DASHRINK: DISAGREEMENT-ADAPTIVE SHRINKAGE FOR MULTIVARIATE FORECAST COMBINATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Combining multivariate forecasters requires deciding how strongly to depart from equal weights. A learned router can exploit differences between experts, but a large weight change can also produce a large forecast change when their predictions disagree. We propose disagreement-adaptive shrinkage (DAShrink), which pulls contextual combination weights toward the uniform average. The shrinkage strength depends on the spread of history-normalized expert predictions. A scalar chosen on validation windows controls the overall departure from equal weighting and can recover the uniform average exactly. The resulting weights have a positive lower bound, and the forecast deviation from the uniform average is bounded by the range of expert predictions. We evaluate a model with three heterogeneous experts on ten multivariate benchmarks with forty dataset–horizon settings. Against eleven baselines, DAShrink achieves an overall average rank of 1.262 and ranks first in 34 settings for MSE and 34 for MAE. With trained parameters fixed, removing either combination control increases both errors in all eight ETT ablation settings.

1 INTRODUCTION

Multivariate forecasting models encode different aspects of an observed history. Linear decomposition extrapolates trend and residual components (Zeng et al., 2023), while attention over variable tokens captures cross-channel dependencies (Liu et al., 2024). Combining their forecasts can use both sources of information. The resulting accuracy depends on how the combination weights are estimated.

Equal averaging is a strong reference because it avoids estimating relative weights (Smith & Wallis, 2009; Wang et al., 2023). A contextual router can adapt the weights to the input (Ni et al., 2024; Liu et al., 2025b), introducing an additional estimation problem. The effect of a weight change also depends on the forecasts being combined: the same change has little effect when experts agree and a larger effect when their predictions are far apart. This suggests controlling the contextual contribution using the current expert predictions.

We develop DAShrink, a disagreement-adaptive shrinkage rule for multivariate forecast combination. The rule extends equal-weight shrinkage (Diebold & Pauly, 1990; Diebold & Shin, 2019) to a coefficient that varies across forecast coordinates: disagreement among history-normalized predictions reduces the contribution of a learned weight proposal. A scalar selected from paired validation losses sets the overall amount of adaptation for each forecasting task. If the eligible set is empty, this scalar is zero and the system uses equal weights. Figure 1 shows the three-expert implementation.

The rule yields explicit bounds on each expert's weight and on forecast displacement from the uniform average. Across ten datasets, the three-expert model achieves the best aggregate rank among twelve methods. The component study measures each control's effect with learned parameters fixed. Horizon-level results and dataset-omission analyses show where the gains occur and how they affect the aggregate rank.

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.

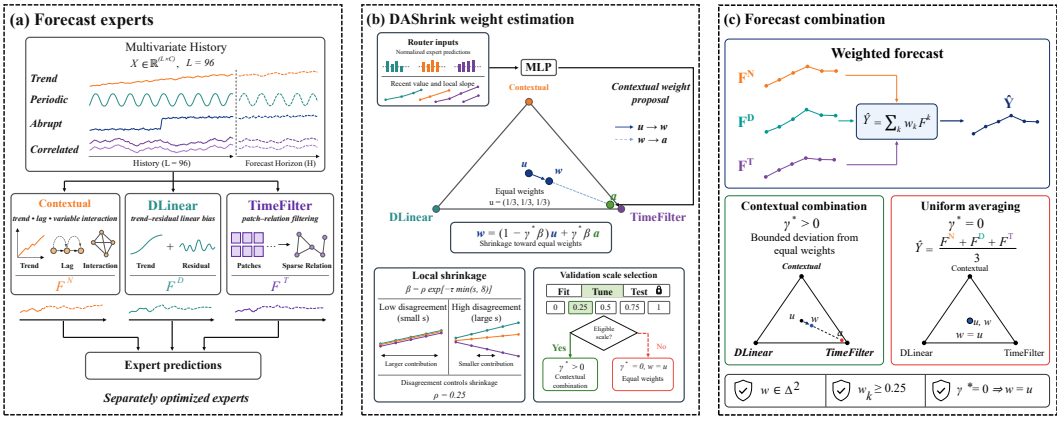


Figure 1: Expert bank and adaptive combination. A contextual router proposes weights, while local disagreement and a validation-selected scalar determine their contribution relative to equal averaging. Superscripts N, D, and T identify the contextual, DLinear, and TimeFilter experts.

2 Related Work

Forecast representations. Forecasting architectures encode different temporal structures. N-BEATS learns basis expansions (Oreshkin et al., 2020); Autoformer and FEDformer combine decomposition with temporal or frequency-domain aggregation (Wu et al., 2021; Zhou et al., 2022). TimesNet maps multiple periods into two-dimensional representations (Wu et al., 2023), while TimeMixer mixes decomposed patterns across scales (Wang et al., 2024). PatchTST builds channel-independent patch tokens (Nie et al., 2023). Crossformer and iTransformer explicitly model relationships between variables (Zhang & Yan, 2023; Liu et al., 2024), and TimeFilter selects patch-level spatial–temporal relations (Hu et al., 2025). These choices affect which patterns each expert can represent. Our bank combines a linear decomposition model with neural forecasters, and its router operates on their predictions.

Weight estimation and shrinkage. Classical forecast combination uses the joint errors of individual forecasts to determine their weights (Bates & Granger, 1969). Regression-based combination (Granger & Ramanathan, 1984) and Mallows model averaging (Hansen, 2008) provide estimation criteria. Empirical studies establish the strength of simple averages (Clemen, 1989; Stock & Watson, 2004), and analyses of the forecast combination puzzle explain how weight estimation affects forecast error (Smith & Wallis, 2009; Claeskens et al., 2016). Equal-weight shrinkage directly addresses this estimation problem: Diebold & Pauly (1990) incorporate prior information on weights, and Diebold & Shin (2019) combine selection with shrinkage toward equality. Wang et al. (2023) review these approaches and their extensions. DAShrink applies this principle at each forecast coordinate, with a coefficient determined by the spread of the current expert predictions.

Adaptive combinations and expert routing. Stacking learns a predictor from model outputs (Wolpert, 1992), and mixtures of experts learn input-dependent gates (Jordan & Jacobs, 1994). Dynamic model averaging updates model probabilities over time (Raftery et al., 2010); RLMC learns combination weights through reinforcement learning (Fu et al., 2022). Recent forecasting systems route among linear experts (Ni et al., 2024), adaptively fuse complete models (Liu et al., 2025b), select multiscale pathways (Chen et al., 2024), or activate sparse expert subnetworks inside a foundation model (Shi et al., 2025). DAShrink uses a dense combination of complete forecasters. Its uniform component sets a weight floor, while the contextual component changes with the forecast input. This makes the strength of adaptation an explicit part of the combination rule.

Normalization and validation selection. RevIN uses instance statistics to normalize and restore time-series values (Kim et al., 2022), and Non-stationary Transformers incorporate stationarization into attention (Liu et al., 2022). DAShrink uses history statistics to express expert disagreement on a common scale. After fitting the networks, a paired validation score selects the task-level multiplier.

Adaptive Learn-then-Test formalizes hyperparameter selection through statistical testing (Zecchin et al., 2025). Our selection rule uses an empirical mean–dispersion criterion over a fixed candidate grid; the resulting scalar remains fixed during evaluation.

3 METHOD

3.1 FORECAST COMBINATION

Given histories $\mathbf{X} \in \mathbb{R}^{B\times L\times C}$, we predict $\mathbf{Y} \in \mathbb{R}^{B\times H\times C}$: B, L, C, H denote batch size, history length, channel count, and horizon. Each of $K \geq 2$ experts produces $F^{(k)} \in \mathbb{R}^{B\times H\times C}$. A router proposes $\mathbf{a}_{bhc} \in \Delta^{K-1}$, where $\Delta^{K-1} = \{\mathbf{v} \in \mathbb{R}^K : \mathbf{v} \geq 0, \sum_k v_k = 1\}$. With $\mathbf{u} = \mathbf{1}/K$, the combination is

$$\mathbf{w}_{bhc}(\gamma) = (1 - \gamma\beta_{bhc})\mathbf{u} + \gamma\beta_{bhc}\mathbf{a}_{bhc}, \tag{1}$$

$$\widehat{Y}_{bhc}(\gamma) = \sum_{k=1}^{K} w_{bhc}^{(k)}(\gamma) F_{bhc}^{(k)}. \tag{2}$$

The local coefficient $\beta_{bhc} \in (0, \rho]$, with $0 < \rho < 1$, limits the contextual contribution at each coordinate. Validation selects $\gamma \in [0, 1]$ once per dataset–horizon task.

Contextual proposal. For each sample and channel, we compute the mean μ_{bc} and standard deviation σ_{bc} from the observed history, with $\sigma_{bc} \geq 10^{-5}$. Expert predictions used by the router are normalized and clipped:

$$\widetilde{F}_{bhc}^{(k)} = \mathrm{clip}\left(\frac{F_{bhc}^{(k)} - \mu_{bc}}{\sigma_{bc}}, -20, 20\right). \tag{3}$$

The input $\boldsymbol{\phi}_{bhc} \in \mathbb{R}^{K+4}$ contains the K normalized predictions, their mean and sample standard deviation, the latest normalized observation, and the slope over the last 24 observations. A width-32 network with horizon and channel embeddings produces $\mathbf{a}_{bhc} = \mathrm{softmax}(\mathrm{clip}(g_\theta(\boldsymbol{\phi}_{bhc}; h, c), -20, 20))$. Appendix B.3 defines these features and the network. Equation (2) combines the original expert predictions.

Local shrinkage. Let $\overline{F}_{bhc} = K^{-1}\sum_k \widetilde{F}_{bhc}^{(k)}$. For $K \geq 2$, define the normalized disagreement and local coefficient as

$$s_{bhc} = \sqrt{\frac{1}{K-1}\sum_{k=1}^{K}\left(\widetilde{F}_{bhc}^{(k)} - \overline{F}_{bhc}\right)^2}, \tag{4}$$

$$\beta_{bhc} = \rho \exp[-\tau\ \mathrm{clip}(\mathrm{sg}(s_{bhc}), 0, 8)]\,. \tag{5}$$

For $\tau > 0$, increasing disagreement pulls the weights toward $\mathbf{u}$, until the clip at eight fixes the coefficient. At $\tau = 0$, $\beta = \rho$. We use $\rho = 0.25$ and $\tau = 1$. Expert predictions are detached during router optimization; sg also removes the backward path through disagreement.

3.2 VALIDATION SCALE SELECTION

After fitting the neural parameters, we evaluate $\Gamma = \{0, 0.25, 0.5, 0.75, 1\}$ on validation windows. For window i, let $\ell_i(\gamma)$ be the squared error averaged over its HC forecast coordinates. Relative to uniform averaging, define

$$d_i(\gamma) = \ell_i(\gamma) - \ell_i(0), \qquad U_\gamma = \bar{d}_\gamma + 1.645\sqrt{\widehat{v}_\gamma/n}, \tag{6}$$

where n counts windows, $\bar{d}_\gamma = n^{-1}\sum_i d_i(\gamma)$, and $\widehat{v}_\gamma = n^{-1}\sum_i (d_i(\gamma) - \bar{d}_\gamma)^2$. With $\mathcal{E} = \{\gamma \in \Gamma \setminus \{0\} : U_\gamma < -10^{-8}\}$, choose

$$\gamma^\star = \begin{cases} \arg\min_{\gamma\in\mathcal{E}} \bar{d}_\gamma, & \mathcal{E} \neq \varnothing, \\ 0, & \mathcal{E} = \varnothing. \end{cases} \tag{7}$$

Ties select the smaller scale. The score U_γ balances mean improvement against dispersion; 1.645 sets the penalty. Tune windows overlap and also serve checkpoint selection. The chosen scalar remains fixed during inference, as illustrated in Figure 3.

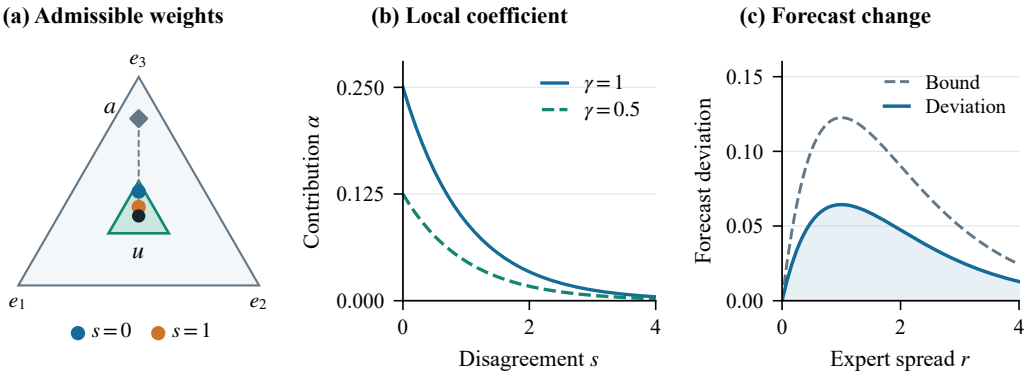


Figure 2: Analytical behavior for $K = 3, \rho = 0.25, \tau = 1$. (a) Admissible weights and a fixed proposal $\mathbf{a} = (0.1, 0.1, 0.8)$. (b) Contextual contribution as disagreement grows. (c) Forecast deviation and its bound at $\gamma = 1$, for predictions $(-r, 0, r)$ and unit history standard deviation.

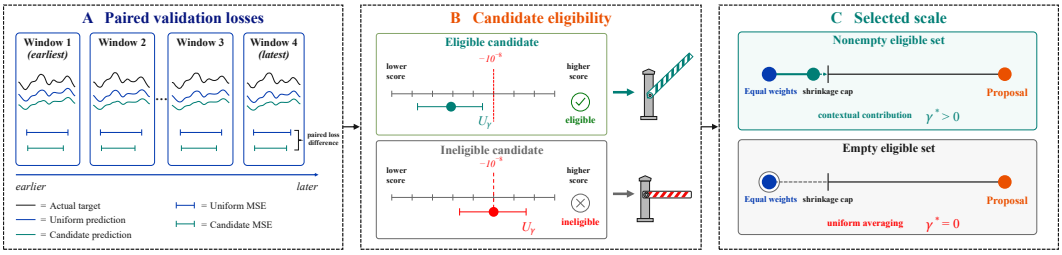


Figure 3: Selection of the task-level scale. Paired losses on tune windows determine the eligible candidates in Equation (7). An empty eligible set gives $\gamma^\star = 0$ and equal weights.

Expert bank and optimization. The bank contains DLinear (Zeng et al., 2023), TimeFilter (Hu et al., 2025), and a contextual expert combining decomposition, lag-based extrapolation, and variable-token attention (Appendix B). Each expert has its own parameters and squared-error objective.

Training uses separate optimizers for the experts and the outer router. Within each fit batch, an expert update is followed by a router update using detached expert predictions. The router is trained at $\gamma = 1$ with

$$\mathcal{L}_{\text{router}} = \frac{1}{BHC} \sum_{b,h,c} \left[(\widehat{Y}_{bhc}(1) - Y_{bhc})^2 + 0.05\, \mathrm{KL}(\mathbf{w}_{bhc}(1) \| \mathbf{u}) \right]. \tag{8}$$

Both objectives use the same fit windows. After checkpoint selection, the neural parameters are fixed for selecting $\gamma^\star$.

3.3 Weight and Forecast Bounds

The shrinkage rule gives the following deterministic bounds. Indices are suppressed for one forecast coordinate, and $R_F = \max_k F^{(k)} - \min_k F^{(k)}$.

Proposition 1. *Let* $\mathbf{a} \in \Delta^{K-1}$, $\mathbf{u} = \mathbf{1}/K$, *and* $\alpha = \gamma\beta \in [0, \rho]$, *where* $\rho < 1$. *Then* $\mathbf{w} = (1-\alpha)\mathbf{u} + \alpha\mathbf{a}$ *lies on the simplex and satisfies*

$$\frac{1-\rho}{K} \le w_k \le \frac{1-\rho}{K} + \rho, \qquad \|\mathbf{w} - \mathbf{u}\|_1 \le 2\alpha(1 - 1/K), \tag{9}$$

$$\left| \widehat{Y}(\gamma) - \widehat{Y}(0) \right| \le \alpha(1 - 1/K) R_F \le \rho\gamma(1 - 1/K) R_F. \tag{10}$$

For $\gamma = 0$, *the weights and forecast equal the uniform combination exactly.*

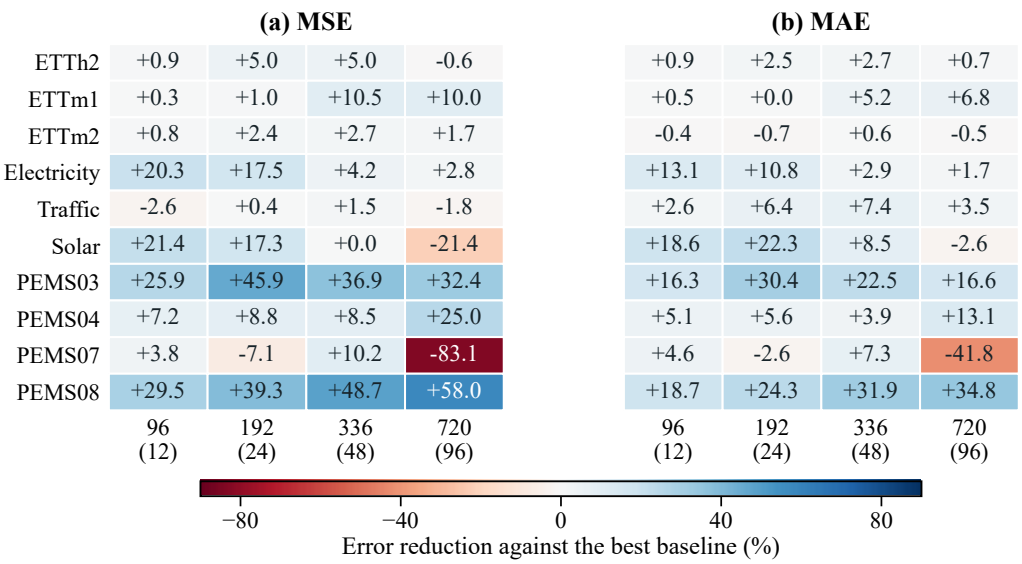


Figure 4: Error reduction against the best of eleven baselines in each setting: $100(e_{\text{best}} - e_{\text{ours}})/e_{\text{best}}$. Positive values favor DAShrink. Parenthesized horizons apply to PEMS. Values are computed from the three-decimal errors in Tables 1–2.

With $K = 3, \rho = 0.25$, every weight lies in $[1/4, 1/2]$. Appendix C gives the proof.

On the range plotted in Figure 2(c), deviation is $0.175re^{-r}$, with bound $re^{-r}/3$. Both peak at $r = 1$, where the expanding forecast range and decreasing coefficient balance.

Computational cost. Inference costs $\sum_k T_k + \mathcal{O}(BLC + BHC[(K+4)d + dK + K])$, where T_k is expert k's cost and d is the router width. The history features cost $\mathcal{O}(BLC)$; predictions and router activations occupy $\mathcal{O}(BHC(K+d))$ memory. Validation reuses expert predictions across all five scales.

4 Experiments

4.1 Setup

We evaluate ten datasets with history length $L = 96$. ETTh2, ETTm1, ETTm2, Electricity, Traffic, and Solar-Energy use horizons $\{96, 192, 336, 720\}$; PEMS03, PEMS04, PEMS07, and PEMS08 use $\{12, 24, 48, 96\}$. Dimensionality ranges from seven channels on ETT to 883 on PEMS07. Mean squared error (MSE) and mean absolute error (MAE) are averaged over all predicted coordinates within each task. Average ranks weight the forty dataset–horizon settings equally and combine both metrics.

The eleven baselines are DLinear, PatchTST, iTransformer, TimeMixer, CycleNet (Lin et al., 2024), TimeFilter, MoLE-DLinear, TimeFuse, TimeBridge (Liu et al., 2025a), TQNet (Lin et al., 2025), and PhaseFormer (Niu et al., 2026). Fit windows supply gradient updates; the subsequent tune segment selects checkpoints and $\gamma^\star$. Standardization uses the official training segment, comprising fit and tune data. The remaining partitions follow chronologically. Appendix A specifies the boundaries and window construction.

4.2 Accuracy across datasets and horizons

Tables 1 and 2 report every result for all twelve methods. DAShrink achieves an overall average rank of 1.262, followed by TQNet at 3.125. It ranks first in 34 MSE settings and 34 MAE settings, with 31 settings in common. The tables report the absolute errors; Figure 4 compares DAShrink with the strongest competing baseline in every setting.

Table 1: Complete MSE results on forty forecasting settings. Bold and underlined entries mark the reported best and second best. DL: DLinear; PT: PatchTST; iT: iTransformer; TM: TimeMixer; CN: CycleNet; TF: TimeFilter; Mo: MoLE-DLinear; Fu: TimeFuse; Br: TimeBridge; TQ: TQNet; PF: PhaseFormer. Joint rank averages both metrics across all settings.

Dataset	H	Ours	DL	PT	iT	TM	CN	TF	Mo	Fu	Br	TQ	PF
ETTh2	96	**0.217**	0.235	0.231	0.234	0.235	0.251	0.219	0.283	0.226	0.235	0.232	0.237
	192	**0.266**	0.340	0.300	0.301	0.303	0.331	0.280	0.384	0.298	0.316	0.322	0.317
	336	**0.362**	0.487	0.419	0.426	0.400	0.503	0.390	0.526	0.387	0.394	0.381	0.389
	720	0.638	0.748	0.673	0.692	**0.634**	0.758	0.652	0.749	0.646	0.648	0.638	0.642
ETTm1	96	**0.387**	0.438	0.426	0.425	0.434	0.457	0.439	0.410	0.388	0.433	0.405	0.416
	192	**0.502**	0.534	0.544	0.550	0.534	0.537	0.566	0.521	0.507	0.549	0.529	0.553
	336	**0.622**	0.817	0.924	1.015	0.753	0.754	0.695	0.997	0.755	0.808	0.738	0.769
	720	**0.902**	1.048	1.225	1.356	1.062	1.002	1.034	1.217	1.067	1.123	1.060	1.090
ETTm2	96	**0.121**	0.132	0.128	0.132	0.132	0.129	0.129	0.140	0.122	0.130	0.124	0.144
	192	**0.164**	0.193	0.171	0.176	0.170	0.202	0.173	0.220	0.169	0.169	0.168	0.179
	336	**0.217**	0.289	0.245	0.252	0.227	0.327	0.237	0.398	0.223	0.234	0.223	0.230
	720	**0.287**	0.425	0.314	0.324	0.293	0.423	0.295	0.487	0.292	0.304	0.295	0.296
Electricity	96	**0.126**	0.416	0.220	0.232	0.199	0.170	0.178	0.266	0.190	0.206	0.158	0.292
	192	**0.137**	0.411	0.221	0.242	0.206	0.176	0.181	0.262	0.188	0.210	0.166	0.270
	336	**0.181**	0.420	0.236	0.261	0.279	0.189	0.193	0.271	0.201	0.230	0.189	0.287
	720	**0.208**	0.437	0.271	0.309	0.248	0.214	0.224	0.295	0.231	0.268	0.221	0.301
Traffic	96	0.473	0.836	0.647	0.664	0.620	0.533	0.524	0.709	0.541	0.549	**0.461**	0.852
	192	**0.454**	0.813	0.595	0.640	0.542	0.499	0.492	0.662	0.504	0.519	0.456	0.729
	336	**0.450**	0.813	0.595	0.650	0.589	0.503	0.489	0.657	0.502	0.523	0.457	0.741
	720	0.508	0.828	0.638	0.714	0.582	0.542	0.537	0.685	0.550	0.577	**0.499**	0.734
Solar	96	**0.158**	0.672	0.316	0.318	0.283	0.236	0.228	0.290	0.236	0.277	0.201	0.709
	192	**0.167**	0.680	0.356	0.367	0.274	0.246	0.272	0.303	0.262	0.303	0.202	0.805
	336	**0.212**	0.657	0.363	0.371	0.314	0.262	0.267	0.307	0.271	0.322	**0.212**	0.673
	720	0.250	0.613	0.351	0.367	0.280	0.255	0.271	0.295	0.266	0.312	**0.206**	0.667
PEMS03	12	**0.063**	0.511	0.142	0.120	0.113	0.115	0.108	0.168	0.110	0.132	0.085	0.421
	24	**0.080**	0.625	0.247	0.201	0.204	0.161	0.170	0.214	0.176	0.221	0.148	0.688
	48	**0.111**	0.822	0.506	0.365	0.419	0.245	0.295	0.273	0.311	0.430	0.176	0.939
	96	**0.140**	1.037	1.036	0.699	0.886	0.317	0.544	0.341	0.562	0.816	0.207	1.350
PEMS04	12	**0.077**	0.573	0.163	0.147	0.134	0.136	0.116	0.193	0.137	0.154	0.083	0.435
	24	**0.103**	0.697	0.288	0.245	0.237	0.189	0.191	0.227	0.223	0.265	0.113	0.601
	48	**0.151**	0.908	0.592	0.442	0.495	0.292	0.349	0.295	0.425	0.526	0.165	1.082
	96	**0.183**	1.126	1.213	0.855	0.862	0.373	0.637	0.425	0.715	0.989	0.244	1.479
PEMS07	12	**0.075**	0.509	0.131	0.113	0.105	0.114	0.091	0.186	0.098	0.100	0.078	0.302
	24	0.120	0.621	0.245	0.208	0.205	0.176	0.164	0.245	0.168	0.189	**0.112**	0.451
	48	**0.202**	0.804	0.505	0.385	0.465	0.286	0.318	0.337	0.326	0.410	0.225	0.964
	96	0.456	0.982	1.031	0.689	0.803	0.401	0.566	0.418	0.552	0.753	**0.249**	1.133
PEMS08	12	**0.086**	0.596	0.161	0.140	0.132	0.153	0.122	0.293	0.130	0.150	0.183	0.417
	24	**0.122**	0.733	0.274	0.243	0.231	0.230	0.201	0.370	0.208	0.244	0.315	0.670
	48	**0.184**	0.972	0.570	0.463	0.506	0.403	0.359	0.545	0.386	0.484	1.112	1.082
	96	**0.291**	1.220	1.229	0.956	1.048	0.730	0.724	0.827	0.693	1.034	1.754	1.525
Joint rank		1.262	10.875	8.412	8.088	6.425	5.750	4.088	8.738	4.275	7.138	3.125	9.825

Horizon-dependent gains. The largest sustained gains occur on PEMS03 and PEMS08. On PEMS08, the MSE reduction grows from 29.5% at $H = 12$ to 58.0% at $H = 96$. Electricity exhibits the reverse pattern, falling from 20.3% to 2.8% across its horizons. ETTm1 improves most at its two longer horizons, whereas ETTm2 has small margins throughout. Figure 5(b) summarizes this heterogeneity: the median and middle half of the gains are computed separately at each horizon position. The width of the band records how differently datasets respond at the same relative forecast length.

Table 2: Complete MAE results on forty forecasting settings. Bold and underlined entries mark the reported best and second best. Abbreviations follow Table 1. Joint rank averages both metrics across all settings.

Dataset	H	Ours	DL	PT	iT	TM	CN	TF	Mo	Fu	Br	TQ	PF
ETTh2	96	**0.318**	0.340	0.326	0.332	0.328	0.353	0.321	0.361	0.323	0.329	0.327	0.331
	192	**0.349**	0.410	0.367	0.371	0.372	0.404	0.358	0.416	0.375	0.374	0.378	0.373
	336	**0.403**	0.492	0.442	0.449	0.422	0.494	0.415	0.509	0.421	0.423	0.414	0.415
	720	**0.544**	0.600	0.574	0.584	0.548	0.602	0.553	0.606	0.555	0.557	0.551	0.550
ETTm1	96	**0.411**	0.446	0.437	0.433	0.440	0.453	0.439	0.423	0.413	0.440	0.420	0.427
	192	**0.472**	0.490	0.495	0.493	0.490	0.494	0.501	0.480	**0.472**	0.496	0.481	0.492
	336	**0.525**	0.633	0.668	0.700	0.590	0.596	0.554	0.705	0.597	0.618	0.582	0.595
	720	**0.633**	0.720	0.769	0.812	0.697	0.684	0.679	0.782	0.709	0.726	0.697	0.711
ETTm2	96	0.240	0.255	0.247	0.249	0.248	0.253	0.247	0.259	0.240	0.249	**0.239**	0.267
	192	0.281	0.312	0.284	0.288	0.284	0.316	0.286	0.322	0.284	0.283	**0.279**	0.291
	336	**0.322**	0.389	0.346	0.352	0.327	0.404	0.332	0.449	0.327	0.334	0.324	0.330
	720	0.371	0.468	0.387	0.396	0.370	0.458	**0.369**	0.496	0.371	0.377	**0.369**	0.371
Electricity	96	**0.219**	0.506	0.307	0.318	0.288	0.275	0.263	0.361	0.278	0.291	0.252	0.342
	192	**0.231**	0.504	0.311	0.329	0.299	0.282	0.270	0.359	0.279	0.298	0.259	0.325
	336	**0.269**	0.510	0.324	0.344	0.372	0.293	0.279	0.366	0.292	0.315	0.277	0.332
	720	**0.298**	0.521	0.349	0.379	0.331	0.316	0.306	0.383	0.314	0.344	0.303	0.345
Traffic	96	**0.333**	0.556	0.438	0.453	0.433	0.374	0.369	0.481	0.368	0.381	0.342	0.482
	192	**0.321**	0.547	0.416	0.450	0.386	0.360	0.351	0.460	0.350	0.368	0.343	0.450
	336	**0.314**	0.546	0.415	0.456	0.411	0.361	0.345	0.458	0.346	0.369	0.339	0.461
	720	**0.334**	0.547	0.429	0.486	0.385	0.373	0.363	0.466	0.361	0.387	0.346	0.467
Solar	96	**0.232**	0.650	0.359	0.351	0.368	0.338	0.285	0.404	0.294	0.331	0.316	0.604
	192	**0.237**	0.654	0.379	0.393	0.335	0.345	0.329	0.411	0.314	0.347	0.305	0.703
	336	**0.280**	0.641	0.384	0.395	0.366	0.359	0.317	0.413	0.316	0.360	0.306	0.565
	720	0.314	0.617	0.380	0.409	0.341	0.350	0.323	0.401	0.309	0.356	**0.306**	0.587
PEMS03	12	**0.174**	0.599	0.266	0.235	0.232	0.246	0.227	0.304	0.229	0.255	0.208	0.473
	24	**0.195**	0.666	0.355	0.312	0.321	0.291	0.291	0.349	0.291	0.332	0.280	0.624
	48	**0.241**	0.774	0.527	0.450	0.469	0.370	0.390	0.401	0.401	0.476	0.311	0.767
	96	**0.276**	0.881	0.791	0.662	0.715	0.426	0.555	0.446	0.574	0.680	0.331	0.953
PEMS04	12	**0.186**	0.633	0.285	0.261	0.252	0.265	0.233	0.317	0.257	0.272	0.196	0.476
	24	**0.219**	0.702	0.379	0.346	0.340	0.316	0.306	0.353	0.328	0.359	0.232	0.568
	48	**0.270**	0.810	0.564	0.488	0.513	0.398	0.425	0.408	0.478	0.521	0.281	0.823
	96	**0.306**	0.916	0.859	0.733	0.706	0.454	0.583	0.486	0.648	0.751	0.352	0.996
PEMS07	12	**0.186**	0.586	0.257	0.232	0.227	0.245	0.208	0.316	0.218	0.219	0.195	0.397
	24	0.237	0.650	0.352	0.324	0.328	0.306	0.282	0.368	0.284	0.302	**0.231**	0.499
	48	**0.318**	0.747	0.512	0.456	0.507	0.399	0.403	0.440	0.409	0.456	0.343	0.768
	96	0.512	0.835	0.773	0.637	0.678	0.479	0.558	0.488	0.556	0.642	**0.361**	0.838
PEMS08	12	**0.187**	0.626	0.275	0.249	0.242	0.270	0.230	0.356	0.245	0.261	0.276	0.457
	24	**0.227**	0.695	0.363	0.339	0.325	0.331	0.300	0.416	0.311	0.337	0.353	0.600
	48	**0.276**	0.802	0.539	0.487	0.503	0.431	0.405	0.492	0.439	0.489	0.524	0.797
	96	**0.344**	0.902	0.827	0.726	0.737	0.528	0.583	0.573	0.602	0.724	0.648	0.956
Joint rank		1.262	10.875	8.412	8.088	6.425	5.750	4.088	8.738	4.275	7.138	3.125	9.825

Metric-dependent error patterns. Most settings improve under both metrics, occupying the upper-right quadrant of Figure 5(a). Traffic improves MAE at every horizon but has higher MSE than TQNet at $H = 96$ and $H = 720$. ETTm2 improves MSE throughout and has higher MAE in three settings. The metric orderings reflect their different weighting of residual magnitudes. The most pronounced loss is PEMS07 at $H = 96$: MSE/MAE is 0.456/0.512, versus TQNet's 0.249/0.361. This setting contributes over half of DAShrink's summed MSE across the four PEMS07 horizons. Solar-Energy also loses at $H = 720$, with MSE 0.250 versus 0.206 for TQNet. On each of these two datasets, the longest horizon has the largest relative loss against the best baseline.

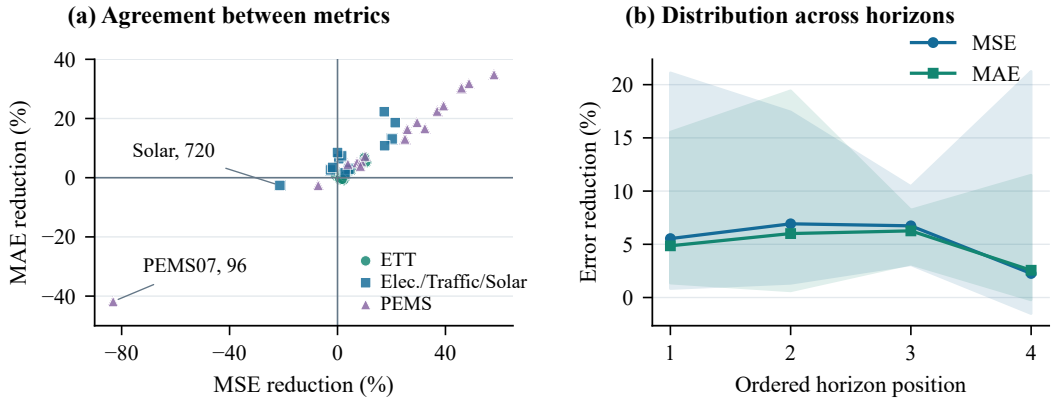


Figure 5: Structure of the benchmark gains. (a) Each point pairs the MSE and MAE reductions for one of forty settings. Labels identify two long-horizon settings, PEMS07-96 and Solar-720. (b) Median reductions and interquartile bands across ten datasets at each ordered horizon position. Both panels use the errors in Tables 1–2.

Table 3: Component results on ETTh2 and ETTm1. No local: $\tau = 0$ with scale reselected; no scale: $\gamma = 1$; neither: $\tau = 0, \gamma = 1$. All variants retain $\rho = 0.25$. Bold marks the smallest error within each row.

Dataset	H	Full		No local		No scale		Neither	
		MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
ETTh2	96	**0.228**	**0.329**	0.233	0.334	0.237	0.337	0.242	0.341
	192	**0.296**	**0.371**	0.301	0.376	0.306	0.380	0.312	0.385
	336	**0.403**	**0.434**	0.409	0.440	0.415	0.446	0.423	0.453
	720	**0.638**	**0.544**	0.649	0.552	0.661	0.560	0.676	0.571
ETTm1	96	**0.387**	**0.411**	0.394	0.417	0.401	0.423	0.409	0.430
	192	**0.502**	**0.472**	0.510	0.479	0.518	0.486	0.527	0.494
	336	**0.622**	**0.525**	0.632	0.533	0.641	0.541	0.652	0.550
	720	**0.902**	**0.633**	0.916	0.643	0.929	0.653	0.944	0.665

4.3 Effects of the combination controls

Table 3 holds the trained experts and router fixed while changing the combination controls. Setting $\tau = 0$ removes local shrinkage and repeats validation scale selection. Setting $\gamma = 1$ removes task-level scale selection. Applying both changes gives $\mathbf{w} = 0.75\mathbf{u} + 0.25\mathbf{a}$, retaining the same contribution cap. Full has the lowest MSE and MAE in all eight rows.

Figure 6 shows a consistent ordering across horizons. Fixing the scale to one increases errors more than removing the disagreement dependence, and their joint removal produces the largest increase. On ETTh2, the four-horizon mean MSE rises by 1.73%, 3.45%, and 5.62%, respectively. The local coefficient allocates adaptation across forecast coordinates; the selected scalar determines its overall magnitude for the task. This study quantifies their effects on the fixed proposal learned with the full rule.

4.4 Rank stability across benchmark compositions

We examine composition dependence at three resolutions in Figure 7. Panel (a) summarizes 20,000 bootstrap draws of ten datasets, keeping all four horizons of each sampled dataset together. DAShrink's observed MSE/MAE ranks are 1.225/1.300, with percentile ranges [1.050, 1.450] and [1.000, 1.650]. TQNet's ranges are [2.025, 5.100] and [1.975, 4.350]. The narrower ranges for DAShrink accompany its more consistent placement across datasets.

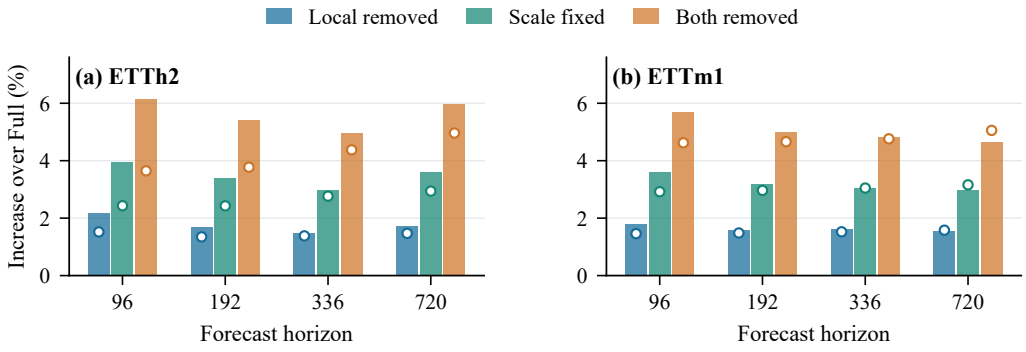


Figure 6: Relative increases over Full in Table 3. Bars show MSE and hollow circles show MAE. Every variant uses the same learned expert and router parameters.

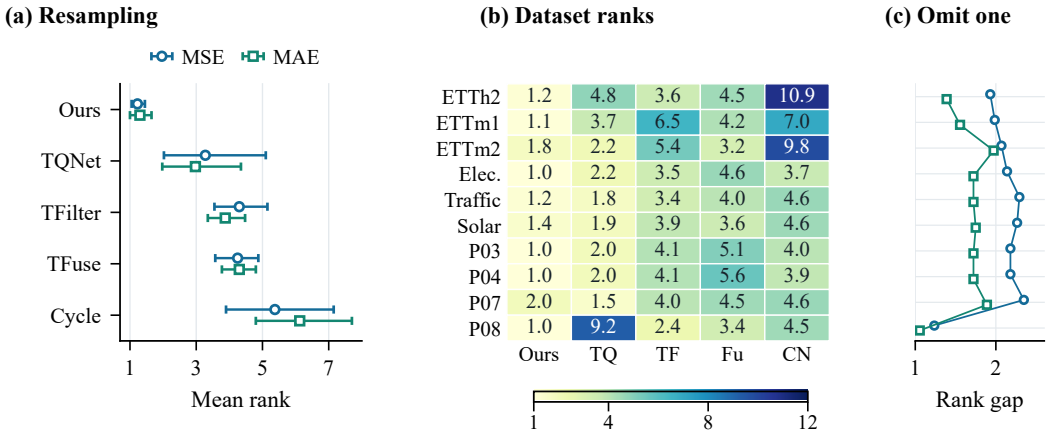


Figure 7: Rank stability at three resolutions. (a) Observed ranks and 2.5th–97.5th percentile ranges under dataset-block resampling. (b) Dataset-level mean ranks across both metrics and four horizons. (c) Gap between DAShrink and the best-ranked competitor after omitting each listed dataset; positive gaps favor DAShrink. Panels (b,c) use the rounded result matrices. All analyses hold the reported errors fixed.

Panel (b) shows that TQNet performs strongly on PEMS07 but ranks much lower on PEMS08. DAShrink leads on most datasets, with its weakest placement on PEMS07. Panel (c) removes each dataset in turn. Every resulting rank gap is positive: the MSE gaps range from 1.24 to 2.35, and the MAE gaps from 1.06 to 1.97. Omitting PEMS08 reduces the margin most. The aggregate ordering therefore persists across all ten omissions, while its size depends on the composition of the benchmark.

5 Conclusion

DAShrink regulates how far contextual forecast weights move from equal averaging. Normalized expert disagreement controls the local contribution of the router, while paired validation losses select a shared scale for each task. The resulting combination preserves a positive weight for every expert, bounds forecast displacement, and recovers equal averaging exactly when no candidate scale satisfies the selection rule. Across ten datasets and forty forecasting settings, the three-expert system achieves the best aggregate rank among twelve methods. Removing either control increases errors in the fixed-parameter ablations on two ETT datasets, and the aggregate ranking remains first after each dataset omission. This formulation separates the estimation of expert preferences from control of their contribution to the final prediction.

Reproducibility Statement

Appendix A specifies the temporal boundaries and preprocessing. Appendix B details the expert bank and training procedure; Appendix C proves the structural bounds. Tables 1 and 2 give the complete benchmark results.

Statement on the Use of Large Language Models

Large language models were used for language polishing, grammar correction, literature search assistance, and manuscript organization. They were also used to improve the clarity of mathematical descriptions and the presentation of existing results. Large language models were not used for research ideation, algorithm design, code generation, experimental execution, or the generation of scientific findings.

References

John M. Bates and Clive W. J. Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969. doi: 10.1057/jors.1969.103.

Peng Chen, Yingying Zhang, Yunyao Cheng, Yang Shu, Yihang Wang, Qingsong Wen, Bin Yang, and Chenjuan Guo. Pathformer: Multi-scale transformers with adaptive pathways for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=lJkOCMP2aW.

Gerda Claeskens, Jan R. Magnus, Andrey L. Vasnev, and Wendun Wang. The forecast combination puzzle: A simple theoretical explanation. *International Journal of Forecasting*, 32(3):754–762, 2016. doi: 10.1016/j.ijforecast.2015.12.005.

Robert T. Clemen. Combining forecasts: A review and annotated bibliography. *International Journal of Forecasting*, 5(4):559–583, 1989. doi: 10.1016/0169-2070(89)90012-5.

Francis X. Diebold and Peter Pauly. The use of prior information in forecast combination. *International Journal of Forecasting*, 6(4):503–508, 1990. doi: 10.1016/0169-2070(90)90028-A.

Francis X. Diebold and Minchul Shin. Machine learning for regularized survey forecast combination: Partially-egalitarian LASSO and its derivatives. *International Journal of Forecasting*, 35(4): 1679–1691, 2019. doi: 10.1016/j.ijforecast.2018.09.006.

Yuwei Fu, Di Wu, and Benoit Boulet. Reinforcement learning based dynamic model combination for time series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 6639–6647, 2022. doi: 10.1609/aaai.v36i6.20618.

Clive W. J. Granger and Ramu Ramanathan. Improved methods of combining forecasts. *Journal of Forecasting*, 3(2):197–204, 1984. doi: 10.1002/for.3980030207.

Bruce E. Hansen. Least squares forecast averaging. *Journal of Econometrics*, 146(2):342–350, 2008. doi: 10.1016/j.jeconom.2008.08.022.

Yifan Hu, Guibin Zhang, Peiyuan Liu, et al. TimeFilter: Patch-specific spatial-temporal graph filtration for time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 24893–24911, 2025. URL https://proceedings.mlr.press/v267/hu25ac.html.

Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. *Neural Computation*, 6(2):181–214, 1994. doi: 10.1162/neco.1994.6.2.181.

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.

Shengsheng Lin, Weiwei Lin, Xinyi Hu, Wentai Wu, Ruichao Mo, and Haocheng Zhong. CycleNet: Enhancing time series forecasting through modeling periodic patterns. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Shengsheng Lin, Haojun Chen, Haijie Wu, Chunyun Qiu, and Weiwei Lin. Temporal query network for efficient multivariate time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 37797–37814, 2025. URL https://proceedings.mlr.press/v267/lin25e.html.

Peiyuan Liu, Beiliang Wu, Yifan Hu, et al. TimeBridge: Non-stationarity matters for long-term time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 39815–39840, 2025a. URL https://proceedings.mlr.press/v267/liu25cb.html.

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, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/4054556fcaa934b0bf76da52cf4f92cb-Abstract-Conference.html.

Yong Liu, Tengge Hu, Haoran Zhang, et al. iTransformer: Inverted transformers are effective for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JePfAI8fah.

Zhining Liu, Ze Yang, Xiao Lin, et al. Breaking silos: Adaptive model fusion unlocks better time series forecasting. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 40022–40042, 2025b. URL https://proceedings.mlr.press/v267/liu25cm.html.

Ronghao Ni, Zinan Lin, Shuaiqi Wang, and Giulia Fanti. Mixture-of-linear-experts for long-term time series forecasting. In *International Conference on Artificial Intelligence and Statistics*, volume 238 of *Proceedings of Machine Learning Research*, pp. 4672–4680, 2024. URL https://proceedings.mlr.press/v238/ni24a.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://openreview.net/forum?id=Jbdc0vTOcol.

Yiming Niu, Jinliang Deng, and Yongxin Tong. PhaseFormer: From patches to phases for efficient and effective time series forecasting. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Lk9SqMQzhX.

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.

Adrian E. Raftery, Miroslav Kárný, and Pavel Ettler. Online prediction under model uncertainty via dynamic model averaging: Application to a cold rolling mill. *Technometrics*, 52(1):52–66, 2010. doi: 10.1198/TECH.2009.08104.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-MoE: Billion-scale time series foundation models with mixture of experts. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/558d48c1f08675daa636e09bfe94a89e-Paper-Conference.pdf.

Jeremy Smith and Kenneth F. Wallis. A simple explanation of the forecast combination puzzle. *Oxford Bulletin of Economics and Statistics*, 71(3):331–355, 2009. doi: 10.1111/j.1468-0084.2008.00541.x.

James H. Stock and Mark W. Watson. Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting*, 23:405–430, 2004. doi: 10.1002/for.928.

Shiyu Wang, Haixu Wu, Xiaoming Shi, et al. TimeMixer: Decomposable multiscale mixing for time series forecasting. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=7oLshfEIC2.

Xiaoqian Wang, Rob J. Hyndman, Feng Li, and Yanfei Kang. Forecast combinations: An over 50-year review. *International Journal of Forecasting*, 39(4):1518–1547, 2023. doi: 10.1016/j.ijforecast.2022.11.005.

David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992. doi: 10.1016/S0893-6080(05)80023-1.

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

Haixu Wu, Tengge Hu, Yong Liu, Hang Zhou, Jianmin Wang, and Mingsheng Long. TimesNet: Temporal 2d-variation modeling for general time series analysis. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=ju_Uqw384Oq.

Matteo Zecchin, Sangwoo Park, and Osvaldo Simeone. Adaptive learn-then-test: Statistically valid and efficient hyperparameter selection. In *International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 74018–74036, 2025. URL https://proceedings.mlr.press/v267/zecchin25a.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. doi: 10.1609/aaai.v37i9.26317.

Yunhao Zhang and Junchi Yan. Crossformer: Transformer utilizing cross-dimension dependency for multivariate time series forecasting. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=vSVLM2j9eiE.

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, 2022. URL https://proceedings.mlr.press/v162/zhou22g.html.

A Evaluation Protocol and Data

A.1 Temporal partitions

Table 4 records the exclusive row boundaries of the data loader. The fit and tune portions together form the official training segment. The later development and test intervals follow it chronologically.

Table 4: Dataset dimensions and exclusive temporal boundaries. The period is the phase cycle supplied to the model. All tasks use history length 96.

Dataset	Channels	Period	Fit end	Tune end	Dev. end	Test end
ETTh2	7	24	6048	8640	11520	14400
ETTm1	7	96	24192	34560	46080	57600
ETTm2	7	96	24192	34560	46080	57600
Electricity	321	168	12888	18412	21043	26304
Traffic	862	168	8596	12280	14035	17544
Solar-Energy	137	144	25754	36792	42048	52560
PEMS03	358	288	11006	15724	20966	26208
PEMS04	307	288	7136	10195	13593	16992
PEMS07	883	288	11853	16934	22579	28224
PEMS08	170	288	7499	10713	14284	17856

For a sequence Z, a window starting at row t uses $Z_t, \ldots, Z_{t+L-1}$ as history and predicts the next H rows. CSV and text datasets may use the preceding L observations as history when entering a new segment; targets remain within that segment. PEMS windows are constructed within each partition. Gradient updates use only fit windows. The tune segment is used both for checkpoint selection and for selecting the combination scale. Dataset-level standardization statistics are fitted through the tune boundary. History normalization in Equation (3) is an additional, window-specific transformation of the router inputs.

A.2 Metrics and aggregation

For N forecast windows, we compute

$$\mathrm{MSE} = \frac{\sum_{i,h,c}(\widehat{Y}_{ihc} - Y_{ihc})^2}{NHC}, \qquad \mathrm{MAE} = \frac{\sum_{i,h,c}|\widehat{Y}_{ihc} - Y_{ihc}|}{NHC}. \tag{11}$$

Each dataset average gives equal weight to its four horizons. For the rank summary, all twelve methods are ranked within each dataset–horizon pair, separately for MSE and MAE, with tied methods assigned their average rank. Ranks are then averaged across settings and metrics. The complete matrices are in Tables 1 and 2; values are displayed to three decimal places.

B Implementation

B.1 History statistics

For each sample and channel, normalization uses

$$\mu_{bc} = \frac{1}{L}\sum_{t=1}^{L} X_{btc}, \qquad \sigma_{bc} = \max\left\{10^{-5}, \sqrt{\frac{1}{L-1}\sum_{t=1}^{L}(X_{btc} - \mu_{bc})^2}\right\}. \tag{12}$$

Write $Z_{btc} = (X_{btc} - \mu_{bc})/\sigma_{bc}$. Let m_{bc} and v_{bc} denote the mean and sample standard deviation of the last $q = \min(24, L)$ normalized observations. With $t_j = -1 + 2(j-1)/(q-1)$, their slope is

$$\xi_{bc} = \frac{\sum_{j=1}^{q} t_j Z_{b,L-q+j,c}}{\sum_{j=1}^{q} t_j^2 + 10^{-6}}. \tag{13}$$

For lag $r < L$, define $A_r(f) = (L-r)^{-1} \sum_{t=r+1}^{L} f(t)$. The normalized lag product is

$$\kappa_{bc,r} = \frac{A_r(Z_{btc} Z_{b,t-r,c})}{\sqrt{A_r(Z_{btc}^2)}\sqrt{A_r(Z_{b,t-r,c}^2)} + 10^{-6}}. \tag{14}$$

B.2 Contextual expert

The contextual expert contains three prediction branches with separate parameters. Its lag and variable-token branches normalize the history by its sample–channel mean and standard deviation. Each branch outputs a forecast on the input scale, and a learned inner router combines these forecasts. This inner combination is part of one expert; the outer rule in Equation (1) combines that expert with DLinear and TimeFilter.

Decomposition branch. Let $\mathbf{X}_{b:,c} \in \mathbb{R}^L$ denote an input channel history. A centered moving average of width 25 with edge replication separates it into trend and residual components. Channel-shared linear maps produce

$$G^{(1)}_{b:,c} = W_{\mathrm{S}} \left(\mathbf{X}_{b:,c} - \mathrm{MA}_{25}(\mathbf{X}_{b:,c})\right) + W_{\mathrm{T}} \, \mathrm{MA}_{25}(\mathbf{X}_{b:,c}) + \mathbf{b}, \tag{15}$$

where $W_{\mathrm{S}}, W_{\mathrm{T}} \in \mathbb{R}^{H \times L}$ and $\mathbf{b} \in \mathbb{R}^H$ combines the two projection biases. The centered operation uses only values in the supplied history.

Lag branch. Let $Z_t = (X_t - \mu)/\sigma$. For valid lags from $\{1, 2, 4, 8, 24, 48\}$, form $\Delta_r Z_t = Z_t - Z_{\max(1,t-r)}$ within the observed window, repeating the first observation when a lag extends beyond its start. The lag paths project these increments over the forecast horizon,

$$P^{(r)}_{b:,c} = Z_{bLc} \mathbf{1}_H + \mathcal{P}_r(\Delta_r \mathbf{Z}_{b:,c}), \qquad G^{(2)}_{bhc} = \mu_{bc} + \sigma_{bc} \sum_{r \in \mathcal{R}} \eta_{bhcr} P^{(r)}_{bhc}. \tag{16}$$

The index set $\mathcal{R}$ contains these six lag paths, persistence $P^{(\mathrm{pers})}_{bhc} = Z_{bLc}$, and an affine history projection $\mathbf{P}^{(\mathrm{id})}_{b:,c} = W_{\mathrm{id}} \mathbf{Z}_{b:,c} + \mathbf{b}_{\mathrm{id}}$. For $L = 96$, all six lags are valid. The path weights are

$$\boldsymbol{\eta}_{bhc} = \mathrm{softmax}(g_\eta([Z_{bLc}, m_{bc}, v_{bc}, \xi_{bc}, \kappa_{bc,1}, \kappa_{bc,2}, \kappa_{bc,4}, \kappa_{bc,8}, \kappa_{bc,24}, \kappa_{bc,48}]) + \mathbf{t}_h). \tag{17}$$

Here g_η is a $10 \to 32 \to 8$ MLP with GELU between its linear layers, and $\mathbf{t}_h \in \mathbb{R}^8$ is a learned horizon bias.

Variable-token branch. Each channel history is projected to a width-64 token with a channel embedding. Two pre-norm Transformer layers exchange information among these tokens. Each layer uses four attention heads and a width-128 GELU feed-forward block, with dropout 0.1. A linear output head maps each resulting token to H values, which are rescaled by σ_{bc} and shifted by μ_{bc} to give $G^{(3)}$. If $\boldsymbol{\pi}_{bhc} \in \Delta^2$ denotes the inner router weights, the contextual expert outputs

$$F^{\mathrm{N}}_{bhc} = \sum_{j=1}^{3} \pi_{bhcj} G^{(j)}_{bhc}. \tag{18}$$

Inner router. Let $\boldsymbol{\delta}_{bc} = (Z_{bLc} - Z_{b,L-1,c}, Z_{bLc} - Z_{b,L-3,c}, Z_{bLc} - Z_{b,L-7,c})^\top$. The inner router uses the ten-dimensional history vector

$$\boldsymbol{\psi}_{bc} = [Z_{bLc}, m_{bc}, v_{bc}, \xi_{bc}, \boldsymbol{\delta}_{bc}^\top, \kappa_{bc,1}, \kappa_{bc,8}, \kappa_{bc,24}]. \tag{19}$$

Normalize and clip the three branch forecasts as $\widetilde{G}^{(j)} = \mathrm{clip}((G^{(j)} - \mu)/\sigma, -20, 20)$. At each forecast coordinate, $\mathbf{r}_{bhc} \in \mathbb{R}^5$ concatenates these three values with their mean $\bar{G}$ and spread $\sqrt{\frac{1}{3}\sum_j (\widetilde{G}^{(j)} - \bar{G})^2 + 10^{-6}}$. Then

$$\mathbf{h}^{\mathrm{N}}_{bhc} = g_{\mathrm{hist}}(\mathrm{clip}(\boldsymbol{\psi}_{bc}, -10, 10)) + W_{\mathrm{br}} \mathbf{r}_{bhc} + \mathbf{b}_{\mathrm{br}} + \mathbf{e}^{\mathrm{N}}_h, \tag{20}$$

$$\boldsymbol{\pi}_{bhc} = \mathrm{softmax}\big(\mathrm{clip}\big(V \, \mathrm{GELU}(\mathrm{LN}(\mathbf{h}^{\mathrm{N}}_{bhc})) + \mathbf{b}_{\mathrm{N}} + \mathbf{c}_{hc}, -30, 30\big)\big). \tag{21}$$

The history MLP has widths $10 \to 64 \to 64$, with GELU between its linear layers. $W_{\mathrm{br}} \in \mathbb{R}^{64 \times 5}$ projects branch features; $\mathbf{e}^{\mathrm{N}}_h \in \mathbb{R}^{64}$ is a horizon embedding. The output map $V \in \mathbb{R}^{3 \times 64}$ and coordinate bias $\mathbf{c}_{hc} \in \mathbb{R}^3$ produce the three branch logits.

B.3 External experts and outer router

DLinear uses channel-shared projections and moving-average width 25. TimeFilter uses non-overlapping patches of length 2 on the seven-channel ETT datasets and 32 on the other datasets in this evaluation. The outer router receives

$$\boldsymbol{\phi}_{bhc} = [\widetilde{F}^{(1)}_{bhc}, \ldots, \widetilde{F}^{(K)}_{bhc}, \overline{F}_{bhc}, s_{bhc}, Z_{bLc}, \xi_{bc}] \in \mathbb{R}^{K+4}. \tag{22}$$

Using learned horizon and channel embeddings $\mathbf{e}_h, \mathbf{e}_c \in \mathbb{R}^d$, the router computes

$$\mathbf{h}_{bhc} = \mathrm{GELU}(\mathrm{LN}(W_1 \boldsymbol{\phi}_{bhc} + \mathbf{b}_1 + \mathbf{e}_h + \mathbf{e}_c)), \tag{23}$$

$$\mathbf{a}_{bhc} = \mathrm{softmax}(\mathrm{clip}(W_2 \mathbf{h}_{bhc} + \mathbf{b}_2, -20, 20)). \tag{24}$$

Here $d = 32$, $W_1 \in \mathbb{R}^{d \times (K+4)}$, and $W_2 \in \mathbb{R}^{K \times d}$; all softmax operations act over the expert or branch index. The disagreement s_{bhc} uses denominator $K - 1$ and the clipped predictions from Equation (3). The forecast and its range bound use the original predictions.

Table 5: Architecture and combination constants used in the described implementation.

Quantity	Value
History length	96
Outer router	$7 \to 32 \to 3$
Contextual expert: token width / heads	64 / 4
Contextual expert: FFN width / layers	128 / 2
Contextual expert: dropout	0.1
Maximum contextual contribution ρ	0.25
Disagreement coefficient τ	1
Outer prediction / logit clip	$[-20, 20]$
Inner-router logit clip	$[-30, 30]$
Disagreement clip	$[0, 8]$
Router KL coefficient	0.05
Gradient-norm limit	1
Eligibility threshold	-10^{-8}

B.4 Optimization and selection order

Expert and router updates use separate AdamW optimizers with mixed-precision scalers and unit gradient-norm clipping. During a fit batch, the expert objective

$$\mathcal{L}_{\mathrm{experts}} = \sum_{k=1}^{K} \frac{1}{BHC} \sum_{b,h,c} (F^{(k)}_{bhc} - Y_{bhc})^2 \tag{25}$$

is evaluated first and updates expert parameters. The router objective is then evaluated using detached expert predictions and updates only router parameters. This alternating procedure is repeated over fit batches. The same fit partition is used by both objectives.

Tune losses select checkpoints. With the selected neural parameters fixed, one pass over tune windows evaluates all scales in $\Gamma = \{0, 0.25, 0.5, 0.75, 1\}$. The per-window errors are paired with the errors of the uniform combination on the same targets. The rule in Equations (6)–(7) chooses a single scale for the task and holds it fixed during evaluation. At each test window, compute the expert forecasts, the features in Equation (22), and $\mathbf{a}$ and β. Equations (1)–(2) then give the weights and output using the selected $\gamma^\star$.

C Proofs and Interpretation

C.1 Proof of Proposition 1

Because both $\mathbf{u}$ and $\mathbf{a}$ are simplex vectors and $0 \leq \alpha < 1$, their convex combination is also a simplex vector. For every coordinate, $0 \leq a_k \leq 1$ gives

$$\frac{1-\alpha}{K} \leq w_k \leq \frac{1-\alpha}{K} + \alpha. \tag{26}$$

The lower limit decreases and the upper limit increases with α. Substituting $\alpha \leq \rho$ proves the uniform coordinate bounds.

For the distance bound, write $\mathbf{a} = \sum_k a_k \mathbf{e}_k$, where $\mathbf{e}_k$ is the k-th simplex vertex. Convexity of the norm gives

$$\|\mathbf{a} - \mathbf{u}\|_1 \leq \sum_k a_k \|\mathbf{e}_k - \mathbf{u}\|_1 = 2(1 - 1/K). \tag{27}$$

Multiplying by α proves the claimed ℓ_1 bound.

For the forecast bound, put $m = \min_k F^{(k)}$, $M = \max_k F^{(k)}$, and $\bar{F}_0 = K^{-1} \sum_k F^{(k)}$. The contextual forecast $\sum_k a_k F^{(k)}$ lies in $[m, M]$. At least one expert attains each endpoint, so

$$M - \bar{F}_0 \leq \frac{K-1}{K}(M-m), \qquad \bar{F}_0 - m \leq \frac{K-1}{K}(M-m). \tag{28}$$

Consequently,

$$|\widehat{Y}(\gamma) - \widehat{Y}(0)| = \alpha \left| \sum_k a_k F^{(k)} - \bar{F}_0 \right| \tag{29}$$

$$\leq \alpha(1 - 1/K)(M-m) \leq \rho\gamma(1-1/K)(M-m). \tag{30}$$

Setting $\gamma = 0$ gives $\alpha = 0$ and exact equality to the uniform combination. This completes the proof.

C.2 What disagreement controls

Within the unclipped region, $\beta(s) = \rho e^{-\tau s}$ has derivative $-\tau\beta(s)$. Thus the coefficient is nonincreasing in disagreement. The forecast displacement bound also contains the raw range $M - m$, which can increase at the same time. Its value therefore depends on both factors. Figure 2(c) gives an explicit example of their interaction.

C.3 Equal averaging and squared error

Equal weights are invariant to permutations of the experts. For any simplex weights, the squared forecast error satisfies the ambiguity identity

$$(\widehat{Y} - Y)^2 = \sum_k w_k (F^{(k)} - Y)^2 - \sum_k w_k (F^{(k)} - \widehat{Y})^2. \tag{31}$$

It follows by expanding both squared terms and using $\sum_k w_k F^{(k)} = \widehat{Y}$. The second term measures the dispersion of expert forecasts around their weighted average.