How Reversible Normalization Changes Linear Forecasting

Anonymous authors
Paper under double-blind review

Abstract

Linear models remain competitive in long-term time series forecasting and can be fit in closed form. They often use reversible instance normalization to address distribution shift between the lookback window and the forecast horizon. We show that this normalization changes the fitted forecaster even when the scale cancels exactly at inference. For a homogeneous head, fitting in normalized coordinates is equivalent to weighted ridge regression. Each squared residual receives a weight equal to the inverse squared context scale, while the mean weight determines the effective ridge penalty. Using the same nominal penalty grid therefore does not isolate the effect of normalization. A learned center can also add one direction to the feature span and change the penalty on the decoded coefficients. We propose an estimator-matched evaluation protocol that controls the attainable coefficient class, relative observation weights, and effective penalty. On long-term forecasting benchmarks, unweighted fitting with a uniform center outperforms instance-normalization weights under a matched effective penalty. This advantage remains after excluding the dataset with the largest improvement. Learning the center provides a smaller gain that disappears with raw lags or long contexts, tying the benefit to the compressed representation. Synthetic experiments show which noise structures make inverse-scale weighting useful.

1 Introduction

Linear predictors have long been used in time series forecasting. Exponential smoothing (Holt, 2004, Winters, 1960, Hyndman et al., 2002) and ARIMA (Box et al., 2015) express forecasts as weighted combinations of past values. Their weights capture trend and seasonal structure without a nonlinear prediction map (Cleveland et al., 1990, Hyndman & Athanasopoulos, 2021). These predictors and their combinations have also performed well in forecasting competitions (Makridakis et al., 2020, Smyl, 2020).

Long-term multivariate benchmarks extend forecasting to much longer horizons (Lai et al., 2018, Zhou et al., 2021). Many neural architectures have been developed for this setting (Salinas et al., 2020, Oreshkin et al., 2020, Wu et al., 2021, Zhou et al., 2022, Challu et al., 2023, Nie et al., 2023). Yet Zeng et al. (2023) showed that a single linear layer applied to the lookback window can match or outperform many of them. Subsequent work has found linear models and shallow MLPs to be competitive as well (Li et al., 2023, Das et al., 2023, Chen et al., 2023, Xu et al., 2024). Linear forecasters also offer a practical advantage for analysis. Their coefficients can be fit in closed form and inspected directly.

Reversible instance normalization is common in these forecasting pipelines. RevIN subtracts the lookback mean, divides by the lookback standard deviation, applies the prediction head, and then reverses the transformation (Kim et al., 2022). Related methods learn the normalization statistics, distinguish input and output distributions, or stationarize the series within the forecaster (Fan et al., 2023, Passalis et al., 2020, Liu et al., 2022, Fan et al., 2025). These methods build on normalization layers in deep learning (Ioffe & Szegedy, 2015, Ulyanov et al., 2016) and aim to address non-stationarity between the lookback window and the forecast horizon. Improvements are usually interpreted as evidence that normalization helps the predictor adapt to changes in scale.

For a linear head, however, the decoded prediction function must be distinguished from the procedure used to fit its coefficients. A scale can cancel from the prediction function and still determine

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.

which observations dominate training. Squared loss in normalized coordinates weights each decoded squared residual by the inverse squared context scale. Unless these weights are normalized, their mean also changes the strength of ridge regularization. Learning the center can change the feature span as well. When these effects vary together, a better score cannot be attributed to scale adaptation alone. Using the same head and nominal penalty grid does not resolve the ambiguity.

We study these effects in a ridge forecaster shared across channels, building on the regression interpretations of Toner & Darlow (2024). Because the head is fit in closed form, the feature map, observation weights, and penalty fully specify the estimator and can be varied separately. The only learned quantities are two distributions over lags, optimized by differentiating through the ridge solver.

The clearest empirical finding concerns the baseline. On a common effective ridge grid, constant weights reduce aggregate error more than the retained learned scales do. We therefore use unweighted fitting with a uniform center as the reference for evaluating normalization weights.

Contributions.

- For a homogeneous head with shared encoding and decoding statistics, we show that fitting in normalized coordinates is weighted regression with an effective penalty. A common *nominal* ridge grid therefore does not isolate normalization.
- We show that a learned center adds at most one direction to a fixed feature map and changes the penalty on the decoded coefficients, even when the function class stays the same.
- We develop an estimator-matched evaluation protocol. On standard benchmarks, unweighted fitting with a uniform center outperforms instance-normalization weights on a common effective grid, both with and without the dataset showing the largest gain. Synthetic controls identify the noise structures that favor inverse-scale weighting.

2 What Changes in the Estimator?

2.1 Setup and decoded predictor

Let $x \in \mathbb{R}^{1\times L}$ denote the context for one channel and $y \in \mathbb{R}^{1\times H}$ its future values. All coordinates are transformed by a scaler fit on the training prefix. The fixed feature map $A \in \mathbb{R}^{L\times F}$ satisfies $\mathbf{1}_L^\top A = \mathbf{1}_F^\top$. Let $p \in \mathbb{R}^L$ be a static center distribution satisfying $p \geq 0$ and $\mathbf{1}_L^\top p = 1$. We write the center as $c = xp$ and take the scale $s(x) > 0$. Features and targets are encoded using the same statistics,

$$z = \frac{xA - c\mathbf{1}_F^\top}{s}, \qquad t = \frac{y - c\mathbf{1}_H^\top}{s}. \tag{1}$$

We fit a homogeneous head zW by minimizing a ridge objective averaged over the fitting rows,

$$W^* = \arg\min_W \frac{1}{n}\sum_{r=1}^{n} \|z_r W - t_r\|_2^2 + \lambda\|W\|_F^2. \tag{2}$$

The loss sums over forecast leads, fixing the convention for λ (Hoerl & Kennard, 1970). Decoding yields

$$\widehat{y} = xB, \qquad B = AW + p\beta, \qquad \beta = \mathbf{1}_H^\top - \mathbf{1}_F^\top W. \tag{3}$$

The scale cancels exactly, including any positive stabilizer, even when it depends on the input. Every column of B sums to one, making the predictor shift equivariant. This identity requires a homogeneous head. An unpenalized intercept in normalized coordinates produces an additional term $s(x)b$ after decoding, while separate output statistics require a different decoder analysis (Appendix A). We use homogeneous heads throughout to exclude the scale-dependent intercept term.

2.2 Scale changes weights and effective regularization

Define $V_p(x) = xA - (xp)\mathbf{1}_F^\top$ and $r_p(x,y) = y - (xp)\mathbf{1}_H^\top$. Substituting Equation (1) into Equation (2) yields

$$\min_W \frac{1}{n}\sum_r w_r \|V_p(x_r)W - r_p(x_r,y_r)\|_2^2 + \lambda\|W\|_F^2, \qquad w_r = s_r^{-2}. \tag{4}$$

The resulting problem is weighted regression in the centered head coordinates, rather than unrestricted ridge regression on B.

Proposition 1 (Loss coordinates and effective penalty)**.** *Fix the center p and the penalty on W. With squared loss in decoded coordinates, the scale disappears from both fitting and prediction. For squared loss in normalized coordinates, write $\bar{w} = n^{-1}\sum_r w_r$ and $\widetilde{w}_r = w_r/\bar{w}$. Equation (4) then has the same minimizer as*

$$\min_W \frac{1}{n}\sum_r \widetilde{w}_r \|V_p(x_r)W - r_p(x_r, y_r)\|_2^2 + \lambda_{\text{eff}}\|W\|_F^2, \qquad \lambda_{\text{eff}} = \frac{\lambda}{\bar{w}}. \tag{5}$$

The first claim follows by decoding the residual. The second follows by dividing the objective by the positive mean weight $\bar{w}$. Appendix A gives both derivations. Learning the scale thus changes the relative observation weights and the effective regularization range at the same time. A common nominal λ grid cannot separate these effects.

A one-point suffix makes the distinction clear. When $c = x_{\text{last}}$ and $s^2 = \epsilon$, every residual receives the same weight, but $\lambda_{\text{eff}} = \epsilon\lambda$. This suffix does not change the relative observation weights. Its selection on an unrescaled grid may reflect a different penalty range as well as a different center. In our controls, we normalize weights to mean one on each fitting subset and search a common effective grid.

2.3 Centering changes the available directions and their penalties

Set $C_p = A - p\mathbf{1}_F^\top$. Equation (3) becomes $B = p\mathbf{1}_H^\top + C_pW$.

Proposition 2 (Center geometry)**.** *For a fixed center p, the attainable coefficient vectors for each forecast lead form the affine set*

$$\mathcal{S}_p = p + \text{col}(C_p) = \{u \in \text{span}(\text{col}(A), p) \mid \mathbf{1}_L^\top u = 1\}, \quad \dim \mathcal{S}_p = \text{rank}([A\ p]) - 1. \tag{6}$$

If P_A projects orthogonally onto $\text{col}(A)$, then

$$(I - P_A)B = ((I - P_A)p)\beta, \tag{7}$$

so the component of the coefficients outside the feature span has rank at most one across forecast leads.

Writing $u = Aw + \alpha p$, the unit-sum constraint gives $\alpha = 1 - \mathbf{1}_F^\top w$. If the uniform center $p_0 = \mathbf{1}_L/L$ belongs to $\text{col}(A)$, a learned center outside that span adds exactly one attainable direction. Optimizing p searches over possible directions, not over a single scalar parameter.

Equation (3) also expresses the predictor as an augmented-feature model on $[xA, xp]$, with coefficients (W, β) constrained by $\mathbf{1}_F^\top W + \beta = \mathbf{1}_H^\top$. Leaving β unpenalized recovers the centered estimator exactly. Penalizing this coefficient leaves the attainable class unchanged but alters the regularized fit. We use this penalized version to check the feature parameterization.

Even when the function class is fixed, the estimator can change. The induced penalty on an attainable B is

$$R_p(B) = \min_{C_pW = B - p\mathbf{1}_H^\top} \|W\|_F^2 = \|C_p^\dagger(B - p\mathbf{1}_H^\top)\|_F^2. \tag{8}$$

For raw lags, $A = I$, all centers yield the same unit-sum class, but their induced penalties take the form

$$R_p(B) = \left\| \left(I - \frac{pp^\top}{\|p\|_2^2}\right) B \right\|_F^2. \tag{9}$$

Raw-lag controls therefore rule out class expansion without fixing the induced regularization. Full derivations appear in Appendix A.

2.4 Matching an estimator, not just a head

Two systems with the same linear head can have different decoded function classes. Even if their classes agree, they may penalize the coefficients differently. And with the same class and penalty,

different observation weights still produce different estimators. Matching the head is therefore only part of a controlled comparison.

For fixed p, rescaling $s_r \mapsto \kappa s_r$ leaves the deployed class unchanged and multiplies all fitting weights by κ^{-2}. If the nominal penalty is held fixed, the effective penalty increases by a factor of κ^2. To preserve the estimator, the penalty must change with the scale,

$$s'_r = \kappa s_r, \qquad \lambda' = \lambda/\kappa^2, \qquad \widetilde{w}'_r = \widetilde{w}_r, \qquad \lambda'_{\text{eff}} = \lambda_{\text{eff}}. \tag{10}$$

A constant change in scale can shift the available regularization range without adding a predictive feature or changing the relative weights. Normalizing the weights to mean one removes this ambiguity. It does not, however, match the relative weights or the penalties induced by different centers.

Raw lags provide another exact reference. For the uniform center $p_0 = \mathbf{1}_L/L$ and any unit-sum B, Equation (9) simplifies to

$$R_{p_0}(B) = \|B\|_F^2 - H/L. \tag{11}$$

The subtracted term is constant over the attainable class. Uniform-centered raw ridge therefore has the same solution as ridge on B subject to unit column sums. A nonuniform p changes the penalty projection but not the class. These identities hold regardless of the training budget. Whether the change produces a useful gain is an empirical question, addressed in Section 4.

3 Controlled Evaluation

A lag-weighted family. We vary the normalization statistics using lag-weighted reversible normalization, which we call LWRN.[1] The static version learns two simplex distributions by differentiating through the ridge solver,

$$p = \mathrm{softmax}(a), \qquad q = \mathrm{softmax}(v), \qquad c = xp, \qquad s^2 = \sum_i q_i (x_i - c)^2 + \epsilon, \tag{12}$$

where $a, v \in \mathbb{R}^L$ are trainable parameters and $\epsilon = 10^{-5}$. Uniform initialization recovers instance normalization (IN). The scale uses a second moment around c, which need not be the variance around the q-weighted mean. For $\mu_q = \sum_i q_i x_i$, we have $s^2 = \mathrm{Var}_q(x) + (c - \mu_q)^2 + \epsilon$. Changing p alone can therefore change both the decoded coefficients and the fitting weights. Constant-scale controls isolate center learning by removing the change in weights. The scale-only control instead fixes $p = p_0$.

Differentiating the fitted estimator. Let θ collect the trainable statistic parameters, and let Z_θ, T_θ contain the encoded inner-fit rows. For each positive penalty, the fitted head solves

$$G_\theta = Z_\theta^\top Z_\theta/n + \lambda I, \qquad K_\theta = Z_\theta^\top T_\theta/n, \qquad G_\theta W_\theta = K_\theta. \tag{13}$$

The outer objective is decoded MSE on the later inner-tune interval. Its gradient includes the dependence of the fitted head on the statistics. Differentiating the linear system gives

$$G_\theta \, dW_\theta = dK_\theta - (dG_\theta) W_\theta. \tag{14}$$

We differentiate through regularized regression rather than jointly optimizing an unconstrained head by gradient descent (Bertinetto et al., 2019). All linear systems are solved without forming matrix inverses. The mean-one controls use the corresponding weighted normal equations, with weights normalized on the fitting subset. Their outer objective remains decoded MSE, so changes to the inner fitting weights are evaluated in the original forecasting coordinates.

Data and fitting. We evaluate on nine datasets, namely ETTh1, ETTh2, ETTm1, ETTm2, Weather, Electricity, Traffic, Solar, and Exchange. We use horizons $H \in \{96, 192, 336, 720\}$ and the standard chronological conventions (Zhou et al., 2021, Wu et al., 2021, Qiu et al., 2024), with context lengths $L \in \{96, 2048\}$. The compressed feature map combines recent lags, older block averages, and seasonal anchors. Raw-lag controls use $A = I$ at $L = 96$. Within each setting, every method uses the same features, channels, and forecast origins.

[1] The released implementation and records use the internal name EWN for this family.

We divide each training prefix chronologically into 70% inner fit and 30% inner tune. Each candidate receives 30 Adam updates through the inner-fit ridge solution (Kingma & Ba, 2015). We select the checkpoint with the lowest decoded inner-tune MSE, allowing the initialization to be selected, and refit the head on training-prefix samples. The official validation set then selects the hyperparameters. The selected model is evaluated on the test set without further refitting. The effective penalty grid is $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-2}, 10^{-1}, 1, 100\}$, and learned models use learning rates $\{0.003, 0.03\}$. All main controls use homogeneous heads and float64 arithmetic.

Inner fit and inner tune are each capped at 20,000 pooled channel windows, and the final fit is capped at 100,000. Sampling retains every channel at deterministic temporal origins, and evaluation includes every eligible origin. We report MSE and MAE in training-standardized coordinates, following standard out-of-sample practice (Tashman, 2000, Hyndman & Koehler, 2006, Godahewa et al., 2021). Appendix B gives the feature definitions, splits, search grids, and the resulting temporal coverage (Table 8).

Estimator-matched controls. U combines a uniform center with a constant scale. C learns the center under the same unweighted loss. IN_1 uses a uniform center and inverse-variance weights normalized to mean one. The conditional scale-only and full LWRN controls keep their previously selected statistics and retune only the heads on the effective grid. In a separate experiment at $H = 96$, we train a gate that interpolates between mean-one learned weights and constant weights. We report this newly trained model separately from the conditional refits.

The feature control F^+ learns xp independently and penalizes both W and β. The original-span control S tunes unit-sum ridge on A, while the restricted control C_{res} removes the added direction at the penalty selected by C. A suffix control T searches mean-one inverse-variance weights on the common effective grid to isolate the normalization component of SearchCast. Two additional temporal splits within the original training prefix assess sensitivity to the fitting window under short compression at $H = 96, 336$.

Aggregation. We summarize error ratios using geometric means over matched dataset–horizon cells. For methods a, b, let ℓ_d denote the mean log error ratio over the horizons for dataset d. The aggregate ratio is

$$\rho(a, b) = \exp\left(\frac{1}{D}\sum_{d=1}^{D} \ell_d\right), \qquad \ell_d = \frac{1}{|\mathcal{H}_d|}\sum_{h\in\mathcal{H}_d} \log \frac{\mathrm{MSE}_{a,d,h}}{\mathrm{MSE}_{b,d,h}}. \tag{15}$$

Each dataset receives equal weight, regardless of its channel count or number of test origins. We express percentage improvement as $100(1 - \rho)$, rather than averaging the percentage changes across cells. The 95% intervals are exact cluster bootstrap intervals obtained by resampling datasets while keeping each dataset's horizons together. We report every aggregate with and without Solar. The absolute-score table instead reports arithmetic mean MSE within each dataset, so the two summaries need not agree numerically.

Interpreting the controls. Table 1 lists the mechanism tested by each comparison. At a fixed center, weighted and unweighted heads have the same attainable class and penalty form. A common effective grid controls for global weight scaling. Comparisons between centers allow both the representation and the induced penalty to change, while raw-lag and restricted-span controls separate the two effects.

Comparison	Matched components	Question addressed
U/IN_1	Center, class, effective grid	Are inverse-scale weights useful?
C/U	Constant weights, feature map	Does adapting the center help?
F^+/C	Attainable class at fixed p	Is the feature interpretation consistent?
C_{res}/C	Selected center and penalty	Is the outside-span direction useful?
Raw C/U	Full unit-sum coefficient class	Is there a gain without class expansion?

Table 1. Controlled comparisons and the mechanisms they test. F^+ and C learn their distributions independently. F^+ also penalizes the feature coefficient. All controls use homogeneous heads.

4 RESULTS

4.1 THE UNWEIGHTED BASELINE CHANGES THE INTERPRETATION

Table 2 compares mean-one weights on a common effective ridge grid. The retained scale-only and full LWRN statistics both outperform IN_1, indicating that their gains are not due solely to the mean weight. But the unweighted, uniform-center model U performs better than either. It reduces aggregate MSE by 4.85%, compared with 1.29% for the retained scale-only statistics, and has lower MSE in 33 of 36 cells. Excluding Solar leaves a 2.57% reduction, with the interval for the error ratio still below one.

Comparison	All datasets	Without Solar
Conditional head refits, $H = 96, 192, 336, 720$		
U/IN_1	0.9515 [0.9030, 0.9858]	0.9743 [0.9562, 0.9898]
Scale-only LWRN/IN_1	0.9871 [0.9702, 0.9980]	0.9951 [0.9908, 0.9989]
Full LWRN/IN_1	0.9760 [0.9564, 0.9921]	0.9841 [0.9709, 0.9945]
Newly trained scale gate, $H = 96$		
Gate/IN_1	0.9451 [0.8929, 0.9836]	0.9694 [0.9468, 0.9889]
Gate/U	1.0006 [0.9998, 1.0015]	1.0003 [0.9997, 1.0009]

Table 2. MSE ratios for compressed features at $L = 96$. Brackets show exact dataset-cluster 95% bootstrap intervals. The conditional refits cover 36 cells, or 32 without Solar. The scale-gate experiment covers nine cells, or eight without Solar. IN_1 and the scale-only control share a uniform center. The full control retains its learned center. Every head is tuned on the same effective penalty grid.

Neither U nor IN_1 learns normalization statistics. Both use a uniform center and tune the head on the common effective grid, leaving the fitting weights as their only difference. This comparison does not depend on the optimization budget discussed in Appendix C. The gate experiment and the center comparisons in Section 4.2 do involve learned parameters. Within those comparisons, all models receive the same budget.

The gate is the only experiment that relearns the weighting rule, whereas the conditional refits keep the previously selected statistics fixed. It uses $\widetilde{w}_{\alpha,r} = (1-\alpha) + \alpha\widetilde{w}_r$, where $\widetilde{w}_r$ has mean one and $\alpha \in (0, 1)$ is learned, so its constant-weight endpoint at $\alpha = 0$ is U. The gate outperforms IN_1 with an MSE ratio of 0.9451, but its ratio to U is 1.0006, with an interval that contains one. Learning α therefore provides no aggregate improvement beyond the unweighted endpoint.

Unweighted fitting has lower geometric-mean MSE than IN_1 on eight of the nine datasets (Table 3). The exception is ETTm2, where the ratio is slightly above one. Solar shows the largest improvement, but the benefit is not confined to that dataset. Table 2 reports the aggregate both with and without it.[2]

Section 4.3 examines when scale weighting helps. Context scale alone is not an estimate of conditional forecast-error variance, and an improvement over IN does not establish that interpretation.

4.2 CENTER LEARNING REVEALS MISSING LAG DIRECTIONS

With short compressed contexts, learning the center under unweighted fitting gives $C/U = 0.9723$, a 2.77% reduction in geometric-mean MSE, and lowers MSE in 32 of 36 cells. Table 4 shows how this gain varies with the representation and the inclusion of Solar. Without Solar, the reduction falls from 2.77% to 1.14%. Removing the added direction yields $C_{\text{res}}/C = 1.0294$. Compared with independently tuned original-span ridge, $C/S = 0.9717$ (Appendix C).

For short compressed contexts, the independently learned feature model F^+ has aggregate MSE within 0.01% of C. This agrees with the feature interpretation in Equation (3). The objectives differ only in the penalty on β, and 31 of the 36 selected C penalties are at most 10^{-4}. We treat the

[2]Solar has the most dispersed inverse-scale weights. Many of its windows are nearly constant, so $s^2 \to \epsilon$ and $w_r = s_r^{-2}$ concentrates fitting weight on a small subset of rows. This effect arises from applying the weighting rule to degenerate contexts, rather than from the series alone.

Dataset	U/IN$_1$	Retained scale-only LWRN/IN$_1$
ETTh1	0.9963	0.9995
ETTh2	0.9923	1.0007
ETTm1	0.9512	0.9913
ETTm2	1.0006	0.9995
Weather	0.9673	0.9866
Electricity	0.9853	0.9999
Traffic	0.9793	0.9981
Solar	0.7875	0.9256
Exchange	0.9247	0.9854

Table 3. Geometric-mean MSE ratios across four horizons for each dataset, using compressed features at $L = 96$. Both comparisons hold the uniform center fixed and retune the head on the common effective grid. Ratios are computed from matched horizon-level errors, not from arithmetic dataset means.

Setting	C/U, all datasets	C/U, without Solar
Compressed, $L = 96$	0.972 [0.938, 0.995]	0.989 [0.979, 0.997]
Raw lags, $L = 96$	1.000	1.000
Compressed, $L = 2048$	1.000 [0.999, 1.001]	1.000 [0.999, 1.001]

Table 4. Unweighted center learning with a budget of 30 updates. Ratios and exact dataset-cluster 95% intervals are rounded to three decimal places. Full-precision values are retained in the companion records. For raw lags, $C/U - 1 = -1.02 \times 10^{-5}$, and both intervals contain one. Proposition 2 rules out class expansion for raw lags, but the induced penalty can still change.

agreement as a check on the parameterization. The difference between the two fits also depends on the regularized design and β, so a small penalty alone does not ensure similar fits if the design is ill-conditioned (Equation (19) in Appendix A).

The direction-removal controls test whether the added representation contributes to the gain. For $p \notin \mathrm{col}(A)$, the constraint $\mathbf{1}_F^\top W = \mathbf{1}_H^\top$ forces $\beta = 0$, removing the entire outside-span component in Equation (7). We impose this constraint and refit the head rather than delete a coefficient from the fitted model. The independently tuned S control also allows the original-span model to choose its own penalty. Together, these comparisons test whether the predictor makes use of the added direction.

Absolute performance and scope. Table 5 compares absolute performance across representations. Raw-lag U has lower dataset-mean MSE than compressed C on eight datasets, with Exchange tied at the reported precision. Compression at $L = 96$ retains 65–85 features, so the reduction in dimension is modest. We use the compressed map to control which lag directions are available. The center comparisons hold that map fixed and compare C with U. The table shows where those comparisons sit relative to absolute performance.

For raw lags, Proposition 2 rules out class expansion for every center. The measured C/U ratios for raw lags and long contexts both round to one at three decimal places. These settings and short compression all use 30 updates, so the representation comparison is budget-matched. The selected penalties are not uniformly small. Of the 36 raw-lag center models, 24 use $\lambda \in \{0.01, 0.1\}$, while 30 of the 36 long-context center models use $\lambda \in \{0.1, 1\}$. Appendix C reports the full distributions (Table 9) and examines sensitivity to the update budget.

The two additional temporal splits give short-compression C/U ratios of 0.9510 and 0.9515. Without Solar, the corresponding ratios are 0.9814 and 0.9870. Center learning therefore retains its within-representation advantage in both splits.

4.3 Scale weighting helps when it matches the noise structure

Our synthetic experiments use independent contexts, hold the unit-sum linear prediction class fixed, and measure error against the known conditional mean. Under heteroskedastic noise, scale-only

Dataset	Compressed, $L = 96$ U	C	Raw, $L = 96$ U	Compressed, $L = 2048$ U
ETTh1	0.4417	0.4407	0.4383	0.4111
ETTh2	0.3743	0.3742	0.3727	0.3405
ETTm1	0.4348	0.4191	0.4116	0.3447
ETTm2	0.2888	0.2871	0.2855	0.2373
Weather	0.2761	0.2747	0.2744	0.2312
Electricity	0.2251	0.2221	0.2162	0.1667
Traffic	0.7125	0.6944	0.6474	0.4437
Solar	0.4329	0.3693	0.3641	0.2010
Exchange	0.3435	0.3435	0.3435	0.2980

Table 5. Absolute MSE, averaged arithmetically across the four horizons for each dataset. All columns use unweighted fitting with homogeneous heads. The long-context column changes both the available history and the representation.

LWRN achieves an excess-MSE ratio to IN of 0.810 [0.791, 0.830] across 50 paired replicates. Oracle weighted least squares achieves 0.742 [0.728, 0.756]. When the true variance uses uniform lag weights, IN is the oracle rule. In that condition, the learned ratio is 0.999, and the unrounded interval contains one.

For a known conditional mean $m(x) = \mathbb{E}[y \mid x]$, excess error measures $\|\widehat{y} - m(x)\|^2$ instead of $\|\widehat{y} - y\|^2$. When the predictor is evaluated independently, expected total error decomposes as

$$\mathbb{E}\|\widehat{y} - y\|^2 = \mathbb{E}\|\widehat{y} - m(x)\|^2 + \mathbb{E}\|y - m(x)\|^2. \tag{16}$$

The second term does not depend on the estimator. A large relative reduction in estimation error can therefore translate into a much smaller relative reduction in total forecasting error. We report both. The normalization scale assigns a single scalar weight to each context–channel row, shared across all forecast leads. It does not model lead-specific uncertainty or a full residual covariance. The synthetic experiments test this scalar weighting mechanism against a known scalar noise structure.

A separate study compares weighting families under different variance structures (Table 6). The conditions use uniform variance weights, suffix-supported weights, dispersed weights, or constant variance. Each condition has the same prediction class and 50 paired replicates. Suffix weights are supported on the last eight lags, while dispersed weights are supported on four separated lags. Appendix D describes the generator and selection protocol.

Variance structure	LWRN/TW-IN, excess MSE	95% paired interval
Uniform	0.9994	[0.9965, 1.0021]
Suffix (8)	0.9356	[0.9213, 0.9514]
Dispersed (4)	0.6926	[0.6748, 0.7108]
Constant	1.1119	[1.0971, 1.1276]

Table 6. Scale-only learning compared with suffix normalization (TW-IN) under different variance structures. Entries are paired geometric-mean excess-MSE ratios across 50 replicates. Intervals are computed by resampling paired seeds.

Dispersed variance weights favor scale-only learning over suffix normalization, whereas constant variance favors unweighted fitting. In the dispersed condition, mean total MSE decreases from 0.231814 to 0.229086, or 1.18%. This reduction is much smaller than the relative reduction in excess error. In the suffix condition, a correctly specified fixed-center suffix rule outperforms learned scale weighting. An improvement over one suffix normalizer should therefore not be read as an improvement over a correctly specified weighting family. As in the benchmark results, the value of the weights depends on the estimation problem.

5 Relation to Prior Work

Normalization and linear forecasting. RevIN and Dish-TS use reversible statistics to address non-stationarity. Dish-TS also introduces separate input and output coefficient networks (Kim et al.,

2022, Fan et al., 2023). DLinear and RLinear established the importance of simple linear architectures and preprocessing (Zeng et al., 2023, Li et al., 2023). Toner & Darlow (2024) characterize their expressible function classes and equivalent regression formulations. Our focus is the regularized estimator under a fixed or compressed feature map. In this setting, normalization can change the fitting weights and induced penalty even when the decoded class remains unchanged.

Learning statistics and fitting weights. SearchCast jointly searches preprocessing and ridge choices, including lookback and local normalization (Huang et al., 2026). Our suffix control isolates its normalization component. IN-Flow learns an invertible normalization through bilevel optimization (Fan et al., 2025). Differentiable closed-form solvers and validation-guided reweighting are established tools (Bertinetto et al., 2019, Ren et al., 2018). We restrict them to lag-weighted moments and a ridge head, which allows the objective to be analyzed in closed form. Weighted least squares accounts for the effect of scale (Aitken, 1936). The center controls distinguish a learned feature from the normalization operation that introduces it.

6 Discussion and Conclusion

A normalization comparison should specify the loss coordinates, account for the mean residual weight, and include a constant-scale baseline. The practical sequence is straightforward. Start with a fixed representation and uniform center, and compare inverse-scale weighting with unweighted fitting on a common effective penalty grid. Then learn the center while holding the scale constant. Where center learning helps, identify the attainable class, fit an explicit feature model, and refit under the original-span restriction to determine how the added direction contributes. These checks distinguish a useful change to the estimator from an improvement that depends on an unnecessarily weighted baseline. A better score alone does not reveal which aspect of normalization produced the gain.

The distinction between fitting and prediction also matters for deployment. A fitted homogeneous head predicts as xB or $(xA)W + (xp)\beta$ without evaluating the scale estimator, yet the fitting weights induced by the scale have already shaped its coefficients. Saved-model checks recover normalized inference to numerical precision (Appendix E).

Reversible normalization changes a fitted linear forecaster through its relative observation weights, effective ridge penalty, and attainable lag directions. Across nine benchmarks, unweighted fitting outperforms mean-one instance-normalization weights both with and without Solar, and learning the center yields a smaller, representation-dependent gain. However, this gain appears only with short compressed contexts, which raw lags of the same length match or outperform in absolute MSE, and the benchmark statistics receive a budget of only 30 updates. Future work will compare the learned center direction with fixed and random directions of the same rank and extend the center comparisons to larger update budgets.

AI USE STATEMENT

Generative AI tools were used for code implementation, data processing, experiment execution, numerical verification, and manuscript editing. The authors specified and verified all derivations, experimental designs, and reported claims.

REPRODUCIBILITY STATEMENT

Section 2 and Appendix A give the assumptions and derivations. Section 3 and Appendix B describe fitting and evaluation, while Appendix C reports the additional controls. The companion records contain full-precision results and selected hyperparameters, together with a deterministic reaggregation script. Code, companion records, selected hyperparameters, and the reaggregation script are available at `anonymous.4open.science/r/XXXXXX`.

REFERENCES

Alexander C. Aitken. On least squares and linear combination of observations. *Proceedings of the Royal Society of Edinburgh*, 55:42–48, 1936. doi: 10.1017/S0370164600014346.

Luca Bertinetto, João F. Henriques, Philip H. S. Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HyxnZh0ct7.

George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Greta M. Ljung. *Time Series Analysis: Forecasting and Control*. Wiley, 5 edition, 2015.

Cristian Challu, Kin G. Olivares, Boris N. Oreshkin, Federico Garza, 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.

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

Robert B. Cleveland, William S. Cleveland, Jean E. McRae, and Irma Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3–73, 1990.

Abhimanyu Das, Weihao Kong, Andrew Leach, Shaan Mathur, Rajat Sen, and Rose Yu. Long-term forecasting with TiDE: Time-series dense encoder. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=pCbC3aQB5W.

Wei Fan, Pengyang Wang, Dongkun Wang, Dongjie Wang, Yuanchun Zhou, and Yanjie Fu. Dish-TS: A general paradigm for alleviating distribution shift in time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 7522–7529, 2023. doi: 10.1609/aaai.v37i6.25914.

Wei Fan, Shun Zheng, Pengyang Wang, Rui Xie, Kun Yi, Qi Zhang, Jiang Bian, and Yanjie Fu. IN-Flow: Instance normalization flow for non-stationary time series forecasting. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 295–306, 2025. doi: 10.1145/3690624.3709260. URL https://arxiv.org/abs/2401.16777.

Rakshitha Godahewa, Christoph Bergmeir, Geoffrey I. Webb, Rob J. Hyndman, and Pablo Montero-Manso. Monash time series forecasting archive. In *NeurIPS Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=wEc1mgAjU-.

Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970. doi: 10.1080/00401706.1970.10488634.

Charles C. Holt. Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1):5–10, 2004. doi: 10.1016/j.ijforecast.2003.09.015.

Lang Huang, Jinglue Xu, and Luke Darlow. How good can linear models be for time-series forecasting? *arXiv preprint arXiv:2606.27282*, 2026. doi: 10.48550/arXiv.2606.27282. URL https://arxiv.org/abs/2606.27282.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, Melbourne, Australia, 3 edition, 2021. URL https://otexts.com/fpp3/.

Rob J. Hyndman and Anne B. Koehler. Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688, 2006. doi: 10.1016/j.ijforecast.2006.03.001.

Rob J. Hyndman, Anne B. Koehler, Ralph D. Snyder, and Simone Grose. A state space framework for automatic forecasting using exponential smoothing methods. *International Journal of Forecasting*, 18(3):439–454, 2002. doi: 10.1016/S0169-2070(01)00110-8.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 448–456. PMLR, 2015. URL https://proceedings.mlr.press/v37/ioffe15.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://openreview.net/forum?id=cGDAkQo1C0p.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

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.

Zhe Li, Shiyi Qi, Yiduo Li, and Zenglin Xu. Revisiting long-term time series forecasting: An investigation on linear mapping. *arXiv preprint arXiv:2305.10721*, 2023. URL https://arxiv.org/abs/2305.10721v1.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. In *Advances in Neural Information Processing Systems*, volume 35, pp. 9881–9893, 2022. URL https://arxiv.org/abs/2205.14415.

Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1):54–74, 2020. doi: 10.1016/j.ijforecast.2019.04.014.

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://openreview.net/forum?id=Jbdc0vTOcol.

Boris N. Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=r1ecqn4YwB.

Nikolaos Passalis, Anastasios Tefas, Juho Kanniainen, Moncef Gabbouj, and Alexandros Iosifidis. Deep adaptive input normalization for time series forecasting. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3760–3765, 2020. doi: 10.1109/TNNLS.2019.2944933.

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.

Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4334–4343. PMLR, 2018. URL https://proceedings.mlr.press/v80/ren18a.html.

David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3): 1181–1191, 2020. doi: 10.1016/j.ijforecast.2019.07.001.

Slawek Smyl. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1):75–85, 2020. doi: 10.1016/j.ijforecast.2019.03.017.

Leonard J. Tashman. Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 16(4):437–450, 2000. doi: 10.1016/S0169-2070(00)00065-0.

William Toner and Luke Nicholas Darlow. An analysis of linear time series forecasting models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48404–48427. PMLR, 2024. URL https://proceedings.mlr.press/v235/toner24a.html.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. URL https://arxiv.org/abs/1607.08022.

Peter R. Winters. Forecasting sales by exponentially weighted moving averages. *Management Science*, 6(3):324–342, 1960. doi: 10.1287/mnsc.6.3.324.

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. URL https://proceedings.neurips.cc/paper_files/paper/2021/hash/bcc0d400288793e8bdcd7c19a8ac0c2b-Abstract.html.

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

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. doi: 10.1609/aaai.v37i9.26317.

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. doi: 10.1609/aaai.v35i12.17325.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. FEDformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 27268–27286. PMLR, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Proofs and Estimator Identities

A.1 Decoding and shift equivariance

Substituting Equation (1) into the homogeneous decoder yields

$$s(zW) + c\mathbf{1}_H^\top = xAW + (xp)(\mathbf{1}_H^\top - \mathbf{1}_F^\top W) = xB.$$

Because $\mathbf{1}_L^\top B = \mathbf{1}_F^\top W + \beta = \mathbf{1}_H^\top$, any scalar d satisfies

$$\widehat{y}(x + d\mathbf{1}_L^\top) = \widehat{y}(x) + d\mathbf{1}_H^\top.$$

The deployment scaler is fixed, and the predictor is linear in training-standardized coordinates. Scale cancellation requires a homogeneous head and the same positive scale for encoding and decoding. The scale estimator itself need not be static or linear. Adding an unpenalized intercept in normalized coordinates gives the decoded predictor $xB + s(x)b$. Separate output statistics or affine transformations within the normalizer require a separate decoder analysis.

A.2 Weighted objective and effective ridge

The homogeneous encoded residual is $(xB - y)/s$. Squaring it gives Equation (4). For the same p and penalty on W, squared loss in decoded coordinates is $n^{-1}\sum_r \|x_rB - y_r\|^2 + \lambda\|W\|_F^2$, with no scale parameter. Dividing Equation (4) by $\bar{w} > 0$ gives Equation (5).

The mean-one experiments compute $\bar{w}$ separately on the inner-fit and final-fit rows. Holding λ_{eff} fixed across these fits is not the same as holding the normalized-coordinate penalty λ fixed. Global mean-one normalization preserves relative weights across channels. All main benchmark fits use a shared head on the pooled rows.

A.3 Attainable coefficients

Since $\mathbf{1}_L^\top C_p = 0$, every vector in $p + \mathrm{col}(C_p)$ has unit sum and lies in $\mathrm{span}(\mathrm{col}(A), p)$. Conversely, a unit-sum vector $u = Aw + \alpha p$ must satisfy $\alpha = 1 - \mathbf{1}_F^\top w$, so $u = p + C_p w$. The unit-sum functional is nonzero on this span because its value at p is one. Intersecting the span with the unit-sum affine hyperplane therefore gives dimension $\mathrm{rank}([A\ p]) - 1$. This proves Equation (6) without requiring the columns of A to be linearly independent.

Projecting $B = AW + p\beta$ onto the orthogonal complement of $\mathrm{col}(A)$ gives Equation (7). The uniform center lies in our compression span because the recent individual lags and non-overlapping older averages cover the entire context. Adding a center p outside this span increases the rank by one. With $A = I$, the span is already $\mathbb{R}^L$, and every center p gives the same unit-sum hyperplane.

A.4 Equivalent features and the additional penalty

For fixed p and positive weights, the centered fit is equivalent to

$$\min_{W,\beta} \frac{1}{n}\sum_r w_r\|x_rAW + (x_rp)\beta - y_r\|^2 + \lambda\|W\|_F^2, \quad \mathbf{1}_F^\top W + \beta = \mathbf{1}_H^\top. \tag{17}$$

Eliminating β recovers the centered objective exactly. The F^+ control adds $\lambda\|\beta\|_2^2$ and learns its center distribution independently, using the same candidate grid and initialization as C.

For unweighted training matrices X, Y, write $G = C_p^\top X^\top X C_p/n + \lambda I$ and $K = C_p^\top X^\top (Y - Xp\mathbf{1}_H^\top)/n$. At fixed p, the two solutions can be written as

$$W_\eta = (G + \eta\lambda\mathbf{1}_F\mathbf{1}_F^\top)^{-1}(K + \eta\lambda\mathbf{1}_F\mathbf{1}_H^\top), \qquad \eta \in \{0, 1\}, \tag{18}$$

where $\eta = 0$ gives C and $\eta = 1$ gives F^+. In the implementation, we solve this system without forming an inverse. A small selected penalty reduces the added rank-one penalty term, consistent with the nearly equal errors in Section 4.2. Writing $W_C = W_0$ and $W_{F^+} = W_1$, subtracting the two normal equations gives

$$(G + \lambda\mathbf{1}_F\mathbf{1}_F^\top)(W_{F^+} - W_C) = \lambda\mathbf{1}_F(\mathbf{1}_H^\top - \mathbf{1}_F^\top W_C). \tag{19}$$

The additional coefficient penalty causes this difference. Its size depends on the regularized design, λ, and β. A small penalty alone does not ensure similar fits if the design is ill-conditioned.

A.5 Induced penalties and restricted fits

For any attainable B, a solution to $C_pW = B - p\mathbf{1}_H^\top$ can be decomposed into the minimum-norm pseudoinverse solution and an orthogonal null-space component. Minimizing the Frobenius norm then gives Equation (8). When $A = I$, $\ker(C_p) = \mathrm{span}(p)$, so valid heads have the form $W = B + p\gamma$. The norm is minimized at $\gamma = -p^\top B/\|p\|^2$, proving Equation (9).

For the uniform center p_0, the penalty is $R_{p_0}(B) = \|B\|_F^2 - H/L$ on the unit-sum class. Uniform-centered raw ridge and canonical unit-sum raw ridge thus have the same optimum for any fixed positive λ. At zero penalty, equal classes give equal optimal fitted values on the training data. Unique coefficients for out-of-sample prediction additionally require sufficient design rank.

The S control independently selects the penalty for $B = AW$ subject to $\mathbf{1}_F^\top W = \mathbf{1}_H^\top$. The conditional control C_{res} instead uses the penalty selected by C and imposes that constraint on the centered head. Writing $G = Z^\top Z/n + \lambda I$, the restricted solution is

$$W_{\mathrm{res}} = W_{\mathrm{full}} - \frac{G^{-1}\mathbf{1}_F(\mathbf{1}_F^\top W_{\mathrm{full}} - \mathbf{1}_H^\top)}{\mathbf{1}_F^\top G^{-1}\mathbf{1}_F}.$$

When the center lies outside the original span, the constraint sets $\beta = 0$ and removes the outside-span component. We compare the restricted refit with the full refit.

B Data, Sampling, and Selection

B.1 Chronological segments and features

Table 7 lists the original split sizes. The scaler is fit only on the designated training prefix. Validation and test targets must fall entirely within their respective segments, although their contexts may include earlier observations. A segment of length T provides $T - H + 1$ eligible forecast origins. These origins are shared across methods and lookbacks.

Dataset	Channels	Train	Validation	Test	F_{96}	F_{2048}
ETTh1	7	8,640	2,880	2,880	70	590
ETTh2	7	8,640	2,880	2,880	70	590
ETTm1	7	34,560	11,520	11,520	65	536
ETTm2	7	34,560	11,520	11,520	65	536
Weather	21	36,887	5,270	10,539	65	530
Electricity	321	18,412	2,632	5,260	70	590
Traffic	862	12,280	1,756	3,508	70	590
Solar	137	36,792	5,256	10,512	65	530
Exchange	8	5,311	760	1,517	85	1,410

Table 7. Observations per channel in each chronological segment and fixed feature dimensions F_L for each context length L.

The feature map keeps the $R = \min(320, L - 32)$ most recent lags individually. Earlier observations contribute through non-overlapping block averages of width 32 and seasonal anchors. Seasonal periods are 24 for ETTh, Electricity, and Traffic, 96 for ETTm, 144 for Weather and Solar, and 5 for Exchange. The corresponding anchor radii are 1, 4, 6, and 1. Anchors begin two periods back and exclude the recent region. We remove duplicates within the seasonal set, but an anchor may also contribute to an older block average. Every normalizer uses the same feature choices. Raw-lag controls replace this map with $A = I$.

Capped samples include all channels at equally spaced temporal origins. Inner sampling rounds single-precision interpolation values, while final sampling rounds double-precision interpolation values. Both procedures remove duplicate origins. All methods follow these conventions. Table 8 reports the resulting temporal coverage for datasets with many channels.

Dataset	Channels	Inner origins, at most	Final origins, at most
Electricity	321	62	311
Traffic	862	23	116
Solar	137	145	729

Table 8. Temporal coverage imposed by the pooled-row caps. For D channels, the bounds are $\lfloor 20{,}000/D \rfloor$ and $\lfloor 100{,}000/D \rfloor$. The recorded Traffic inner fits contain 23 origins, or 19,826 rows.

B.2 Learned candidates and weight controls

Each learned homogeneous model uses seven penalties and two learning rates, giving 14 optimized candidates. Every candidate starts from uniform lag weights and receives 30 updates. We retain the best inner-tune checkpoint, refit the head on training-prefix samples, and use official validation to select the candidate. Near-ties are resolved by the original deterministic grid order, with tolerance $10^{-12}\max(1, |\mathrm{MSE}|)$. Validation observations are never included in the fitting set. Averaging the ridge objective over fitting rows ensures that a change in row count alone does not change the nominal penalty.

The conditional weighting study retains the previously selected statistics of the homogeneous scale-only and full normalizers. Their heads are retuned with mean-one weights over a controlled effective penalty range, but the statistics are not retrained under the new objective. The new study at $L = H = 96$ trains a sigmoid gate initialized at $\alpha = 0.5$, using

$$\widetilde{w}_{\alpha,r} = (1-\alpha) + \alpha w_r/\bar{w}.$$

We evaluate the uniform-weight and constant-weight endpoints separately. The center-only model C and feature model F^+ are learned independently under unweighted fitting.

The suffix control T uses mean-one weights and a homogeneous head. For short and long contexts, respectively, it searches the suffix grids

$$\{1, 2, 3, 5, 7, 10, 15, 22, 32, 46, 66, 96\},$$
$$\{1, 2, 4, 6, 8, 12, 18, 26, 37, 53, 77, 111, 160, 231, 332, 478, 688, 989, 1423, 2048\}.$$

With seven penalties, the two grids produce 84 and 140 validation candidates, compared with 14 optimized candidates per learned model. The raw-lag controls do not include T.

B.3 Temporal follow-ups and statistical calculation

Let N denote the original training-prefix length. In the first additional split, training, validation, and evaluation end at $\lfloor 0.4N \rfloor$, $\lfloor 0.5N \rfloor$, and $\lfloor 0.6N \rfloor$. The second split uses $\lfloor 0.6N \rfloor$, $\lfloor 0.7N \rfloor$, and $\lfloor 0.8N \rfloor$. Each split refits the scaler and repeats the inner chronological split at $L = 96$ and $H = 96, 336$.

The studies were conducted sequentially, beginning with the normalizer comparison, followed by the conditional weight study, and then the representation controls and temporal follow-ups. Each study selected its hyperparameters on official validation before evaluating its own test results. Test scores were never used for selection. The temporal follow-ups reuse the same series and assess stability across fitting windows.

For methods a, b, dataset d contributes $\ell_d = |\mathcal{H}_d|^{-1} \sum_{h \in \mathcal{H}_d} \log(\mathrm{MSE}_{a,d,h}/\mathrm{MSE}_{b,d,h})$, giving aggregate ratio $\exp(D^{-1}\sum_d \ell_d)$. We enumerate every count vector $k \in \mathbb{N}_0^D$ satisfying $\sum_d k_d = D$. Each vector assigns probability $D!/(D^D \prod_d k_d!)$ to $\exp(D^{-1}\sum_d k_d \ell_d)$. The exact bootstrap interval is given by the weighted 2.5th and 97.5th percentiles. Enumeration covers 24,310 count vectors for nine datasets and 6,435 without Solar, eliminating Monte Carlo variation. No benchmark model is retrained or reselected for this calculation.

C Additional Controls and Optimization Scope

C.1 Selected penalties and near-equal feature fits

Table 9 counts the selected penalties in the frozen representation records. The companion CSV records each cell's selection, including its learning rate and checkpoint. The small penalties account for the close agreement between C and F^+ under short compression.

Setting	Model	10^{-8}	10^{-6}	10^{-4}	10^{-2}	10^{-1}	1
Compressed, $L = 96$	U	9	0	6	12	9	0
	C	18	4	9	5	0	0
	F^+	18	9	8	1	0	0
Raw, $L = 96$	U	6	0	6	14	10	0
	C	6	0	6	14	10	0
	F^+	6	0	6	14	10	0
Compressed, $L = 2048$	U	0	1	0	4	21	10
	C	0	1	0	5	20	10
	F^+	0	1	0	4	21	10

Table 9. Selected effective penalties across the 36 dataset–horizon cells in each row. No model shown here selects 100, so that grid value is omitted.

For compressed $L = 96$, the unrounded geometric ratios are $F^+/C - 1 = 4.46 \times 10^{-5}$, $C_{\text{res}}/C = 1.0294$, and $C/S = 0.9717$. For raw $L = 96$, the deviation is $C/U - 1 = -1.02 \times 10^{-5}$. For compressed $L = 2048$, it is 1.09×10^{-5}. Under short compression, center learning also improves MAE, with $C/U = 0.9819$. The MAE ratio $C/T = 0.9858$ has a dataset-bootstrap interval containing one in the original Monte Carlo analysis. The corresponding MSE ratio is $C/T = 0.9716$.

C.2 Temporal sensitivity

Table 10 reports the two additional chronological splits, both with and without Solar.

Setting	C/U, all datasets	C/U, without Solar
Temporal split 1	0.9510 [0.8894, 0.9907]	0.9814 [0.9667, 0.9939]
Temporal split 2	0.9515 [0.8826, 0.9912]	0.9870 [0.9807, 0.9929]

Table 10. Additional chronological evaluations, each covering 18 cells or 16 without Solar. Both use the exact dataset-cluster bootstrap and the same 30-update protocol.

C.3 Update budget and its sensitivity

Every learned model receives 30 updates. All comparisons in Section 4 are budget-matched, and the weight comparison in Section 4.1 does not learn statistics. Among the representation controls, 35 of the 36 short-compression C models retain step 30. All raw-lag and long-context C models also retain step 30, as do the selected normalizers in the original study.

A longer-budget study evaluates the original normalizers on ETTh1 and Weather at $H = 96$ for both lookbacks. It uses four learning rates, $\{3 \times 10^{-5}, 3 \times 10^{-4}, 0.003, 0.03\}$, and budgets of 30 or 120 updates, holding the original head and penalty fixed. Checkpoints are selected on inner tune and scored on official validation. For static normalization on Weather at $L = 2048$ and learning rate 0.03, validation MSE decreases from 0.4265 to 0.4143. On ETTh1 at the same lookback and rate, it increases from 0.7192 to 0.7366. The larger update budget therefore does not help uniformly.

C.4 Scope of earlier baseline adaptations

Earlier common-head adaptations of RevIN and Dish-TS share parameters across channels and use the same ridge head. The Dish-TS adaptation also omits its auxiliary guidance loss. These adaptations are outside the homogeneous-decoder comparison and are not used to rank the published systems.

On the unrescaled nominal grid, the original short-context LWRN/TW-IN ratio is 1.0223. Proposition 1 shows that a suffix normalizer with $s^2 = \epsilon$ leaves relative weights constant but rescales the available penalty range by ϵ. Comparing methods on a nominal grid therefore combines changes in the penalty range with changes in the weighting rule. This is why Section 4 uses a common effective grid. The earlier results remain in the original experimental records.

D Synthetic Weighting Experiments

The independent-context study uses $L = 32$, $H = 8$, and a realizable unit-sum linear conditional mean. Noise variance is proportional to a known second moment around the uniform center. The 50 paired replicates use independent inner-fit, tune, validation, and test samples containing 256, 512, 2,048, and 8,192 examples, respectively. Scale-only learning searches rates $\{0.003, 0.03\}$ and budgets $\{30, 100, 300\}$. All methods refit on the combined 768 fit-and-tune examples, using unregularized heads in a common 31-coordinate centered representation. Test samples are generated after selection. Excess MSE is measured against the known conditional mean, not the noisy target.

Under heteroskedastic noise, the excess-MSE ratio for unweighted OLS/IN is 1.122 [1.109, 1.135]. Scale-only LWRN/IN achieves 0.810 [0.791, 0.830], while oracle WLS/IN achieves 0.742 [0.728, 0.756]. When the true variance uses uniform lag weights, IN is the oracle rule and LWRN/IN is 0.999 [0.997, 1.000]. Its unrounded interval contains one. These intervals use paired seed bootstraps rather than the dataset-cluster bootstraps used for the benchmarks.

The variance-structure extension uses AR-shaped contexts with coefficient 0.6 and noise standard deviation $0.5\sqrt{v(x)}$. Variance weights are either uniform, supported on the last eight lags, or supported on lags $0, 8, 16, 24$. A fourth condition has constant variance. Every condition uses 50 paired replicates and the same sample sizes. The suffix baseline is tail-window instance normalization (TW-IN), which estimates the center and scale from the last k lags and searches all k from 1 through 32. The fixed-center suffix control changes only the scale weights, keeping the generator's uniform center. Table 11 reports all four conditions.

	Mean total MSE		Mean excess MSE	
Variance structure	TW-IN	LWRN	TW-IN	LWRN
Uniform	0.229469	0.229468	0.008780	0.008776
Suffix (8)	0.232041	0.231539	0.008059	0.007544
Dispersed (4)	0.231814	0.229086	0.008932	0.006195
Constant	0.260799	0.261989	0.010614	0.011798

Table 11. Arithmetic means across 50 paired replicates in the variance-structure experiment. LWRN learns only the scale. Values are shown to six decimal places to make the small differences in excess risk visible.

The paired geometric excess-MSE ratio LWRN/TW-IN is 0.9994 [0.9965, 1.0021] for uniform weights, 0.9356 [0.9213, 0.9514] for suffix support, 0.6926 [0.6748, 0.7108] for dispersed support, and 1.1119 [1.0971, 1.1276] for constant variance. In the suffix condition, the correctly specified fixed-center suffix rule achieves excess MSE of 0.006757. This is below LWRN's 0.007544 and close to the oracle WLS value of 0.006732. With constant variance, unweighted OLS is the oracle rule and coincides with the singleton endpoint of TW-IN.

In the dispersed condition, total MSE decreases by 1.18%. Irreducible noise makes this improvement smaller than the relative reduction in excess risk. The experiments show that scale weighting can help within a fixed prediction class when the weights match the noise structure.

E Numerical Checks and Companion Records

The saved representation study compares the profiled and direct solvers across 144 cells. The maximum coefficient difference is 6.25×10^{-11}, the maximum difference in selected validation MSE is 5.28×10^{-13}, and the maximum backward residual is 2.07×10^{-15}. The unpenalized augmented-feature identity holds to within 8.05×10^{-16}.

Independent float64 checks on a random-matrix example test decoding, the weighted-regression equivalence, mean-one rescaling, and the raw induced penalty. Every absolute error is below 4×10^{-15}. The companion manifest records the frozen source hashes.

A homogeneous model can be deployed as xB or $(xA)W + (xp)\beta$ without evaluating the scale. We checked saved static and contextual ETTh1 models at $H = 96$ for both lookbacks. Their predictions differ from normalized inference by at most 9.59×10^{-14}. Dense deployment stores $L \times H$ coefficients, while the factorized form retains the feature map.