Theorem A.1 (Scalar invariance in the unconstrained regime)**.** *Let $K_t > 0$ and let the same rate loss $g : I \to \mathbb{R}$ apply at every time step. If g has a unique minimiser $r^\star \in I$, the unique unconstrained optimizer is $r_t^\star = r^\star$ for all t. If instead $g = D$ is strictly decreasing on a compact interval $I = [r_{\min}, r_{\max}]$, the unconstrained optimizer is the boundary allocation $r_t^\star = r_{\max}$ for all t; on an interval unbounded above the pure decreasing problem has no finite optimum. In each case, the set of pointwise minimizers is independent of the positive weights.*

Intuition. A positive scalar K_t rescales the loss surface but does not move its argmin. Thus an unconstrained common-loss objective either selects the same common minimizer everywhere or, for a decreasing loss on a compact interval, the same upper rate boundary.

Corollary A.2 (Necessity of coupling or landscape variation)**.** *Under the positive-weight and common-unique-minimiser assumptions of Theorem A.1, no non-constant allocation is an unconstrained minimiser. A non-constant optimum must leave that regime. Possible mechanisms include:*

1. Budget coupling: *a total-resolution constraint forces time steps into a shared resource problem; under the assumptions of Theorem A.3, the KKT condition $K_t D'(r_t^\star) + \lambda = 0$ yields a K_t-dependent optimum.*
2. Time-varying loss landscape: *different time steps have different $\ell_t(r)$, whose per-step optima may differ independently of coupling.*
3. Additional structural constraints: *token limits, boundary constraints, nonlocal dependencies, or computational costs that create further coupling.*

For decreasing strictly convex D, positive heterogeneous K_t, and an interior constrained minimiser, Theorem A.3 gives strictly ordered optimal rates. For the different objective $\mathcal{L}$ with a common global minimiser $r^\star$, a budget with $\bar{r} = r^\star$ instead makes uniform allocation optimal (Theorem 4.10).

Proof of Theorem A.1. Without a coupling constraint, the global problem separates into independent per-step subproblems, as in Equation (7). For a common per-step loss g and $K_t > 0$,

$$\arg\min_{r_t \in I} K_t g(r_t) = \arg\min_{r_t \in I} g(r_t),$$

because multiplication by a positive scalar preserves the argmin. If g has a unique minimiser $r^\star$, each subproblem therefore selects $r_t^\star = r^\star$. If $g = D$ is strictly decreasing on a compact interval $[r_{\min}, r_{\max}]$, each subproblem is minimised at $r_{\max}$; on an unbounded interval the decreasing objective has no finite minimiser when the interval is unbounded above. The positive weights do not change the pointwise minimizer sets.

□

A.2 Proof of Proposition 4.5 (nonimprovement under independence)

Under independence, K and $D(r)$ are independent, so

$$\mathcal{J}_{\text{dyn}}(r) = \mathbb{E}[K D(r)] = \bar{K}\,\mathbb{E}[D(r)].$$

Because D is convex and $\mathbb{E}[r] = \bar{r}$, Jensen's inequality gives $\mathbb{E}[D(r)] \geq D(\bar{r})$, hence $\mathcal{J}_{\text{dyn}}(r) \geq \bar{K} D(\bar{r}) = \mathcal{J}_{\text{uni}}$. If D is strictly convex on I, equality requires $r_t \equiv \bar{r}$ almost surely.

A.3 Proof of Theorem 4.4 (exact threshold)

Treat t as a uniform random index and write $K = K_t$ and $r = r_t$. The dynamic scheme improves on the uniform baseline if and only if

$$\mathbb{E}[K D(r)] < \mathbb{E}[K] D(\bar{r}). \tag{8}$$

Using the covariance identity,

$$\mathbb{E}[K D(r)] = \mathrm{Cov}(K, D(r)) + \mathbb{E}[K]\,\mathbb{E}[D(r)]. \tag{9}$$

Substituting and rearranging gives

$$-\mathrm{Cov}(K, D(r)) > \mathbb{E}[K]\big(\mathbb{E}[D(r)] - D(\bar{r})\big), \tag{10}$$

which is exactly Equation (2).

A.4 Proof of Theorem 4.7 (quadratic-surrogate maximum)

Using the local quadratic approximation from Equation (3), define

$$\Delta(r) := (-D'(\bar{r}))\mathrm{Cov}(K, r) - \tfrac{1}{2}\bar{K} D''(\bar{r})\mathrm{Var}(r). \tag{11}$$

Write $\mathrm{Cov}(K, r) = \rho\sigma_K\sigma_r$ and $\mathrm{Var}(r) = \sigma_r^2$. Then

$$\Delta(r) = (-D'(\bar{r}))\rho\sigma_K\sigma_r - \tfrac{1}{2}\bar{K}D''(\bar{r})\sigma_r^2. \tag{12}$$

For fixed ρ and σ_K, this is a concave quadratic in σ_r. If $\rho \le 0$, the linear term is non-positive for all $\sigma_r \ge 0$, so the maximum is attained at $\sigma_r = 0$ with value 0. If $\rho > 0$, differentiating with respect to σ_r and setting to zero gives

$$\sigma_r^\star = \frac{(-D'(\bar{r}))\rho\sigma_K}{\bar{K}D''(\bar{r})}. \tag{13}$$

Substituting $\sigma_r^\star$ back into $\Delta(r)$ yields

$$\Delta_{\max}^{\text{quad}} \;=\; \frac{\big(-D'(\bar{r})\rho\sigma_K\big)^2}{2\bar{K}D''(\bar{r})}, \tag{14}$$

which is the positive-correlation case of Equation (4). This is the maximum of the quadratic surrogate, not an exact gain bound for arbitrary allocations.

A.5 Proof of Theorem A.3 (KKT characterisation of the optimal constrained allocation)

Theorem A.3 (KKT characterisation of the optimal constrained allocation)**.** *Assume $D \in C^1$ is strictly convex with $D'(r) < 0$ for all r, and $K_t > 0$. If the minimiser $r^\star$ of $\min_{r\in\mathcal{F}(\bar{r},I)} \mathbb{E}[K\,D(r)]$ is interior, there exists $\lambda > 0$ such that*

$$K_t\,D'(r_t^\star) + \lambda = 0, \qquad t = 0, \dots, T{-}1. \tag{15}$$

Equivalently $D'(r_t^\star) = -\lambda/K_t$, and the allocation is monotone *in the weights: $K_s > K_t \Rightarrow r_s^\star > r_t^\star$.*

Proof. The constrained optimisation problem is

$$\min_{\{r_t\}\in\mathcal{F}(\bar{r},I)} \frac{1}{T}\sum_{t=0}^{T-1} K_t D(r_t), \quad \mathcal{F}(\bar{r}, I) := \Big\{\{r_t\} \in I^T : \tfrac{1}{T}\sum_{t=0}^{T-1} r_t = \bar{r}\Big\}. \tag{16}$$

Form the Lagrangian

$$\mathcal{L}\big(\{r_t\}, \lambda\big) \;=\; \frac{1}{T}\sum_{t=0}^{T-1} K_t D(r_t) + \lambda\left(\frac{1}{T}\sum_{t=0}^{T-1} r_t - \bar{r}\right). \tag{17}$$

The gradient of the mean-rate constraint is nonzero since $T \ge 1$. The Lagrange multiplier rule therefore applies at an interior minimiser. Differentiating $\mathcal{L}$ with respect to r_t gives

$$\frac{1}{T}\big(K_t\,D'(r_t^\star) + \lambda\big) = 0, \qquad t = 0, \dots, T{-}1, \tag{18}$$

which is (15). Since $D'(r) < 0$ for all r and $K_t > 0$, the equation $K_t D'(r_t^\star) = -\lambda$ can only hold with $\lambda > 0$. Rearranging gives $D'(r_t^\star) = -\lambda/K_t$. A differentiable strictly convex function on an interval has a strictly increasing derivative, so

$$K_s > K_t \;\Rightarrow\; \frac{-\lambda}{K_s} > \frac{-\lambda}{K_t} \;\Rightarrow\; D'(r_s^\star) > D'(r_t^\star) \;\Rightarrow\; r_s^\star > r_t^\star. \tag{19}$$

Uniqueness of $r_t^\star$ for each t follows from strict convexity of D. □

A.6 Local existence of an improving aligned allocation

Proof. Define the perturbed allocation $r_t^{(\varepsilon)} := \bar{r} + \varepsilon(K_t - \bar{K})$ for $\varepsilon > 0$. The mean is $\mathbb{E}[r^{(\varepsilon)}] = \bar{r} + \varepsilon(\bar{K} - \bar{K}) = \bar{r}$, so the budget constraint is satisfied. Because $\bar{r}$ is an interior point and there are finitely many coordinates, sufficiently small $\varepsilon > 0$ also keeps every $r_t^{(\varepsilon)}$ in I.

Expand $D(r_t^{(\varepsilon)})$ to first order in ε:

$$D\big(\bar{r} + \varepsilon(K_t - \bar{K})\big) = D(\bar{r}) + D'(\bar{r})\,\varepsilon(K_t - \bar{K}) + o(\varepsilon). \tag{20}$$

Multiplying by K_t, summing over t, and dividing by T:

$$\mathcal{J}_{\text{dyn}}\big(r^{(\varepsilon)}\big) = D(\bar{r})\,\mathbb{E}[K] + D'(\bar{r})\,\varepsilon\,\mathbb{E}\big[K(K - \bar{K})\big] + o(\varepsilon). \tag{21}$$

Note that $\mathbb{E}[K(K - \bar{K})] = \mathbb{E}[K^2] - \bar{K}^2 = \mathrm{Var}(K) > 0$ by assumption. Therefore

$$\mathcal{J}_{\text{dyn}}\big(r^{(\varepsilon)}\big) - \mathcal{J}_{\text{uni}} = D'(\bar{r})\,\varepsilon\,\mathrm{Var}(K) + o(\varepsilon). \tag{22}$$

Since $D'(\bar{r}) < 0$ and $\mathrm{Var}(K) > 0$, the linear coefficient is strictly negative. For all sufficiently small $\varepsilon > 0$ the linear term dominates and $\mathcal{J}_{\text{dyn}}(r^{(\varepsilon)}) < \mathcal{J}_{\text{uni}}$. □

A.7 Exact second-order expansion (Proposition A.4)

Proposition A.4 (Exact second-order expansion with remainder)**.** *Let $\delta_t = r_t - \bar{r}$ and assume $D \in C^3$ on an open neighborhood of the segments joining $\bar{r}$ to each r_t. With $R_3(\delta) := D(\bar{r} + \delta) - D(\bar{r}) - D'(\bar{r})\delta - \frac{1}{2}D''(\bar{r})\delta^2$,*

$$\Delta_D(r) = -D'(\bar{r})\mathrm{Cov}(K, r) - \tfrac{1}{2}D''(\bar{r})\mathbb{E}[K\delta^2] - \mathbb{E}[KR_3(\delta)], \tag{23}$$

and $\mathbb{E}[K\delta^2] = \bar{K}\mathrm{Var}(r) + \mathrm{Cov}(K, \delta^2)$. If $|D'''| \le M_3$ throughout those segments, then $|\mathbb{E}[KR_3(\delta)]| \le \frac{M_3}{6}\mathbb{E}[K|\delta|^3]$.

Proof. By Taylor's theorem applied to $D \in C^3$:

$$D(r_t) = D(\bar{r}) + D'(\bar{r})\,\delta_t + \tfrac{1}{2}D''(\bar{r})\,\delta_t^2 + R_3(\delta_t), \tag{24}$$

where $\delta_t = r_t - \bar{r}$ and $R_3(\delta_t) = D(r_t) - D(\bar{r}) - D'(\bar{r})\delta_t - \frac{1}{2}D''(\bar{r})\delta_t^2$. Multiply by K_t, sum over t, and divide by T:

$$\frac{1}{T}\sum_t K_t D(r_t) = D(\bar{r})\,\mathbb{E}[K] + D'(\bar{r})\,\mathbb{E}[K\delta] + \tfrac{1}{2}D''(\bar{r})\,\mathbb{E}[K\delta^2] + \mathbb{E}[KR_3(\delta)]. \tag{25}$$

Since $\mathbb{E}[\delta] = \mathbb{E}[r - \bar{r}] = 0$ and $\mathbb{E}[K\delta] = \mathbb{E}[K(r - \bar{r})] = \mathbb{E}[Kr] - \bar{r}\,\mathbb{E}[K] = \mathrm{Cov}(K, r)$, we obtain

$$\Delta_D(r) = \mathcal{J}_{\text{uni}} - \mathcal{J}_{\text{dyn}}(r) = -D'(\bar{r})\,\mathrm{Cov}(K, r) - \tfrac{1}{2}D''(\bar{r})\,\mathbb{E}[K\delta^2] - \mathbb{E}[KR_3(\delta)], \tag{26}$$

which is (23).

For the expansion of $\mathbb{E}[K\delta^2]$, write $K = \bar{K} + (K - \bar{K})$:

$$\mathbb{E}[K\delta^2] = \bar{K}\,\mathbb{E}[\delta^2] + \mathbb{E}[(K - \bar{K})\delta^2] = \bar{K}\,\mathrm{Var}(r) + \mathrm{Cov}(K, \delta^2). \tag{27}$$

For the remainder bound: by Taylor's theorem with the Lagrange remainder, $|R_3(\delta)| \le \frac{M_3}{6}|\delta|^3$. Multiplying by $K \ge 0$ and taking expectations gives $|\mathbb{E}[KR_3(\delta)]| \le \frac{M_3}{6}\,\mathbb{E}[K|\delta|^3]$. □

A.8 Proof of Theorem 4.8 (global gain bound under strong convexity)

Proof. Starting from the exact decomposition implicit in Theorem 4.4,

$$\Delta_D(r) = -\mathrm{Cov}(K, D(r)) - \mathbb{E}[K]\big(\mathbb{E}[D(r)] - D(\bar{r})\big). \tag{28}$$

Step 1: upper-bounding the covariance term. Writing $-\mathrm{Cov}(K, D(r)) = \mathrm{Cov}(K, -D(r))$, we use $\mathrm{Cov}(K, -D(r)) \le \alpha_+(r)\,\sigma_K\,\sigma_{D(r)}$, where $\alpha_+(r) = \max\{\mathrm{Corr}(K, -D(r)), 0\}$ (the non-negative part of the correlation is the only part that can benefit). Since $|D'| \le L_D$ on I, D is L_D-Lipschitz, so $\sigma_{D(r)} \le L_D\,\sigma_r$. Therefore $-\mathrm{Cov}(K, D(r)) \le \alpha_+(r)\,L_D\,\sigma_K\,\sigma_r$.

Step 2: lower-bounding the Jensen penalty. Under m_D-strong convexity, for any r with $\mathbb{E}[r] = \bar{r}$,

$$\mathbb{E}[D(r)] \;\ge\; D(\bar{r}) + \tfrac{m_D}{2}\,\mathbb{E}[(r-\bar{r})^2] = D(\bar{r}) + \tfrac{m_D}{2}\,\sigma_r^2, \tag{29}$$

so $\mathbb{E}[D(r)] - D(\bar{r}) \ge \frac{m_D}{2}\sigma_r^2$ and $-\mathbb{E}[K](\mathbb{E}[D(r)] - D(\bar{r})) \le -\frac{1}{2}\bar{K}\,m_D\,\sigma_r^2$.

Step 3: combining.

$$\Delta_D(r) \;\le\; \alpha_+(r)\,L_D\,\sigma_K\,\sigma_r - \tfrac{1}{2}\,\bar{K}\,m_D\,\sigma_r^2. \tag{30}$$

Step 4: completing the square. For each fixed allocation, regard $\alpha_+(r)$ as a fixed coefficient. The scalar quadratic in a variable $s \ge 0$ is bounded by its value at $s^\star = \alpha_+(r)L_D\sigma_K/(\bar{K}m_D)$. Applying this algebraic inequality at $s = \sigma_r$ gives

$$\Delta_D(r) \;\le\; \frac{\alpha_+(r)^2\,L_D^2\,\sigma_K^2}{2\,\bar{K}\,m_D} \;\le\; \frac{L_D^2\,\sigma_K^2}{2\,\bar{K}\,m_D}, \tag{31}$$

where the last step uses $\alpha_+(r) \le 1$, giving (5). □

A.9 Proof of Theorem 4.10 (uniform optimality at a common loss minimum)

Let $r^\star$ be a global minimiser of $\mathcal{L}$ on I. For any feasible allocation, $\mathcal{J}_{\mathcal{L}}(r) \ge \bar{K}\mathcal{L}(r^\star)$, and hence

$$\Delta_{\mathcal{L}}(r) = \bar{K}\mathcal{L}(\bar{r}) - \mathcal{J}_{\mathcal{L}}(r) \le \bar{K}[\mathcal{L}(\bar{r}) - \mathcal{L}(r^\star)].$$

This proves Equation (6). Now suppose $\bar{r} = r^\star$. For every time step t,

$$\mathcal{L}(r_t) \;\ge\; \mathcal{L}(r^\star) \quad \forall\, r_t, \tag{32}$$

since $r^\star$ is the global minimiser. Since $K_t \ge 0$,

$$\mathcal{J}_{\mathcal{L}}(r) = \mathbb{E}[K\,\mathcal{L}(r)] \;\ge\; \mathbb{E}[K\,\mathcal{L}(r^\star)] = \bar{K}\,\mathcal{L}(r^\star) = \bar{K}\,\mathcal{L}(\bar{r}) = \mathcal{J}_{\mathcal{L}}^{\mathrm{unif}}. \tag{33}$$

For the equality condition: $\mathcal{J}_{\mathcal{L}}(r) = \mathcal{J}_{\mathcal{L}}^{\mathrm{unif}}$ requires $K_t(\mathcal{L}(r_t) - \mathcal{L}(r^\star)) = 0$ for every t. When $K_t > 0$ and $\mathcal{L}(r_t) \ge \mathcal{L}(r^\star)$, this forces $\mathcal{L}(r_t) = \mathcal{L}(r^\star)$. If $r^\star$ is the unique minimiser of $\mathcal{L}$ (ensured by Assumption 4.9), then $r_t = r^\star$ for all t.

A.10 Proof of Proposition A.5 (local expansion of the shared loss)

Theorem 4.10 is the global statement. Expanding $\mathcal{L}$ around $\bar{r}$ gives the corresponding local terms.

Proposition A.5 (Local expansion of the shared loss)**.** *Let $\delta_t = r_t - \bar{r}$ and assume $\mathcal{L} \in C^3$ on a neighborhood of each segment between $\bar{r}$ and r_t. Then*

$$\Delta_{\mathcal{L}}(r) \;=\; \underbrace{-\mathcal{L}'(\bar{r})\,\mathrm{Cov}(K, r)}_{\textit{first-order alignment}} \;-\; \underbrace{\tfrac{1}{2}\,\mathcal{L}''(\bar{r})\,\mathbb{E}[K\delta^2]}_{\textit{weighted curvature term}} \;-\; \mathbb{E}[K\,R_{\mathcal{L},3}(\delta)]. \tag{34}$$

The first-order term vanishes when $\mathcal{L}'(\bar{r}) = 0$ exactly. If $\mathcal{L}'(\bar{r}) = 0$ and $|\mathcal{L}^{(3)}| \le M_3$ on those segments, then

$$\big|\Delta_{\mathcal{L}}(r) + \tfrac{1}{2}\mathcal{L}''(\bar{r})\mathbb{E}[K\delta^2]\big| \le \tfrac{M_3}{6}\mathbb{E}[K|\delta|^3]. \tag{35}$$

When $\mathcal{L}'(\bar{r})$ is only small, its term must be retained and can yield positive gain. The exact near-minimum restriction is Equation (6)*, not nonpositivity throughout a neighborhood.*

At a stationary scalar operating point, $\mathcal{L}'(\bar{r}) = 0$. Away from that point, the first-order term remains part of the gain, even when its coefficient is small.

Apply Proposition A.4 with D replaced by $\mathcal{L}$. With $R_{\mathcal{L},3}(\delta) := \mathcal{L}(\bar{r}+\delta) - \mathcal{L}(\bar{r}) - \mathcal{L}'(\bar{r})\delta - \frac{1}{2}\mathcal{L}''(\bar{r})\delta^2$, the exact expansion is

$$\Delta_{\mathcal{L}}(r) = -\mathcal{L}'(\bar{r})\mathrm{Cov}(K, r) - \tfrac{1}{2}\mathcal{L}''(\bar{r})\mathbb{E}[K\delta^2] - \mathbb{E}[KR_{\mathcal{L},3}(\delta)],$$

which is (34). At exact stationarity $\mathcal{L}'(\bar{r}) = 0$, the first term vanishes. Applying the weighted remainder bound from Proposition A.4 yields Equation (35).

A.11 Conditional amplification of the curvature term

An optional construction is a base loss plus an auxiliary scalar cost:

$$\mathcal{L}(r) \;:=\; D(r) + \lambda\, S(r), \qquad \lambda \geq 0, \tag{36}$$

Here $S : I \to \mathbb{R}$ is an auxiliary cost. The weights multiply the entire loss $\mathcal{L}$ in $\mathbb{E}[K\mathcal{L}(r)]$. This construction does not enter the proof of uniform optimality, which only uses the common minimum.

Apply the exact Taylor expansion of Proposition A.4 with $\mathcal{L}$ in place of D:

$$\Delta_{\mathcal{L}}(r) = -\mathcal{L}'(\bar{r})\mathrm{Cov}(K, r) - \tfrac{1}{2}\,\mathcal{L}''(\bar{r})\,\mathbb{E}[K\delta^2] - \mathbb{E}[KR_{\mathcal{L},3}(\delta)]. \tag{37}$$

This is valid under the segment smoothness assumptions of Proposition A.5. At $\mathcal{L}'(\bar{r}) = 0$, the quadratic contribution is $-\frac{1}{2}\mathcal{L}''(\bar{r})\mathbb{E}[K\delta^2]$, with the weighted cubic remainder bounded as in Equation (35). When $\bar{r} = r^\star$ is the common global minimiser, exact nonimprovement follows separately from Theorem 4.10.

Finally, if D and S are C^2 near $\bar{r}$, the relation $\mathcal{L}(r) = D(r) + \lambda S(r)$ implies

$$\mathcal{L}''(\bar{r}) = D''(\bar{r}) + \lambda\, S''(\bar{r}). \tag{38}$$

When $\lambda > 0$ and the auxiliary cost has $S''(\bar{r}) > 0$, the weighted quadratic term has strictly larger curvature than the base loss for the same rate perturbation. This compares the quadratic curvature terms, not the complete gains, whose first derivatives and remainders can also differ.

A.12 Lean formal verification

All theoretical results in the paper, comprising the 11 named mathematical results and the finite patch-budget identity, are formally verified in Lean 4.19.0 with mathlib v4.19.0. The formalization covers scalar invariance and its coupling corollary, nonimprovement under independence, the exact improvement criterion, local improving allocations, the quadratic-surrogate maximum, the global gain bound, KKT conditions, uniform optimality and remaining absolute gain at a common loss minimum, and the Taylor expansions with cubic remainder bounds. The complete Lean project is provided as supplementary material. An axiom audit checked all 111 public theorems and supporting lemmas; their only axiom dependencies were `propext`, `Classical.choice`, and `Quot.sound`. No proof uses `sorry`, `admit`, or a custom axiom.

B Significance of the validation-selected comparison

Estimands and variability. For each method, dataset, and horizon, both candidates are fixed using three-seed validation results before five fresh paired runs are evaluated. Tables report sample means and standard deviations over those five runs. The MSE gain is $100(D - U)/D$, where D and U are arm means; it is not the mean of per-seed percentage changes. A dataset-level point averages each arm's four horizon means equally before computing the gain or time ratio. Method summaries average the eight resulting relative effects equally. Absolute MSE and MAE are never averaged across different datasets into a single benchmark score.

Setting-level paired tests. We test the five paired raw MSE differences within each setting by exhaustively enumerating all $2^5 = 32$ sign assignments to their mean. The two-sided p-value is

Table 2: **Dataset-level uncertainty and directional tests.** Each method contributes eight dataset-level MSE gains, defined as in Table 1. The median interval resamples the eight datasets 20,000 times. p is the raw value from a two-sided exact signed-rank test for each method. Wins count strictly lower uniform seed-mean MSE across 32 horizon settings and eight dataset averages; they are descriptive.

Method	Setting wins	Dataset wins	Median (%)	95% interval	p
EntroPE	18/32	6/8	+0.22	$[-0.67, +1.00]$	0.461
HDMixer	23/32	5/8	+0.71	$[-0.53, +2.02]$	0.195
TimeMosaic	12/32	2/8	-0.16	$[-0.66, +0.03]$	0.250

the fraction whose absolute mean is at least the observed absolute mean. Benjamini–Hochberg correction is applied jointly to all 96 tests. The smallest raw p is 0.0625 and the smallest adjusted q is 0.2727; none reaches 0.05. A percentile interval that excludes zero is not interpreted as exact-test significance.

Dataset-level analysis. Horizons from the same dataset are dependent, so the main directional check uses the eight dataset-level gains rather than treating 32 horizons as independent observations. We report a two-sided exact signed-rank test for each method. The test uses the symmetric, independent dataset-effect null. Related ETT benchmarks may remain dependent, and this suite is a fixed collection, so population-level interpretation remains limited. For a descriptive interval around the median dataset gain, we resample the eight complete dataset effects with replacement 20,000 times. This is the median of dataset-level gains, not the median of 32 horizon-level gains. The resulting summaries are reported in Table 2.

Interpretation. All three median intervals include zero and all method-wise raw p-values exceed 0.05. The observed means, win counts, and efficiency ratios are reported as descriptive results.

C Complete validation-selected accuracy and efficiency results

The validation-selected adaptive and uniform configurations for all eight datasets and four horizons are reported for EntroPE in Tables 3 and 4, for HDMixer in Tables 5 and 6, and for TimeMosaic in Tables 7 and 8, with five final runs per arm. Candidate identifiers map to the candidate grids in Section E. Selection is performed independently for every method–dataset–horizon–arm combination. No test-best candidate is chosen. Mean test MSE and MAE are accompanied by across-run sample standard deviations; boldface marks the lower unrounded mean, not significance. The accompanying efficiency table uses exactly the same configurations.

Table 3: **EntroPE: selected configurations and test accuracy.** Both arms are selected by mean validation MSE over three tuning seeds and evaluated on five fresh seeds. Entries are final-run mean ± sample standard deviation. c_D identifies the adaptive candidate in Section E; p_U is the selected uniform length. Bold indicates the smaller unrounded mean within each metric pair; $G = 100(D - U)/D$.

				Test MSE ↓		Test MAE ↓		
Dataset	H	c_D	p_U	Adaptive	Uniform	Adaptive	Uniform	G (%)
ETTh1	96	E2	24	0.3844 ± 0.0025	**0.3810** ± 0.0022	0.4008 ± 0.0009	**0.3989** ± 0.0010	+0.89
	192	E11	32	**0.4269** ± 0.0041	0.4321 ± 0.0068	**0.4257** ± 0.0012	0.4267 ± 0.0019	−1.23
	336	E8	32	0.4752 ± 0.0075	**0.4650** ± 0.0031	0.4482 ± 0.0032	**0.4438** ± 0.0013	+2.14
	720	E11	4	0.4890 ± 0.0106	**0.4796** ± 0.0153	0.4738 ± 0.0073	**0.4685** ± 0.0080	+1.92
ETTh2	96	E1	8	**0.3232** ± 0.0071	0.3233 ± 0.0079	0.3654 ± 0.0050	**0.3642** ± 0.0050	−0.04
	192	E6	8	0.4018 ± 0.0133	**0.3989** ± 0.0047	**0.4123** ± 0.0083	0.4136 ± 0.0044	+0.72
	336	E0	8	**0.4281** ± 0.0074	0.4323 ± 0.0062	**0.4417** ± 0.0052	0.4427 ± 0.0039	−0.99
	720	E4	32	0.4606 ± 0.0068	**0.4537** ± 0.0113	0.4668 ± 0.0070	**0.4616** ± 0.0074	+1.50
ETTm1	96	E10	24	**0.3282** ± 0.0056	0.3316 ± 0.0078	**0.3630** ± 0.0054	0.3637 ± 0.0049	−1.03
	192	E5	16	**0.3715** ± 0.0048	0.3716 ± 0.0052	0.3877 ± 0.0026	**0.3866** ± 0.0033	−0.02
	336	E9	16	**0.4020** ± 0.0050	0.4050 ± 0.0056	**0.4064** ± 0.0041	0.4099 ± 0.0021	−0.76
	720	E1	4	**0.4572** ± 0.0054	0.4611 ± 0.0036	**0.4417** ± 0.0052	0.4444 ± 0.0021	−0.85
ETTm2	96	E7	24	**0.1840** ± 0.0015	0.1885 ± 0.0034	**0.2682** ± 0.0014	0.2706 ± 0.0026	−2.41
	192	E1	4	**0.2531** ± 0.0029	0.2539 ± 0.0037	**0.3122** ± 0.0011	0.3137 ± 0.0025	−0.29
	336	E3	32	**0.3161** ± 0.0034	0.3320 ± 0.0053	**0.3530** ± 0.0029	0.3604 ± 0.0024	−5.02
	720	E6	4	0.4261 ± 0.0046	**0.4222** ± 0.0054	0.4184 ± 0.0018	**0.4164** ± 0.0047	+0.92
Electricity	96	E11	4	0.1679 ± 0.0015	**0.1661** ± 0.0022	0.2638 ± 0.0031	**0.2608** ± 0.0046	+1.06
	192	E8	16	0.1786 ± 0.0015	**0.1770** ± 0.0010	0.2735 ± 0.0031	**0.2715** ± 0.0013	+0.88
	336	E10	16	0.1959 ± 0.0017	**0.1951** ± 0.0022	0.2906 ± 0.0019	**0.2894** ± 0.0032	+0.44
	720	E8	16	**0.2337** ± 0.0007	0.2367 ± 0.0035	**0.3205** ± 0.0016	0.3238 ± 0.0044	−1.30
Exchange	96	E0	4	0.0901 ± 0.0030	**0.0899** ± 0.0028	0.2107 ± 0.0034	**0.2106** ± 0.0028	+0.20
	192	E11	8	**0.1896** ± 0.0059	0.1913 ± 0.0042	**0.3104** ± 0.0048	0.3119 ± 0.0037	−0.89
	336	E8	32	0.3721 ± 0.0129	**0.3549** ± 0.0083	0.4405 ± 0.0083	**0.4303** ± 0.0046	+4.63
	720	E10	8	**0.9526** ± 0.0614	0.9638 ± 0.0621	**0.7361** ± 0.0235	0.7400 ± 0.0253	−1.18
Traffic	96	E11	4	0.5364 ± 0.0056	**0.5162** ± 0.0066	0.3214 ± 0.0034	**0.3121** ± 0.0046	+3.78
	192	E9	4	0.5302 ± 0.0124	**0.5263** ± 0.0129	0.3220 ± 0.0033	**0.3141** ± 0.0031	+0.73
	336	E10	8	0.5448 ± 0.0110	**0.5409** ± 0.0044	0.3330 ± 0.0041	**0.3228** ± 0.0057	+0.71
	720	E11	4	0.5904 ± 0.0065	**0.5777** ± 0.0093	0.3455 ± 0.0049	**0.3342** ± 0.0028	+2.16
Weather	96	E3	4	0.1707 ± 0.0012	**0.1697** ± 0.0009	0.2174 ± 0.0014	**0.2155** ± 0.0006	+0.63
	192	E1	4	**0.2153** ± 0.0009	0.2166 ± 0.0011	**0.2562** ± 0.0007	0.2570 ± 0.0012	−0.59
	336	E8	16	0.2708 ± 0.0017	**0.2702** ± 0.0009	**0.2953** ± 0.0013	0.2958 ± 0.0007	+0.19
	720	E3	32	0.3433 ± 0.0014	**0.3429** ± 0.0005	**0.3435** ± 0.0011	0.3440 ± 0.0005	+0.13

Table 4: **EntroPE: training and inference costs for the same selected configurations.** Each value averages five runs. Training is measured from steady-state timed epochs after optimizer warm-up; inference is median steady-state full-batch latency. $S = D/U$; values above one favor uniform.

		Training (s/epoch) ↓			Inference (ms) ↓		
Dataset	H	Adaptive	Uniform	S_{tr}	Adaptive	Uniform	S_{inf}
ETTh1	96	2.01	1.42	1.42	7.04	3.44	2.05
	192	1.75	1.39	1.26	5.87	3.48	1.69
	336	1.82	1.38	1.32	6.29	3.48	1.81
	720	1.68	1.42	1.18	5.86	3.59	1.63
ETTh2	96	2.25	1.58	1.42	7.63	3.77	2.02
	192	2.06	1.54	1.33	6.91	3.76	1.84
	336	2.22	1.52	1.46	7.68	3.77	2.04
	720	2.05	1.41	1.45	7.32	3.71	1.97
ETTm1	96	10.53	7.88	1.34	4.66	2.49	1.87
	192	10.80	7.92	1.36	4.99	2.51	1.99
	336	10.46	7.85	1.33	4.74	2.51	1.89
	720	11.24	8.09	1.39	5.22	2.60	2.01
ETTm2	96	11.11	8.08	1.37	5.04	2.59	1.94
	192	11.73	8.44	1.39	5.39	2.69	2.00
	336	11.71	8.11	1.44	5.31	2.63	2.02
	720	11.21	8.49	1.32	5.15	2.72	1.89
Electricity	96	158.02	141.11	1.12	102.82	84.66	1.21
	192	170.56	123.48	1.38	115.83	74.11	1.56
	336	159.07	123.46	1.29	104.72	74.33	1.41
	720	169.60	123.44	1.37	116.64	74.91	1.56
Exchange	96	1.93	1.45	1.33	4.80	2.06	2.33
	192	1.84	1.40	1.31	4.28	2.07	2.06
	336	1.74	1.34	1.29	4.43	2.04	2.17
	720	1.63	1.30	1.26	4.30	2.06	2.08
Traffic	96	110.41	61.25	1.80	30.51	13.43	2.27
	192	114.28	61.54	1.86	32.59	13.45	2.42
	336	111.62	57.59	1.94	31.17	12.40	2.51
	720	110.44	63.60	1.74	30.61	13.65	2.24
Weather	96	22.09	16.10	1.37	32.95	18.21	1.81
	192	22.43	16.06	1.40	33.92	18.24	1.86
	336	19.18	14.17	1.35	27.56	16.06	1.72
	720	22.26	14.09	1.58	33.18	15.77	2.10

Table 5: **HDMixer: selected configurations and test accuracy.** Both arms are selected by mean validation MSE over three tuning seeds and evaluated on five fresh seeds. Entries are final-run mean ± sample standard deviation. c_D identifies the adaptive candidate in Section E; p_U is the selected uniform length. Bold indicates the smaller unrounded mean within each metric pair; $G = 100(D-U)/D$.

				Test MSE ↓		Test MAE ↓		
Dataset	H	c_D	p_U	Adaptive	Uniform	Adaptive	Uniform	G (%)
ETTh1	96	H1	4	0.3688 ± 0.0007	**0.3646** ± 0.0002	0.3946 ± 0.0005	**0.3917** ± 0.0005	+1.14
	192	H2	4	0.4088 ± 0.0013	**0.4023** ± 0.0005	0.4159 ± 0.0009	**0.4118** ± 0.0008	+1.60
	336	H3	32	0.3971 ± 0.0017	**0.3892** ± 0.0015	0.4197 ± 0.0009	**0.4123** ± 0.0024	+1.98
	720	H0	4	0.4577 ± 0.0086	**0.4442** ± 0.0019	0.4612 ± 0.0049	**0.4564** ± 0.0012	+2.95
ETTh2	96	H0	8	0.2651 ± 0.0020	**0.2625** ± 0.0007	0.3314 ± 0.0019	**0.3286** ± 0.0009	+0.95
	192	H0	8	0.3208 ± 0.0026	**0.3179** ± 0.0018	0.3723 ± 0.0031	**0.3686** ± 0.0025	+0.93
	336	H0	8	0.3109 ± 0.0041	**0.3104** ± 0.0036	0.3724 ± 0.0033	**0.3709** ± 0.0028	+0.16
	720	H0	8	0.4156 ± 0.0196	**0.3952** ± 0.0137	0.4434 ± 0.0124	**0.4308** ± 0.0091	+4.92
ETTm1	96	H3	16	**0.2985** ± 0.0020	0.3020 ± 0.0010	0.3501 ± 0.0012	**0.3464** ± 0.0009	−1.17
	192	H15	4	**0.3360** ± 0.0011	0.3387 ± 0.0020	0.3715 ± 0.0010	**0.3677** ± 0.0016	−0.81
	336	H2	4	**0.3714** ± 0.0023	0.3725 ± 0.0021	0.3926 ± 0.0030	**0.3877** ± 0.0017	−0.29
	720	H0	4	**0.4305** ± 0.0016	0.4308 ± 0.0022	0.4215 ± 0.0016	**0.4195** ± 0.0018	−0.08
ETTm2	96	H0	4	0.1658 ± 0.0015	**0.1650** ± 0.0008	0.2551 ± 0.0015	**0.2540** ± 0.0005	+0.45
	192	H12	8	0.2214 ± 0.0010	**0.2201** ± 0.0010	0.2929 ± 0.0010	**0.2915** ± 0.0009	+0.63
	336	H12	8	**0.2774** ± 0.0024	0.2777 ± 0.0009	0.3298 ± 0.0016	**0.3292** ± 0.0014	−0.08
	720	H1	4	**0.3764** ± 0.0203	0.3840 ± 0.0322	**0.3895** ± 0.0090	0.3926 ± 0.0138	−2.00
Electricity	96	H4	16	0.1619 ± 0.0104	**0.1487** ± 0.0010	0.2605 ± 0.0124	**0.2503** ± 0.0013	+8.15
	192	H0	16	0.1798 ± 0.0143	**0.1667** ± 0.0014	0.2700 ± 0.0080	**0.2660** ± 0.0029	+7.29
	336	H0	8	0.1839 ± 0.0051	**0.1680** ± 0.0026	0.2822 ± 0.0038	**0.2662** ± 0.0028	+8.65
	720	H4	16	0.2404 ± 0.0524	**0.2040** ± 0.0009	0.3259 ± 0.0374	**0.2979** ± 0.0012	+15.15
Exchange	96	H13	32	0.0871 ± 0.0011	**0.0845** ± 0.0011	0.2063 ± 0.0012	**0.2009** ± 0.0013	+2.96
	192	H5	32	0.1854 ± 0.0040	**0.1809** ± 0.0037	0.3057 ± 0.0039	**0.2998** ± 0.0031	+2.44
	336	H4	32	0.3520 ± 0.0111	**0.3443** ± 0.0051	0.4303 ± 0.0084	**0.4220** ± 0.0039	+2.21
	720	H9	24	1.1067 ± 0.0149	**1.1053** ± 0.0104	0.7955 ± 0.0052	**0.7951** ± 0.0040	+0.13
Traffic	96	H9	24	**0.4057** ± 0.0018	0.4093 ± 0.0024	**0.2821** ± 0.0017	0.2856 ± 0.0048	−0.90
	192	H10	24	0.4254 ± 0.0032	**0.4220** ± 0.0002	0.3012 ± 0.0069	**0.2917** ± 0.0014	+0.80
	336	H9	32	0.4480 ± 0.0052	**0.4422** ± 0.0055	0.3230 ± 0.0064	**0.3140** ± 0.0081	+1.29
	720	H9	32	**0.4661** ± 0.0035	0.4738 ± 0.0037	**0.3319** ± 0.0059	0.3373 ± 0.0042	−1.67
Weather	96	H8	32	0.1545 ± 0.0025	**0.1515** ± 0.0014	0.2106 ± 0.0028	**0.2072** ± 0.0029	+1.94
	192	H8	24	0.2004 ± 0.0008	**0.1991** ± 0.0018	0.2518 ± 0.0016	**0.2500** ± 0.0014	+0.64
	336	H8	24	**0.2544** ± 0.0032	0.2562 ± 0.0071	**0.2927** ± 0.0021	0.2928 ± 0.0072	−0.74
	720	H10	24	0.3336 ± 0.0051	**0.3315** ± 0.0155	0.3463 ± 0.0027	**0.3433** ± 0.0077	+0.65

Table 6: **HDMixer: training and inference costs for the same selected configurations.** Each value averages five runs. Training is measured from steady-state timed epochs after optimizer warm-up; inference is median steady-state full-batch latency. $S = D/U$; values above one favor uniform.

		Training (s/epoch) ↓			Inference (ms) ↓		
Dataset	H	Adaptive	Uniform	S_{tr}	Adaptive	Uniform	S_{inf}
ETTh1	96	1.78	0.30	6.02	11.90	2.59	4.59
	192	1.75	0.30	5.82	11.87	2.59	4.58
	336	0.95	0.21	4.56	15.20	1.96	7.75
	720	1.77	0.61	2.92	27.96	10.15	2.75
ETTh2	96	1.79	0.22	8.17	11.81	1.45	8.16
	192	1.61	0.18	8.82	23.52	2.73	8.61
	336	1.63	0.20	7.96	23.50	2.75	8.54
	720	1.54	0.26	6.01	23.40	2.80	8.35
ETTm1	96	3.75	0.68	5.49	6.59	0.88	7.50
	192	3.74	1.12	3.35	6.58	2.59	2.54
	336	7.38	1.15	6.40	11.95	2.62	4.55
	720	7.40	1.27	5.85	11.98	2.65	4.52
ETTm2	96	7.37	1.11	6.63	11.92	2.59	4.60
	192	3.75	0.81	4.63	6.58	1.45	4.54
	336	3.76	0.90	4.19	6.60	1.47	4.49
	720	7.42	1.26	5.87	11.98	2.65	4.52
Electricity	96	119.37	9.12	13.10	35.56	3.82	9.30
	192	197.64	9.39	21.05	67.36	3.84	17.54
	336	197.27	16.41	12.02	66.42	6.11	10.87
	720	117.79	15.11	7.79	35.89	3.97	9.04
Exchange	96	1.07	0.56	1.91	1.36	0.44	3.07
	192	1.04	0.55	1.90	1.36	0.44	3.12
	336	1.02	0.55	1.86	1.35	0.43	3.12
	720	0.75	0.52	1.45	1.03	0.42	2.47
Traffic	96	95.80	14.72	6.51	13.70	2.55	5.38
	192	129.09	24.50	5.27	18.47	3.53	5.23
	336	260.55	32.52	8.01	32.84	6.30	5.21
	720	185.95	30.46	6.10	28.17	5.38	5.24
Weather	96	9.79	4.15	2.36	1.98	0.44	4.47
	192	9.85	4.28	2.31	1.98	0.47	4.21
	336	10.14	4.43	2.29	2.00	0.47	4.22
	720	8.70	5.03	1.73	1.78	0.48	3.71

Table 7: **TimeMosaic: selected configurations and test accuracy.** Both arms are selected by mean validation MSE over three tuning seeds and evaluated on five fresh seeds. Entries are final-run mean ± sample standard deviation. c_D identifies the adaptive candidate in Section E; p_U is the selected uniform length. Bold indicates the smaller unrounded mean within each metric pair; $G = 100(D - U)/D$.

				Test MSE ↓		Test MAE ↓		
Dataset	H	c_D	p_U	Adaptive	Uniform	Adaptive	Uniform	G (%)
ETTh1	96	T8	32	**0.3611** ± 0.0061	0.3639 ± 0.0067	**0.3869** ± 0.0047	0.3886 ± 0.0036	−0.79
	192	T6	16	0.4271 ± 0.0150	**0.4238** ± 0.0184	0.4307 ± 0.0121	**0.4269** ± 0.0146	+0.78
	336	T8	8	**0.4494** ± 0.0181	0.4498 ± 0.0225	0.4451 ± 0.0106	**0.4446** ± 0.0150	−0.09
	720	T8	8	**0.4467** ± 0.0043	0.4578 ± 0.0094	**0.4600** ± 0.0028	0.4693 ± 0.0047	−2.50
ETTh2	96	T4	32	**0.2947** ± 0.0054	0.2986 ± 0.0051	0.3464 ± 0.0048	**0.3444** ± 0.0037	−1.35
	192	T5	16	**0.3451** ± 0.0089	0.3507 ± 0.0037	**0.3792** ± 0.0040	0.3831 ± 0.0026	−1.61
	336	T8	32	**0.3809** ± 0.0064	0.3817 ± 0.0113	0.4058 ± 0.0034	**0.4046** ± 0.0043	−0.22
	720	T8	16	0.4217 ± 0.0126	**0.4133** ± 0.0141	0.4399 ± 0.0077	**0.4315** ± 0.0061	+1.99
ETTm1	96	T2	16	**0.2880** ± 0.0015	0.2913 ± 0.0017	**0.3310** ± 0.0017	0.3319 ± 0.0006	−1.15
	192	T5	16	**0.3327** ± 0.0032	0.3336 ± 0.0029	**0.3593** ± 0.0016	0.3600 ± 0.0010	−0.27
	336	T8	8	0.3662 ± 0.0022	**0.3656** ± 0.0048	**0.3795** ± 0.0006	0.3800 ± 0.0009	+0.18
	720	T6	8	**0.4340** ± 0.0120	0.4367 ± 0.0077	**0.4195** ± 0.0031	0.4221 ± 0.0021	−0.62
ETTm2	96	T4	8	**0.1702** ± 0.0039	0.1731 ± 0.0050	**0.2533** ± 0.0036	0.2550 ± 0.0034	−1.66
	192	T5	16	**0.2295** ± 0.0056	0.2320 ± 0.0053	**0.2930** ± 0.0046	0.2940 ± 0.0023	−1.07
	336	T5	16	**0.2732** ± 0.0025	0.2752 ± 0.0042	**0.3219** ± 0.0014	0.3251 ± 0.0039	−0.76
	720	T5	16	**0.3677** ± 0.0100	0.3698 ± 0.0048	**0.3817** ± 0.0056	0.3823 ± 0.0030	−0.57
Electricity	96	T0	32	**0.1645** ± 0.0005	0.1654 ± 0.0006	**0.2507** ± 0.0007	0.2507 ± 0.0008	−0.55
	192	T1	8	**0.1770** ± 0.0002	0.1779 ± 0.0013	0.2619 ± 0.0005	**0.2613** ± 0.0015	−0.51
	336	T3	8	0.1935 ± 0.0012	**0.1928** ± 0.0010	0.2784 ± 0.0014	**0.2769** ± 0.0013	+0.35
	720	T1	8	0.2234 ± 0.0014	**0.2231** ± 0.0008	0.3035 ± 0.0015	**0.3023** ± 0.0009	+0.11
Exchange	96	T2	32	**0.0892** ± 0.0032	0.0930 ± 0.0053	**0.2100** ± 0.0036	0.2148 ± 0.0054	−4.21
	192	T7	16	0.2139 ± 0.0178	**0.2007** ± 0.0207	0.3325 ± 0.0121	**0.3223** ± 0.0162	+6.18
	336	T5	8	0.4021 ± 0.0474	**0.3637** ± 0.0190	0.4611 ± 0.0278	**0.4410** ± 0.0120	+9.53
	720	T0	16	**0.9518** ± 0.1153	0.9732 ± 0.1340	**0.7210** ± 0.0369	0.7328 ± 0.0491	−2.25
Traffic	96	T0	32	0.4240 ± 0.0029	**0.4216** ± 0.0009	**0.2657** ± 0.0028	0.2675 ± 0.0019	+0.56
	192	T1	32	0.4431 ± 0.0007	**0.4429** ± 0.0004	**0.2733** ± 0.0031	0.2768 ± 0.0012	+0.04
	336	T0	32	0.4618 ± 0.0018	**0.4614** ± 0.0006	**0.2797** ± 0.0024	0.2817 ± 0.0005	+0.08
	720	T7	32	**0.4905** ± 0.0011	0.4929 ± 0.0014	**0.2968** ± 0.0018	0.2987 ± 0.0004	−0.48
Weather	96	T7	32	0.1553 ± 0.0010	**0.1547** ± 0.0015	0.1964 ± 0.0017	**0.1957** ± 0.0027	+0.38
	192	T5	8	0.2016 ± 0.0014	**0.2008** ± 0.0013	0.2390 ± 0.0009	**0.2374** ± 0.0010	+0.36
	336	T0	16	**0.2508** ± 0.0012	0.2514 ± 0.0011	0.2775 ± 0.0005	**0.2772** ± 0.0016	−0.21
	720	T3	8	**0.3233** ± 0.0052	0.3257 ± 0.0056	**0.3278** ± 0.0041	0.3282 ± 0.0032	−0.74

Table 8: **TimeMosaic: training and inference costs for the same selected configurations.** Each value averages five runs. Training is measured from steady-state timed epochs after optimizer warm-up; inference is median steady-state full-batch latency. $S = D/U$; values above one favor uniform.

		Training (s/epoch) ↓			Inference (ms) ↓		
Dataset	H	Adaptive	Uniform	S_{tr}	Adaptive	Uniform	S_{inf}
ETTh1	96	10.92	10.07	1.08	12.37	11.20	1.10
	192	15.15	14.29	1.06	16.63	15.42	1.08
	336	8.10	7.27	1.11	9.57	8.33	1.15
	720	14.58	13.79	1.06	17.62	16.37	1.08
ETTh2	96	11.19	10.41	1.08	12.33	11.14	1.11
	192	6.29	5.45	1.16	7.34	6.15	1.19
	336	4.63	3.84	1.21	5.79	4.61	1.26
	720	14.97	14.17	1.06	18.11	16.92	1.07
ETTm1	96	25.58	22.27	1.15	7.39	6.20	1.19
	192	44.08	40.65	1.08	11.85	10.65	1.11
	336	18.82	15.33	1.23	5.61	4.34	1.29
	720	25.98	22.59	1.15	7.70	6.44	1.20
ETTm2	96	47.47	43.89	1.08	12.86	11.61	1.11
	192	46.33	42.95	1.08	12.32	11.12	1.11
	336	32.44	29.09	1.12	9.15	7.95	1.15
	720	27.26	23.99	1.14	7.99	6.80	1.17
Electricity	96	509.51	506.75	1.01	75.56	74.64	1.01
	192	507.55	506.87	1.00	75.33	75.94	0.99
	336	338.71	337.36	1.00	50.95	51.03	1.00
	720	496.38	495.86	1.00	76.07	76.70	0.99
Exchange	96	10.80	10.33	1.05	22.14	20.99	1.06
	192	9.55	9.09	1.05	16.44	15.29	1.07
	336	2.77	2.31	1.20	5.68	4.46	1.27
	720	6.20	5.80	1.07	12.67	11.54	1.10
Traffic	96	769.40	662.94	1.16	61.06	59.43	1.03
	192	477.26	471.18	1.01	41.24	40.47	1.02
	336	320.88	315.02	1.02	28.27	27.48	1.03
	720	462.55	457.14	1.01	41.81	41.03	1.02
Weather	96	242.02	237.96	1.02	57.03	55.82	1.02
	192	249.42	245.62	1.02	60.50	59.29	1.02
	336	116.74	112.72	1.04	28.99	27.74	1.05
	720	171.07	167.22	1.02	42.59	41.36	1.03

D MODEL ADAPTATIONS FOR THE UNIFORM-PATCH ABLATIONS

The controlled study requires a uniform-patch counterpart for each dynamic method. In all cases, the design principle is the same: modify the patch generation mechanism while leaving the downstream forecasting backbone, prediction head, and training budget as close as possible to the original model. The model implementations used for these ablations are publicly released open-source research codebases. Our ablation implementations for EntroPE, HDMixer, and TimeMosaic are provided as supplementary material.

EntroPE. The EntroPE ablation uses the model's pre-existing static patching mode with a fixed patch length. Relative to the original method, the only conceptual change is that entropy-based boundary selection is disabled and all segments are forced to have equal length. The rest of the forecasting pipeline remains unchanged. This comparison uses the original implementation's entropy-guided segmentation and fixed-length alternative within the same architecture.

TimeMosaic. TimeMosaic ordinarily learns, for each temporal region, which patch length to use from a small candidate set. The uniform variant removes this routing decision and applies a single fixed patch length across all regions. The fixed-length variant reuses the same patch-embedding operator associated with that patch size in the original multi-scale embedding bank rather than introducing a new projection module. The downstream encoder and prediction head are unchanged, while the auxiliary routing regularization is disabled because no patch-size classification is performed. The uniform implementation repeat-pads its patch tokens to the adaptive materialized token width, preserving the forecast-head input width. This keeps that tensor dimension fixed while the token representations retain the content of their respective patch lengths.

HDMixer. HDMixer ordinarily uses deformable patch extraction, in which the model predicts patch offsets and widths and applies an auxiliary patch-entropy regularizer. The uniform variant replaces this deformable extraction stage with standard non-overlapping fixed-length patches. Under this ablation, the deformable localization mechanism and the associated auxiliary regularizer are inactive. The mixer architecture is retained, and its native flattened forecast-head width varies with the number of patches, allowing parameter count and representation capacity to vary with the selected patch length. The number of patches is recomputed from the chosen fixed length so that the downstream forecast module receives a tensorization matched to the selected granularity.

Common interpretation. The ablations retain each method's architectural family and shared training settings while replacing adaptive patch selection and its associated losses. Downstream tensors follow each method's patch-count convention, so token counts and parameter counts can differ between variants. The comparison evaluates the accuracy and execution cost of the resulting adaptive and tuned uniform systems at their respective resource requirements.

E EXPERIMENTAL HYPERPARAMETERS AND HARDWARE

Validation-only selection and independent evaluation. For each method, arm, dataset, and forecast horizon, every candidate is trained with three independent tuning seeds. We rank candidates by their mean best validation MSE; the predeclared candidate order breaks exact ties. The selector has no access to test metrics. The selected adaptive and uniform configurations are then trained with freshly initialized forecasting networks using five new seeds, paired across the two arms. EntroPE retains its frozen pretrained entropy estimator; that auxiliary model is not retrained in these runs. Test MSE and MAE average those five final runs. This is an expanded adaptive search: the adaptive arm receives more candidate configurations than the uniform arm, as summarized in Table 9. Across eight datasets and four horizons, the reported comparison uses 4,800 tuning runs and 960 selected final runs.

Table 9: **Search budget and final evaluation.** Each candidate has three tuning runs; each selected arm has five final runs.

Method	Adaptive	Uniform	Tuning runs	Final runs
EntroPE	12	5	1,632	320
HDMixer	16	5	2,016	320
TimeMosaic	9	3	1,152	320

Search spaces. For EntroPE, q is the empirical entropy quantile used to set the boundary threshold, and $L_{\max}$ is the maximum patch length. The adaptive grid is $q \in \{0.55, 0.75, 0.95\}$ crossed with $L_{\max} \in \{8, 16, 24, 32\}$; consecutive boundaries are disallowed. The uniform grid is $p \in \{4, 8, 16, 24, 32\}$. TimeMosaic varies router temperature $\tau \in \{0.25, 0.5, 1\}$ and balance-loss weight $\lambda_{\text{bal}} \in \{10^{-4}, 10^{-3}, 10^{-2}\}$, keeping the patch bank $\{8, 16, 32\}$ fixed. Its uniform grid is $p \in \{8, 16, 32\}$. Uniform candidate Pp denotes fixed patch length p. The complete adaptive candidate grids are listed in Table 10.

For HDMixer, the first twelve candidates cross patch length $p \in \{8, 16, 24, 32\}$ with deformation range $d \in \{0.125, 0.25, 0.5\}$, using stride $s = p/2$, length-regularization weight $\lambda = 0.1$, and radius parameter $r = 0.01$. Four additional candidates vary λ or r around the $p/s = 16/8$, $d = 0.25$

Table 10: **EntroPE and TimeMosaic adaptive candidates.** IDs are used in the selected-result tables.

EntroPE	q	$L_{\max}$	**TimeMosaic**	τ	λ_{bal}
E0	0.55	8	T0	0.25	10^{-4}
E1	0.55	16	T1	0.25	10^{-3}
E2	0.55	24	T2	0.25	10^{-2}
E3	0.55	32	T3	0.5	10^{-4}
E4	0.75	8	T4	0.5	10^{-3}
E5	0.75	16	T5	0.5	10^{-2}
E6	0.75	24	T6	1	10^{-4}
E7	0.75	32	T7	1	10^{-3}
E8	0.95	8	T8	1	10^{-2}
E9	0.95	16			
E10	0.95	24			
E11	0.95	32			

configuration. The uniform grid is $p \in \{4, 8, 16, 24, 32\}$; the complete adaptive grid is listed in Table 11.

Table 11: **Complete HDMixer adaptive search.** All sixteen candidates are evaluated independently for every dataset and horizon.

ID	p	s	d	λ	r
H0	8	4	0.125	0.1	0.01
H1	8	4	0.25	0.1	0.01
H2	8	4	0.5	0.1	0.01
H3	16	8	0.125	0.1	0.01
H4	16	8	0.25	0.1	0.01
H5	16	8	0.5	0.1	0.01
H6	24	12	0.125	0.1	0.01
H7	24	12	0.25	0.1	0.01
H8	24	12	0.5	0.1	0.01
H9	32	16	0.125	0.1	0.01
H10	32	16	0.25	0.1	0.01
H11	32	16	0.5	0.1	0.01
H12	16	8	0.25	0.01	0.01
H13	16	8	0.25	1	0.01
H14	16	8	0.25	0.1	0.001
H15	16	8	0.25	0.1	0.1

Fixed architecture and optimization settings. Only the patch/router parameters listed above are searched. Non-patching settings, preprocessing, optimizer, epoch cap, and stopping rule remain fixed within each method–dataset–horizon comparison. All three methods use Adam and validation-based early stopping, restoring the checkpoint with the lowest validation MSE. Forecaster training disables automatic mixed precision. TimeMosaic's uniform counterpart preserves the adaptive model's materialized token/head width. HDMixer retains its native geometry, so patch length can change token count, prediction-head width, parameter count, and compute.

The effective fixed settings are given in Tables 12 to 15. T is lookback length, L is encoder depth, h is attention-head count, and B is batch size. The training caps are 20 epochs for EntroPE, 50 for HDMixer, and 10 for TimeMosaic; these are caps, not the number actually completed by every run.

Table 12: **EntroPE fixed configurations.** Shared by all four horizons and both arms; $T = 96$, patience $= 10$, and learning-rate schedule `TST`.

Dataset	d_{model}	d_{ff}	h	L	B	**LR**	**Dropout**
ETTh1	8	2048	2	1	64	10^{-3}	0.05
ETTh2	8	2048	2	2	64	10^{-2}	0.1
ETTm1	16	2048	2	1	32	10^{-2}	0.1
ETTm2	16	2048	4	1	32	10^{-3}	0.1
Electricity	32	2048	4	2	32	10^{-2}	0.1
Traffic	8	256	2	3	8	10^{-2}	0.1
Weather	16	2048	2	2	128	10^{-2}	0.2
Exchange	8	256	2	3	32	10^{-2}	0.1

`fc_dropout` and `head_dropout` are both 0.1. Traffic and Exchange use $d_{\text{ff}} = 256$; the other six datasets use $d_{\text{ff}} = 2{,}048$.

Table 13: **HDMixer fixed configurations.** Both arms use $T = 336$, $d_{\text{model}} = 16$, $d_{\text{ff}} = 32$, $h = 4$, and learning-rate schedule `type3`.

Dataset	H	L	B	**LR**	**Drop.**	**FC drop.**	**Patience**
ETTh1	96,192	1	256	5×10^{-4}	0.8	0.3	20
ETTh1	336,720	2	512	5×10^{-3}	0.8	0.3	20
ETTh2	96	1	256	5×10^{-4}	0.8	0.3	10
ETTh2	192,720	1	512	5×10^{-4}	0.8	0.3	10
ETTh2	336	1	512	10^{-3}	0.8	0.3	10
ETTm1	All	1	256	10^{-3}	0.8	0.3	20
ETTm2	All	1	256	5×10^{-4}	0.8	0.3	20
Electricity	All	2	32	10^{-3}	0.8	0.3	20
Traffic	All	2	8	10^{-3}	0.8	0.3	20
Weather	All	2	32	5×10^{-4}	0.3	0.1	20
Exchange	All	2	32	5×10^{-4}	0.8	0.3	20

`head_dropout` is 0 throughout. All denotes all four horizons.

Table 14: **TimeMosaic fixed configurations: ETT datasets.** Both arms use $d_{\text{model}} = 512$, LR $= 10^{-4}$, patience $= 3$, and learning-rate schedule `type1`.

Dataset	H	T	d_{ff}	L	h	B	Channel
ETTh1	96	320	2048	2	8	32	CI
ETTh1	192	320	2048	3	1	32	CI
ETTh1	336	320	2048	2	16	32	CDA
ETTh1	720	320	2048	3	2	32	CD
ETTh2	96	320	2048	2	1	32	CDA
ETTh2	192	320	2048	1	1	32	CDA
ETTh2	336	320	2048	1	2	32	CDA
ETTh2	720	320	2048	3	16	32	CI
ETTm1	96	320	2048	1	8	32	CI
ETTm1	192	320	2048	2	1	32	CI
ETTm1	336	320	2048	1	3	32	CI
ETTm1	720	320	2048	1	4	32	CD
ETTm2	96	320	2048	2	2	32	CDA
ETTm2	192	320	2048	2	1	32	CDA
ETTm2	336	320	2048	2	16	32	CI
ETTm2	720	320	2048	1	2	32	CDA

Dropout and `fc_dropout` are both 0.1. Channel labels preserve the executed codebase modes.

Table 15: **TimeMosaic fixed configurations: remaining datasets.** Both arms use $d_{\text{model}} = 512$, LR $= 10^{-4}$, patience $= 3$, and learning-rate schedule `type1`.

Dataset	H	T	d_{ff}	L	h	B	Channel
Electricity	96	96	2048	2	8	16	CDP
Electricity	192	96	2048	2	8	16	CDP
Electricity	336	96	2048	2	8	16	CI+
Electricity	720	96	2048	2	8	16	CDP
Traffic	96	96	512	2	8	4	CDP
Traffic	192	96	512	2	8	4	CDP
Traffic	336	96	512	2	8	4	CDP
Traffic	720	96	512	2	8	4	CDP
Weather	96	320	2048	3	1	32	CDA
Weather	192	320	2048	3	4	32	CDA
Weather	336	320	2048	2	8	32	CDA
Weather	720	320	2048	2	8	32	CDA
Exchange	96	320	2048	3	16	32	CI
Exchange	192	320	2048	3	1	32	CDA
Exchange	336	320	2048	1	4	32	CDA
Exchange	720	320	2048	2	2	32	CDA

Dropout and `fc_dropout` are both 0.1. Channel labels preserve the executed codebase modes.

Training and inference measurements. All selected checkpoints are profiled on NVIDIA H200 GPUs with paired arms on the same batch size. Training cost is measured from complete timed optimization epochs after warm-up and reported in seconds per epoch and per step. It excludes validation and checkpoint management and does not measure time to convergence. Inference is steady-state full-batch forward latency with inputs resident on the GPU and synchronized CUDA-event timing. The reported per-run value is the median over repeated measurements.

Native adaptive timing includes route construction and, for EntroPE, the frozen entropy estimator. To estimate routing overhead, we first run the adaptive model and save the patch tokens it produces. We then time a second run with the same checkpoint and inputs, bypassing adaptive token generation and feeding the saved tokens to the forecasting backbone. Routing overhead is the normal adaptive

run time minus the time of this second run, expressed as a percentage of the normal adaptive run time. Dataset-level figures average the four horizon means equally. Speedup is adaptive cost divided by uniform cost, so values above one favor uniform patching.

F BENCHMARK DATASET NOTES

The main experiments use eight standard multivariate long-horizon forecasting benchmarks. The suite combines the ETT variants and Weather from the Informer benchmark (Zhou et al., 2021) with Electricity, Exchange, and Traffic from the LSTNet benchmark collection (Lai et al., 2018). Across HDMixer, EntroPE, and TimeMosaic, the ETT loaders use the first 12, next 4, and following 4 months for training, validation, and testing, respectively, with 30-day months. The custom-data loaders use chronological 70%/10%/20% splits for Weather, Electricity, Traffic, and Exchange. Standardization is fit on the training segment only.

Dataset descriptions. **Electricity** records hourly electricity consumption for 321 clients and is used for multivariate load forecasting. **ETTh1**, **ETTh2**, **ETTm1**, and **ETTm2** track electricity transformer temperature and load, with the 'h' variants sampled hourly and the 'm' variants sampled every 15 minutes. **Exchange** contains daily exchange rates for multiple currencies relative to the U.S. dollar. **Traffic** records hourly road occupancy measurements from a freeway sensor network. **Weather** contains 10-minute atmospheric measurements from a weather station.

Table 16 reports raw time observations assigned to each split, before constructing forecast windows. Given lookback T and horizon H, the training window count is $n_{\text{train}} - T - H + 1$; validation and test use the preceding T observations as context and have $n_{\text{split}} - H + 1$ windows. Thus window counts depend on the method and horizon. The native EntroPE and HDMixer loaders discard incomplete final batches; TimeMosaic keeps them. These loader choices are identical between the adaptive and uniform arms of each method.

Table 16: **Benchmark dimensions and chronological splits.** Counts are raw observations per split, not forecast windows. ETT uses the fixed 20-month benchmark interval.

Dataset	Dim.	Train	Validation	Test	Frequency
ETTh1	7	8,640	2,880	2,880	Hourly
ETTh2	7	8,640	2,880	2,880	Hourly
ETTm1	7	34,560	11,520	11,520	15 min
ETTm2	7	34,560	11,520	11,520	15 min
Electricity	321	18,412	2,632	5,260	Hourly
Traffic	862	12,280	1,756	3,508	Hourly
Weather	21	36,887	5,270	10,539	10 min
Exchange	8	5,311	760	1,517	Daily

Hardware. All experiments were conducted on NVIDIA H200 GPUs with 141 GB of VRAM each.