PRICING STRUCTURAL RESTRICTIONS IN TIME-SERIES FORECASTING

Anonymous authors
Paper under double-blind review

ABSTRACT

The same architectural choice can improve some time-series predictions and worsen others. We examine this variation by comparing the predictive cost of structural restrictions with the learning cost they can save. Here, learning cost is the prediction loss above a structure's best attainable performance due to imperfect estimation and optimization with limited data and resources. In linear prediction settings, we express structural restriction costs through properties of the data distribution, termed data requirements. An orthogonal-design ridge comparison then identifies when estimation savings outweigh the cost of restricting inputs. These results motivate held-out comparisons of restricted linear reference predictors to estimate selected data requirements. In controlled experiments, larger measured requirements tend to accompany larger gains from relaxing the corresponding restriction. Combining these measurements with candidate configurations and a finite-sample approximation to the estimation component of learning cost yields recommendations without training candidates or using performance results from other datasets. Across ten forecasters and 14 datasets, the three recommended candidates have a mean excess test error of 5.9% over the best candidate overall, comparable to the strongest training-free proxy (6.8%) and selectors trained on other datasets.

1 INTRODUCTION

Forecasting requires more than extracting a pattern from a window. Convolutions combine observations within a bounded span, recurrent models carry the past through a state, and attention connects observations by content. Their usefulness depends on the data and prediction target. Building on empirical studies of data characteristics and architectural choices (Wang et al., 2025a; Zhao et al., 2026; Liang et al., 2026), we examine which data properties determine the value of a forecaster's information and computations, and whether they can be measured before candidate training.

Average error can conceal where a structural choice helps or hurts. Two forecasters with the same MLP-mixer layers on Weather illustrate this: variable tokens represent each variable's full window, whereas patch tokens represent shorter segments. Pressure and density move together, while solar radiation follows a daily cycle (Figure 1a). Overall test errors differ by only 2%, but variable tokens do better on pressure and density and patch tokens on solar radiation, averaging over forecast steps (Figure 1b). Averaging across variables instead, patch tokens lead at early steps and variable tokens at later ones (Figure 1c). These reversals motivate examining both what a prediction needs and what a forecasting structure can use.

We compare the predictive cost of structural restrictions with the learning cost they can save. A structural restriction constrains information access or computation; its corresponding *data requirement* is a property of the data distribution describing how much predictable signal depends on what it removes. For channel independence, the requirement is other channels' added predictive value given a channel's own history. Relaxing a restriction can make more signal usable but demand more capacity, data, or computation to learn to use it. We call the loss above a structure's best attainable prediction its *learning cost*, reflecting imperfect estimation and optimization with available data and resources. A restricted model can perform better after training when learning-cost savings exceed the restriction's predictive cost.

Our analysis motivates held-out comparisons of restricted linear references to measure selected data requirements. We combine these measurements with candidate access, sharing, and fitted dimension

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

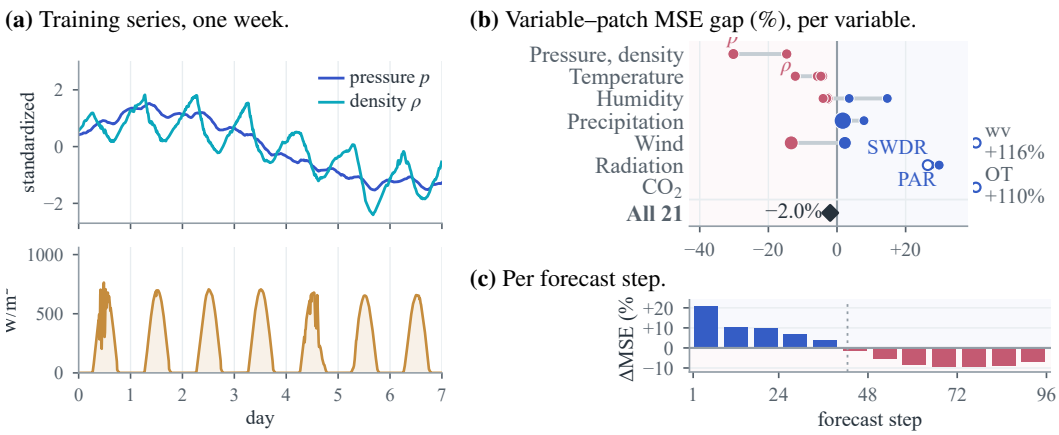


Figure 1: **Similar overall error, different winners across variables and forecast steps.** Variable versus patch tokens with the same MLP-mixer layers on Weather. **(a)** One training week: standardized pressure and density, and solar radiation. **(b)** Test MSE gaps by variable. **(c)** Gaps by forecast step, shown as eight-step means. Gaps are variable minus patch MSE, divided by their mean; negative favors variable tokens. In (b), dot area scales with MSE; open dots mark variables containing missing-value codes in training. Two off-scale values are labeled.

in a ridge-motivated score before candidate training. Controlled structural changes test whether the measurements track gains, while comparisons with training-free proxies and cross-dataset selectors assess recommendation quality.

The contributions are:

- **Restrictions priced by data requirements.** In linear prediction settings, we express restriction costs as functions of data requirements and structure. An orthogonal-design ridge comparison identifies when estimation savings outweigh the predictive cost of restricting inputs (Section 4).
- **Theory-guided measurement and scoring.** Held-out linear-reference comparisons measure selected data requirements. We combine them with candidate structure and a finite-sample approximation to the estimation component of learning cost in a score set using controlled synthetic experiments and held unchanged thereafter (Section 5).
- **Tests of structural effects and recommendation quality.** Measured requirements track the gains of controlled structural changes. Recommendation quality is comparable to training-free proxies (mean top-three regret: 5.9% vs. 6.8% for the strongest proxy) and selectors trained on other datasets. Replacing a dataset's requirements with another's degrades recommendation quality (Section 6).

2 Related Work

Architectural choices and data properties in forecasting. Benchmarks document variation in forecasting performance across datasets (Shao et al., 2025; Qiu et al., 2024). ARIES relates time-series properties to modeling strategies (Wang et al., 2025a), while TimeRecipe benchmarks module effects and TSCOMP learns component choices from model–dataset evaluations (Zhao et al., 2026; Liang et al., 2026). Studies of channel independence examine capacity–robustness trade-offs and the role of cross-channel dependence (Han et al., 2024; Abdelmalak et al., 2026). Analyses of foundation-model design likewise connect architectural choices to biases in the functions learned (Yu et al., 2026).

Structural restrictions and learning costs. Sequence approximation theory links architectural capabilities to memory decay, spectral regularity, and target smoothness (Li et al., 2022; Jiang et al., 2021; Jiang & Li, 2024; Takakura & Suzuki, 2023). Recall and state-tracking studies expose

task-dependent limits of attention, gated convolutions, and state-space models (Arora et al., 2024; Merrill et al., 2024); other results concern local connectivity and shared or equivariant representations (Alexander et al., 2023; Elesedy & Zaidi, 2021; Maurer et al., 2016). Our analysis uses conditional-expectation and reduced-rank tools (Banerjee et al., 2005; Izenman, 1975) to price restrictions. Related ridge-risk analyses (Dobriban & Wager, 2018) provide context for our orthogonal-design comparison of restriction and estimation costs.

Measuring predictive information. Predictive $\mathcal{V}$-information measures information usable by a predictor class (Xu et al., 2020; Ethayarajh et al., 2022); predictive heterogeneity measures its increase from distinguishing subpopulations (Liu et al., 2023). Variable importance and Granger causality assess the value of additional variables or histories (Lei et al., 2018; Williamson et al., 2023; Granger, 1969). Landmarking characterizes datasets through simple learners (Pfahringer et al., 2000). Our linear-reference contrasts are guided by restriction costs, with their connection to structural gains assessed through controlled interventions.

Training-free architecture scoring. Forecasting-specific architecture search evaluates combinations of modules through candidate training (Deng & Lindauer, 2025; Cao et al., 2026). Zero-cost NAS evaluates candidates at initialization using network statistics, including criteria adapted from pruning (Abdelfattah et al., 2021; Lee et al., 2019; Tanaka et al., 2020) and input-Jacobian statistics (Mellor et al., 2020). TE-NAS and ZiCo provide theoretical motivation for such scores (Chen et al., 2021; Li et al., 2023); MRS-based search and TFAS apply training-free evaluation to time-series settings (Camero et al., 2021; Trirat & Lee, 2025). Transferability measures such as LEEP and LogME evaluate already pretrained models (Nguyen et al., 2020; You et al., 2021). The requirement score uses candidate configurations and requirements measured by fitted linear references, so scoring does not require running candidate networks.

Meta-learning for model selection. Meta-learning recommends algorithms using performance on previous tasks (Brazdil et al., 2003; Bischl et al., 2016). In forecasting, FFORMS, FFORMPP, and AutoForecast learn model choices or expected errors (Talagala et al., 2023; 2022; Abdallah et al., 2025), whereas FFORMA learns combination weights (Montero-Manso et al., 2020). MetaOD selects outlier models, ZAP selects pretrained pipelines (Zhao et al., 2021; Öztürk et al., 2022), and TabRepo supports portfolio construction from tabular-model evaluations (Salinas & Erickson, 2024). We compare with FFORMA weights used as rankings and simplified selectors based on MetaOD's factorization and ZAP's pairwise ranking (Appendix D.5). These require cross-dataset candidate performance histories; our score does not.

3 Problem Setup

At forecast origin t, the available information O_t contains the last L observations of C channels and any covariates known at t. The target $Y_t \in \mathbb{R}^{C\times H}$ contains the values of the same channels over the next H steps.

Suppressing t, we use total squared-error risk $\mathcal{R}(f) = \mathbb{E}\|Y - f(O)\|^2$. For a fixed evaluation target, averaging over the same coordinates rescales this risk without changing candidate ordering; experiments report MSE. For $\mathbb{E}\|Y\|^2 < \infty$, the Bayes predictor $\mu(O) = \mathbb{E}[Y \mid O]$ attains the minimum risk given O, denoted by R_B.

A model configuration m specifies the complete input-to-output computation, including information access, processing, and parameter sharing. Its realizable predictors form a class $\mathcal{F}_m$ with *class risk* $\mathcal{R}^*_m = \inf_{f\in\mathcal{F}_m} \mathcal{R}(f)$. A trained predictor $\hat{f}_m \in \mathcal{F}_m$ has risk

$$\mathcal{R}(\hat{f}_m) = \underbrace{\mathcal{R}^*_m}_{\text{class risk}} + \underbrace{\left(\mathcal{R}(\hat{f}_m) - \mathcal{R}^*_m\right)}_{\text{learning cost}}.$$

The *learning cost* is the excess risk within the class, reflecting imperfect estimation and optimization with the available data and computational resources. Comparisons hold O, the target, loss, and evaluation origins fixed.

A *restriction* narrows a predictor class $\mathcal{F}_1$ to a subclass $\mathcal{F}_0 \subseteq \mathcal{F}_1$ through a structural constraint. Its predictive cost is the class-risk difference $\Delta = \mathcal{R}^*_0 - \mathcal{R}^*_1 \geq 0$. A *data requirement* is a functional

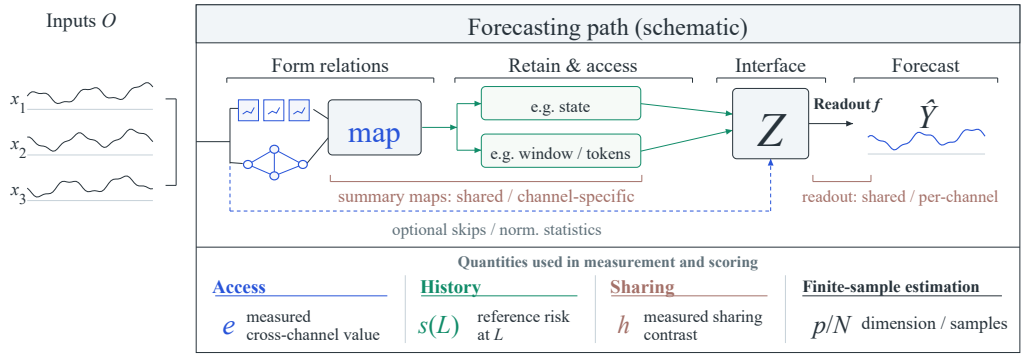


Figure 2: **Structural restrictions along a schematic forecasting path.** The interface Z collects all inputs to the readout f, including optional skips and normalization statistics. Sharing may differ between summary maps and the readout. The lower panel lists quantities used in measurement and scoring (Section 5); $s(L)$ denotes reference risk and p/N the dimension-to-sample ratio.

of the data distribution alone that characterizes the predictive signal whose use is limited by the restriction. We *price* the restriction by expressing Δ in terms of this requirement and the structural parameters of the two classes.

Section 4 derives restriction costs for nested predictor classes. For the generally non-nested classes of published forecasters, Section 6 evaluates a score based on restrictions identified from their configurations (Section 5.2).

4 Pricing Restrictions Along the Path

We price restrictions along the forecasting path (Figure 2); an orthogonal-design ridge comparison then shows when estimation savings outweigh the cost of restricting inputs. Proofs appear in Appendix A, with structural extensions in Appendix B.

4.1 Information at the Readout

Let $Z = \phi(O)$ collect all readout inputs, including queries, skips, and normalization statistics.

Proposition 1 (Interface decomposition)**.** *For every measurable readout f with $\mathbb{E}\|f(Z)\|^2 < \infty$,*

$$\mathcal{R}(f \circ \phi) - R_B = \underbrace{\mathbb{E}\|\mu - \mathbb{E}[\mu \mid Z]\|^2}_{D(Z)} + \mathbb{E}\|\mathbb{E}[\mu \mid Z] - f(Z)\|^2. \tag{1}$$

Here $D(Z)$ is the cost of observing Z rather than O; the second term is readout excess risk given Z. For fixed ϕ, unrestricted readouts attain class risk $R_B + D(Z)$; trainable paths also optimize over admissible interfaces and readout classes (Banerjee et al., 2005). Adding an input helps only if it changes the conditional mean given Z (Corollary 6, Appendix A.1).

4.2 Sharing a Prediction Rule

Let channel j have weight $\pi_j > 0$, $\sum_j \pi_j = 1$, centered input X_j with common covariance Σ, and conditional target mean $\beta_j^\top X_j$. Write $\bar{\beta} = \sum_j \pi_j \beta_j$ and $\|v\|_\Sigma^2 = v^\top \Sigma v$.

Proposition 2 (Shared rule)**.** *The common coefficient $\bar{\beta}$ minimizes the weighted risk. Its excess class risk over channel-specific linear rules is*

$$\sum_j \pi_j \|\beta_j - \bar{\beta}\|_\Sigma^2. \tag{2}$$

The cost is the covariance-weighted dispersion of channel-specific coefficients. Fitted-rule comparisons also include estimation differences: sharing helps when its estimation savings exceed this cost (Appendix A.2).

4.3 SHARING A SUMMARY

A shared summary can discard information even with channel-specific readouts. Let $U_j \sim N(0, I_d)$ have conditional target mean $A_j U_j$, and let positive channel weights π_j sum to one. Set $K_j = A_j^\top A_j$; $\mathrm{tail}_r(K)$ sums eigenvalues beyond the largest r, for $1 \le r \le d$.

Proposition 3 (Shared linear summaries)**.** *With unrestricted channel-specific readouts, the minimum excess risks over the Bayes risk for shared and channel-specific rank-r linear summaries are, respectively,*

$$\mathrm{tail}_r\left(\sum_j \pi_j K_j\right) \quad \textit{and} \quad \sum_j \pi_j \,\mathrm{tail}_r(K_j).$$

They coincide if and only if a rank-r subspace is optimal for every channel.

Their difference prices sharing at fixed rank: channel-specific readouts cannot recover directions discarded by the summary. Appendix A.3 gives the reduced-rank calculation (Izenman, 1975) and examples.

4.4 RETAINING HISTORY

Relative to a longer reference history, a window W_r incurs class-risk cost $q_{\mathrm{win}}(r) = D(W_r)$. For the stationary fading state $S_t = aS_{t-1} + \xi_t$, with $0 < a < 1$ and iid centered innovations of variance σ_ξ^2, predicting S_t from the last r innovations costs $\sigma_\xi^2 a^{2r}/(1-a^2)$ relative to the full innovation history. Thus more persistent states require longer windows to achieve the same cost (Boyd & Chua, 1985); Proposition 9 and extensions appear in Appendices A.4–B.

4.5 FORMING RELATIONS

Observing context is insufficient if it cannot change how inputs are combined.

Proposition 4 (Context-dependent coefficients)**.** *Let X be centered with $\mathrm{Cov}(X) = \sigma^2 I$, independent of an observed context c, and let $\mu = \beta(c)^\top X$, with $\mathbb{E}\|\beta(c)\|^2 < \infty$. Over predictors $w^\top X + b(c)$ with fixed w,*

$$\inf_{w,b} \mathbb{E}[\mu - w^\top X - b(c)]^2 = \sigma^2 \textstyle\sum_k \mathrm{Var}(\beta_k(c)). \tag{3}$$

The cost measures variation in the required coefficients: $b(c)$ cannot represent the context–input interaction. A separate restriction arises when compression precedes context access. In a Gaussian coordinate-query example, a fixed low-rank summary loses information that a query-aware encoder retains in one scalar, even though both readouts see the query (Proposition 8, Appendix A.3). These mechanisms complement studies of associative recall and copying (Arora et al., 2024; Jelassi et al., 2024) and motivate the synthetic tests in Appendix E.2.

4.6 LEARNING WHAT IS ACCESSIBLE

To compare restriction costs with the estimation component of learning cost, we use fixed orthogonal designs and scalar ridge penalties minimizing expected risk.

Proposition 5 (Ridge recovery)**.** *Let $y = X\beta + \epsilon$, with fixed $X^\top X = nI_p$, $\|\beta\|^2 = \gamma > 0$, and $\epsilon \sim N(0, \sigma^2 I_n)$, $\sigma^2 > 0$. For an independent test input with mean zero and covariance I_p, ridge regression with penalty $\lambda^* = \sigma^2 p/\gamma$ has expected estimation cost $\gamma\nu/(\gamma+\nu)$, where $\nu = \sigma^2 p/n$. Its reduction in signal prediction error relative to the zero predictor is $r_\nu(\gamma) = \gamma^2/(\gamma+\nu)$.*

Now let $X_1 = [X_0, X_a]$ satisfy $X_1^\top X_1 = nI_{p_1}$, with $1 \le p_0 < p_1 \le n$. For $y = X_0\beta_0 + X_a\beta_a + \epsilon$, set $\gamma = \|\beta_0\|^2 > 0$ and $g = \|\beta_a\|^2 \ge 0$. Under the same noise and independent isotropic test-input assumptions, fitting both models at their expected-risk-minimizing penalties gives

$$\mathbb{E}\mathcal{R}(\hat{f}_0) - \mathbb{E}\mathcal{R}(\hat{f}_1) = g + \frac{\gamma\nu_0}{\gamma+\nu_0} - \frac{(\gamma+g)\nu_1}{\gamma+g+\nu_1}, \qquad \nu_i = \sigma^2 p_i/n. \tag{4}$$

Thus restricting inputs helps exactly when estimation savings exceed g; the difference tends to g as n grows. Corollary 10 in Appendix A.6 gives the full comparison. These calculations motivate the measurements and approximate score below.

5 From Requirements to Predictions

We measure selected requirements with linear references and combine them with candidate configurations. These *measured requirements* are held-out error contrasts, reflecting both predictive differences captured by the references and finite-sample fitting and evaluation variability.

5.1 Measuring requirements

For each dataset, history L, and horizon H, ridge references exchange fitting and evaluation roles across two separated blocks within the training split. Let $R_{\bullet,c}$ denote channel c's reference MSE averaged over the two folds, and $R_\bullet$ its mean across sampled channels. The *reference level* $s = R_{\text{own}}$ uses channel-specific rules with own-channel history. We define

$$h = R_{\text{shared}} - s, \qquad e = \text{median}_c\big(R_{\text{shifted},c} - R_{\text{aligned},c}\big). \tag{5}$$

The *sharing requirement* h compares a jointly fitted shared rule with channel-specific rules. Its sign includes estimation effects, so a positive class-risk cost of sharing need not imply positive h.

The *external requirement* e compares a reference that adds other channels' aligned histories with a time-shifted control, retaining own-channel inputs and equal input dimension. It measures the value of temporal alignment within the references, motivated by the conditional value of additional inputs (Corollary 6). History enters through $s(L)$ and its change with L. Appendix C specifies the fits, shifts, and error aggregation.

5.2 Reading the restriction

Configuration indicators a_e, a_h, a_p mark cross-channel access, channel-specific parameters, and a shared rule, respectively. With $x_+ = \max(x, 0)$, the *assigned gains* are

$$g_e = a_e \min(e_+, s), \qquad g_h = a_h \min(h_+, s), \qquad g_p = a_p \min((-h)_+, s). \tag{6}$$

Each gain is capped at s; indicators may coexist in different parts of a computation. The *fitted dimension* p is a configuration-based proxy for estimation dimension, and nominal N counts training windows times channels. Appendix C.4 gives the assignments. Neither mapping nor scoring runs the candidate network.

5.3 The requirement score

Motivated by Proposition 5, we use $r_\nu(g) = g^2/(g+\nu)$ with $\nu = sp/N$ and $r_0(0) = 0$. Substituting reference error s, proxy dimension p, and nominal N is a modeling approximation. Let $G = \sum_{j\in\{e,h,p\}} g_j$, and rescale only the recovery inputs: $\tilde{g}_j = g_j \min(1, s/G)$ for $G > 0$, and zero otherwise. The score is

$$Q_m = \underbrace{(s - G)}_{\text{requirement term}} + \underbrace{\tfrac{1}{2}\big(G - \textstyle\sum_j r_\nu(\tilde{g}_j) + D_{\text{ro}}\big)}_{\text{learning term}}. \tag{7}$$

The readout adjustment D_{ro} compares flattened and patch-aggregated dimensions where applicable (Appendix C.4). The raw G rewards assigned gains; incomplete recovery and readout dimension adjust this reward. The score is not a guaranteed attainable risk. Mappings and weights were set in controlled synthetic experiments and held unchanged thereafter; lower Q_m predicts lower test risk.

Predictions. The score predicts: (i) access gains increase with positive e until capped, and are zero for $e \le 0$; (ii) $h > 0$ favors channel-specific rules and $h < 0$ a shared rule; (iii) larger decreases in $s(L)$ favor longer histories, all else equal; (iv) with other score inputs fixed and positive recoverable gain, larger p raises the learning term and predicts higher test error; (v) equal score inputs tie, although unrepresented relation requirements can distinguish test errors. Section 6.2 tests (i)–(v), and Section 6.4 examines the score terms.

6 Experiments

Controlled changes test structural predictions (Section 6.2); published forecasters test recommendation quality and its drivers (Sections 6.3–6.4).

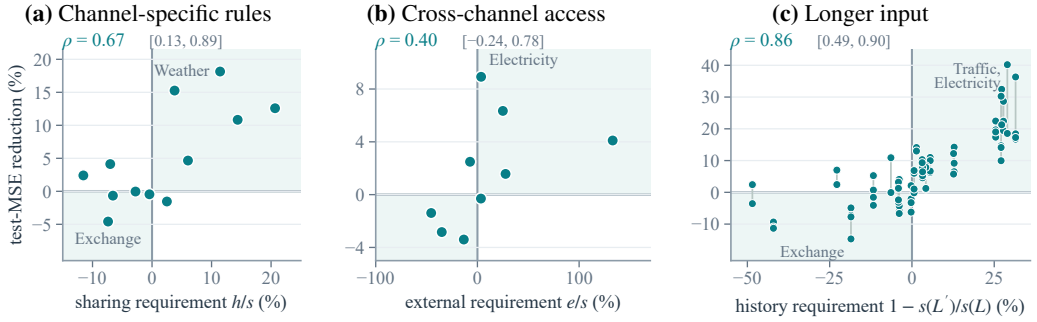


Figure 3: **Measured requirements and test gains.** Horizontal axes: training-split requirements; vertical axes: test-MSE reduction relative to the restricted configuration. (a) Channel-specific versus shared linear rules; (b) cross-channel access for an MLP mixer, one point per dataset; (c) history extended from 96 to 192–720 steps for four cores, one point per dataset, core, and length. Here $s(L')$ is the longer-window reference level; joins group cores at one dataset and length. Above: Spearman ρ and 95% dataset-bootstrap intervals; shading marks matching signs. In (b), dots show raw e/s, but ρ uses the clipped score input. Three datasets in (b) and two cases in (c) are off scale but included in statistics.

6.1 Setup

Data. We use 14 datasets: four ETT datasets, Weather, Electricity, Traffic (Zhou et al., 2021; Wu et al., 2021), Exchange, and six UCI series (Appendix D.2), with chronological splits and $L = H = 96$ unless varied.

Controlled experiments. Six *cores*, from a linear map to attention, relate time-step tokens on 12 datasets. We vary cross-channel access (i), channel-specific rules (ii), history (iii), readout (iv), or core (v) separately. Patch or variable tokens change assigned gains and fitted dimension together. Each variant is re-tuned under its parent's budget (Appendix D.3).

Published forecasters. The ten candidates are DLinear and NLinear (Zeng et al., 2023), FITS (Xu et al., 2024), N-HiTS (Challu et al., 2023), TSMixer (Chen et al., 2023), PatchTST (Nie et al., 2023), iTransformer (Liu et al., 2024), S-Mamba (Wang et al., 2025b), ModernTCN (Luo & Wang, 2024), and TimeMixer (Wang et al., 2024). Each is searched over twelve configurations and trained with three seeds; the score and every comparison method see only the default configuration, and their rankings are evaluated against the mean test MSE after search (Appendix D.4).

Comparison methods. Eight proxies evaluate untrained networks: GradNorm, SNIP, GraSP (Wang et al., 2020), SynFlow, JacobCov (Mellor et al., 2020), ZiCo, neural tangent kernel condition number, and parameter count. Seven selectors additionally use candidate test results from other datasets (Appendix D.5). Figure 4 shows the four proxies with highest Spearman correlation, single best by average rank (Bischl et al., 2016), and kNN average rank.

Metrics. The main measure, *top-three regret*, averages the three recommendations' percentage excess test error over the best candidate. Spearman correlation evaluates the whole ranking (Abdelfattah et al., 2021; Krishnakumar et al., 2022); *best in top three* measures inclusion of the best candidate. *Direction accuracy* counts correctly ordered pairs with test errors differing by more than 1%. Ties count as uniform draws; metrics are dataset averages with 95% dataset-bootstrap intervals (Appendix D.6).

6.2 Requirements and the effect of one structural change

Measured requirements order the test gains of channel-specific rules and of longer inputs, and the gains of cross-channel access in the point estimate (Figure 3). Channel-specific linear rules gain more as the sharing requirement grows (prediction (ii); Spearman 0.67, interval 0.13–0.89, 12 datasets). Cross-channel access gains more as the external requirement grows (prediction (i); 0.40, interval

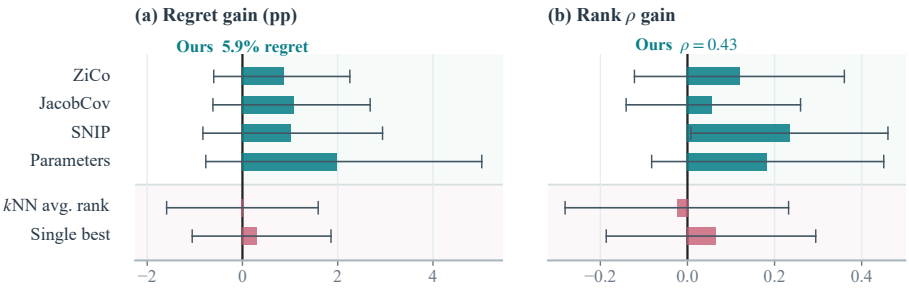


Figure 4: **Recommendations comparable to strong proxies and selectors before training.** Ten forecasters, 14 datasets. Bars show mean paired advantages: (a) comparator minus score top-three regret (percentage points); (b) score minus comparator Spearman ρ. Whiskers: 95% dataset-bootstrap intervals; positive values favor the score. Upper four methods use only the new dataset's training series; lower two also use candidate test results from other datasets.

-0.24–0.78, 12 datasets) and lowers test error on five of the six datasets with $e > 0$. A longer input gains more where the reference risk decreases more (prediction (iii); 0.86, interval 0.49–0.90, 80 dataset–core–length cases on eight datasets).

Predictions (iv) and (v) concern the fitted dimension, which only the learning term prices, and the core, which the score leaves out. Token-layout changes vary the assigned gains and the fitted dimension together; across 63 pairs, direction accuracy is 60% for the requirement term, 54% for the learning term, and 74% for the full score (Figure 7). Readout changes vary only the fitted dimension: the flattened readout, which has the larger fitted dimension, has the lower test error on eight of 12 pairs, against prediction (iv), and the score's direction accuracy is 38% (Figure 6). Changing only the core at fixed tokens, readout, access, and sharing changes test error by a median 1.7%; the score ties 92% of exchanges differing by more than 1%, illustrating prediction (v). On synthetic series with planted relation requirements, cores unable to form the relation have up to 84% higher test error, consistent with Propositions 4 and 8 (Appendices E.1 and E.2).

6.3 Recommending forecasters before training

Before any candidate is trained, the score's recommendations match those of the strongest training-free proxies and exceed several others (Figure 4). The score's top-three regret is 5.9%, against 6.8% for ZiCo, the strongest proxy, and 10.9% for uniform choice; its Spearman correlation is 0.43, against 0.38 for JacobCov. Both measures are best among methods using only the new dataset, including when any dataset or the ETT family is omitted. Differences from ZiCo and JacobCov have intervals including zero; the score's regret is lower than that of each on 10 of 14 datasets. Against the other six proxies, intervals exclude zero in the score's favor for three in regret and five in correlation (Appendices E.3 and E.4).

Selectors that also learn from candidate test results on other datasets are comparable to the score, with 5.9–8.7% regret. For five of seven, intervals of both differences include zero; Portfolio in regret and FFORMS in correlation fall behind the score with intervals excluding zero (Appendix E.3).

The score can also precede search: training its top five for 1,024 updates each, then fully searching the one with lowest tuning error, gives 4.0% regret, versus 3.8% for all ten at twice the updates and 5.0% for five random candidates (Appendix E.5). Requirement measurement takes 2–41 CPU seconds per dataset.

6.4 What drives the recommendations

The ranking comes from each dataset's own requirements. Permuting the requirements at random across datasets raises top-three regret from 5.9% to 11.1%, the level of uniform choice (10.9%), and lowers the Spearman correlation from 0.43 to 0.07; both intervals exclude zero, and 12 of the 14 datasets lose on both measures (Table 1).

Table 1: **Recommendations depend on dataset-specific requirements.** Δ: variant minus full score, with 95% dataset-bootstrap intervals (percentage points for percentage metrics). Gain-type ranking counts available gains, breaking ties by smaller fitted dimension. Reassignment permutes normalized requirements across datasets, including self-assignments (10,000 permutations). Ablations remove external or sharing gains. Values and changes are rounded separately.

	Top-3 regret (%) ↓		Spearman ρ ↑		Best in top 3 (%) ↑	
Variant	Value	Δ [95% CI]	Value	Δ [95% CI]	Value	Δ [95% CI]
Ranking by						
Uniform choice	10.9	+5.0 [1.8, 8.5]	–	–	30	−25 [−49, 0]
Gain types and fitted dimension	6.4	+0.5 [−0.8, 2.0]	0.10	−0.33 [−0.46, −0.19]	50	− 5 [−29, 19]
Requirement term	6.0	+0.1 [−0.5, 0.8]	0.33	−0.10 [−0.23, 0.02]	46	− 9 [−27, 8]
Full score	5.9	–	0.43	–	55	–
Wrong requirements						
Reassigned across datasets	11.1	+5.2 [2.3, 7.4]	0.07	−0.36 [−0.46, −0.18]	41	−14 [−28, 2]
One part removed						
External gain g_e	9.8	+3.9 [1.5, 6.6]	0.25	−0.18 [−0.30, −0.08]	41	−13 [−30, 0]
Sharing gains g_h, g_p	6.1	+0.2 [−0.4, 0.8]	0.31	−0.12 [−0.27, 0.02]	44	−11 [−30, 6]

Best-candidate shortlists vary nearly as much as random ones: two datasets share one of their three best candidates on average, versus 0.9 at random. Proxies share 1.8–3.0; ranking by gain types and fitted dimension selects one candidate everywhere and achieves only 0.10 correlation. The score shares 1.4 candidates and selects three distinct first choices (Appendix E.6).

The measured requirements carry most ranking information; the learning term orders their ties. It raises correlation from 0.33 to 0.43, with a difference interval including zero; the requirement term alone reaches 6.0% regret. Candidates access nested gains (shared rule, then external input, then channel-specific rules), so requirement signs determine their requirement-term ordering and magnitudes enter through the learning term. Removing external gains has the largest effect: regret rises to 9.8%, with a difference interval excluding zero (Table 1).

7 Discussion and Conclusion

Architectural choices govern both usable predictive signal and the cost of learning to use it. Our analysis expresses selected restriction costs through data requirements and makes the trade-off with estimation explicit in an orthogonal-design ridge comparison. This perspective guides measurements that connect to controlled structural gains and support competitive recommendations without candidate training or cross-dataset performance histories.

The reassignment experiment clarifies why the measurements matter. A fixed preference for a larger model or a more permissive structure cannot explain recommendations that deteriorate when the same requirements are assigned to different datasets. What matters is whether the information made usable by a candidate has predictive value for the target dataset. The controlled comparisons examine this connection one restriction at a time, while the published-forecaster study tests whether it remains useful when several choices coexist.

The distinction between a restriction cost and a measured reference gain is important here. Finite-sample reference comparisons include estimation effects, and the score uses nominal sample counts and configuration-based dimensions rather than the effective quantities governing each trained network. The readout results show where this approximation can fail; core exchanges show effects that the current measurements do not resolve. The score is therefore most useful as a basis for forming a shortlist and identifying which structural differences warrant evaluation. The search experiment illustrates this role by combining inexpensive screening with subsequent candidate training and tuning.

Richer nonlinear and context-dependent measurements, together with more precise finite-resource learning models, would extend this account. For time-series foundation models, a natural next step is to examine how pretraining changes usable signal and the cost of adaptation to a target dataset. Linking these changes to target-data requirements could guide both architecture and adaptation choices.

Reproducibility Statement

Appendix C gives every reference fit, requirement, and restriction rule, and Appendix D lists the data splits, the search space and budget of every forecaster, the settings of the comparison methods, the metrics, and the seeds (Tables 4–6). Requirement scores and proxy scores read only the training split and the configuration; each stored score, configuration, and test result is recorded with its SHA-256, and all reported metrics are recomputed from the stored scores and test errors by evaluation scripts released with them (Appendix D.7). We will release the code, the stored scores, the selected configurations, and the per-seed test errors.

AI Use Statement

Large language model assistants were used for language editing and drafting assistance, literature search and reference checking, and routine code and figure preparation, including assistance with synthetic data generators and descriptive statistics. The authors reviewed the assisted work, interpreted the results, and take responsibility for the content of this paper.

References

Mustafa Abdallah, Ryan A. Rossi, Kanak Mahadik, Sungchul Kim, Handong Zhao, and Saurabh Bagchi. Evaluation-Free Time-Series Forecasting Model Selection via Meta-Learning. *ACM Transactions on Knowledge Discovery from Data*, 19(3):64:1–64:41, 2025. doi: 10.1145/3715149. URL https://engineering.purdue.edu/dcsl/wp-content/uploads/2025/02/AutoForecast_ACM_TKDD.pdf.

Mohamed S. Abdelfattah, Abhinav Mehrotra, Łukasz Dudziak, and Nicholas D. Lane. Zero-cost proxies for lightweight NAS. In *International Conference on Learning Representations*, 2021.

Ibram Abdelmalak, Kiran Madhusudhanan, Jungmin Choi, Christian Klötergens, Vijaya Krishna Yalavarthi, Maximilian Stubbemann, and Lars Schmidt-Thieme. Channel dependence, limited lookback windows, and the simplicity of datasets: How biased is time series forecasting? In *Advances in Knowledge Discovery and Data Mining (PAKDD)*, volume 16598 of *Lecture Notes in Computer Science*, pp. 585–597. Springer, 2026. doi: 10.1007/978-981-92-1462-4_46.

Yotam Alexander, Nimrod De La Vega, Noam Razin, and Nadav Cohen. What Makes Data Suitable for a Locally Connected Neural Network? A Necessary and Sufficient Condition Based on Quantum Entanglement. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/html/2303.11249v5.

Simran Arora, Sabri Eyuboglu, Aman Timalsina, Isys Johnson, Michael Poli, James Zou, Atri Rudra, and Christopher Ré. Zoology: Measuring and Improving Recall in Efficient Language Models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2312.04927.

Arindam Banerjee, Xin Guo, and Hui Wang. On the optimality of conditional expectation as a Bregman predictor. *IEEE Transactions on Information Theory*, 51(7):2664–2669, 2005.

Bernd Bischl, Pascal Kerschke, Lars Kotthoff, Marius Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, and Joaquin Vanschoren. ASlib: A benchmark library for algorithm selection. *Artificial Intelligence*, 237:41–58, 2016.

Stephen Boyd and Leon O. Chua. Fading Memory and the Problem of Approximating Nonlinear Operators with Volterra Series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, 1985. doi: 10.1109/TCS.1985.1085649.

Pavel B. Brazdil, Carlos Soares, and Joaquim Pinto da Costa. Ranking learning algorithms: Using IBL and meta-learning on accuracy and time results. *Machine Learning*, 50(3):251–277, 2003.

Andrés Camero, Hao Wang, Enrique Alba, and Thomas Bäck. Bayesian neural architecture search using a training-free performance metric. *Applied Soft Computing*, 106:107356, 2021. doi: 10.1016/j.asoc.2021.107356.

Luis Candanedo. Appliances Energy Prediction. Dataset, UCI Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/dataset/374/appliances+energy+prediction. doi: 10.24432/C5VC8G.

Luis M. Candanedo, Véronique Feldheim, and Dominique Deramaix. Data driven prediction models of energy use of appliances in a low-energy house. *Energy and Buildings*, 140:81–97, 2017. doi: 10.1016/j.enbuild.2017.01.083.

Qianying Cao, Shanqing Liu, Alan John Varghese, Jérôme Darbon, Michael S. Triantafyllou, and George Em Karniadakis. Automatic selection of the best neural architecture for time series forecasting. *Nature Communications*, 17(1):7058, 2026. doi: 10.1038/s41467-026-73687-9.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza Ramirez, Max Mergenthaler Canseco, and Artur Dubrawski. NHITS: Neural hierarchical interpolation for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 6989–6997, 2023. doi: 10.1609/aaai.v37i6.25854.

S. Chen. Beijing Multi-Site Air Quality. Dataset, UCI Machine Learning Repository, 2017. URL https://archive.ics.uci.edu/dataset/501/beijing+multi+site+air+quality+data.

Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O. Arik, and Tomas Pfister. TSMixer: An all-MLP architecture for time series forecasting. *Transactions on Machine Learning Research*, 2023. URL https://research.google/pubs/tsmixer-an-all-mlp-architecture-for-time-series-forecasting/.

Wuyang Chen, Xinyu Gong, and Zhangyang Wang. Neural architecture search on ImageNet in four GPU hours: A theoretically inspired perspective. In *International Conference on Learning Representations*, 2021.

S. De Vito, E. Massera, M. Piga, L. Martinotto, and G. Di Francia. On field calibration of an electronic nose for benzene estimation in an urban pollution monitoring scenario. *Sensors and Actuators B: Chemical*, 129(2):750–757, 2008. doi: 10.1016/j.snb.2007.09.060.

Saverio De Vito. Air Quality. Dataset, UCI Machine Learning Repository, 2008. URL https://archive.ics.uci.edu/dataset/360/air+quality. doi: 10.24432/C59K5F.

Difan Deng and Marius Lindauer. Optimizing time series forecasting architectures: A hierarchical neural architecture search approach. *Transactions on Machine Learning Research*, 2025. URL https://openreview.net/forum?id=Ym2wqojm4e.

Lee H. Dicker. Ridge Regression and Asymptotic Minimax Estimation over Spheres of Growing Dimension. *Bernoulli*, 22(1):1–37, 2016.

Edgar Dobriban and Stefan Wager. High-dimensional asymptotics of prediction: Ridge regression and classification. *The Annals of Statistics*, 46(1):247–279, 2018. doi: 10.1214/17-AOS1549.

Carl Eckart and Gale Young. The Approximation of One Matrix by Another of Lower Rank. *Psychometrika*, 1(3):211–218, 1936.

Bryn Elesedy and Sheheryar Zaidi. Provably Strict Generalisation Benefit for Equivariant Models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 2959–2969, 2021.

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 5988–6008, 2022. URL https://proceedings.mlr.press/v162/ethayarajh22a.html.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.

C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):424–438, 1969. doi: 10.2307/1912791.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Lu Han, Han-Jia Ye, and De-Chuan Zhan. The Capacity and Robustness Trade-off: Revisiting the Channel Independent Strategy for Multivariate Time Series Forecasting. *IEEE Transactions on Knowledge and Data Engineering*, 36(11):7129–7142, 2024. doi: 10.1109/TKDE.2024.3400008.

Georges Hebrail and Alice Berard. Individual Household Electric Power Consumption. Dataset, UCI Machine Learning Repository, 2006. URL https://archive.ics.uci.edu/dataset/235/individual+household+electric+power+consumption. doi: 10.24432/C58K54.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

John Hogue. Metro Interstate Traffic Volume. Dataset, UCI Machine Learning Repository, 2019. URL https://archive.ics.uci.edu/dataset/492/metro+interstate+traffic+volume. doi: 10.24432/C5X60B.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248–264, 1975.

Samy Jelassi, David Brandfonbrener, Sham M. Kakade, and Eran Malach. Repeat After Me: Transformers are Better than State Space Models at Copying. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 21502–21521, 2024.

Haotian Jiang and Qianxiao Li. Approximation Rate of the Transformer Architecture for Sequence Modeling. In *Advances in Neural Information Processing Systems*, volume 37, pp. 68926–68955, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7f64034009f4a5fa417a57e1a987c5cd-Abstract-Conference.html.

Haotian Jiang, Zhong Li, and Qianxiao Li. Approximation Theory of Convolutional Architectures for Time Series Modelling. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 4961–4970, 2021. URL https://proceedings.mlr.press/v139/jiang21d.html.

Taesung Kim, Jinhee Kim, Yunwon Tae, Cheonbok Park, Jang-Ho Choi, and Jaegul Choo. Reversible instance normalization for accurate time-series forecasting against distribution shift. In *International Conference on Learning Representations*, 2022. URL https://iclr.cc/virtual/2022/poster/6034.

Arjun Krishnakumar, Colin White, Arber Zela, Renbo Tu, Mahmoud Safari, and Frank Hutter. NAS-Bench-Suite-Zero: Accelerating research on zero cost proxies. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 35, pp. 28037–28051, 2022.

Guokun Lai, Wei-Cheng Chang, Yiming Yang, and Hanxiao Liu. Modeling Long- and Short-Term Temporal Patterns with Deep Neural Networks. In *Proceedings of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 95–104, 2018. doi: 10.1145/3209978.3210006.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*, 2019.

Jing Lei, Max G'Sell, Alessandro Rinaldo, Ryan J. Tibshirani, and Larry Wasserman. Distribution-Free Predictive Inference for Regression. *Journal of the American Statistical Association*, 113 (523):1094–1111, 2018.

Guihong Li, Yuedong Yang, Kartikeya Bhardwaj, and Radu Marculescu. ZiCo: Zero-shot NAS via inverse coefficient of variation on gradients. In *International Conference on Learning Representations*, 2023.

Zhong Li, Jiequn Han, Weinan E, and Qianxiao Li. Approximation and Optimization Theory for Linear Continuous-Time Recurrent Neural Networks. *Journal of Machine Learning Research*, 23 (42):1–85, 2022. URL https://www.jmlr.org/papers/v23/21-0368.html.

Shuang Liang, Chaochuan Hou, Xu Yao, Shiping Wang, Hailiang Huang, Songqiao Han, and Minqi Jiang. Beyond Holistic Models: Systematic Component-level Benchmarking of Deep Multivariate Time-Series Forecasting. In *Proceedings of the 32nd ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 9392–9402, 2026. doi: 10.1145/3770855.3817551. URL https://arxiv.org/html/2605.26562v1.

Jiashuo Liu, Jiayun Wu, Renjie Pi, Renzhe Xu, Xingxuan Zhang, Bo Li, and Peng Cui. Measure the predictive heterogeneity. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=g2oB_k-18b.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/2ea18fdc667e0ef2ad82b2b4d65147ad-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Carl H. Lubba, Sarab S. Sethi, Philip Knaute, Simon R. Schultz, Ben D. Fulcher, and Nick S. Jones. catch22: CAnonical time-series CHaracteristics. *Data Mining and Knowledge Discovery*, 33(6): 1821–1852, 2019.

Donghao Luo and Xue Wang. ModernTCN: A modern pure convolution structure for general time series analysis. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/86b1437c1e4c3b3c4debff98234a67e7-Abstract-Conference.html.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Joseph Mellor, Jack Turner, Amos Storkey, and Elliot J. Crowley. Neural architecture search without training. *arXiv preprint arXiv:2006.04647v1*, 2020. URL https://arxiv.org/abs/2006.04647v1. First version, which introduced the Jacobian covariance score.

William Merrill, Jackson Petty, and Ashish Sabharwal. The Illusion of State in State-Space Models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35492–35506, 2024.

Pablo Montero-Manso, George Athanasopoulos, Rob J. Hyndman, and Thiyanga S. Talagala. FFORMA: Feature-based Forecast Model Averaging. *International Journal of Forecasting*, 36 (1):86–92, 2020. doi: 10.1016/j.ijforecast.2019.02.011. URL https://robjhyndman.com/papers/fforma.pdf.

Cuong Nguyen, Tal Hassner, Matthias Seeger, and Cedric Archambeau. LEEP: A New Measure to Evaluate Transferability of Learned Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7294–7305, 2020.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8162–8171, 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Yuqi Nie, Nam H. Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. A time series is worth 64 words: Long-term forecasting with transformers. In *International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/poster/10876.

Ekrem Öztürk, Fabio Ferreira, Hadi Jomaa, Lars Schmidt-Thieme, Josif Grabocka, and Frank Hutter. Zero-shot AutoML with pretrained models. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 17138–17155, 2022.

Bernhard Pfahringer, Hilan Bensusan, and Christophe Giraud-Carrier. Meta-learning by landmarking various learning algorithms. In *International Conference on Machine Learning*, pp. 743–750, 2000.

Xiangfei Qiu, Jilin Hu, Lekui Zhou, Xingjian Wu, Junyang Du, Buang Zhang, Chenjuan Guo, Aoying Zhou, Christian S. Jensen, Zhenli Sheng, and Bin Yang. TFB: Towards Comprehensive and Fair Benchmarking of Time Series Forecasting Methods. *Proceedings of the VLDB Endowment*, 17 (9):2363–2377, 2024. doi: 10.14778/3665844.3665863. URL https://www.vldb.org/pvldb/vol17/p2363-hu.pdf.

Abdulwahed Salam and Abdelaaziz El Hibaoui. Comparison of machine learning algorithms for the power consumption prediction: Case study of Tetouan city. In *2018 6th International Renewable and Sustainable Energy Conference (IRSEC)*, pp. 1–5. IEEE, 2018a. doi: 10.1109/IRSEC.2018.8703007.

Abdulwahed Salam and Abdelaaziz El Hibaoui. Power Consumption of Tetouan City. Dataset, UCI Machine Learning Repository, 2018b. URL https://archive.ics.uci.edu/dataset/849/power+consumption+of+tetouan+city. doi: 10.24432/C5B034.

David Salinas and Nick Erickson. TabRepo: A large scale repository of tabular model evaluations and its AutoML applications. In *Proceedings of the Third International Conference on Automated Machine Learning*, volume 256 of *Proceedings of Machine Learning Research*, pp. 19/1–30, 2024. URL https://proceedings.mlr.press/v256/salinas24a.html.

Michael Scheuerer and Thomas M. Hamill. Variogram-based proper scoring rules for probabilistic forecasts of multivariate quantities. *Monthly Weather Review*, 143(4):1321–1334, 2015.

Zezhi Shao, Fei Wang, Yongjun Xu, Wei Wei, Chengqing Yu, Zhao Zhang, Di Yao, Tao Sun, Guangyin Jin, Xin Cao, Gao Cong, Christian S. Jensen, and Xueqi Cheng. Exploring Progress in Multivariate Time Series Forecasting: Comprehensive Benchmarking and Heterogeneity Analysis. *IEEE Transactions on Knowledge and Data Engineering*, 37(1):291–305, 2025. doi: 10.1109/TKDE.2024.3484454. URL https://zezhishao.github.io/assets/pdf/2024_TKDE_BasicTS.pdf.

Shokichi Takakura and Taiji Suzuki. Approximation and estimation ability of transformers for sequence-to-sequence functions with infinite dimensional input. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 33416–33447. PMLR, 2023.

Thiyanga S. Talagala, Feng Li, and Yanfei Kang. FFORMPP: Feature-based forecast model performance prediction. *International Journal of Forecasting*, 38(3):920–943, 2022. doi: 10.1016/j.ijforecast.2021.07.002. URL https://arxiv.org/abs/1908.11500.

Thiyanga S. Talagala, Rob J. Hyndman, and George Athanasopoulos. Meta-learning how to forecast time series. *Journal of Forecasting*, 42(6):1476–1501, 2023. doi: 10.1002/for.2963.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. Pruning neural networks without any data by iteratively conserving synaptic flow. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6377–6389, 2020.

Patara Trirat and Jae-Gil Lee. TFAS: Zero-shot NAS for general time-series analysis with time-frequency aware scoring. *Machine Learning*, 114(10):223, 2025. doi: 10.1007/s10994-025-06832-y.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SkgsACVKPH.

Fei Wang, Yujie Li, Zezhi Shao, Chengqing Yu, Yisong Fu, Zhulin An, Yongjun Xu, and Xueqi Cheng. ARIES: Relation Assessment and Model Recommendation for Deep Time Series Forecasting. *arXiv preprint arXiv:2509.06060*, 2025a. URL https://arxiv.org/html/2509.06060.

Shiyu Wang, Haixu Wu, Xiaoming Shi, Tengge Hu, Huakun Luo, Lintao Ma, James Y. Zhang, and Jun Zhou. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/a7ac8a21e5a27e7ab31a5f42a0117bdb-Abstract-Conference.html.

Zihan Wang, Fanheng Kong, Shi Feng, Ming Wang, Xiaocui Yang, Han Zhao, Daling Wang, and Yifei Zhang. Is Mamba effective for time series forecasting? *Neurocomputing*, 619:129178, 2025b. doi: 10.1016/j.neucom.2024.129178.

Satosi Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, 1960. doi: 10.1147/rd.41.0066.

Brian D. Williamson, Peter B. Gilbert, Noah R. Simon, and Marco Carone. A General Framework for Inference on Algorithm-Agnostic Variable Importance. *Journal of the American Statistical Association*, 118(543):1645–1658, 2023.

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

Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A Theory of Usable Information under Computational Constraints. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/html/2002.10689.

Zhijian Xu, Ailing Zeng, and Qiang Xu. FITS: Modeling time series with $10k$ parameters. In *International Conference on Learning Representations*, 2024. URL https://www.iclr.cc/virtual/2024/poster/18329.

Kaichao You, Yong Liu, Jianmin Wang, and Mingsheng Long. LogME: Practical Assessment of Pre-trained Models for Transfer Learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 12133–12143, 2021. URL https://proceedings.mlr.press/v139/you21b.html.

Annan Yu, Danielle Maddix, Boran Han, Xiyuan Zhang, Abdul Fatir Ansari, Oleksandr Shchur, Christos Faloutsos, Andrew Gordon Wilson, Michael W. Mahoney, and Bernie Wang. Understanding the Implicit Biases of Design Choices for Time Series Foundation Models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/753d9584b57ba01a10482f1ea7734a89-Abstract-Conference.html.

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. URL https://ojs.aaai.org/index.php/AAAI/article/view/26317.

Shuyi Zhang, Bin Guo, Anlan Dong, Jing He, Ziping Xu, and Song Xi Chen. Cautionary tales on air-quality improvement in Beijing. *Proceedings of the Royal Society A*, 473(2205):20170457, 2017. doi: 10.1098/rspa.2017.0457.

Yue Zhao, Ryan A. Rossi, and Leman Akoglu. Automatic unsupervised outlier model selection. In *Advances in Neural Information Processing Systems*, volume 34, pp. 4489–4502, 2021.

Zhiyuan Zhao, Juntong Ni, Shangqing Xu, Haoxin Liu, Wei Jin, and B. Aditya Prakash. TimeRecipe: A Time-Series Forecasting Recipe via Benchmarking Module Level Effectiveness. In *International Conference on Learning Representations*, 2026. URL https://iclr.cc/virtual/2026/poster/10010822.

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.

Overview. Appendices A and B give the proofs and structural extensions of Section 4. Appendix C specifies the reference measurements and score; Appendix D documents the experimental protocol. Appendix E follows the empirical questions in Section 6: controlled changes, recommendation quality, and the information driving the rankings. Appendix F extends the restriction-cost analysis to distributional forecasts.

A Proofs of the Main Results

This appendix follows the order of Section 4. Risks refer to a common target, observation, and squared loss within each comparison. All predictors used in an L^2 projection are square integrable. We write $\mu = \mathbb{E}[Y \mid O]$ and $\mu_Z = \mathbb{E}[\mu \mid Z]$ for an interface Z measurable with respect to O. Expectations over a trained predictor refer to its training randomness and an independent evaluation observation unless stated otherwise. The reference measurements in Appendix C are empirical contrasts, whose interpretation is discussed separately.

A.1 Information at the readout

The first identity separates information lost at an interface from error in its readout. The corollary then isolates the value of making an additional input available.

Proof of Proposition 1. The decomposition

$$Y - f(Z) = (Y - \mu) + (\mu - \mu_Z) + (\mu_Z - f(Z))$$

is orthogonal in L^2. Indeed, $\mathbb{E}[Y - \mu \mid O] = 0$, while the last two terms are measurable with respect to O; also $\mathbb{E}[\mu - \mu_Z \mid Z] = 0$ and $\mu_Z - f(Z)$ is measurable with respect to Z. Taking squared norms and expectations, and subtracting $R_B = \mathbb{E}\|Y - \mu\|^2$, proves Eq. (1). □

Corollary 6 (Value of an additional input)**.** *If Z_1 is a function of Z_2, then $D(Z_1) - D(Z_2) = \mathbb{E}\|\mathbb{E}[\mu \mid Z_2] - \mathbb{E}[\mu \mid Z_1]\|^2$. In particular, adding an available input W to Z reduces the minimum risk by $\mathbb{E}\|\mathbb{E}[\mu \mid Z, W] - \mathbb{E}[\mu \mid Z]\|^2$.*

Proof of Corollary 6. For Z_1 measurable with respect to Z_2, the tower property gives $\mu_{Z_1} = \mathbb{E}[\mu_{Z_2} \mid Z_1]$. Thus

$$\mu - \mu_{Z_1} = (\mu - \mu_{Z_2}) + (\mu_{Z_2} - \mu_{Z_1})$$

is an orthogonal sum. Subtracting the squared norms yields the claim. The additional-input comparison sets $Z_1 = Z$ and $Z_2 = (Z, W)$, with W available in O. □

Paths with trainable interfaces. For admissible encoders ϕ and readout classes $\mathcal{G}_\phi$, taking infima in the identity gives

$$\inf_{\phi, f \in \mathcal{G}_\phi} \mathcal{R}(f \circ \phi) - R_B = \inf_{\phi} \left\{ D(\phi(O)) + \inf_{f \in \mathcal{G}_\phi} \mathbb{E}\|\mu_{\phi(O)} - f(\phi(O))\|^2 \right\}. \tag{8}$$

The inner term may contain readout approximation error even before a training procedure is specified. For fixed ϕ and unrestricted square-integrable readouts it vanishes.

Nested stages. If $O = Z_0, Z_1, \ldots, Z_K$ satisfy $\sigma(Z_k) \subseteq \sigma(Z_{k-1})$, repeated application of the corollary gives

$$D(Z_K) = \sum_{k=1}^{K} \mathbb{E}\|\mu_{Z_{k-1}} - \mu_{Z_k}\|^2.$$

Every bypass needed later must be retained in the earlier interface for this nesting to hold. This is an accounting identity for a specified path, not a sum of independently measured architectural effects.

A.2 SHARED PREDICTION RULES

We first compute the cost of imposing a common linear rule. The subsequent decomposition distinguishes this population class cost from the risk difference between fitted predictors.

Proof of Proposition 2. The common covariance and conditional-mean assumptions imply that the risk above the weighted Bayes risk for coefficient w is $\sum_j \pi_j \|\beta_j - w\|_\Sigma^2$. Since $\sum_j \pi_j(\beta_j - \bar{\beta}) = 0$,

$$\sum_j \pi_j \|\beta_j - w\|_\Sigma^2 = \sum_j \pi_j \|\beta_j - \bar{\beta}\|_\Sigma^2 + \|w - \bar{\beta}\|_\Sigma^2. \tag{9}$$

The last term is minimized by $w = \bar{\beta}$, proving the claim. Uniqueness is not required when Σ is singular. □

For coefficients fitted independently of evaluation inputs, the expected shared-minus-specific risk is Eq. (2) plus

$$\mathbb{E}\|\hat{\beta}_{\text{shared}} - \bar{\beta}\|_\Sigma^2 - \sum_j \pi_j \mathbb{E}\|\hat{\beta}_j - \beta_j\|_\Sigma^2.$$

This expression identifies the estimation difference; it does not assert that sharing always reduces it.

Proposition 7 (Expected fitted contrasts)**.** *For two predictor classes evaluated on the same distribution, let $\mathcal{R}_i^* = \inf_{f \in \mathcal{F}_i} \mathcal{R}(f)$ and $\delta_i = \mathbb{E}\mathcal{R}(\hat{f}_i) - \mathcal{R}_i^*$, with finite terms and $\hat{f}_i \in \mathcal{F}_i$. Then*

$$\mathbb{E}[\mathcal{R}(\hat{f}_0) - \mathcal{R}(\hat{f}_1)] = (\mathcal{R}_0^* - \mathcal{R}_1^*) + \delta_0 - \delta_1.$$

If $\mathcal{F}_0 \subseteq \mathcal{F}_1$, the class-risk difference is nonnegative.

Proof. Add and subtract the two class risks. Nesting is needed for their ordering, not for the identity. □

An empirical held-out contrast additionally has evaluation variability. Shifted-input controls also change the input–target alignment and are not automatically a nested-class comparison; see Appendix C.

A.3 SHARED SUMMARIES AND QUERY ACCESS

Gaussian projection turns the loss of a linear summary into a spectral calculation. We use it first for a summary shared across targets, then for an encoder that receives only part of a query.

Gaussian projection. For $U \sim N(0, I_d)$, a linear summary with row-space projector P generates the same information as PU. Thus $\mathbb{E}[U \mid PU] = PU$, and an unrestricted readout for target mean AU has excess risk

$$\mathbb{E}\|A(I-P)U\|^2 = \operatorname{tr}((I-P)A^\top A).$$

Minimizing this expression over rank-r projectors retains the leading r eigenvectors of $A^\top A$.

Proof of Proposition 3. For a shared projector, the weighted excess risk is $\operatorname{tr}((I-P)\sum_j \pi_j K_j)$, with minimum $\operatorname{tail}_r(\sum_j \pi_j K_j)$. Separate projectors yield $\sum_j \pi_j \operatorname{tail}_r(K_j)$. Let $f_j(P) = \operatorname{tr}((I-P)K_j)$ and $v_j = \min_P f_j(P)$. Every shared projector satisfies $\sum_j \pi_j f_j(P) \geq \sum_j \pi_j v_j$. Equality holds at a common minimizer. Conversely, a shared minimizer exists by compactness of the rank-r projector set. Equality of the minimum values and $\pi_j > 0$ force $f_j(P) - v_j = 0$ for every channel. □

For two equally weighted unit directions a, b at angle $0 \leq \theta \leq \pi/2$, the eigenvalues of $(aa^\top + bb^\top)/2$ on their span are $(1 \pm \cos\theta)/2$. The rank-one shared excess is $(1 - \cos\theta)/2$, whereas each channel's own rank-one summary is sufficient.

Query information available before compression. Let $\eta(Q)$ denote the query information available to the encoder: a constant represents no access, while $\eta(Q) = Q$ gives full access. The readout receives the full query in either case.

Proposition 8 (Compression before a query)**.** *Let $U \sim N(0, I_d)$ be independent of a finite query Q, and let the target mean be $A_Q U$. For each value k of $\eta(Q)$ with positive probability, the encoder uses a fixed linear map $B_k U$ of rank at most r_k, where $0 \le r_k \le d$. The readout receives both the summary and Q. Then the minimum excess risk is*

$$\sum_k P(\eta(Q) = k)\, \mathrm{tail}_{r_k}\left(\mathbb{E}[A_Q^\top A_Q \mid \eta(Q) = k]\right). \tag{10}$$

Proof. Let P_k project onto the row space of B_k. Conditional on Q, the optimal readout is $A_Q P_k U$. Averaging its error conditional on $\eta(Q) = k$ gives $\mathrm{tr}((I - P_k)K_k)$ with $K_k = \mathbb{E}[A_Q^\top A_Q \mid \eta(Q) = k]$. Maximizing $\mathrm{tr}(P_k K_k)$ retains the leading r_k eigenvectors, independently for each k, giving the stated sum. □

Coordinate query. For a uniform coordinate query, $A_Q = e_Q^\top$. Without query access at the encoder, $K = I_d/d$ and the minimum cost is $1 - r/d$. With access to Q, $B_Q U = U_Q$ suffices in one scalar. Both readouts know Q.

A common summary across blocks. The same projection argument also applies when a single summary map is reused across independent input blocks. For independent Gaussian blocks $X_b \sim N(0, \Sigma_0)$ with $\Sigma_0 \succ 0$ and target mean $\sum_b A_b X_b$, one common rank-r linear map applied separately to each block has cost

$$\mathrm{tail}_r\left(\sum_b \Sigma_0^{1/2} A_b^\top A_b \Sigma_0^{1/2}\right),$$

provided the readout receives every block summary. Whiten each block and apply the same trace minimization. Additional aggregation of the summaries can introduce a further loss.

A.4 Retaining history

This example compares a finite window of innovations with the full innovation history. The omitted state gives an explicit information cost before any predictor is trained.

Proposition 9 (Window over a fading state)**.** *Consider the stationary solution of $S_t = aS_{t-1} + \xi_t$, where $0 < a < 1$ and the innovations are iid, centered, and have variance σ_ξ^2. Let $\mu = S_t$, with the full innovation history $\xi_{\le t}$ as the reference observation and $W_r = (\xi_t, \ldots, \xi_{t-r+1})$. Then*

$$q_{\mathrm{win}}(r) = \frac{\sigma_\xi^2 a^{2r}}{1 - a^2}. \tag{11}$$

Proof of Proposition 9. The stationary solution is $S_t = \sum_{j \ge 0} a^j \xi_{t-j}$ in L^2. The observed window determines $c_r = \sum_{j=0}^{r-1} a^j \xi_{t-j}$, while

$$S_t = c_r + a^r S_{t-r}.$$

The residual state is independent of the window, centered, and has variance $\sigma_\xi^2/(1 - a^2)$. Hence $\mathbb{E}[S_t \mid W_r] = c_r$ and $D(W_r) = a^{2r}\sigma_\xi^2/(1 - a^2)$. □

A recurrence reproduces S_t when initialized at the corresponding state and run with the correct coefficient and innovations. A recurrence started from an arbitrary finite-time state has an additional initialization error. Extensions to nonlinear readouts and bounded gated states are given in Appendix B.

A.5 CONTEXT-DEPENDENT COEFFICIENTS

The calculation below isolates an interaction restriction: the context can add an offset, but cannot change the coefficients applied to the input.

Proof of Proposition 4. Independence and centering give $\mathbb{E}[X \mid c] = 0$ and $\mathbb{E}[XX^\top \mid c] = \sigma^2 I$. For any finite-risk predictor in the stated class,

$$\mathbb{E}[(\beta(c)^\top X - w^\top X - b(c))^2] = \sigma^2 \mathbb{E}\|\beta(c) - w\|^2 + \mathbb{E}b(c)^2.$$

The minimum is attained at $b = 0$, $w = \mathbb{E}\beta(c)$, with value $\sigma^2 \sum_k \mathrm{Var}(\beta_k(c))$. □

If $\beta(c)$ selects coordinate k with probability p_k, this value is $\sigma^2(1 - \sum_k p_k^2)$. The comparison concerns the stated additive, fixed-coefficient predictor class; a nonlinear readout or an additional interaction path can change that class.

A.6 RIDGE RECOVERY AND THE FINITE-SAMPLE COMPARISON

We begin with a general fixed-design risk identity, specialize it to the orthogonal design in Section 4, and compare the restricted and full fits. The final lemma varies the covariance within a matched training/test design family.

General fixed-design identity. Let β and X be fixed, $\mathbb{E}[\epsilon \mid X, \beta] = 0$, and $\mathrm{Cov}(\epsilon \mid X, \beta) = \Omega$. For $\lambda > 0$, write $K_\lambda = (X^\top X + \lambda I)^{-1}$. At an independent centered test input with covariance Σ_Q, the expected signal prediction error is

$$\lambda^2 \beta^\top K_\lambda \Sigma_Q K_\lambda \beta + \mathrm{tr}(\Sigma_Q K_\lambda X^\top \Omega X K_\lambda). \tag{12}$$

This follows by substituting $\hat{\beta} - \beta = -\lambda K_\lambda \beta + K_\lambda X^\top \epsilon$ and using the zero conditional noise mean. Independent response noise adds its variance to the test risk.

Proof of Proposition 5. Conditioning on the fixed design, the ridge estimator satisfies

$$\hat{\beta}_\lambda - \beta = -\frac{\lambda}{n+\lambda}\beta + \frac{X^\top \epsilon}{n+\lambda}.$$

The independent test input has covariance I_p, so its expected signal prediction error is

$$\mathbb{E}\|\hat{\beta}_\lambda - \beta\|^2 = \frac{\lambda^2 \gamma + \sigma^2 np}{(n+\lambda)^2}.$$

Differentiation gives $2n(\lambda\gamma - \sigma^2 p)/(n+\lambda)^3$, hence the minimum at $\lambda^* = \sigma^2 p/\gamma$. The minimum is $\gamma\nu/(\gamma+\nu)$; subtracting it from the zero predictor's signal error γ gives $r_\nu(\gamma)$. The calculation also holds after averaging over any coefficient distribution independent of training noise with $\mathbb{E}\|\beta\|^2 = \gamma$ and a penalty fixed across coefficient draws. The separate isotropy assumption in Lemma 11 is used when comparing non-orthogonal designs. □

Corollary 10 (When restricting inputs helps ridge prediction)**.** *Let $X_1 = [X_0, X_a]$ satisfy $X_1^\top X_1 = nI_{p_1}$, with $1 \le p_0 < p_1 \le n$, and let $y = X_0\beta_0 + X_a\beta_a + \epsilon$, where $\|\beta_0\|^2 = \gamma > 0$ and $\|\beta_a\|^2 = g \ge 0$. Use the same training-noise model and an independent test pair following this linear model, with centered input covariance I_{p_1} and independent centered noise of variance σ^2. Fit scalar-penalty ridge on X_0 and X_1, each at its expected-risk-minimizing penalty. With $\nu_i = \sigma^2 p_i/n$, their expected trained-risk difference is given by Eq. (4). The restricted fit has lower expected risk exactly when the reduction in estimation cost exceeds its class-risk cost g.*

Proof of Corollary 10. Since $X_0^\top X_a = 0$, the restricted ridge estimator is

$$\hat{\beta}_{0,\lambda} = \frac{n}{n+\lambda}\beta_0 + \frac{X_0^\top \epsilon}{n+\lambda}.$$

The omitted signal contributes no coefficient-estimation noise under this fixed orthogonal design. For an independent centered test input $x = (x_0, x_a)$ with covariance I_{p_1}, the omitted signal contributes

risk g, and its cross term with the retained-coordinate estimation error is zero. Including independent test noise of variance σ^2, the two expected risks are

$$\bar{R}_0 = \sigma^2 + g + \frac{\gamma\nu_0}{\gamma+\nu_0}, \qquad \bar{R}_1 = \sigma^2 + \frac{(\gamma+g)\nu_1}{\gamma+g+\nu_1}.$$

Their penalties are respectively $\sigma^2 p_0/\gamma$ and $\sigma^2 p_1/(\gamma+g)$. Subtracting yields Eq. (4). The class risks are σ^2+g and σ^2. For fixed p_0, p_1, the estimation terms vanish as $n \to \infty$.

For $g = 0$ and $p_0 < p_1$, the restricted fit has strictly lower expected risk for finite n. For $g > 0$, multiplying by the positive denominators shows that the gap is negative exactly when

$$n < \frac{\sigma^2\{\gamma^2 p_1 - (\gamma+g)^2 p_0\}}{\gamma(\gamma+g)g}, \qquad n \geq p_1.$$

The upper bound must exceed p_1 for the restricted-win regime to be nonempty. For $\gamma = \sigma^2 = 1$, $g = 0.1$, $p_0 = 2$, and $p_1 = 8$, the bound is approximately 50.73: the gap is about -0.22381 at $n = 8$ and 0.01806 at $n = 64$. □

Lemma 11 (Isotropic covariance maximizes the optimized risk in the matched design setting)**.** *Let $1 \leq p \leq n$, $X^\top X = n\Sigma$, $\Sigma \succeq 0$, and $\operatorname{tr}\Sigma = p$. Suppose the independent test input has covariance Σ, the random coefficient has second moment $\mathbb{E}\beta\beta^\top = (\gamma/p)I_p$, and training noise is centered, independent of β, with covariance $\sigma^2 I_n$, where $\gamma, \sigma^2 > 0$. Among scalar ridge penalties, the minimum expected signal prediction error is at most $\gamma\nu/(\gamma+\nu)$, where $\nu = \sigma^2 p/n$. Equality holds at $\Sigma = I_p$.*

Proof. For eigenvalues κ_j of Σ, averaging Eq. (12) over β gives

$$\sum_j \frac{(\gamma/p)\lambda^2\kappa_j + \sigma^2 n\kappa_j^2}{(n\kappa_j+\lambda)^2}.$$

Every nonzero-eigenvalue summand is minimized at $\lambda = \sigma^2 p/\gamma$. At that penalty the risk is $\sum_j f(\kappa_j)$ with $f(x) = \sigma^2 x/(nx+\lambda)$, a concave function on $[0,\infty)$. Since $p^{-1}\sum_j \kappa_j = 1$, Jensen's inequality gives $\sum_j f(\kappa_j) \leq p f(1) = \gamma\nu/(\gamma+\nu)$. The isotropic design attains equality. □

The comparison fixes the relation between the training Gram matrix and test covariance and averages over an isotropic coefficient distribution. It is not a worst-case prediction bound for arbitrary forecasters or a guarantee for the proxy dimensions used in Section 5.3.

A.7 Mathematical tools and scope

Conditional expectation provides the L^2 projection identity (Banerjee et al., 2005); reduced-rank projection provides the spectral tail calculations (Eckart & Young, 1936; Izenman, 1975). Fading memory motivates the history examples (Boyd & Chua, 1985). The fixed-design ridge calculations above are derived directly; related random-design and asymptotic analyses appear in Dicker (2016); Dobriban & Wager (2018). Table 2 records the assumptions and role of each result.

B Further Structural Results

The following results extend the restriction-cost analysis to nonlinear targets, alternative readouts, finite storage, and log loss. Each comparison specifies the information, computation, or storage constraint whose cost it isolates.

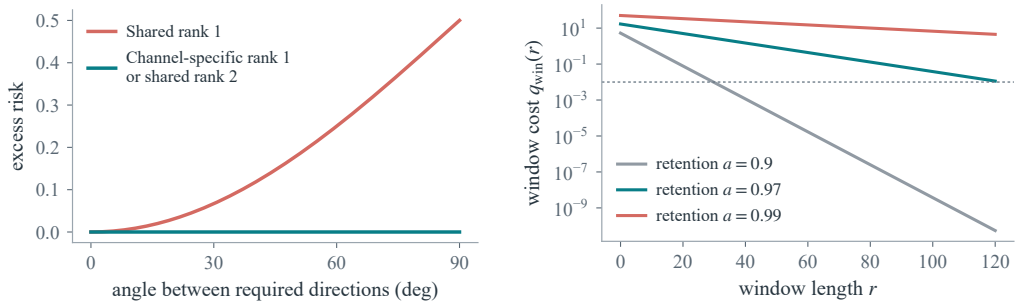
B.1 History extensions and local gated updates

We first ask how a nonlinear target changes the finite-window cost, then give a related bound for a state updated by local gates. For the stationary state of Proposition 9, suppose $\mu = g(S_t)$ and $\ell|x-y| \leq |g(x)-g(y)| \leq L_g|x-y|$ on its support. Then

$$\ell^2 \frac{\sigma_\xi^2 a^{2r}}{1-a^2} \leq D(W_r) \leq L_g^2 \frac{\sigma_\xi^2 a^{2r}}{1-a^2}. \tag{13}$$

Table 2: Scope of the results supporting Section 4. Measurement links refer to the construction of reference contrasts, not equality with their finite-sample values.

Result	Setting and mathematical tool	Role
Prop. 1, Cor. 6	Square-integrable targets; conditional expectation	Interface loss and conditional input value
Prop. 2	Linear means; common covariance	Shared-rule class cost
Prop. 3, Prop. 8	Gaussian inputs; linear summaries; spectral projection	Shared summaries and query timing
Prop. 9	Stationary scalar state; observed innovations	Finite-window cost
Prop. 4	Independent context and centered inputs	Fixed-coefficient interaction cost
Prop. 5, Cor. 10	Fixed orthogonal design; oracle scalar ridge	Recovery and finite-sample risk comparison
Lemma 11	Matched Gram/test covariance; isotropic coefficients	Covariance comparison within the stated ridge family



(a) Sharing a linear summary

(b) A window over a fading state

Figure 5: Exact costs in two illustrative settings. (a) Shared rank-one summaries for equally weighted unit directions at angle $\theta \le 90°$: the excess is $(1 - \cos\theta)/2$, while channel-specific rank-one summaries have zero excess. (b) A length-r innovation window in the stationary state model with $\sigma_\xi^2 = 1$: cost $a^{2r}/(1-a^2)$; dotted line: tolerance 10^{-2}.

Proof. Let S, S' be conditionally independent copies of S_t given W_r. Conditional variance gives $D(W_r) = \frac{1}{2}\mathbb{E}(g(S) - g(S'))^2$. The proof of Proposition 9 gives $\frac{1}{2}\mathbb{E}(S - S')^2 = a^{2r}\sigma_\xi^2/(1-a^2)$. Apply the two Lipschitz inequalities. □

Thus cost at most $\varepsilon > 0$ requires $r \ge \log[\ell^2\sigma_\xi^2/((1-a^2)\varepsilon)]/[2\log(1/a)]$ when this bound is positive.

Receptive-field consequence. A causal convolutional stack with kernel size k, dilations $1, \delta, \ldots, \delta^{m-1}$, and a readout restricted to its final position has receptive-field span at most $\rho_m = 1 + (k-1)\sum_{j=0}^{m-1}\delta^j$. If no other path brings earlier inputs to that readout, its risk is bounded below by $R_B + D(W_{\rho_m})$.

Local gated updates. Let $h_t = a(x_t)h_{t-1} + (1 - a(x_t))v(x_t)$, with $a(x_t) \in [0,1]$, $|v(x_t)| \le 1$, and $|h_0| \le 1$. Let the target mean be $\mu = \psi(h_t)$ with an L_ψ-Lipschitz readout. For $W_r = (x_{t-r+1}, \ldots, x_t)$ and $r \le t$,

$$D(W_r) \le L_\psi^2 \mathbb{E} \prod_{j=t-r+1}^{t} a(x_j)^2.$$

Proof. All states lie in $[-1, 1]$. Resetting the state at $t - r$ to zero gives a predictor measurable with respect to W_r, because both gate and update value use only the current input. The state difference at t is $h_{t-r}\prod_j a(x_j)$, of squared magnitude at most $\prod_j a(x_j)^2$. Apply the Lipschitz bound and minimize over all W_r-measurable predictors. □

This locality is essential: gates or update values computed from an unobserved earlier history need not yield a window-measurable reset predictor.

Numerical stability. The preceding bounds concern missing history. A separate calculation controls error in computing a retained state: a state map contractive by $\tau < 1$ and per-step numerical error at most ε satisfies $e_t \le \tau e_{t-1} + \varepsilon$, hence $e_t \le \tau^t e_0 + \varepsilon(1-\tau^t)/(1-\tau)$.

B.2 Nonlinear targets through a linear interface

A linear encoder can feed a nonlinear readout. For quadratic Gaussian targets, the resulting interface cost has an exact expression and a spectral surrogate with a factor-two guarantee. Let $V \sim N(0, I_d)$ and $\mu_h = a_h^\top V + V^\top B_h V - \operatorname{tr} B_h$, with symmetric B_h. A rank-r projector P and an unrestricted readout have interface cost

$$\Phi(P) = \sum_h \|(I-P)a_h\|^2 + 2\sum_h (\|B_h\|_F^2 - \|PB_hP\|_F^2).$$

With $K = \sum_h (a_h a_h^\top + 2B_h^2)$ and $\ell(P) = \operatorname{tr}((I-P)K)$, one has $\ell(P) \le \Phi(P) \le 2\ell(P)$. Consequently, a leading rank-r eigenspace of K has cost at most twice the minimum over rank-r linear interfaces.

Proof. Write $V = PV + (I-P)V$, whose components are independent. The conditional mean given PV is $a_h^\top PV + V^\top PB_hPV - \operatorname{tr}(PB_hP)$. Gaussian linear and centered quadratic polynomials are orthogonal, with variances $\|a\|^2$ and $2\|B\|_F^2$, respectively, giving the exact cost by orthogonal projection. In coordinates adapted to P, write $B_h = \left(\begin{smallmatrix} B_{11} & B_{12} \\ B_{12}^\top & B_{22} \end{smallmatrix}\right)$. Its contributions to Φ and ℓ are $4\|B_{12}\|_F^2 + 2\|B_{22}\|_F^2$ and $2\|B_{12}\|_F^2 + 2\|B_{22}\|_F^2$. Adding the identical nonnegative linear contributions proves the bounds. For a minimizer P_ℓ of ℓ and a minimizer P_* of Φ, $\Phi(P_\ell) \le 2\ell(P_\ell) \le 2\ell(P_*) \le 2\Phi(P_*)$. □

B.3 Combining inputs and restricting the readout

Making another input available and retaining its information after aggregation are distinct operations. Their contributions separate directly in the interface identity. For $T = F(Z_1, Z_2)$, define $I_{\mathrm{add}} = D(Z_1) - D(Z_1, Z_2)$ and $L_{\mathrm{agg}} = D(T) - D(Z_1, Z_2)$. Both are nonnegative by Corollary 6, and subtraction gives

$$D(Z_1) - D(T) = I_{\mathrm{add}} - L_{\mathrm{agg}}.$$

The combined representation improves on Z_1 exactly when the information retained from the additional input exceeds the subsequent aggregation loss. For independent uniform signs U, V and mean UV, either input alone has cost one and the pair has cost zero.

Additive readouts. Even when both inputs reach the readout, restricting how they can interact can leave a positive cost. Let $\mathcal{A}$ be the L^2 closure of square-integrable additive predictors on specified input regions. The infimum excess risk in that class is $\|\mu - \Pi_{\mathcal{A}}\mu\|_{L^2}^2$. For standard Gaussian U, V of correlation ρ and $\mu = UV$, take

$$u = UV - \rho - \frac{\rho}{1+\rho^2}(U^2 + V^2 - 2).$$

The conditional Gaussian moments give $\mathbb{E}[u \mid U] = \mathbb{E}[u \mid V] = 0$, so u is orthogonal to the additive class. Since $UV - u$ is additive, it is the projection. Using $\operatorname{Var}(UV) = 1+\rho^2$, $\operatorname{Cov}(UV, U^2+V^2) = 4\rho$, and $\operatorname{Var}(U^2+V^2) = 4(1+\rho^2)$ yields cost $(1-\rho^2)^2/(1+\rho^2)$.

The readout can change the preferred summary. Let Q be uniform on $\{1, 2\}$ and independent of $V \sim N(0, I_2)$, with target mean $A_Q V$, where

$$A_1 = \begin{pmatrix} \sqrt{2} & \sqrt{3/2} \\ 0 & 0 \end{pmatrix}, \qquad A_2 = \begin{pmatrix} 0 & \sqrt{3/2} \\ \sqrt{2} & 0 \end{pmatrix}.$$

Compare only the summaries V_1 and V_2, with a readout that knows Q but must output in one common k-dimensional subspace. Total signal variance is $7/2$. Retaining V_1 gives conditional-mean

covariance I_2; retaining V_2 gives $\mathrm{diag}(3/2, 0)$. Retained signal after a rank-k output projection is the sum of the leading k eigenvalues. Hence at $k = 1$ the costs are $5/2$ and 2, whereas at $k = 2$ they are $3/2$ and 2. This compares the two specified coordinate summaries, not all rank-one encoders.

B.4 Finite storage and query access

Here the resource constraint is the number of stored bits rather than the rank of a real-valued summary. The first construction gives an achievable squared-error bound; the second gives an exact optimum for a discrete log-loss problem. Let a history $\mathcal{H}$ be independent of a query $J \sim \pi$, and let the scalar target mean have an L^2 expansion $\mu(\mathcal{H}, J) = \sum_l a_l(\mathcal{H}) b_l(J)$ in an orthonormal basis of $L^2(\pi)$. Suppose $|a_l| \le A_l$. Uniform midpoint quantization of each of the first r coefficients with $B_l \ge 0$ integer bits has squared prediction error at most

$$\tau_r^2 + \sum_{l=1}^{r} A_l^2 2^{-2B_l}, \qquad \tau_r^2 = \mathbb{E} \sum_{l>r} a_l^2.$$

Proof. Each of the 2^{B_l} equal bins on $[-A_l, A_l]$ has half-width $A_l 2^{-B_l}$. Independence of the query and Parseval's identity separate the coefficient errors and the omitted tail. Taking expectations proves the bound. This is an achievable storage construction, not a lower bound for every encoding. □

A discrete log-loss comparison. Let independent fair bits $V_1, \dots, V_d$ be encoded into a message S of at most M bits, where $0 \le M \le d$ is an integer, before an independent query $J \sim \pi$ is observed. A decoder receiving (S, J) has minimum expected log loss

$$\left(1 - \sum_{j=1}^{M} \pi_{(j)}\right) \log 2, \qquad \pi_{(1)} \ge \cdots \ge \pi_{(d)}.$$

Proof. The optimal conditional log loss is $H(V_J \mid S, J) = \log 2 - \sum_j \pi_j I(V_j; S)$. Independence of the bits and conditional entropy subadditivity give $\sum_j I(V_j; S) \le I(V; S) \le H(S) \le M \log 2$. Also $0 \le I(V_j; S) \le \log 2$. The weighted sum is therefore at most $(\sum_{j=1}^{M} \pi_{(j)}) \log 2$, attained by storing the M most probable queried bits. □

This result uses discrete log loss and an explicit bit budget, unlike the squared-loss, real-valued linear summaries of Section 4.

B.5 A log-loss counterpart of the interface identity

For a predictive conditional density or mass function $q(Y \mid Z)$, under conditions making the following quantities finite,

$$\mathbb{E}_O \, \mathrm{KL}(P_{Y|O} \| q_{Y|Z}) = I(Y; O \mid Z) + \mathbb{E}_Z \, \mathrm{KL}(P_{Y|Z} \| q_{Y|Z}).$$

Splitting the log likelihood ratio through $P_{Y|Z}$ separates information unavailable at the interface from mismatch of the predictive distribution to the conditional law at that interface. Taking expectations gives the two terms. At the conditional optimum an additional available input W reduces log loss by $I(Y; W \mid Z)$. This need not agree with its value under squared loss. For $Y = B\epsilon$, $B \in \{1, 2\}$, and an independent uniform sign ϵ, observing B leaves the conditional mean zero but reveals the magnitude: its value is zero under squared loss and $H(B)$ under log loss. Distributional forecasts are discussed separately in Appendix F.

C Measurements and the Score

This appendix specifies the finite-sample reference contrasts used in Section 5, then the configuration mapping and score arithmetic. Population class costs, measured contrasts, and assigned score gains are distinct quantities.

C.1 Reference data and fitting

Data panel. For the training-only measurements, the reference routine uses at most the final 17,280 rows of the designated training interval and at most 12 channels, selected by rounded evenly spaced channel indices with duplicates removed. All references for a dataset, history L, and horizon H use the same channel panel. The dataset's training-scale normalization and input imputation are applied before reference construction; these preprocessing choices do not use the test split. They need not be fitted separately within each reference fold.

Folds. Let n be the number of valid origins in the retained segment, numbered $0, \dots, n-1$, and $k = \lfloor n/2 \rfloor$. The two fitting/evaluation index pairs are

$$([0, k-L-H), [k, n)), \qquad ([k+L+H, n), [0, k)).$$

Thus the fitting side is purged next to the fold boundary. The second fold fits on the later block and evaluates on the earlier block; this is a within-training-split diagnostic, not a forward-only deployment evaluation. Separation reduces overlap but does not establish independence of the time-series observations.

Fitting and evaluation. Within each fitting fold, each input column is centered and scaled by its fitting mean and standard deviation plus 10^{-6}. For target coordinate h, let I_h be the rows where that target is observed, $n_h = |I_h|$, and $\bar{y}_h$ its fitting mean. With standardized design Z, the fitted coefficient is

$$\hat{w}_h = (Z_{I_h}^\top Z_{I_h} + 10^{-3} n_h I)^{-1} Z_{I_h}^\top (y_{I_h,h} - \bar{y}_h).$$

Predictions restore $\bar{y}_h$. The standardization of inputs is fitted on all fitting rows; target missingness selects the rows used in each coefficient fit. Evaluation excludes unobserved targets, computes MSE per forecast step over observed origins, and averages the available forecast-step errors. Fold errors are averaged equally.

C.2 Reference contrasts and their interpretation

For each sampled channel, the own reference uses its L lagged values. The shared reference pools the sampled channels' own-history training examples and fits one coefficient matrix. The aligned reference appends the most recent $\min(24, L)$ values of the other sampled channels. Its shifted control rolls the rows of the complete external-history design matrix by half the row count, separately within the fitting and evaluation sets. Own-history inputs and targets stay fixed. This preserves the empirical multiset of external-history rows and their dimension while changing alignment with the target and own history.

$$s = R_{\text{own}}, \qquad h = R_{\text{shared}} - s, \qquad e = \text{median}_c \left(R_{\text{shifted},c} - R_{\text{aligned},c} \right). \tag{14}$$

Here s and R_{shared} average over sampled channels and both folds. For e, differences are first averaged across folds within each channel, then aggregated by the channel median. If no other channel is available, its external contrast is set to zero. With $J > 1$ sampled channels, each fold uses J own fits, one shared fit, and $2J$ external fits, totaling $3J + 1$; the single-channel case needs no external fits.

Interpretation. The sharing contrast includes finite-sample fitting effects as well as differences between the reference classes. The external contrast measures the benefit of aligned external histories relative to the shifted control within these linear references. A circular shift does not generally produce an input independent of the target conditional on own history; persistent or periodic dependence and changes in the joint input distribution can affect the contrast. Consequently, Corollary 6 motivates the comparison but does not identify its finite-sample value with a Bayes-risk difference. The history contrast $s(L) - s(L')$ likewise compares fitted references, and can be negative.

Summary diagnostics. The analysis suggests a further diagnostic $q_r = R_{\text{rank-}r} - R_{\text{own}}$, using a shared rank-$r$ linear summary followed by channel-specific references. Its population analogue in the Gaussian setting is the shared-summary tail cost in Proposition 3. This diagnostic separates summary restrictions from shared prediction rules. The reported measurement routine computes only s, h, e, and q_r does not enter the score.

C.3 NOMINAL SAMPLE COUNT

With full training interval length T_{tr}, $n_w = T_{\mathrm{tr}} - H - L + 1$ is the nominal number of forecast windows and $N = Cn_w$ counts all native channels. This count is distinct from the smaller reference segment and sampled channel panel. Overlapping windows and cross-channel dependence mean that N is not an asserted number of independent observations.

C.4 CONFIGURATION MAPPING AND SCORE ARITHMETIC

Configuration indicators assign gains $g_e = a_e \min(e_+, s)$, $g_h = a_h \min(h_+, s)$, and $g_p = a_p \min((-h)_+, s)$. With $G = \sum_j g_j$, set $\tilde{g}_j = g_j \min(1, s/G)$ for $G > 0$ and zero otherwise. The recovery scale is $\nu = sp/N$, and $r_\nu(g) = g^2/(g + \nu)$, with $r_0(0) = 0$.

The score uses the unrescaled G in its requirement term and rescaled gains only in its recovery term:

$$Q_m = s - \tfrac{1}{2}G - \tfrac{1}{2}\sum_j r_\nu(\tilde{g}_j) + \tfrac{1}{2}D_{\mathrm{ro}}.$$

Interpretation and limiting behavior. The two quantities G and $\sum_j \tilde{g}_j$ can differ. In particular, $s - G$ can be negative and is a ranking-score component rather than a nonnegative attainable risk. For fixed gains and dimensions as $N \to \infty$, the score tends to $s - \frac{1}{2}G - \frac{1}{2}\min(G, s)$, because $D_{\mathrm{ro}} \to 0$ and recovery tends to $\sum_j \tilde{g}_j$. This equals $s - G$ only when $G \le s$.

The cap and joint rescaling are modeling choices for combining measured gains, not a proof that those gains represent disjoint predictable signals. The fixed-design ridge results motivate the recovery function; substituting a reference error scale, a configuration proxy p, and nominal N does not transfer their exact-risk guarantees to nonlinear candidates.

Readout adjustment. For configurations where a flattened time-step readout is compared with patch aggregation,

$$D_{\mathrm{ro}} = s\left(\frac{p_{\mathrm{flat}}}{p_{\mathrm{flat}} + N} - \frac{p_{\mathrm{patch}}}{p_{\mathrm{patch}} + N}\right). \tag{15}$$

The assigned dimensions are $p_{\mathrm{flat}} = Ld$ and $p_{\mathrm{patch}} = (\text{number of patches})d$; both are zero when the adjustment does not apply. The functional form follows the orthogonal ridge expression with both signal energy and noise scale set to s. These substitutions define a proxy adjustment, not the measured excess risk of the candidate's readout.

Mapping scope. Table 3 specifies the assignments used by the score. Indicators can coexist at different parts of a model, and p is not a count of all trainable parameters. Properties not distinguished by these assignments can still affect test error. The mapping therefore does not imply that configurations with the same score are equivalent predictor classes. Architecture-specific assignments and the fixed weights are retained from the controlled-synthetic design and subsequent evaluation protocol; the protocol and sensitivity checks are given in Appendices D.7 and E.6.

Table 3: Score assignments from the configuration: assigned gains (g_e external, g_h channel-specific, g_p shared rule), fitted dimension p (width d, C channels), and readout dimensions $(p_{\text{flat}}, p_{\text{patch}})$ where they apply. Time-step cores with cross-channel access also receive g_e; TimeMixer uses three resolutions.

Architecture	Gains	Dimension proxy and readout
DLinear, NLinear	g_p	$p = L$
FITS	g_p	$p = 2 \times$ cut-off frequency
N-HiTS	g_p	$p =$ stacks $\times\, d$
PatchTST	g_p	$p =$ patches $\times\, d$
TSMixer	g_e, g_p	$p = L$
iTransformer, S-Mamba, variable tokens	g_e, g_p	$p = Ld + d$
ModernTCN	g_e, g_h, g_p	$p = (L/4)\, d$
TimeMixer	g_p	$p = (L + L/2 + L/4)\, d$
Channel-specific linear map (controlled study)	g_h	$p = LC$
Time-step core, last-token readout	g_p	$p = d$
Time-step core, flattened readout	g_p	$p = Ld$; $(p_{\text{flat}}, p_{\text{patch}}) = (Ld, \text{patches} \times d)$

D Experimental Design and Reproducibility

D.1 Overview of the studies

The controlled experiments on cores and token layouts test whether measured requirements order the test gains of single structural changes and what the fitted dimension and the core add (predictions (i)–(v); Section 6.2; Appendix D.3). Ten published forecasters on 14 datasets evaluate the score's recommendations before training (Sections 6.3–6.4; search in Appendix D.4). Synthetic series with planted relation requirements test prediction (v) (Appendix E.2), and joint forecasts test the extension to distributional forecasts (Appendix F).

All studies use chronological splits and select configurations on the tuning split. Test results average three seeds, with two seeds for some cells of the controlled experiments (Appendix D.3).

D.2 Data

The real-data studies use eight standard benchmark series and six UCI series (Table 4). ETT, Weather, Electricity, and Traffic are the standard long-term forecasting files (Zhou et al., 2021; Wu et al., 2021), and Exchange is the series of Lai et al. (2018). The UCI series are Appliances (Candanedo, 2017; Candanedo et al., 2017), Beijing Multi-Site Air Quality, $PM_{2.5}$ at 12 stations (Chen, 2017; Zhang et al., 2017), Metro Interstate Traffic Volume (Hogue, 2019), Tetouan power consumption (Salam & El Hibaoui, 2018b;a), Air Quality (De Vito, 2008; De Vito et al., 2008), and Individual Household Electric Power Consumption (Hebrail & Berard, 2006), aggregated to hourly values.

ETT uses the standard 12/4/4-month training, validation, and test split, and the other series use a 70/10/20 split. The first three quarters of the validation span form the tuning split; the last quarter is a development block held out from every reported result. Targets lie entirely inside one split, and histories may reach into the preceding split.

Channels are standardized with training statistics; missing inputs are forward-filled causally, with the training mean for leading gaps, and missing targets are excluded from evaluation. AirQuality treats the -200 sentinel as missing and drops channels with at least half their entries missing; the other files, including Weather, are used as released. Inputs are the histories of the observed channels, with no calendar features or future covariates.

Table 4: Datasets: channels C, sampling interval, time steps in the training, tuning, and test splits, and domain.

Dataset	C	Interval	Train / tune / test	Domain
ETTh1, ETTh2	7	1 h	8,640 / 2,160 / 2,880	electricity transformer
ETTm1, ETTm2	7	15 min	34,560 / 8,640 / 11,520	electricity transformer
Weather	21	10 min	36,887 / 3,952 / 10,539	meteorology
Exchange	8	1 day	5,311 / 570 / 1,517	exchange rates
Electricity	321	1 h	18,412 / 1,974 / 5,260	electricity consumption
Traffic	862	1 h	12,280 / 1,317 / 3,508	road occupancy
Appliances	26	10 min	13,814 / 1,480 / 3,947	building energy
Beijing Multi-Site	12	1 h	24,544 / 2,631 / 7,012	air quality, 12 stations
Metro Traffic	5	1 h	36,785 / 3,942 / 10,510	traffic volume, weather
Tetouan	8	10 min	36,691 / 3,931 / 10,483	power consumption, weather
AirQuality	12	1 h	6,549 / 702 / 1,871	air-quality sensors
Household Power	7	1 h	24,212 / 2,595 / 6,917	household power

D.3 Controlled experiments

The controlled experiments test whether a measured requirement orders the test gain of the single structural change that accesses it, and what the fitted dimension and the core add (Section 6.2).

Core study. The core study trains a linear map and five nonlinear cores (MLP mixer, dilated convolution, linear state space, selective state space, attention) on time-step tokens with a last-token readout and reversible instance normalization (Kim et al., 2022), at $L = H = 96$ on 12 real datasets (all except AirQuality and Household Power). Each change relaxes or tightens one element: cross-channel access, channel-specific linear rules, instance normalization, the input length ($L \in \{192, 336, 720\}$), or the horizon ($H \in \{192, 336, 720\}$); exchanging the core at fixed tokens, readout, access, and sharing tests prediction (v). The input-length and horizon analyses use eight datasets (the four ETT datasets, Weather, Electricity, Traffic, and Exchange), and the input-length analysis of Figure 3(c) uses the linear map, the MLP mixer, the selective state space, and attention.

Token-layout study. The token-layout study replaces time-step tokens by patch tokens (length 16, stride 8) or variable tokens for the five nonlinear cores, which alters both the assigned gains and the fitted dimension. It repeats several of the changes above on its own configurations and adds a readout change, which replaces the flattened readout of patch tokens by the last patch for the selective state space and attention and so alters only the fitted dimension (prediction (iv)); its horizon changes use ETTh1, ETTm1, and Weather.

Search and refitting. Each core is searched over twelve configurations ($d \in \{32, 64, 128\}$, lr in $[10^{-5}, 10^{-2}]$, including the default); all are trained for 2,048 updates, the best three and one independent draw for 8,192, and the best for 16,384 (32,768 on Electricity and Traffic). Token layouts use 24 configurations ($d \in \{64, 128, 256\}$, $\ell \in \{2, 3, 4\}$), of which the best five continue.

Each changed configuration is re-tuned under its parent's update budget, around the parent's two best configurations with learning-rate multipliers 0.5, 1, and 2 and two fresh draws. Weight decay is zero with probability 0.2 and otherwise log-uniform in $[10^{-6}, 10^{-2}]$, and $p_{\mathrm{dr}} \in \{0, .05, .1, .2\}$. Each core's default configuration, used by the proxies, is one of the searched configurations and is fixed before the search. The selected configuration of each cell is trained with three seeds, or two for some cells of the changes.

D.4 Published forecasters

The ten published forecasters are the candidates that the score, the proxies, and the selectors rank, and each ranking is evaluated against the candidates' test errors after search (Sections 6.3–6.4). They span the five families of Table 5, linear, MLP, convolutional, Transformer, and state-space forecasters, and differ in cross-channel access, channel sharing, and fitted dimension, the restrictions that the score reads.

DLinear, NLinear, FITS, PatchTST, iTransformer, ModernTCN, and TimeMixer run their official PyTorch code. TSMixer is a PyTorch port of the official Keras model, N-HiTS uses the NeuralForecast blocks[1], and S-Mamba uses a PyTorch implementation of the selective scan (Gu & Dao, 2024). Table 5 lists the search spaces and the default configurations.

The staged search adds the default to eleven random configurations. All twelve are trained for 2,048 updates, the best four by tuning MSE for 8,192, and the best with seeds 101–103 for 16,384 (32,768 on Electricity and Traffic), using AdamW (Loshchilov & Hutter, 2019) with batches of 32 windows (16 for S-Mamba on Traffic). Each seed keeps the checkpoint with the lowest tuning MSE.

Table 5: Search spaces and default configurations of the published forecasters. d, d_{ff}: width and feed-forward width, in powers of two over the stated range; ℓ: layers or blocks; p_{dr}: dropout; –: none. Learning rates (lr) are log-uniform; weight decay is zero with probability 0.2 and otherwise log-uniform in $[10^{-7}, 3 \cdot 10^{-2}]$; other entries are uniform over the listed values. The default, which the score and the proxies evaluate, is one of the twelve searched configurations: the official script for the dataset at $H = 96$, or the listed values with lr 10^{-3} and no weight decay.

Family	Forecaster	lr	Architecture choices	Default
Linear	DLinear	$[10^{-4}, 5 \cdot 10^{-2}]$	moving average 25	official
	NLinear	$[10^{-5}, 10^{-2}]$	–	–
	FITS	$[10^{-5}, 10^{-2}]$	cut-off frequency $\{4, 8, 12, 24, 40, 48\}$	cut-off frequency 24
MLP	N-HiTS	$[10^{-5}, 10^{-2}]$	$d \in [64, 512]$, $\ell \in \{1, 2\}$, pooling $\{221, 421, 841\}$, $p_{\text{dr}} \in \{0, .1, .2\}$	d=512, ℓ=1, pooling 221
	TSMixer	$[10^{-5}, 10^{-2}]$	$d_{\text{ff}} \in [16, 512]$, $\ell \in \{2, 4, 6, 8\}$, $p_{\text{dr}} \in \{.1, .3, .5, .7, .9\}$	official
	TimeMixer	$[10^{-5}, 10^{-2}]$	$d \in [16, 64]$, $\ell \in \{1, 2\}$, scales $\{1, 2, 3\}$, $p_{\text{dr}} \in \{0, .1, .2\}$	d=32, ℓ=2, 2 scales, p_{dr}=.1
Convolution	ModernTCN	$[10^{-5}, 10^{-2}]$	$d \in [16, 64]$, $\ell \in \{1, 2\}$, kernel $\{15, 31, 51\}$, $p_{\text{dr}} \in \{0, .1, .2\}$	d=32, ℓ=1, kernel 31, p_{dr}=.1
Transformer	PatchTST	$[3 \cdot 10^{-5}, 3 \cdot 10^{-3}]$	$d \in [16, 256]$, $\ell \in \{1, 2, 3\}$, patch $\{8, 16, 24\}$, $p_{\text{dr}} \in \{0, .1, .2, .3\}$	official
	iTransformer	$[3 \cdot 10^{-5}, 3 \cdot 10^{-3}]$	$d \in [32, 512]$, $\ell \in \{1, \ldots, 4\}$, $p_{\text{dr}} \in \{0, .1, .2\}$	official
State space	S-Mamba	$[3 \cdot 10^{-5}, 3 \cdot 10^{-3}]$	$d \in [32, 512]$, $\ell \in \{1, \ldots, 4\}$, state $\{2, 8, 16, 32\}$, $p_{\text{dr}} \in \{0, .1, .2\}$	official

D.5 Comparison methods

Two families of methods rank architectures before training on a new dataset, and both serve as comparisons (Section 6.3; Table 6). Training-free proxies score each network at initialization and use the same information as the score, the new dataset's training series and each candidate's default configuration. Selectors from algorithm selection learn a ranking from every candidate's test results on the other datasets. The proxies cover gradient, pruning, Jacobian, kernel, and size criteria, and the selectors cover average ranking, portfolio construction, classification, combination weights, factorization, and pairwise ranking.

Training-free proxies. Each proxy scores a candidate at initialization, with the forecasting loss where it needs one; it is oriented as in its original paper and averaged over three initializations on the same training windows.

Selectors that learn from results on other datasets. For each target dataset, a selector learns from the test results of the ten candidates on the other datasets, excluding the ETT sibling of an ETT target, so from 12 or 13 datasets. Each dataset is described by the channel mean and standard deviation of the 22 catch22 features (Lubba et al., 2019) and by the length, the number of channels,

[1] https://github.com/Nixtla/neuralforecast

the lag-one autocorrelation, and the mean absolute cross-correlation, computed on the training split and standardized over the training datasets. Table 6 specifies the implementations and settings, including the factorization and pairwise-ranking adaptations. A second variant appends $\log s$, e/s, h/s, and the logarithms of the sample size and the number of channels to these features, to test whether the measured requirements also inform the selectors (Appendix E.3).

Table 6: Comparison methods: quantity or ranking rule, with inputs or settings in parentheses (4×32: four batches of 32 training windows). The proxies use every trainable parameter. Relative error: test error divided by the best test error on the dataset, minus one.

Method	Quantity or ranking rule (inputs or settings)
Training-free proxies	
GradNorm, SNIP	gradient norm; connection sensitivity $\lvert\theta \odot \nabla\ell\rvert$ (4×32 windows)
ZiCo	mean-to-standard-deviation ratio of per-batch gradients (4×32 windows)
GraSP	Hessian–gradient product at initialization (1×8 windows)
Jacobian covariance (JacobCov)	correlation of input–output Jacobians (1×8 windows)
NTK condition	log condition number of the ridge-regularized, trace-normalized kernel of the forecast sum (1×8 windows)
SynFlow	product of absolute parameters along paths, double precision (all-ones input)
Parameter count	number of trainable parameters
Selectors that learn from results on other datasets	
Single best (Bischl et al., 2016)	mean test rank on the training datasets
Portfolio (Salinas & Erickson, 2024)	order of greedy addition to the set with the lowest mean best-in-set relative error (ties broken by mean rank)
kNN average rank (Brazdil et al., 2003)	mean test rank on the nearest training datasets (3 neighbors, Euclidean distance)
FFORMS (Talagala et al., 2023)	class probability of a random-forest classifier of the best candidate (500 trees)
FFORMA (Montero-Manso et al., 2020)	weight from gradient boosting that minimizes the softmax-weighted relative error (94 rounds, learning rate 0.58, depth 14, row and column subsampling 0.92 and 0.77)
Factorization (Zhao et al., 2021)	rank matrix reconstructed from dataset factors regressed on the features (rank 3, ridge penalty 1)
ZAP (Öztürk et al., 2022)	pairwise ranking surrogate on the features and a candidate indicator (300 trees, depth 4, learning rate 0.1)

D.6 Metrics

Each method produces a ranking on each dataset. Scores equal to relative precision 10^{-10} are tied. For shortlist metrics, we compute the exact expectation over uniform resolutions of ties: top-k regret averages $100(R_m/\min_{m'} R_{m'} - 1)$ over the k recommended candidates, and best in top k is the probability that the shortlist contains the best candidate. Thus top-k regret corresponds to choosing uniformly from the recommendations, rather than evaluating only their best member.

Spearman and Kendall correlations evaluate the whole ranking and use average ranks for ties; an all-tied ranking contributes zero to the dataset mean. Direction accuracy (pair accuracy) evaluates pairs whose relative test gap $2|R_a - R_b|/(R_a + R_b)$ exceeds 1%, with half credit for tied scores. Metrics are averaged over datasets with equal weight; intervals resample datasets with replacement (5,000 draws). Dataset win/loss counts compare values at the precision shown in the tables, with equal values counted as neither.

D.7 Protocol, stored results, and compute

Each split serves one role in every study: the requirement score and training-free proxies use the training split and candidate configurations, configurations are selected on the tuning split, and the test split is evaluated once. The cross-dataset selectors additionally use results from other datasets as specified in Appendix D.5. The reference fits and the score, including its weights, caps, sample size,

and fitted-dimension rule, are set using controlled synthetic experiments with planted requirements, separate from the reported studies, and held unchanged for every real dataset.

Every stored score, configuration, checkpoint, and test result is recorded with its SHA-256; evaluation scripts recompute every reported metric from the stored scores and per-seed test errors, and a validation script reproduces every stored requirement score exactly. Training used NVIDIA RTX A6000, RTX 3090, and RTX 5090 GPUs; with $J > 1$ sampled channels, the requirement score needs $2(3J + 1)$ ridge fits per dataset on a CPU.

E Additional Results

We first examine controlled structural changes and planted relation requirements, then report the complete recommendation comparisons. The final analyses separate the contributions of dataset-specific measurements, configuration assignments, and fixed score choices.

E.1 Single structural changes

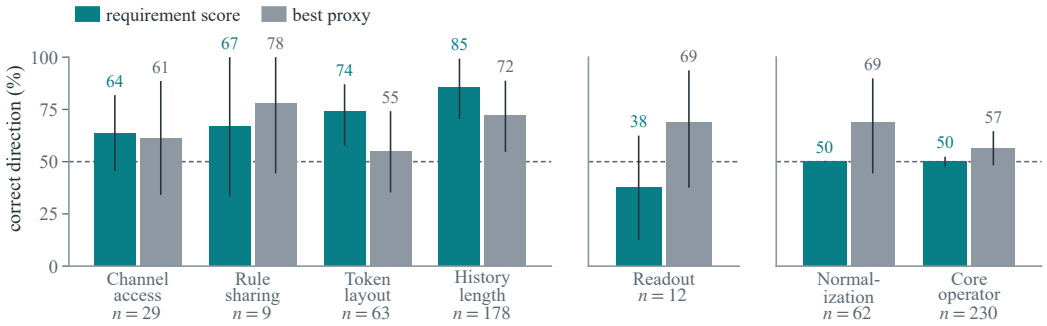
Measured requirements and test gains. Figure 3(b) tests prediction (i) with the external requirement as the score uses it, $\min(e_+, s)/s$: it orders the dataset-level gains from cross-channel access with Spearman 0.40 (-0.24–0.78) over 12 datasets, and access lowers test error on five of the six datasets with $e > 0$. The measured e depends on the control that shifts the alignment of the other channels with the target, so a variant e' uses a second control: a circular shift within each block by a random multiple of 24 steps no smaller than 480, on at most ten channels and on the 11 datasets other than Weather. It orders the same gains with Spearman 0.58 (-0.18–0.97) over these datasets and 0.75 (0.11–0.88) over the 32 dataset–core cases of six cores. Both correlations have the predicted positive sign, although both dataset-level intervals include zero.

Reference risk along the horizon. We test whether changes in the reference level s track changes in forecasting difficulty as the horizon grows. Across the horizon changes of the core study, the increase of the reference risk orders the increase of test error (Spearman 0.84, 95% dataset-bootstrap interval 0.67–0.92, 138 dataset–core–horizon cases on eight datasets). On the 72 horizon pairs of the token-layout study, the score predicts the direction of 93% and SNIP of all.

Direction of each change. Each single structural change forms a pair with its parent configuration, and a score that ranks configurations should predict which of the two has the lower test error. Figure 6 groups the pairs of both studies by the kind of change and compares the score with the training-free proxies of Figure 4 at their default configurations. On changes that alter the assigned gains or the history length (panel a), the score is correct on 64% (cross-channel access), 74% (token layout), and 85% (history length) of the pairs, against at most 61%, 55%, and 72% for a proxy; on the nine rule-sharing pairs, three proxies reach 78% against 67% for the score.

The 12 readout pairs (panel b) alter only the fitted dimension and so test prediction (iv) alone. The flattened readout, which has the larger fitted dimension, has the lower test error on eight of them; the score, which ties six, is correct on 38%, below chance, and three proxies on 69%.

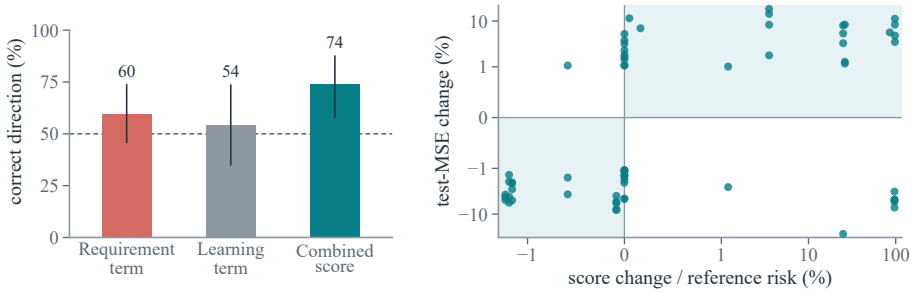
Configurations with equal access, sharing, fitted dimension, and readout share a score (prediction (v); panel c), so the score ties every normalization pair and 92% of the core exchanges at a fixed interface, on which the Jacobian covariance reaches 69% and 57%. Exchanging the core changes test error by 1.7% at the median over all 348 core exchanges, and Appendix E.2 plants the relation requirements that separate cores at a fixed interface.



(a) Assigned gains and history length (b) Fitted dimension (c) Relations, normalization

Figure 6: Correct direction (%) of the requirement score and, for each kind of change, the most accurate of the four training-free proxies of Figure 4, on pairs from single structural changes of both controlled studies whose test errors differ by more than 1% (n: number of pairs; horizon pairs are reported in the text). Values are averaged within and then across datasets, with 95% dataset-bootstrap intervals; ties earn half credit; dashed line: chance.

The two terms of the score. Token-layout changes alter both the assigned gains and the fitted dimension, so they show what the learning term, the score's finite-sample approximation to the learning cost, adds to the requirement term (Section 5.3). On these 63 pairs, the full score predicts more directions than either term alone (Figure 7(a); Section 6.2). Across all single structural changes other than the horizon, most of which move only one of the two terms, the requirement term, the learning term, and the full score are correct on 61%, 49%, and 63%: the requirement term carries most of the accuracy, and where both terms move, the learning term adds to it.



(a) Token-layout changes ($n = 63$) (b) Full score, individual pairs

Figure 7: The two terms of the score on the 63 token-layout pairs whose test errors differ by more than 1%. (a) Correct direction (%) of the requirement term, the learning term, and the full score, averaged within and then across datasets; ties earn half credit; dashed line: chance. (b) Predicted change of the full score relative to the reference risk against the observed change of test MSE, one point per pair (symmetric log axes).

E.2 Planted requirements

At a fixed interface the score ties the cores. To examine relation requirements beyond its inputs (prediction (v)), we plant two mechanisms motivated by Propositions 4 and 8, each with a null control. Two additional laws, a long window and an observed regime, test requirements that a configuration can meet through its history and inputs (Table 7). These are empirical comparisons of trained cores, rather than direct tests of the propositions' exact risk formulas.

Each core is searched jointly over history, tokenization, readout, width, and optimizer: 97 configurations, of which 24 continue after 4,096 updates and 6 after 16,384 to a final 32,768, with weight decay and dropout as in Appendix D.3.

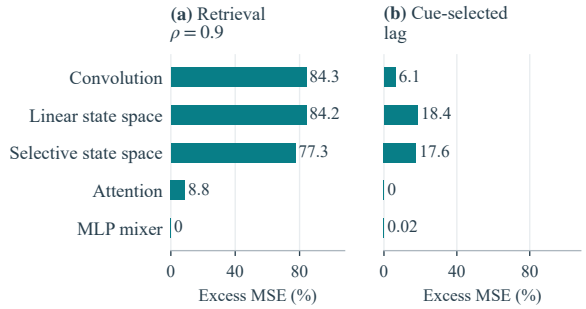


Figure 8: Test MSE above the lowest mean in each planted setting (%, three seeds; 0 marks the lowest): (a) retrieval at $\rho = 0.9$; (b) cue-selected lag. Both panels share a linear scale.

Core	Retrieval (null)	Long window	Observed regime
Convolution	0	0.036	0
Linear state space	0.003	0.097	0.172
Selective state space	0.025	0.403	0.271
Attention	0.022	0	0.560
MLP mixer	0.007	0.195	0.327

Table 8: The same quantity in the null retrieval setting and the two control laws.

Table 7: Synthetic laws and their null controls; innovations are independent standard Gaussians. The experiments use retrieval at $\rho \in \{0, 0.9\}$, cue-selected lag at strength 0.4, window strength $b = 0.3$, and regime persistence 0.97, all with $H = 24$.

Requirement	Generating dependence	Null
Retrieval	in 240-step episodes, a hidden period $P \in \{30, \dots, 50\}$ sets $x_t = \rho x_{t-P} + \sqrt{1-\rho^2}\epsilon_t$	$\rho = 0$
Cue-selected lag	$y_t = x_{t-\tau_t} + 0.3\epsilon_t$, $\tau_t \in \{24, 28, \dots, 44\}$ chosen by an observed cue	fixed lag
Long window	$x_t = 0.5x_{t-1} + bx_{t-48} + \epsilon_t$	$b = 0$
Observed regime	$x_t = a_{c_t}x_{t-1} + \sqrt{1-a_{c_t}^2}\epsilon_t$ with an observed Markov regime c_t	equal a_c

Figure 8 reports the two planted relation requirements, and Table 8 the null retrieval and the two control laws. With retrieval strength $\rho = 0$, the five cores are within 0.03% of one another. At $\rho = 0.9$, the MLP mixer, which mixes all patches before its readout, is best; attention is 8.8% above it, and the convolution and both state-space cores are 77–84% above it. Here the relevant value depends on a period inferred from the window. The result is consistent with the query-timing mechanism of Proposition 8: compression before identifying the relevant value must preserve information for multiple possible queries.

With a cue that selects the lag, both state-space cores are about 18% above the best. Convolution, with fixed coefficients but direct access to the window, is 6.1% above, while attention and the MLP mixer, which combine cue and window, tie. These gaps motivate considering both what a summary retains before the cue and whether the computation can form cue-dependent relations (Propositions 8 and 4). They do not isolate either mechanism on their own.

In the two control laws, every selected configuration covers the 48-step history and receives the regime, and all cores are within 0.6% of one another. Across these laws, the cores are close where the configuration meets the requirement and separate where a relation requirement is planted.

E.3 Comparison with every method

Figure 4 shows four of the eight training-free proxies and two of the seven selectors; Table 9 lists every method with its difference from the score. Among the methods that use only the new dataset, the score has the lowest top-three regret and the highest Spearman correlation, also when any single dataset or the ETT family is left out; the interval of the difference excludes zero for three proxies in regret and five in correlation. The selectors, which also use every candidate's test results on the other datasets, have bootstrap intervals for their differences from the score that include zero, except the portfolio in regret and FFORMS in Spearman correlation.

Table 9: Every comparison method, ten published forecasters on 14 datasets. Top-3: top-three regret (%); ρ: Spearman correlation; Best: best in top three (%); Δ: method minus our score, with 95% dataset-bootstrap interval. Positive Δ top-3 and negative $\Delta\rho$ favor our score. Bold: best and underlined: second best among all methods except uniform choice. Values and differences are rounded separately.

Method	Top-3 ↓	Δ top-3	ρ ↑	Δρ	Best ↑
Requirement score	**5.9**	–	0.43	–	55
Training-free proxies					
JacobCov	7.0	+1.1 [−0.6, +2.7]	0.38	−0.06 [−0.26, +0.14]	36
ZiCo	6.8	+0.9 [−0.6, +2.3]	0.31	−0.12 [−0.36, +0.12]	43
SNIP	6.9	+1.0 [−0.8, +2.9]	0.20	−0.23 [−0.46, −0.01]	36
GradNorm	7.6	+1.7 [+0.3, +3.3]	0.19	−0.24 [−0.47, −0.01]	36
SynFlow	8.0	+2.1 [−0.8, +5.3]	0.11	−0.32 [−0.60, −0.02]	29
Parameter count	7.9	+2.0 [−0.8, +5.0]	0.25	−0.18 [−0.45, +0.08]	36
NTK condition	9.2	+3.3 [+0.7, +6.2]	−0.03	−0.46 [−0.65, −0.24]	43
GraSP	13.2	+7.3 [+1.3, +13.5]	−0.18	−0.61 [−0.93, −0.28]	43
Selectors that learn from results on other datasets					
Single best	<u>6.2</u>	+0.3 [−1.1, +1.9]	0.36	−0.07 [−0.29, +0.19]	<u>57</u>
Portfolio	8.7	+2.8 [+0.5, +5.2]	0.17	−0.26 [−0.48, +0.00]	29
kNN average rank	**5.9**	−0.0 [−1.6, +1.6]	<u>0.45</u>	+0.02 [−0.23, +0.28]	50
FFORMS	7.6	+1.7 [−1.0, +4.9]	0.21	−0.22 [−0.39, −0.03]	21
FFORMA	6.4	+0.5 [−1.7, +3.6]	0.41	−0.02 [−0.22, +0.20]	43
Factorization	6.5	+0.6 [−0.8, +2.2]	0.37	−0.06 [−0.24, +0.13]	**64**
ZAP	6.5	+0.6 [−1.5, +3.3]	**0.47**	+0.04 [−0.14, +0.25]	43
Uniform choice	10.9	+5.0 [+1.8, +8.5]	–	–	30

Shortlist size and whole ranking. For shortlists of $k = 1, 2, 4, 5$ candidates, the mean regret of the score is 5.0, 5.5, 6.2, and 7.5%, against 6.8, 6.4, 6.2, and 6.5% for the best proxy of Figure 4 at each k: the score's lead is largest for one recommendation, vanishes at $k = 4$, and reverses at $k = 5$. Its Kendall correlation with the test errors is 0.36, against at most 0.27 for a proxy.

Dataset weighting and configuration. The four ETT datasets share one source, enter the mean four times, and are easy for every method: uniform choice has 3.9% regret on them and 13.7% elsewhere. Counting the ETT family as one dataset, the score has 6.6% regret and a Spearman correlation of 0.52, against 6.9% for kNN average rank and 0.51 for ZAP, the next methods on each measure. Scored at the configuration selected by search in place of the default, the score gives 6.0% regret and the proxies of Figure 4 change by at most one point.

Requirements as selector features. Appended to the dataset features of the selectors (Appendix D.5), the measured requirements lower the top-three regret of three of the five feature-based selectors, kNN average rank to 4.9%, and raise that of the other two by at most half a point.

E.4 Recommendations on each dataset

A mean over datasets can hide datasets with high regret. Table 10 gives the top-three regret of each method on each dataset. The score is lower on most datasets against the four proxies and uniform choice, and on 4 to 8 of the 14 datasets against the four selectors.

kNN average rank, which has the same mean regret as the score, is lower on nine datasets, by at most 1.3 points on seven of them and by 2.6 and 6.5 points on ETTm2 and Weather; the score is lower on four, by 3.2–6.9 points on AirQuality, Electricity, and Exchange, and ETTm1 is a tie.

The score's largest deficits, 5.2–6.5 points above the lowest regret, are on Weather, Traffic, and Exchange, where parameter count (Traffic, Exchange) or three of the selectors (Weather) are lowest.

Table 10: Top-three regret (%) per dataset (Params: parameter count; kNN: kNN average rank). Bold: lowest and underlined: second lowest among the score and the four proxies of Figure 4. Last row: number of datasets on which the score's regret is lower or higher than that of each column; values equal at the precision shown count as neither.

	New dataset only					Learn from other datasets				
Dataset	Score	ZiCo	JacobCov	SNIP	Params	Single best	kNN	FFORMA	ZAP	Uniform
AirQuality	**0.6**	3.4	4.9	2.6	5.2	0.3	3.8	0.3	0.3	2.0
Appliances	**4.0**	6.0	7.2	12.8	17.7	4.3	2.7	4.3	5.5	14.6
Beijing Multi-Site	15.2	19.3	19.3	18.7	**14.2**	14.3	14.3	15.9	16.0	14.9
ETTh1	2.8	2.9	3.0	**2.6**	3.3	2.3	2.3	2.3	3.2	2.9
ETTh2	2.3	**1.8**	3.0	2.0	9.9	1.3	1.3	1.2	1.2	4.4
ETTm1	**2.1**	2.8	4.2	3.8	3.2	1.2	2.1	1.2	1.2	4.2
ETTm2	5.2	**3.5**	4.9	4.2	5.1	2.1	2.6	2.6	2.6	4.3
Electricity	5.5	7.9	**3.1**	**3.1**	4.9	10.8	9.3	10.8	9.3	18.8
Exchange	12.7	18.7	19.8	19.3	**7.5**	20.1	19.6	9.1	9.1	10.2
Household Power	**0.8**	1.6	0.9	2.7	1.5	1.1	0.9	1.6	1.6	2.3
Metro Traffic	4.8	**4.1**	6.9	5.3	6.5	5.3	4.2	4.2	5.3	11.3
Tetouan	**5.5**	6.8	7.6	6.8	17.5	5.9	4.7	5.9	5.9	23.6
Traffic	13.2	7.1	7.1	7.1	**7.0**	14.1	12.9	29.4	28.0	28.7
Weather	7.7	8.8	**5.6**	**5.6**	6.6	3.4	1.2	1.2	1.2	10.4
Mean	**5.9**	6.8	7.0	6.9	7.9	6.2	5.9	6.4	6.5	10.9
Score lower / higher	–	10/4	10/4	8/6	8/6	7/7	4/9	6/8	8/6	11/3

E.5 Selection with training

Ranking by tuning error. Trained candidates can instead be ranked by their tuning error after a short training run (Table 11, upper part). Training the default configuration of every candidate for 1,024 updates gives 5.1% top-three regret, 2,048 updates give 4.7%, and the full search gives 3.3%, all below the score's 5.9%. The score's Spearman correlation, 0.43, is higher than those of the two short runs, 0.34 and 0.40, and the score trains no candidate. Measuring the requirements takes 2–41 CPU seconds per dataset; by the per-update timings recorded for six candidates, training all ten for 1,024 updates costs at least 3.7 times as much at the median.

Training a shortlist. This comparison uses short training runs to choose a candidate identity, then evaluates that candidate's full-search test error (Table 11, lower part). Each shortlisted candidate is trained for 1,024 updates at its default configuration. The candidate with the lowest tuning error is selected, and regret compares its full-search test error with the best candidate's. The score's top five give 4.0% with half the screening updates of training all ten, which give 3.8%; five random candidates give 5.0%, and the top five of the other methods 3.7–6.1%. The score's top three give 4.9%, against 6.7% for three random candidates.

Narrowing a full search. The score's top three contain the candidate that a full search of all ten selects by tuning error with probability 0.67 (95% interval 0.49–0.83), against 0.30 for three random candidates, and searching only these three gives a lower test regret than searching three random candidates on 10 of the 14 datasets.

Table 11: Selection with training, ten published forecasters on 14 datasets. Upper part: rankings of all ten by the score and by the tuning error after training the default configuration of every candidate for the stated number of updates or after the full search of Appendix D.4. Lower part: each shortlisted candidate is trained for 1,024 updates at its default configuration, and the candidate with the lowest tuning error is selected. Updates count this screening stage only; regret compares the selected candidate's full-search test error with the best candidate's, with 95% dataset-bootstrap intervals. Random: expectation over uniformly drawn candidates.

Ranking all ten	Top-3 regret (%) ↓	Spearman ρ ↑
Requirement score, no training	5.9	0.43
Tuning error, 1,024 updates	5.1	0.34
Tuning error, 2,048 updates	4.7	0.40
Tuning error, full search	3.3	0.67
Training a shortlist	Updates	Regret (%) ↓
All ten	10,240	3.8 [1.4, 6.9]
Score, top five	5,120	4.0 [1.7, 7.0]
Random five	5,120	5.0 [2.9, 7.4]
Score, top three	3,072	4.9 [2.5, 7.7]
Random three	3,072	6.7 [4.2, 9.6]

E.6 What drives the recommendations

How the rankings change across datasets. Table 12 measures how much each ranking changes across datasets. The test-error rankings of two datasets have a mean Kendall correlation of 0.14, and their top threes share one candidate on average, so the best candidates change across datasets. The proxies order candidates largely by size and depth and change less (Kendall correlation 0.39–0.94, or 0.59–0.94 without the NTK condition number; 1.8–3.0 shared candidates), and JacobCov and SynFlow recommend the same three candidates on every dataset. The ranking by gain types and fitted dimension of Table 1 barely changes (0.99) and puts the same candidate first on every dataset.

The score, which follows each dataset's requirements, changes more (0.33; 1.4 shared candidates) and has a higher Spearman correlation with the test errors than any proxy (Table 9); on each dataset it forms 2 to 8 groups of equal score, and its first choice falls on three different candidates across the 14 datasets. Ranking variability alone does not explain recommendation quality: the two proxies that change most, the NTK condition number and GraSP, have the lowest Spearman correlations, and with the requirements permuted at random across datasets the score is worse on both measures on 12 of the 14 datasets (Table 1).

Table 12: How much each ranking changes across datasets (ten published forecasters, 14 datasets, 91 pairs): mean Kendall correlation between the rankings of two datasets, and mean number of candidates that their top threes share (uniform tie resolution; three random candidates share 0.9).

Ranking	Kendall	Shared in top 3
Test errors	0.14	1.0
Full score	0.33	1.4
Gain types and fitted dimension	0.99	2.5
Training-free proxies		
JacobCov	0.90	3.0
ZiCo	0.86	2.2
SNIP	0.65	2.1
GradNorm	0.64	2.2
SynFlow	0.94	3.0
Parameter count	0.84	2.5
NTK condition	0.39	1.8
GraSP	0.59	1.9
Selectors that learn from results on other datasets		
Single best	0.86	3.0
Portfolio	0.73	2.0
kNN average rank	0.47	1.7
FFORMS	0.59	1.5
FFORMA	0.46	1.9
Factorization	0.21	1.1
ZAP	0.35	1.6

Simpler rankings. Table 13 replaces the score by rankings that keep only part of its information. Because the candidates access nested sets of gains, the requirement term of Table 1 ranks them as the count $a_e\mathbf{1}_{e>0} + a_h\mathbf{1}_{h>0} + a_p\mathbf{1}_{h<0}$ of the gains that the measured signs make positive; the external-sign rule keeps only the sign of e, favoring cross-channel access when $e > 0$ and breaking ties by fewer parameters. Both reach a comparable top-three regret, 6.0% (0.14 points above the score, 95% interval -0.53–0.78) and 5.9%, with lower Spearman correlations, 0.33 and 0.24 against 0.43: the count ties all candidates on six datasets, and the learning term orders the candidates that the signs leave tied.

Replacing $e > 0$ by a threshold on the mean absolute training cross-correlation gives 5.9–19.8% over the 15 thresholds between observed values and 14.0% at the median threshold; reaching the score's 5.9% requires choosing the threshold by test regret. The threshold $e = 0$ instead denotes equal error for aligned and shifted external-history references (Appendix C); it does not require calibrating a cross-correlation threshold. Fixing every requirement at its median over the other datasets gives 16.3%, higher than the score on 12 of the 14 datasets.

Fixed choices of the score. Table 14 recomputes the score from the stored measurements with one fixed choice changed at a time. The component weight w on assigned gains ($1 - w$ on ridge-motivated recovery), the readout weight c on D_{ro} (both $\frac{1}{2}$ in Eq. (7)), the variance scale ν, and the caps keep the top-three regret within 5.9–6.1%; $w = 1$ removes the learning term of the gains, as in Table 1.

The nominal count N includes overlapping windows and is not an effective independent-sample count; scaling N by 0.3 to 100 keeps the regret between 5.9% and 6.1%, and counting windows without channels or only non-overlapping windows gives 6.1%. Reassigning the sample size across datasets changes the mean regret by less than 0.1 points. Taking the fitted dimension from the parameter count, its square root, or the count divided by C gives 5.2–5.6% with Spearman correlations of 0.40–0.41.

With circular shifts by $\frac{1}{4}$ or $\frac{3}{4}$ of the block or a blocked permutation as the control for e, the sign of e agrees with the stored value on 12 or 13 of the 14 datasets, and every dataset with $e > 0$ keeps $e > 0$. Treating scores within 1% as tied changes the regret by less than 0.1 points. Setting requirements below 5% of s to zero has the largest effect, 7.1% and 0.27, indicating that discarding small measured contrasts can reduce recommendation quality in this comparison.

Table 13: Simpler rankings: top-three regret (%), Spearman ρ, and best in top three (%).

Ranking	Top-3	ρ	Best
Full score	5.9	0.43	55
Signs of the measured requirements			
Requirement term	6.0	0.33	46
External-sign rule	5.9	0.24	50
Cross-correlation in place of e			
Median threshold	14.0	−0.12	29
Requirements fixed across datasets			
Median of other datasets	16.3	−0.26	33

Table 14: The score with one fixed choice changed; columns as in Table 13, ranges over the listed settings.

Setting	Top-3	ρ	Best
As specified	5.9	0.43	55
Weights and caps			
Weight $w \in [0, 0.9]$	5.9–6.1	0.43–0.44	55
Weight $c \in [0, 1]$	5.9	0.43	55
Variance scale $\nu \times \frac{1}{4}$–4	5.9–6.1	0.43–0.44	55
Gains without caps	5.9	0.43	55
Sample size and fitted dimension			
$N \times 0.3$–100	5.9–6.1	0.43–0.44	55
$N = n_w$, windows only	6.1	0.43	55
N, non-overlapping windows	6.1	0.41–0.42	55
p from parameter count	5.2–5.6	0.40–0.41	50
Measurement and ties			
Other controls for e	5.9–6.4	0.37–0.40	48–57
Requirements $< 0.05\, s$ set to 0	7.1	0.27	52
Scores within 1% tied	5.9	0.35	50

F Extension: Distributional Forecasts

The interface accounting extends to distributional forecasts. There the structure of the output imposes a restriction of its own, and its requirement is the dependence among future coordinates. The point-forecast score of Section 5 does not use this result.

Write the future invertibly as an anchor A and blocks $R_1, \dots, R_B$; a factorized output models the anchor given the observation and each block given the anchor and the observation. Here TC denotes conditional total correlation: the KL divergence of the joint conditional block distribution from the product of its conditional marginals.

Proposition 12 (Factorized futures). *For $q = q_A(A \mid O) \prod_b q_b(R_b \mid A, O)$,*

$$\mathbb{E}_O \operatorname{KL}(P_{Y|O} \| q) = \mathbb{E}\ \operatorname{TC}(R_1, \dots, R_B \mid A, O) + \mathbb{E} \operatorname{KL}(P_{A|O} \| q_A) + \textstyle\sum_b \mathbb{E} \operatorname{KL}(P_{R_b|A,O} \| q_b). \quad (16)$$

Proof. KL divergence is invariant under the invertible measurable map from Y to $(A, R_1, \dots, R_B)$. Writing $p(a, r \mid o) = p(a \mid o) p(r \mid a, o)$,

$$\log \frac{p(a, r \mid o)}{q_A(a \mid o) \prod_b q_b(r_b \mid a, o)} = \log \frac{p(r \mid a, o)}{\prod_b p(r_b \mid a, o)} + \log \frac{p(a \mid o)}{q_A(a \mid o)} + \sum_b \log \frac{p(r_b \mid a, o)}{q_b(r_b \mid a, o)},$$

and taking expectations gives the conditional total correlation (Watanabe, 1960), the anchor KL, and the block KLs. □

The restriction is an output whose blocks are independent given the anchor and the observation, and its requirement is the conditional total correlation, the first term of Eq. (16). This term is the smallest divergence any factorized output attains and remains when the anchor and every block are exact, so joint and marginal scores measure different requirements.

For an ordered factorization in which coordinate i conditions on parents $\mathrm{pa}(i) \subseteq \{1, \dots, i-1\}$, the same argument gives $\sum_i I(Y_i; Y_{<i} \mid Y_{\mathrm{pa}(i)}, O) + \sum_i \mathbb{E} \operatorname{KL}(P_{Y_i|Y_{\mathrm{pa}(i)},O} \| q_i)$; the requirement is then the information each coordinate shares with its earlier coordinates beyond its parents and the observation.

Empirical check. The KL decomposition motivates evaluating joint and marginal quality separately; it does not guarantee improvements in other scoring rules for fitted models. We compare three output constructions on the same MLP-mixer conditioner at $L = H = 96$: independent Gaussian

coordinates, a joint Gaussian head, and block diffusion with 100 steps. The diffusion head predicts Gaussian noise over each channel's full future block using the DDPM objective (Ho et al., 2020) and a cosine noise schedule (Nichol & Dhariwal, 2021). Each is selected by the tuning variogram score and refitted with three seeds, with 64 samples per origin. Energy and variogram scores evaluate the joint forecast, and CRPS evaluates each marginal (Gneiting & Raftery, 2007; Scheuerer & Hamill, 2015).

On the four ETT datasets, Weather, Electricity, Traffic, Exchange, Appliances, and Tetouan, block diffusion lowers the energy score relative to independent Gaussian coordinates on 9 of 10 datasets (mean -9.1%) and the variogram score on 10 of 10 (-28.1%), while marginal CRPS improves on 6 of 10 (-5.3%). On Tetouan and Electricity, the dependence-sensitive scores improve while CRPS worsens slightly, illustrating why both evaluations are needed. The joint Gaussian head improves the energy score on 5 of 10 datasets and raises CRPS by 8.7% on average. In this comparison, block diffusion improves joint quality on more datasets while also reducing mean marginal error.