LABEL-FREE COMBINATION OF ZERO-SHOT TIME-SERIES FOUNDATION MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Zero-shot time-series foundation models (TSFMs) forecast unseen series without training, and since no single model dominates, practitioners pool several. On a new domain, however, no target label is available to decide how to combine them or which of them to include. We argue that both decisions follow from one observation, that without labels a member's error can only be contained, not corrected. Averaging distributions rather than quantiles contains it, and on 97 GIFT-Eval and 98 TIME configurations the quantile mean trails the linear opinion pool (LOP) on nearly all 120 subsets of seven TSFMs, while the best setting of every other rule family lands within half a percent of LOP. Which members enter the pool matters far more, yet a subset chosen in advance, even with labels from other sources, recovers only part of the gain of choosing with the target's labels, so selection has to happen inside the combination, one forecast at a time. We propose HuberLOP, which replaces the equal weights of LOP with Huber weights computed from each member's distance to a robust center of the pool. It improves on LOP on both benchmarks, and when three naive forecasters join the pool its conservative variant degrades by 0.9% on both benchmarks, where LOP degrades by 14.0% on GIFT-Eval and 12.0% on TIME. With no training and one threshold, it is a simple yet effective baseline for label-free TSFM combination. Code is available at `https://anonymous.4open.science/r/ICLR27-TSFM`.

1 INTRODUCTION

Time-series foundation models (TSFMs) forecast series they have never seen from the observed context alone, which makes them zero-shot forecasters (Das et al., 2024; Ansari et al., 2025; Auer et al., 2025; Woo et al., 2024). Since they differ in architecture and training data, no single model performs best across datasets, frequencies and horizons (Aksu et al., 2024; Qiao et al., 2026). A practitioner therefore deploys several models and combines their forecasts, and as Figure 1 shows, the combination usually outperforms its best member. Existing combiners for TSFMs, such as learned gates (Datadog, 2026), FFORMA-style meta-learners (Montero-Manso et al., 2020) and adaptive arbitration (Das et al., 2025), learn how to combine from labeled data. They require target labels against which the members' forecasts can be scored, inference with every TSFM on that data to produce the forecasts they learn from, and a new round of training whenever the domain changes.

These requirements conflict with how TSFMs are deployed. A series from a new domain arrives with its context and no label, so there is nothing to learn the combination from. Forecasting systems in practice also serve large numbers of series and refresh their forecasts as new observations arrive, so any step that trains or tunes the combination per series or per domain adds cost and latency that such systems cannot absorb. Tuning on the observed context does not escape this, since backtesting on the history recovers pseudo-labels only through repeated inference with every model and a fitting step for every series, and it consumes a history that is often only a few horizons long. The combination must therefore be both label-free and training-free, decided from the member forecasts alone at the moment they are produced.

We argue that such a rule cannot correct a member's error, because without labels the forecasts cannot distinguish a model that is wrong from one that is right but different, and that it can only limit how far any single member moves the combined forecast. We take this limit as the premise of the paper

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.

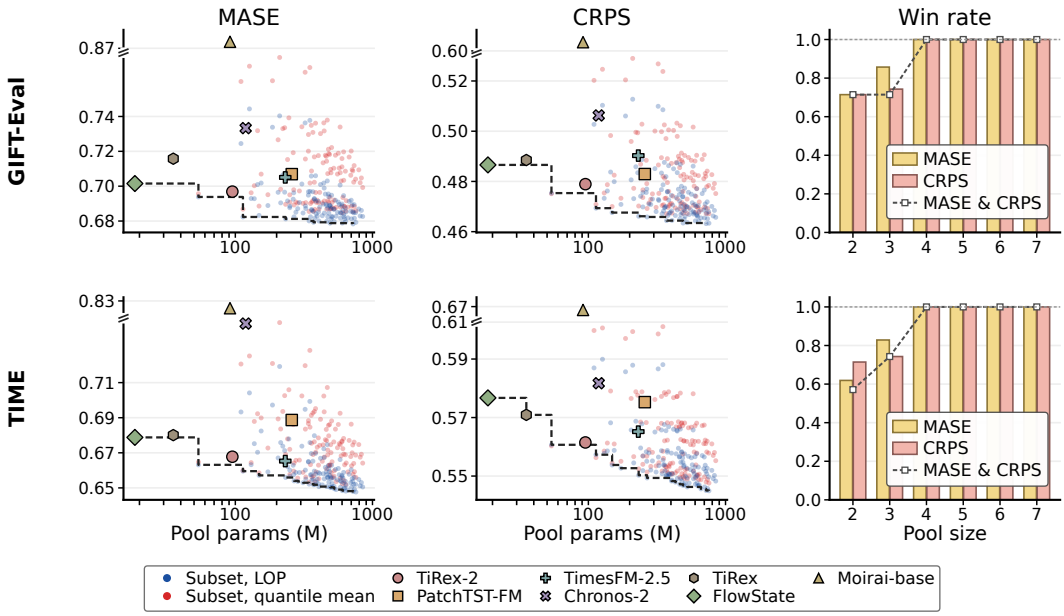

Figure 1: **Combining helps, and which models are combined matters most.** MASE and CRPS against total parameter count for the seven TSFMs (markers) and all 120 subsets combined by LOP (blue) or the quantile mean (red), on GIFT-Eval (top) and TIME (bottom). The dashed line is the Pareto frontier. Right, the share of subsets whose LOP beats their best member.

and examine its consequences for the two decisions a practitioner faces, namely how to combine the members and which members to combine.

For how to combine, the premise favors averaging distributions over averaging quantiles. Quantile averaging lets a member whose scale is off shift the combination at every level, whereas the linear opinion pool (LOP) caps the influence of each member at its mass $1/K$, so that an outlying member widens the forecast rather than shifting it. Once every member is capped in this way, the rules that cap their members are expected to differ only in details. For which members to combine, however, the uniform share $1/K$ becomes a limitation, since a member that performs poorly on the target domain still contributes its full share. The pool thus becomes the main lever, yet it is difficult to choose in advance. Whether a member helps depends on the target domain, which a new series does not reveal, and rankings measured on other domains transfer only as far as the domains resemble each other. Ranking members by their agreement with the consensus helps only in part, because a member may differ from the rest for being wrong or for being right, and a fixed ranking discards both alike.

Motivated by this, we ask how the share of each member can be set without labels. To this end, we propose Huber-weighted LOP (HuberLOP), a simple yet effective baseline that moves the choice of members inside the combination. Specifically, at each series and horizon step, a robust center of the member quantiles assigns every member a Huber weight that remains one within a band around the center and decreases once the member leaves it, and LOP pools the member CDFs with these weights instead of $1/K$. Members that differ within the band thus keep their full share, which preserves the diversity of the pool, whereas a member that departs is discounted in that forecast only. This formulation can be seen as a generalization of LOP, since it reduces to LOP whenever the members agree. HuberLOP requires no label, no training and a single threshold, and its conservative variant $\text{HuberLOP}_{\text{con}}$ discounts a member as soon as any of its quantiles leaves the band.

We conduct extensive experiments on GIFT-Eval and TIME to examine each step of this argument. Section 4.2 shows that distributions outperform quantiles and that the rules capping their members lie within a narrow band. Section 4.3 shows that the pool moves accuracy far more than the rule and cannot be chosen well in advance. Section 4.4 shows that HuberLOP discounts the member a label-chosen subset would drop, varies that discount from forecast to forecast, and, in its conservative variant, remains nearly unchanged when naive forecasters enter the pool while LOP degrades by

Table 1: **What each combination method requires.** Synapse and model portfolios score members on withheld context, which serves as labels. Adaptive weights change with the series or the forecast, and a rule keeps agreeing members when it discards none of them while they agree.

Method	No labels	No training	No extra inference	Adaptive weights	Keeps agreeing members
Toto Family-and-Friends (Datadog, 2026)	×	×	×	✓	×
FFORMA (Montero-Manso et al., 2020)	×	×	×	✓	×
TimeCopilot ensemble (TimeCopilot, 2026)	×	✓	×	×	×
Synapse (Das et al., 2025)	×	✓	×	✓	×
Model portfolios (Kayaalp et al., 2026)	×	✓	×	✓	×
Quantile averaging (Lichtendahl et al., 2013)	✓	✓	✓	×	✓
Linear opinion pool (Genest & Zidek, 1986)	✓	✓	✓	×	✓
Trimmed opinion pool (Jose et al., 2014)	✓	✓	✓	✓	×
HuberLOP (ours)	✓	✓	✓	✓	✓

more than ten percent. Our claim, in one sentence, is that label-free combination of zero-shot TSFMs should average distributions rather than quantiles, and should select forecasts rather than models.

The main contributions of this paper are summarized as follows.

- We propose HuberLOP, a simple yet effective label-free and training-free baseline for combining zero-shot TSFMs, which selects forecasts rather than models by reweighting the linear opinion pool with Huber weights.
- We provide an extensive analysis on two benchmarks and 32 combination settings, showing that quantile averaging loses to LOP on nearly every subset, that the rules capping their members stay within half a percent of LOP in CRPS, and that the choice of members moves accuracy several times more than the choice among these rules.
- We show that what a model gains from the pool is not what it gives, and that choosing the pool in advance with labels from other sources recovers only part of the gain of choosing with the target's labels, whereas HuberLOP comes close to it without labels, without selecting any model and without choosing how many to keep, and, in its conservative variant, remains nearly unaffected by contamination.

2 Related Work

Zero-shot time-series foundation models. Pretrained forecasters now span several architectures and training recipes, including patched decoders such as TimesFM (Das et al., 2024; Google Research, 2026) and Lag-Llama (Rasul et al., 2023), tokenized language modeling in Chronos (Ansari et al., 2024; 2025), cross-frequency universal models such as Moirai (Woo et al., 2024), long-context in-context learners such as TiRex (Auer et al., 2025), state-space and patch-based encoders such as FlowState (Graf et al., 2025) and PatchTST-FM (Wen et al., 2026), and billion-parameter models trained at scale (Liu et al., 2026; Khwaja et al., 2026). Benchmarks such as GIFT-Eval (Aksu et al., 2024) and TIME (Qiao et al., 2026) rank them by accuracy averaged over many datasets, frequencies and horizons, but the ranking changes from one configuration to the next and the leaderboard does not say which model to trust on a new series. We study the setting that instability creates, in which several public zero-shot forecasters face one unlabeled series with no budget to fine-tune or validate.

Label-free combination of zero-shot TSFMs. Combining forecasts is a classical improvement over selection (Bates & Granger, 1969; Timmermann, 2006; Wang et al., 2023), with the long-standing observation that simple equal weights are hard to beat once weights must be estimated (Stock & Watson, 2004; Smith & Wallis, 2009). For predictive distributions the two families are quantile averaging (Lichtendahl et al., 2013; Busetti, 2017) and opinion pools (Genest & Zidek, 1986; Gneiting & Ranjan, 2013), and a recent line of work reweights or trims the members of either from observed errors (Taylor & Meng, 2026; Taylor, 2026; Yu et al., 2026; Fakoor et al., 2023), from latent reliability models (Dawid & Skene, 1979; Guha et al., 2024), or by trimming outlying members (Jose et al., 2014; Wang et al., 2024). A trimmed pool fixes in advance how many members to discard, whereas the weights we use fall continuously with a member's distance from a robust center and equal the

uniform pool exactly when the members agree, which is why, as Appendix B.1 shows, they absorb more contamination than a fixed trim count. Krüger (2026) relates the linear pool's advantage to member disagreement. Systems built specifically for TSFMs go further and use information beyond the forecasts at hand, such as learned gates (Datadog, 2026), FFORMA-style meta-learners (Montero-Manso et al., 2020), adaptive arbitration among experts (Das et al., 2025), test-time portfolios (Kayaalp et al., 2026), ensemble entries on public leaderboards (TimeCopilot, 2026), and tool-selecting or self-validating agents (USTC-AGI, 2026; Garza & Rosillo, 2025; Cao et al., 2025; Dalpha Agent Team, 2026). Table 1 summarizes what each method needs at deployment, and Appendix A.4 separates them further by what they are allowed to observe. We address the strictest case, in which deployment provides only unlabeled forecast outputs, and ask a question this literature mostly leaves implicit, namely how much of the accuracy is in the rule at all and how much in the choice of members. Our premise answers the first part. Without labels a rule can only bound each member's influence, the opinion pool bounds it at the member's mass, and once members are bounded the combination puzzle reappears in distributional form, since no fixed weighting beats equal weights by more than half a percent. Huber estimation (Huber, 1964; 1981) supplies the one ingredient our rule adds to that equal-weight pool.

3 Label-Free Combination and Huber-Weighted LOP

3.1 Problem Setting

At one target variable and horizon step, which we call a forecast cell, each of K pretrained forecasters predicts Q quantiles $\mathbf{q}_k = (q_k(\tau_1), \ldots, q_k(\tau_Q))$ at common levels $0 < \tau_1 < \cdots < \tau_Q < 1$, from the observed context alone. The desired output is a combined forecast $\widehat{\mathbf{q}}$ with nondecreasing quantiles. No forecast-period target is observed, and no parameter is learned from target-series errors. The setting we target is a new series from a domain the models have not seen, where the rule reads only the current forecasts. Rankings of the members measured on other series appear in Section 4.3 only as a diagnostic of how far a choice of members could reach. Every rule below is applied independently at each forecast cell, so a rule is a map from K quantile vectors to one.

3.2 Quantile Averaging and Linear Opinion Pool

Quantile averaging combines values. At each level τ_j it returns a location estimate of the K member values, such as their mean, median, trimmed mean or Huber center. The linear opinion pool (LOP) combines probabilities instead. Each member's quantiles define a CDF F_k by linear interpolation, with the remaining mass placed at the outermost quantiles, and LOP returns the quantiles of the pooled CDF

$$\bar{F} = \frac{1}{K} \sum_{k=1}^{K} F_k. \tag{1}$$

A member can move $\bar{F}$ at any value by at most $1/K$, so an outlying member widens the pooled forecast rather than shifting it. The same bound holds for every member alike, however, so a member that is far from all the others keeps its full share at every value.

3.3 Huber-Weighted LOP

We propose Huber-weighted LOP (HuberLOP), which keeps Eq. (1) but lets each member's share depend on how far it sits from the rest of the pool, recomputed in every forecast cell. It proceeds in three steps.

Robust center. At each level τ_j, we compute a robust center h_j of the K member values, the Huber M-estimate started from their median, and a robust scale s_j, the median absolute deviation. Both ignore a minority of outlying members.

Per-level weights. Each member's standardized distance $d_{kj} = |q_k(\tau_j) - h_j|/s_j$ gives it the Huber weight

$$\omega_{kj} = \min\big(1,\, c/d_{kj}\big), \tag{2}$$

which is one inside the band $d_{kj} \le c$ and decays as c/d_{kj} outside it.

Weighted pool. LOP multiplies each member's whole CDF by one weight, so we collapse the Q per-level weights into $w_k = \frac{1}{Q}\sum_j \omega_{kj}$ and return the quantiles of

$$\bar{F}_w = \frac{\sum_k w_k F_k}{\sum_k w_k}. \tag{3}$$

Three properties follow from this construction. When every member lies within the band at every level, all weights are one and HuberLOP is exactly LOP, so it changes nothing in a pool whose members agree. When one member moves away from the others while most stay together, the center and scale stay put, its distances grow and its weight falls toward zero, so its share of $\bar{F}_w$ vanishes, whereas in LOP it would stay at $1/K$ however far the member went. Because the weights are recomputed in every cell, a member discounted in one forecast keeps its full weight in the next, so the rule selects forecasts rather than models.

Two choices remain. The threshold c sets how far a member may stray before it loses weight. Since a weight here discounts a whole distribution rather than one value, we use $c = 3$, wider than the constant 1.345 that robust location estimation conventionally uses. The collapse over levels sets how strictly a partly outlying member is treated. The mean discounts it by how far and at how many levels it strays, whereas the conservative variant HuberLOP$_{\texttt{con}}$ takes the minimum, discounting a member as soon as one of its quantiles leaves the band, and keeps $c = 1.345$. Section 4.4 studies the threshold as a dial between the two. Neither variant learns anything, and each adds KQ weight evaluations to one LOP inversion per cell.

4 Experiments

4.1 Setup

The curated pool holds seven public TSFMs in their zero-shot setting, namely TiRex-2-ZS and TiRex (Auer et al., 2025), PatchTST-FM-r1 (Wen et al., 2026), TimesFM-2.5 (Das et al., 2024), Chronos-2 in its `autogluon/chronos-2-synth` release (Ansari et al., 2025), FlowState-r1.1 (Graf et al., 2025), and Moirai-base (Woo et al., 2024), which we call by their short names below. They were chosen by model class rather than by observed accuracy and restricted to models whose reported pretraining data exclude GIFT-Eval, so that every forecast we combine is zero-shot on the benchmark rather than on data the model has seen. TIME (Qiao et al., 2026) is a second benchmark of 98 configurations from 39 source groups, and because it normalizes and averages scores by its own convention, absolute scores are not comparable across the two benchmarks. Every rule receives the same archived nine-quantile forecasts on all 97 GIFT-Eval configurations (Aksu et al., 2024), each a dataset at one sampling frequency and horizon, and on the 98 configurations of TIME, and the subset analysis enumerates all 120 multi-model subsets. MASE (mean absolute scaled error) scores the median and CRPS (continuous ranked probability score, as the mean weighted quantile loss over the nine levels) the full forecast. Each configuration score is divided by that of the seasonal-naive forecaster and the geometric mean over configurations is reported. The label-free baselines are the quantile mean, median and trimmed mean, quantile-wise Huber, angular combination, LOP, and the depth-weighted and trimmed opinion pools, with protocols and implementation in Appendix A.

4.2 Combination Rules on a Curated Pool

Comparison of combination rules. Every combination rule in the literature is justified by beating the previous one, which suggests that the rule is where accuracy is decided. To test this, we hold the seven-TSFM pool fixed and compare the label-free rules on both benchmarks, next to the seven members themselves. As shown in Table 2, the quantile mean is the outlier at 0.4752 CRPS on GIFT, while the robust quantile rules and LOP land between 0.4657 and 0.4672, a span of 0.3%, and most of the mean's gap is one member, since dropping Moirai-base moves it to 0.4672. Across the 120 subsets of Figure 1 the quantile mean loses to LOP in nearly every case. No ordering among the quantile rules and LOP survives both benchmarks, as LOP leads on GIFT and the trimmed mean on TIME. HuberLOP is ahead of LOP in every column, by 0.29% CRPS on GIFT and 0.25% on TIME, winning 56 of 97 and 83 of 98 configurations, and HuberLOP$_{\texttt{con}}$ leads by 0.49% on TIME, winning 75 of 98. Across 32 settings of these rule families, the best setting of each family stays within half a

Table 2: **Combination rules on the seven-TSFM pool.** Geometric means of seasonal-naive-normalized MASE and CRPS over 97 GIFT-Eval and 98 TIME configurations. Lower is better, best in bold. Settings are in Appendices A.1 and B.7.

		GIFT-Eval		TIME	
Category	Method	MASE (↓)	CRPS (↓)	MASE (↓)	CRPS (↓)
Single TSFM	TiRex-2 (Auer et al., 2025)	0.6969	0.4789	0.6678	0.5615
	PatchTST-FM (Wen et al., 2026)	0.7069	0.4829	0.6887	0.5753
	TimesFM-2.5 (Das et al., 2024)	0.7050	0.4903	0.6653	0.5652
	Chronos-2 (Ansari et al., 2025)	0.7333	0.5062	0.7437	0.5817
	TiRex (Auer et al., 2025)	0.7158	0.4885	0.6801	0.5709
	FlowState (Graf et al., 2025)	0.7015	0.4866	0.6788	0.5767
	Moirai-base (Woo et al., 2024)	0.8737	0.6035	0.8259	0.6689
Quantile-based	Quantile averaging (Lichtendahl et al., 2013)	0.6992	0.4752	0.6641	0.5553
	Quantile median (Busetti, 2017)	0.6862	0.4672	0.6519	0.5474
	Trimmed mean	0.6844	0.4662	0.6501	0.5470
	Quantile-wise Huber (Huber, 1964)	0.6852	0.4665	0.6517	0.5476
Hybrid approach	Angular combination (Taylor & Meng, 2026)	0.6860	0.4665	0.6534	0.5489
Distribution-based	Linear opinion pool (Genest & Zidek, 1986)	0.6826	0.4657	0.6505	0.5483
	Depth-weighted pool (Taylor, 2026)	0.6829	0.4648	0.6499	0.5468
	Trimmed opinion pool (Jose et al., 2014)	0.6842	0.4657	0.6500	0.5466
	HuberLOP (ours)	**0.6818**	**0.4644**	0.6497	0.5469
	HuberLOP$_{\text{con}}$ (ours)	0.6838	0.4657	**0.6489**	**0.5456**

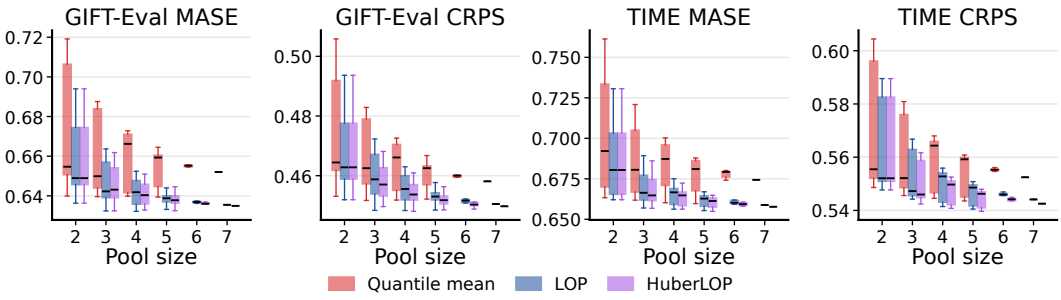

Figure 2: **Composition matters most in small pools.** Accuracy of all subsets of each size under the quantile mean, LOP and HuberLOP, scored on every fifth forecast window.

percent of LOP on both benchmarks, as Appendix B.7 shows. This implies that once a rule bounds its members, the choice among such rules keeps CRPS within half a percent of LOP.

4.3 Pool Composition and Selection

Effect of pool composition. If the rule cannot move accuracy, the lever left is which models to deploy, and the vertical spread of the subsets in Figure 1 already suggests how much it moves. To measure it, we score all 120 multi-model subsets under the quantile mean, LOP and HuberLOP. As shown in Figure 2, accuracy improves as members are added and the boxes narrow as the pool grows, while at every size the quantile mean sits above the distribution-based rules and HuberLOP at or below LOP. With the rule fixed, the best and worst of the 35 fixed trios differ by 5.2% CRPS on GIFT and 3.6% on TIME, and a trio chosen per configuration spans 15.7% on GIFT, many times the 0.3% spread among the robust quantile rules and LOP. This implies that swapping one member of a small pool changes accuracy more than replacing the rule does, which raises the question of what each member is worth.

Gain and contribution of each member. The heuristics a leaderboard suggests for choosing members, keeping the accurate models, the large ones or the ones the pool improves most, assume that a member's visible properties say what it adds to a pool. To test this, we score every member's

Table 3: **Choosing the pool in advance.** Pool selection by pool size k on both benchmarks, every subset combined by LOP. *Oracle* chooses with the target's labels and *Profiled* with labels of the other source datasets, so both serve as references that the label-free setting cannot reach. Rules (3) and (4), like HuberLOP, use no labels, and HuberLOP also needs no k, so its score is the same for every k.

	GIFT-Eval						TIME					
	MASE			CRPS			MASE			CRPS		
Selection rule	$k=3$	$k=5$	$k=6$	$k=3$	$k=5$	$k=6$	$k=3$	$k=5$	$k=6$	$k=3$	$k=5$	$k=6$
(1) *Oracle*	0.6525	0.6671	0.6739	0.4444	0.4533	0.4586	0.6370	0.6407	0.6447	0.5346	0.5388	0.5422
(2) *Profiled*	0.6910	0.6783	0.6792	0.4686	0.4639	0.4634	0.6540	0.6486	0.6476	0.5547	0.5451	0.5450
(3) Random	0.6928	0.6855	0.6839	0.4765	0.4686	0.4669	0.6615	0.6536	0.6518	0.5577	0.5511	0.5494
(4) Largest	0.6862	0.6869	0.6861	0.4701	0.4701	0.4685	0.6557	0.6548	0.6530	0.5496	0.5519	0.5502
HuberLOP	**0.6818**	**0.6818**	**0.6818**	**0.4644**	**0.4644**	**0.4644**	0.6497	0.6497	0.6497	0.5469	0.5469	0.5469
HuberLOP$_{\texttt{con}}$	0.6838	0.6838	0.6838	0.4657	0.4657	0.4657	**0.6489**	**0.6489**	**0.6489**	**0.5456**	**0.5456**	**0.5456**

peer gain and leave-one-out contribution. The peer gain of member k is the percent CRPS reduction when its forecast is replaced by the uniform LOP of the other $K-1$ members, and its contribution is the reduction the peers' LOP obtains when the member is added, both scored on the target. As shown in Table 10 of Appendix B.5, peer gain orders the members exactly as their own accuracy does, because it measures how far a member sits from the consensus, while contribution follows accuracy only partly and model size not at all. The two disagree most at the ends. Moirai-base gains by far the most from its peers, 23.2% CRPS on GIFT, yet it is the only member whose contribution is negative on both benchmarks, while FlowState, the smallest model, gains little and contributes the most. This implies that a member the pool can repair is not a member the pool should keep.

Choosing the pool in advance. The practical question is how much of the composition headroom a choice of members can recover before forecasting and without labels. To answer it, we compare two label-free rules for choosing k members against two references that need labels, all combined by LOP and evaluated leave-one-source-out on both benchmarks. (1) The *oracle* picks the best subset for each configuration with its own labels, an unreachable upper bound. (2) The *profiled* subset fixes the subset with the best average score on the other source datasets, as a practitioner reading a leaderboard would, which requires their labels and a pass of every TSFM over them. (3) *Random* averages over all subsets of size k. (4) *Largest* keeps the k members with the most parameters. As shown in Table 3, the oracle improves on a random trio by 6.7% CRPS on GIFT and 4.1% on TIME, and even the profiled subset recovers only part of that. Keeping the largest models is worse than random from five members on, since it adds Moirai-base before the small FlowState and TiRex. HuberLOP on GIFT and HuberLOP$_{\texttt{con}}$ on TIME beat every label-free rule at every size and come within 0.2% CRPS of the profiled subset without its labels and without choosing k. This implies that most of the headroom is configuration-specific, as the gap to the oracle shows, and that weighting forecasts recovers what a choice of members made with labels recovers.

4.4 Per-Forecast Selection with HuberLOP

HuberLOP and the best fixed subset. Section 4.3 leaves the pool unchosen, and HuberLOP moves the choice inside the combination. To see whether it recovers what a choice of members would, we compare the three rules on all seven members and on each rule's best fixed subset. As shown in Table 9 of Appendix B.4, every best subset leaves out Moirai-base, and on those subsets HuberLOP and LOP meet at 0.4632 CRPS on GIFT and differ by 0.06% on TIME. On all seven members, HuberLOP discounts Moirai-base without being told which member it is, its average weight falling to 0.92 while the other six stay at 0.96 to 0.99, and recovers 54% of the best subset's CRPS gain on GIFT, with HuberLOP$_{\texttt{con}}$ recovering 80% on TIME. This implies that the gain of HuberLOP is the selection the best subset makes, carried out forecast by forecast.

Per-forecast weights. A rule that only removed Moirai-base would be a fixed selection in disguise. To see whether HuberLOP selects forecasts instead, we record the weight it gives each member in every forecast cell of the curated pool, 40.7 million cells on GIFT and 7.8 million on TIME. As shown in Figure 3, Moirai-base keeps its full weight in about half of its forecasts and is strongly

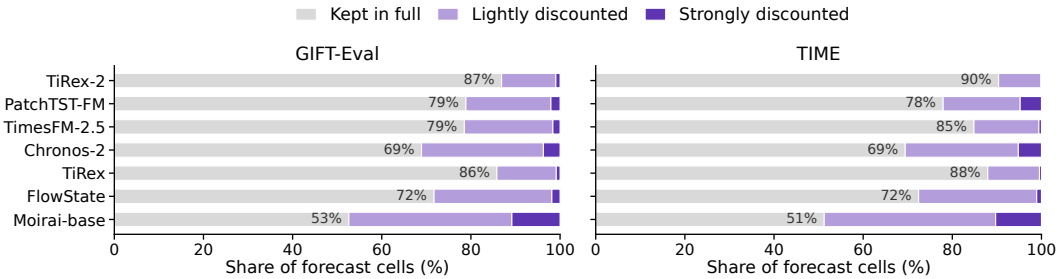

Figure 3: **HuberLOP selects forecasts, not models.** Share of each member's forecast cells kept in full ($w_k \geq 0.999$), lightly discounted or strongly discounted ($w_k < 0.75$), averaged over configurations.

Table 4: **Ablation of HuberLOP.** *CDF* averages distributions instead of quantiles, and *Huber* replaces equal weights with Huber weights. Contaminated pools add three naive forecasters. Best in bold, second best underlined.

			GIFT-Eval				TIME			
			Curated Pool		Contaminated		Curated Pool		Contaminated	
Rule	CDF	Huber	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Quantile mean	×	×	0.6992	0.4752	0.9190	0.6271	0.6641	0.5553	0.8884	0.7205
Linear opinion pool	✓	×	<u>0.6826</u>	<u>0.4657</u>	0.7336	0.5308	0.6505	0.5483	0.6859	0.6140
Quantile-wise Huber	×	✓	0.6852	0.4665	0.7214	<u>0.4872</u>	0.6517	0.5476	0.6821	<u>0.5667</u>
HuberLOP	✓	✓	**0.6818**	**0.4644**	<u>0.7157</u>	0.4916	<u>0.6497</u>	<u>0.5469</u>	<u>0.6744</u>	0.5690
HuberLOP$_{\text{con}}$	✓	✓	0.6838	<u>0.4657</u>	**0.6936**	**0.4701**	**0.6489**	**0.5456**	**0.6573**	**0.5504**

discounted in about a tenth, while every other member is discounted in 10 to 31% of its forecasts. This implies that HuberLOP does not drop a model but down-weights individual forecasts, which no fixed subset can do.

Contamination with naive forecasters. A pool on a new domain cannot be vetted, so a bad member may enter unnoticed. To stress each rule, we add three naive forecasters built from the context alone, namely the last value, the history mean and a Gaussian fitted to the context, whose seasonal-naive-normalized CRPS of 1.07, 1.59 and 1.23 is worse than seasonal naive itself. As shown in Figure 6 of Appendix B.3 and in Table 4, CRPS rises by 32.0% for the quantile mean and 14.0% for LOP on GIFT, and by only 1.8% for the quantile median. This implies that LOP's fragility lies not in averaging distributions but in the uniform weight $1/K$, which gives a displaced member its full share of probability at every value, and the ablation below isolates exactly that weight.

Ablation of HuberLOP. HuberLOP makes two changes to the quantile mean, averaging distributions instead of quantiles and replacing equal weights with Huber weights. To separate them, we switch each on in turn and score the four combinations on the curated pool and under the three naive forecasters. As shown in Table 4, averaging distributions improves curated-pool CRPS, from 0.4752 to 0.4657 on GIFT, but leaves LOP fragile, while Huber weights cut the CRPS increase from 32.0% to 4.4% for quantiles and from 14.0% to 5.9% for distributions, and the MASE increase in the same way. Only the combination is both the most accurate on the curated pool and robust under contamination, and HuberLOP$_{\text{con}}$, which discounts members more strictly, is the most robust of all. This implies that the two components fix different failures and that HuberLOP needs both.

Per-forecast against fixed weights. Varying weights could still be no better than giving each member one well-chosen weight. To test this, we give every member its mean HuberLOP weight over all forecast cells of a configuration, which again uses the forecasts alone, and apply LOP with that fixed weight, on the curated pool and with the three naive forecasters. As shown in Table 5, per-forecast weights lower CRPS in seven of eight comparisons, most clearly under contamination, where HuberLOP beats its fixed counterpart on 80 of 97 GIFT and 91 of 98 TIME configurations. This implies that the gain of HuberLOP comes from weighting forecasts, not only from weighting models.

Table 5: **Per-forecast against fixed weights.** A fixed rule gives each member, in every forecast cell of a configuration, its mean HuberLOP or HuberLOP$_{\text{con}}$ weight over that configuration. Best in bold.

		GIFT-Eval				TIME			
		Curated Pool		Contaminated		Curated Pool		Contaminated	
Rule	Per-forecast	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Linear opinion pool	×	0.6826	0.4657	0.7336	0.5308	0.6505	0.5483	0.6859	0.6140
Fixed HuberLOP weights	×	0.6822	0.4651	0.7127	0.5009	0.6501	0.5476	0.6736	0.5869
HuberLOP	✓	**0.6818**	0.4644	0.7157	0.4916	0.6497	0.5469	0.6744	0.5690
Fixed HuberLOP$_{\text{con}}$ weights	×	**0.6818**	**0.4642**	**0.6918**	0.4733	**0.6486**	0.5461	**0.6559**	0.5541
HuberLOP$_{\text{con}}$	✓	0.6838	0.4657	0.6936	**0.4701**	0.6489	**0.5456**	0.6573	**0.5504**

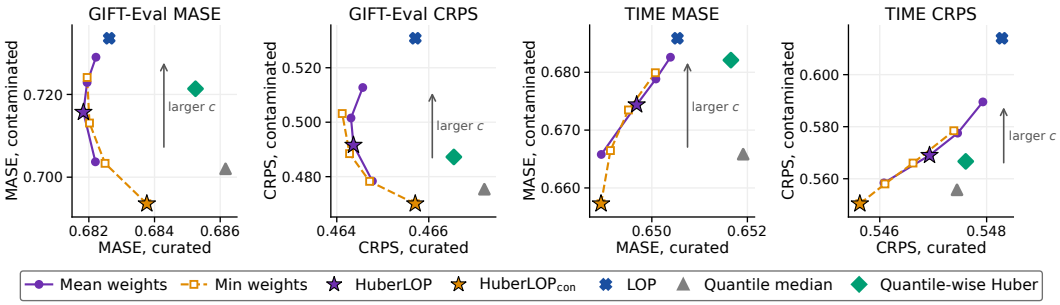

Figure 4: **Accuracy against robustness as c varies.** Each line joins $c = 1.345$, 3, 5 and 10 for mean or min weights, and stars mark HuberLOP and HuberLOP$_{\text{con}}$. Lower left is better.

Effect of threshold c. The one free choice in HuberLOP is the threshold. To see what it trades, we sweep c and the collapse of the per-level weights on both benchmarks. As shown in Figure 4, lowering c to 1.345 with the minimum over levels gives HuberLOP$_{\text{con}}$, which matches LOP on the curated GIFT pool, leads by 0.49% on TIME and degrades by only 0.9% under the three naive forecasters. Raising c to 10 with the minimum over levels moves the curated-pool score 0.34% ahead of LOP but returns the fragility, 8.4%. One member displaced by $20s_j$ in Appendix B.2 gives the same order, raising CRPS by 6.2% for LOP and by 1.7% and 1.2% for the two variants. This implies that c trades accuracy on a curated pool against robustness to a bad member, and that HuberLOP$_{\text{con}}$ is the choice when the pool cannot be vetted.

Context backtesting. A natural competitor scores the members on the masked last stretch of the context, which needs no future label but requires repeated inference with every model. To compare with it, Appendix B.6 evaluates 24 such recipes on both benchmarks. As shown in Table 12, the best recipe gains 3.3% CRPS over LOP on GIFT at the cost of three additional inference passes of all seven TSFMs but falls 0.3% below LOP on TIME, while HuberLOP improves on LOP on both benchmarks with no additional inference. This implies that the gain of backtesting depends on the benchmark and is paid for with repeated inference.

5 Conclusion

Without labels a member's error can be bounded but not corrected, and this premise settles both decisions of label-free TSFM combination. For how to combine, averaging distributions bounds each member at its mass, so the quantile mean loses to LOP on nearly every subset while bounded rules stay within half a percent of LOP in CRPS. For which members to combine, the pool matters far more than the rule, yet on a new domain a choice made in advance recovers only part of what the target's labels would. The choice therefore belongs inside the combination, one forecast at a time. HuberLOP makes it with one threshold, recovering much of the best fixed subset's gain without choosing any model, and HuberLOP$_{\text{con}}$ degrades less than any other rule when the pool is contaminated. We propose it as a simple yet effective baseline for label-free combination of zero-shot TSFMs.

AI USE STATEMENT

We used generative AI tools mainly in two parts of this work. The first is writing, where the tools helped draft, edit and translate the manuscript and prepare figures and tables. The second is experimentation, where they helped implement and run the analysis scripts. The research ideas and hypotheses are the authors' own. Every AI-assisted output was reviewed and verified by the authors, and we take responsibility for the final content of this work.

ETHICS STATEMENT

This work evaluates combination rules on two public forecasting benchmarks, GIFT-Eval and TIME, and on seven publicly released pretrained forecasting models, each used under the license posted on its release page as Appendix A.7 lists. It involves no human subjects, no personal data and no newly collected data. The rules we study only combine forecasts that the models already produce, so they add no risk beyond that of the underlying forecasters, whose outputs should still be validated before they inform high-stakes decisions.

REPRODUCIBILITY STATEMENT

To support reproducibility, we release the code, the per-configuration scores and the scripts behind every table and figure at `https://anonymous.4open.science/r/ICLR27-TSFM`. Every result comes from a fixed archive of nine-quantile forecasts produced once per member and reused by all rules, and Appendix A gives the implementation, the forecast protocols, the information each compared method may use, the compute and the licenses. Both benchmarks and all seven models are publicly available.

REFERENCES

Taha Aksu, Gerald Woo, Juncheng Liu, Xu Liu, Chenghao Liu, Silvio Savarese, Caiming Xiong, and Doyen Sahoo. GIFT-Eval: A benchmark for general time series forecasting model evaluation. *arXiv preprint arXiv:2410.10393*, 2024.

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

Abdul Fatir Ansari, Oleksandr Shchur, Jaris Küken, Andreas Auer, Boran Han, Pedro Mercado, Syama Sundar Rangapuram, Huibin Shen, Lorenzo Stella, et al. Chronos-2: From univariate to universal forecasting. *arXiv preprint arXiv:2510.15821*, 2025.

Andreas Auer, Patrick Podest, Daniel Klotz, Sebastian Böck, Günter Klambauer, and Sepp Hochreiter. TiRex: Zero-shot forecasting across long and short horizons with enhanced in-context learning. *arXiv preprint arXiv:2505.23719*, 2025.

John M Bates and Clive WJ Granger. The combination of forecasts. *Journal of the Operational Research Society*, 20(4):451–468, 1969.

Fabio Busetti. Quantile aggregation of density forecasts. *Oxford Bulletin of Economics and Statistics*, 79(4):495–512, 2017. doi: 10.1111/obes.12163.

Defu Cao, Michael Gee, Jinbo Liu, Hengxuan Wang, Wei Yang, Rui Wang, and Yan Liu. Conversational time series foundation models: Towards explainable and effective forecasting. *arXiv preprint arXiv:2512.16022*, 2025.

Dalpha Agent Team. Cobra Agent: ML engineering without a human in the loop, 2026. URL `https://dalphakr.github.io/CobraAgent/`.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *International Conference on Machine Learning*, 2024.

Sarkar Snigdha Sarathi Das, Palash Goyal, Mihir Parmar, Yiwen Song, Long T. Le, Lesly Miculicich, Jinsung Yoon, Rui Zhang, Hamid Palangi, and Tomas Pfister. Synapse: Adaptive arbitration of complementary expertise in time series foundational models. *arXiv preprint arXiv:2511.05460*, 2025. URL https://arxiv.org/abs/2511.05460.

Datadog. Toto-2.0-Family-and-Friends. Hugging Face model card, 2026. URL https://huggingface.co/Datadog/Toto-2.0-Family-and-Friends.

Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28 (1):20–28, 1979.

Rasool Fakoor, Taesup Kim, Jonas Mueller, Alexander J. Smola, and Ryan J. Tibshirani. Flexible model aggregation for quantile regression. *Journal of Machine Learning Research*, 24(162):1–45, 2023. URL https://www.jmlr.org/papers/v24/22-0799.html.

Azul Garza and Reneé Rosillo. TimeCopilot. *arXiv preprint arXiv:2509.00616*, 2025.

Christian Genest and James V Zidek. Combining probability distributions: A critique and an annotated bibliography. *Statistical Science*, 1(1):114–135, 1986.

Tilmann Gneiting and Roopesh Ranjan. Combining predictive distributions. *Electronic Journal of Statistics*, 7:1747–1782, 2013. doi: 10.1214/13-EJS823.

Google Research. TimesFM-3: A zero-shot foundation model for multivariate forecasting. Official model release, 2026.

Lars Graf, Thomas Ortner, Stanisław Woźniak, and Angeliki Pantazi. FlowState: Sampling-rate-equivariant time-series forecasting. *arXiv preprint arXiv:2508.05287*, 2025.

Neel Guha, Mayee F Chen, Trevor Chow, Ishan S Khare, and Christopher Ré. Smoothie: Label free language model routing. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35 (1):73–101, 1964.

Peter J. Huber. *Robust Statistics*. Wiley, New York, 1981.

Victor Richmond R. Jose, Yael Grushka-Cockayne, and Kenneth C. Lichtendahl, Jr. Trimmed opinion pools and the crowd's calibration problem. *Management Science*, 60(2):463–475, 2014. doi: 10.1287/mnsc.2013.1781.

Mert Kayaalp, Caner Turkmen, Oleksandr Shchur, Pedro Mercado, Abdul Fatir Ansari, Michael Bohlke-Schneider, and Bernie Wang. Test-time efficient pretrained model portfolios for time series forecasting. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.06419.

Emaad Khwaja, Chris Lettieri, Gerald Woo, Eden Belouadah, Marc Cenac, Guillaume Jarry, Enguerrand Paquin, Xunyi Zhao, Viktoriya Zhukov, Othmane Abou-Amal, et al. Toto 2.0: Time series forecasting enters the scaling era. *arXiv preprint arXiv:2605.20119*, 2026.

Fabian Krüger. A kernel score perspective on forecast disagreement and the linear pool, 2026. URL https://arxiv.org/abs/2412.09430v4. Version 4.

Kenneth C Lichtendahl, Jr., Yael Grushka-Cockayne, and Robert L Winkler. Is it better to average probabilities or quantiles? *Management Science*, 59(7):1594–1611, 2013.

Yong Liu, Xingjian Su, Shiyu Wang, Haoran Zhang, Haixuan Liu, Yuxuan Wang, Zhou Ye, Yang Xiang, Jianmin Wang, and Mingsheng Long. Timer-S1: A billion-scale time series foundation model with serial scaling. *arXiv preprint arXiv:2603.04791*, 2026.

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

Zhongzheng Qiao, Sheng Pan, Anni Wang, Viktoriya Zhukova, Yong Liu, Xudong Jiang, Qingsong Wen, Mingsheng Long, Ming Jin, and Chenghao Liu. It's TIME: Towards the next generation of time series forecasting benchmarks. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2602.12147.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Hena Ghonia, Rishika Bhagwatkar, Arian Khorasani, Mohammad Javad Darvishi Bayazi, George Adamopoulos, Roland Riachi, Nadhir Hassen, Marin Biloš, Sahil Garg, Anderson Schneider, Nicolas Chapados, Alexandre Drouin, Valentina Zantedeschi, Yuriy Nevmyvaka, and Irina Rish. Lag-Llama: Towards foundation models for probabilistic time series forecasting. *arXiv preprint arXiv:2310.08278*, 2023.

Peter J. Rousseeuw and Christophe Croux. Alternatives to the median absolute deviation. *Journal of the American Statistical Association*, 88(424):1273–1283, 1993. doi: 10.1080/01621459.1993.10476408.

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.

James H Stock and Mark W Watson. Combination forecasts of output growth in a seven-country data set. *Journal of Forecasting*, 23(6):405–430, 2004.

James W. Taylor. Probabilistic forecast aggregation with statistical depth. *European Journal of Operational Research*, 328(2):460–476, 2026. doi: 10.1016/j.ejor.2025.06.028.

James W Taylor and Xiaochun Meng. Angular combining of forecasts of probability distributions. *Management Science*, 72(3):2111–2133, 2026. doi: 10.1287/mnsc.2024.05558.

TimeCopilot. Foundation model ensemble (GIFT-Eval), 2026. URL https://timecopilot.dev/examples/gift-eval/.

Allan Timmermann. Forecast combinations. In *Handbook of Economic Forecasting*, volume 1, pp. 135–196. Elsevier, 2006.

USTC-AGI. CastStar: An agentic time-series forecasting system. Hugging Face model card, 2026. URL https://huggingface.co/USTC-AGI/CastStar.

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.

Xiaoqian Wang, Yanfei Kang, and Feng Li. Another look at forecast trimming for combinations: Robustness, accuracy and diversity, 2024. URL https://arxiv.org/abs/2208.00139v2. Version 2.

Yunshi Wen, Wesley M. Gifford, Chandra Reddy, Lam M. Nguyen, Jayant Kalagnanam, and Anak Agung Julius. Revisiting the generic transformer: Deconstructing a strong baseline for time series foundation models. *arXiv preprint arXiv:2602.06909*, 2026.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. In *International Conference on Machine Learning*, pp. 53140–53164, 2024.

Ke Yu, James W Taylor, and Xiaochun Meng. Quantile-adaptive probabilistic forecast combining. *European Journal of Operational Research*, 330(3):885–899, 2026. doi: 10.1016/j.ejor.2025.11.031.

A Experimental Setup

This appendix gives the implementation, the forecast protocols of both benchmarks, the information each compared method may use, the compute and the licenses. Appendix B collects the additional results.

A.1 Implementation Details

All methods use the nine quantile levels specified by GIFT-Eval, $\{0.1, 0.2, \ldots, 0.9\}$. The robust scale of Section 3.3 is $s_j = b_{\mathrm{MAD}} \operatorname{median}_k |q_k(\tau_j) - \operatorname{median}_k q_k(\tau_j)| + 10^{-12}$ with $b_{\mathrm{MAD}} = 1/\Phi^{-1}(0.75) \approx 1.4826$, the standard factor that makes the MAD consistent for Gaussian standard deviation (Rousseeuw & Croux, 1993), and the small constant avoids division by zero when members coincide. The Huber center starts at the median and is approximated by two reweighting updates. HuberLOP takes its per-level weights ω_{kj} of Eq. (2) from the second update and inverts the weighted pooled CDF exactly, keeping its left and right limits at every predicted value. It uses $c = 3$ with mean weights, and HuberLOP$_{\mathrm{con}}$ uses $c = 1.345$ with minimum weights. Appendix B.1 reports every threshold we swept on both benchmarks. In Table 2 the trimmed mean and the trimmed opinion pool drop one member per side, the depth-weighted pool weights members by band depth, and the angular combination uses 75°. The settings of these last three are the best of their families on GIFT-Eval CRPS among the 32 settings of Appendix B.7 and are applied unchanged to TIME. For Table 9, the best fixed subset of HuberLOP is searched over 13 candidate subsets that include the leaders of LOP and HuberLOP, and that of the other rules over all 120 subsets. Quantile-wise Huber aggregation uses the conventional $c = 1.345$, fixed in advance.

A.2 Base Forecast Protocol and Leaderboard Comparison

Motivation. Every combination rule reads the same saved member forecasts, so the reader needs to know how this pool relates to the public GIFT-Eval leaderboard. Our protocol is univariate, and each target variable is predicted from its own history, even when the source dataset contains several variables. The rules themselves can equally combine marginal quantiles produced by models that use multivariate inputs.

Comparison with the public results. Table 6 compares all seven members with the public results retrieved on September 23, 2026, after rescoring all 679 model and configuration pairs with the same seasonal-naive denominators and geometric aggregation. Five members agree with their public scores within 0.001. Chronos-2 and Moirai-base differ, and their public recipes differ from ours in input handling and numerical precision.

Table 6: **Leaderboard comparison.** Member accuracy under our input protocol and in the public GIFT-Eval results. Each entry is a geometric mean of seasonal-naive-normalized scores over 97 configurations.

	MASE		CRPS	
Member	Ours	Public	Ours	Public
TiRex-2-ZS	0.6969	0.6973	0.4789	0.4781
PatchTST-FM-r1	0.7069	0.7069	0.4829	0.4829
TimesFM-2.5	0.7050	0.7050	0.4903	0.4903
Chronos-2-Synth	0.7333	0.7203	0.5062	0.4958
TiRex	0.7158	0.7158	0.4885	0.4885
FlowState-r1.1	0.7015	0.7015	0.4866	0.4866
Moirai-base	0.8737	0.9010	0.6035	0.6095

Implication. Every rule in the paper combines the same forecasts under one shared protocol, which is exactly what a combination rule sees in deployment, so the comparisons between rules do not depend on how closely each member reproduces its leaderboard entry.

Table 7: **Information requirements.** What the reported rules and related system families may read. Source labels are used only to fit or choose the combiner on other datasets. Target validation means scoring members on withheld parts of the target history. Auxiliary passes are TSFM forecasts at deployment beyond the candidate forecasts that are eventually combined.

Protocol	Source labels	Target validation	Auxiliary passes
Quantile mean, median, trimmed mean and Huber	No	No	0
LOP and HuberLOP	No	No	0
Trimmed, depth-based and angular pools	No	No	0
Profiled-subset selection	Yes	No	0
Learned gates, FFORMA-style meta-learners	Yes	No	0
Leaderboard ensembles	Yes	No	0
History-validated portfolios and agents	No	Yes	≥ 1

A.3 TIME Benchmark Protocol and Composition

TIME (Qiao et al., 2026) has 98 configured tasks, $111{,}177$ univariate rolling windows from 39 named source groups, and its own scoring convention, and we evaluate the same seven frozen models on it.

We generate float32 forecasts under the official split rule, evaluating multivariate data by their univariate channels with the nine quantiles fixed at $0.1, \ldots, 0.9$. TIME averages normalized nine-quantile WQL over windows, whereas GIFT uses a ratio of summed losses to summed target magnitudes, so we follow each benchmark's own convention and do not compare normalized scores across them.

Composition on TIME. The bottom row of Figure 1 repeats the analysis on TIME with the same 120 subsets combined by uniform LOP, and the picture is the one GIFT-Eval gave. Mean CRPS falls from 0.5654 with two members to 0.5483 with all seven, 102 of the 120 subsets beat the strongest member inside them on both metrics and 105 on CRPS, every subset of size four or larger does so, and at three members the best fixed trio sits 1.5% CRPS below the mean trio and the worst 2.0% above it. Pool membership, not the rule, is again where the spread lies.

A.4 Information Each Compared Protocol Uses

Motivation. A comparison that ignores what each protocol is allowed to read can mistake extra information for a better rule. All rows keep the forecasting models frozen during the evaluated task, and fitting a combiner on source labels is a separate operation from updating a TSFM. Likewise, using the observed target history as input differs from withholding part of that history and using its known values to validate models, which is target-specific validation even though no future test label is revealed. Table 7 records these requirements for the rules we report and for the families of TSFM combination systems discussed in the Related Work section.

Contents of the table. Every combination rule compared in the main text reads only the candidate forecasts of the current task, namely the quantile-wise rules, LOP, HuberLOP, and the trimmed, depth-based and angular pools of Appendix B.7. The profiled subsets of Section 4.3 read labels on the sources they are estimated on, none on the target, and add no auxiliary pass at deployment. Learned gates and meta-learners are trained on source labels, and a leaderboard ensemble is a portfolio fixed on benchmark labels. Test-time portfolios, adaptive arbitration and self-validating agents score members on withheld target history, which requires extra forecasts from truncated contexts.

Implication. HuberLOP sits in the first group, with the rules that read nothing but the current forecasts, which is the setting a new series from an unseen domain leaves the practitioner.

A.5 Compute Resources

Zero-shot inference of the seven TSFMs over all GIFT-Eval and TIME configurations takes on the order of a hundred GPU-hours in aggregate, dominated by the largest models. Every combination rule, including HuberLOP, runs on CPU and finishes in seconds per configuration, and nothing in the pipeline updates the underlying TSFMs.

Table 8: **Threshold sweep.** Every Huber-weighted LOP setting on the curated seven-TSFM pool and with three naive forecasters added. Geometric means over 97 GIFT-Eval and 98 TIME configurations. Best in bold.

		GIFT-Eval				TIME			
		Curated Pool		Contaminated		Curated Pool		Contaminated	
Weights	c	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Mean	1.345	0.6822	0.4648	0.7037	0.4784	**0.6489**	0.5461	0.6658	0.5584
Mean	3 (HuberLOP)	**0.6818**	0.4644	0.7157	0.4916	0.6497	0.5469	0.6744	0.5690
Mean	5	0.6819	0.4643	0.7229	0.5015	0.6501	0.5475	0.6788	0.5775
Mean	10	0.6822	0.4646	0.7290	0.5127	0.6504	0.5479	0.6826	0.5895
Min	1.345 (HuberLOP$_{\texttt{con}}$)	0.6838	0.4657	**0.6936**	**0.4701**	**0.6489**	**0.5456**	**0.6573**	**0.5504**
Min	3	0.6825	0.4647	0.7033	0.4783	0.6491	0.5461	0.6665	0.5580
Min	5	0.6820	0.4643	0.7131	0.4885	0.6495	0.5466	0.6735	0.5660
Min	10	0.6819	**0.4641**	0.7241	0.5032	0.6501	0.5474	0.6799	0.5784

A.6 Computation and Deployment

Every rule runs after the TSFMs have produced their forecasts and needs no gradients, model access, or persistent state. Per forecast cell, HuberLOP computes T Huber reweighting updates over the KQ predicted values, $O(TKQ)$, collapses them to K member weights, and inverts the weighted pooled CDF, which evaluates the K member CDFs at the KQ predicted values as LOP does. On one pinned CPU core an optimized LOP combines a single forecast window in 0.13 ms at horizon 24 and 3.1 ms at horizon 720. With $T = 2$ and $Q = 9$ the weights add little to this, and both are negligible beside TSFM inference.

A.7 Licenses of Existing Assets

We use the GIFT-Eval benchmark (Aksu et al., 2024), the TIME benchmark (Qiao et al., 2026), and the public releases of seven time-series foundation models (Auer et al., 2025; Wen et al., 2026; Das et al., 2024; Ansari et al., 2025; Graf et al., 2025; Woo et al., 2024), each under the open-source license posted on its release page (mostly Apache-2.0), for zero-shot research evaluation only. Seasonal naive is taken from the GIFT-Eval leaderboard as the metric normalization reference and is not a pool member, and the three naive forecasters of Section 4.4 are implemented from scratch.

B Additional Results

B.1 Threshold Sweep of Huber-Weighted LOP

Figure 4 plots every threshold we swept, and Table 8 lists the values behind it. Each member's nine per-level weights of Eq. (2) are collapsed to one weight per member and forecast cell by their mean or their minimum, and the weighted pooled CDF of Eq. (3) is inverted at the requested levels. LOP and the quantile rules appear in Table 4.

Threshold and robustness. Across the eight settings the curated-pool CRPS spans 0.3%, while the CRPS degradation under three naive forecasters spans 0.9 to 10.4% on GIFT-Eval, so the threshold matters for robustness rather than for accuracy on a curated pool. A larger c or the mean over levels moves a setting toward LOP, and the ordering of the settings under contamination is the same on both benchmarks.

Trimmed opinion pools. The trimmed opinion pools of Jose et al. (2014) discard a fixed number of CDFs from each side before averaging. On GIFT-Eval HuberLOP$_{\texttt{con}}$ is level with them on the curated pool, within 0.18% CRPS with two trimmed per side, and ahead under three naive forecasters by 6.67% with one trimmed and 3.33% with two, since a count fixed in advance cannot absorb more outliers than it trims while a weight that falls with distance discounts all three.

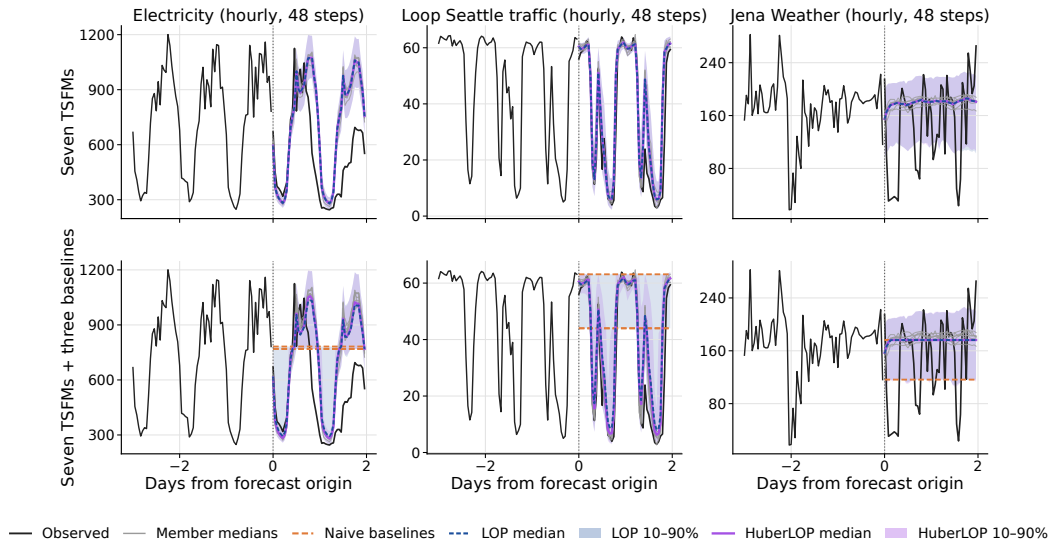

Figure 5: **Forecast examples.** Three hourly GIFT-Eval configurations. Black lines show the context and observed future, grey lines the seven member medians and dashed orange lines the three naive baselines. Blue is the uniform LOP and purple HuberLOP, each drawn as its median and 10 to 90% band. The top row pools the seven TSFMs and the bottom row adds the three baselines to the same window. Windows are selected without targets at the median displacement of the LOP median caused by the baselines.

B.2 One Displaced Member

Motivation. The naive forecasters of Section 4.4 displace several members at once. Here we translate a single TSFM forecast while preserving its quantile spacing, in units of the cross-model MAD scale s_j of Section 3.3, and report the largest CRPS ratio over the seven translated members within each configuration, geometrically averaged across configurations.

Observations. At a $20s_j$ displacement CRPS rises by 12.2% for the quantile mean, 6.2% for LOP, and 1.2% for both the quantile median and quantile-wise Huber. HuberLOP rises by 1.7% and HuberLOP$_{\text{con}}$ by 1.2%, so both sit with the robust rules rather than with LOP. At displacements of $0.5s_j$, s_j and $2s_j$ HuberLOP rises less than any robust rule, because it keeps LOP's behavior until the displaced member leaves the band.

B.3 Forecast Examples

Motivation. Aggregate degradation numbers do not show what a contaminated pool does to one forecast, so Figure 5 draws three hourly GIFT-Eval configurations with a 48-step horizon, the seven TSFMs alone (top) and with the three naive forecasters of Section 4.4 added (bottom), under uniform LOP and HuberLOP. The window is chosen without targets, at the median displacement of the LOP median that the naive forecasters cause, so each panel shows a typical contamination rather than the most damaging one.

Observations. On the curated pool the two rules coincide. Once the naive forecasters enter, the LOP band stretches to cover their flat levels because every member keeps weight $1/K$, while the Huber-weighted band stays where the band on the curated pool was, because members far from the pool's consensus receive small weights at every level.

B.4 Best Fixed Subsets

Table 9 compares the quantile mean, LOP and HuberLOP on all seven members and on each rule's best fixed subset, the subset with the lowest geometric-mean CRPS over all configurations when chosen with target labels.

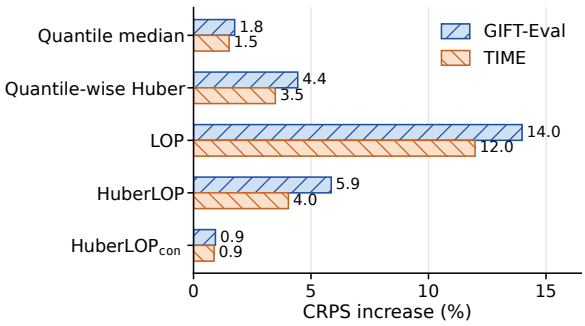

Figure 6: **Degradation under contamination.** Percent CRPS increase when three naive forecasters join the seven-TSFM pool. The quantile mean, not shown, degrades by 32% on GIFT-Eval and 30% on TIME.

Table 9: **Best fixed subsets.** The three rules on all seven members and on each rule's best fixed subset chosen with target labels. Best in bold, second best underlined.

	GIFT-Eval				TIME			
	All seven		Best fixed subset		All seven		Best fixed subset	
Rule	MASE	CRPS	MASE	CRPS	MASE	CRPS	MASE	CRPS
Quantile mean	0.6992	0.4752	0.6857	0.4671	0.6641	0.5553	0.6560	0.5476
Linear opinion pool	<u>0.6826</u>	<u>0.4657</u>	**0.6783**	**0.4632**	<u>0.6505</u>	<u>0.5483</u>	<u>0.6476</u>	<u>0.5450</u>
HuberLOP (ours)	**0.6818**	**0.4644**	<u>0.6787</u>	**0.4632**	**0.6497**	**0.5469**	**0.6473**	**0.5446**

B.5 Member Roles

Table 10 gives the peer gain and leave-one-out contribution of every member that Section 4.3 summarizes. Contribution follows own accuracy with Spearman correlations of -0.71 on MASE and -0.75 on CRPS on GIFT, where rank one is best, but not model size, with a correlation of -0.04. Moirai-base contributes -0.51% CRPS on GIFT and -0.61% on TIME.

Table 10: **Member roles on GIFT-Eval.** What each member gains from its peers and what it gives them. Peer LOP is the uniform LOP of the other six scored in the member's place, and gain is its reduction from Raw. Contribution is the percent improvement of the seven-member LOP (0.6826 MASE, 0.4657 CRPS) over the six-member LOP without the member. Rows are ordered by raw CRPS, and bold marks the largest and the negative contributions.

GIFT-Eval	MASE (↓)			CRPS (↓)			Contribution (%)	
Model	Raw	Peer LOP	Gain	Raw	Peer LOP	Gain	MASE	CRPS
TiRex-2	0.6969	0.6848	+1.7%	0.4789	0.4679	+2.3%	+0.32	+0.47
PatchTST-FM	0.7069	0.6846	+3.2%	0.4829	0.4676	+3.2%	+0.29	+0.40
FlowState	0.7015	0.6861	+2.2%	0.4866	0.4685	+3.7%	**+0.51**	**+0.59**
TiRex	0.7158	0.6829	+4.6%	0.4885	0.4663	+4.5%	+0.04	+0.12
TimesFM-2.5	0.7050	0.6838	+3.0%	0.4903	0.4668	+4.8%	+0.18	+0.23
Chronos-2	0.7333	0.6847	+6.6%	0.5062	0.4673	+7.7%	+0.31	+0.35
Moirai-base	0.8737	0.6792	+22.3%	0.6035	0.4634	+23.2%	**−0.50**	**−0.51**
Average	0.7312	0.6837	+6.5%	0.5038	0.4668	+7.3%		

Roles on TIME. Table 10 is measured on GIFT-Eval alone. We repeat it on the 98 TIME configurations under TIME's scoring convention.

Peer gain in Table 11 again orders the members exactly as their own CRPS does, from 2.1% for TiRex-2 to 18.5% for Moirai-base, and Moirai-base is again the one member with a negative contribution,

Table 11: **Member roles on TIME.** Peer gain and leave-one-out contribution on TIME, in the format of Table 10. Scores are geometric means of seasonal-naive-normalized MASE and CRPS under TIME's convention over 98 configurations, and the seven-member LOP scores 0.6505 MASE and 0.5483 CRPS.

TIME	MASE (↓)			CRPS (↓)			Contribution (%)	
Model	Raw	Peer LOP	Gain	Raw	Peer LOP	Gain	MASE	CRPS
TiRex-2	0.6678	0.6520	+2.4%	0.5615	0.5497	+2.1%	+0.23	+0.26
TimesFM-2.5	0.6653	0.6529	+1.9%	0.5652	0.5502	+2.7%	+0.36	+0.34
TiRex	0.6801	0.6512	+4.2%	0.5709	0.5491	+3.8%	+0.10	+0.15
PatchTST-FM	0.6887	0.6536	+5.1%	0.5753	0.5508	+4.3%	**+0.46**	**+0.46**
FlowState	0.6788	0.6530	+3.8%	0.5767	0.5502	+4.6%	+0.38	+0.35
Chronos-2	0.7437	0.6520	+12.3%	0.5817	0.5508	+5.3%	+0.22	**+0.46**
Moirai-base	0.8259	0.6476	+21.6%	0.6689	0.5450	+18.5%	**−0.45**	**−0.61**
Average	0.7052	0.6518	+7.6%	0.5848	0.5494	+6.1%		

Table 12: **Context backtesting.** Backtesting recipes against HuberLOP on the same subsample of windows, as the percent CRPS improvement over uniform LOP, a geometric mean of per-configuration ratios where positive is better, and the number of configurations where the rule beats LOP. Extra inference counts the additional runs of all seven TSFMs. The first three recipes are the ones chosen on GIFT-Eval by source-held-out selection within each family.

		GIFT-Eval (97)		TIME (98)	
Rule	Extra inference	CRPS gain	Wins	CRPS gain	Wins
Backtest simplex weights ($\lambda = 0.1$, $B = 3$)	3×	**+3.30%**	64	−0.27%	52
Backtest subset within 5% of best ($B = 3$)	3×	+3.13%	61	−0.37%	42
Backtest inverse-loss weights ($B = 1$)	1×	+1.72%	66	+0.43%	68
Backtest top five ($B = 1$)	1×	+0.62%	55	+0.47%	54
HuberLOP	none	+0.62%	65	+0.29%	71
HuberLOP$_{\texttt{con}}$	none	+0.85%	57	**+0.74%**	68

−0.45% MASE and −0.61% CRPS, and the largest contributor is PatchTST-FM on MASE, tied with Chronos-2 on CRPS. Both readings transfer, since gain follows own accuracy and the weakest member costs the pool.

B.6 Context Backtesting

Motivation. Future labels are unavailable at combination time, but the observed context is not. Truncating the context and forecasting its own last stretch gives each member a score on this series without any future label, and that score can set combination weights. This is the most natural competitor to a rule that uses the forecasts alone, and it has a cost, since every backtest reruns all seven TSFMs.

Protocol. For a context x of length L and horizon H, backtest j feeds the frozen models the input $x[:L-jH]$ and scores their forecasts against $x[L-jH:L-(j-1)H]$, which lies inside the observed context, and B backtests use $j = 1, \ldots, B$ and need $L \geq (B+1)H$. The backtest scores choose a subset, namely the single best, the top three or five, the best subset or the subsets within a tolerance of the best, or they set weights, either inverse weighted quantile loss or simplex weights that minimize backtest CRPS with shrinkage λ. The chosen subset or weights are then applied to the forecasts from the full context. A window with no usable backtest falls back to LOP. We ran 24 recipes on the first forecast origin of up to 128 series per configuration, 7,195 windows on GIFT-Eval and 1,621 on TIME, a matched subsample rather than the full benchmark. On GIFT-Eval the recipe of each family is chosen on the other source datasets, and the same recipes are then applied to TIME unchanged. HuberLOP is scored on the same windows.

Table 13: **Pooling controls.** Gains are relative to uniform LOP, and positive is better.

Method	GIFT CRPS	GIFT gain (%)	TIME CRPS	TIME gain (%)
Uniform LOP	0.4657	Reference	0.5483	Reference
Quantile mean	0.4752	−2.031	0.5553	−1.270
Quantile median	0.4672	−0.322	0.5474	+0.154
Quantile-wise Huber, two steps	0.4665	−0.180	0.5476	+0.124
Quantile-wise Huber, converged	0.4666	−0.181	0.5476	+0.118
CDF trim 1/7	0.4657	−0.009	0.5466	+0.300
CDF trim 2/7	0.4665	−0.178	0.5468	+0.279
Mean trim 1/7	0.4662	−0.105	0.5470	+0.243
Mean trim 2/7	0.4678	−0.449	0.5481	+0.038

Table 14: **Depth rules.** All ten fixed settings. Gains are relative to uniform LOP, and positive is better.

Method	GIFT CRPS	GIFT gain (%)	TIME CRPS	TIME gain (%)
Half-region deepest	0.4747	−1.937	0.5561	−1.422
Half-region trim 1/7	0.4672	−0.331	0.5485	−0.042
Half-region trim 2/7	0.4664	−0.151	0.5478	+0.089
Half-region weights	0.4651	+0.123	0.5471	+0.209
Half-region squared weights	0.4658	−0.029	0.5471	+0.209
Band deepest	0.4691	−0.730	0.5490	−0.123
Band trim 1/7	0.4670	−0.279	0.5470	+0.239
Band trim 2/7	0.4662	−0.103	0.5468	+0.278
Band weights	0.4648	+0.200	0.5468	+0.266
Band squared weights	0.4651	+0.121	0.5464	+0.350

Observations. Table 12 shows that the gain of backtesting depends on the benchmark. The recipes that lead on GIFT-Eval, simplex weights and a subset within five percent of the best, both fitted on three backtests, fall below LOP on TIME, while the two recipes fitted on a single backtest stay above LOP on both, at one extra run of every model. HuberLOP and HuberLOP$_{\texttt{con}}$ improve on LOP on both benchmarks, and HuberLOP$_{\texttt{con}}$ is the best rule on TIME, without any extra inference.

Implication. Backtesting pays when the recent context predicts which member will do well on the next horizon, and it needs one to three extra runs of every model to find out. HuberLOP asks nothing of the context and improves on LOP on both benchmarks, which is why we propose it as the default and leave backtesting as a complement when contexts are long and inference is cheap.

B.7 Unified Forecast-Only Baseline Table

Motivation. A competitor we did not choose, a depth-based, angular, radial or trimmed pool, might separate from LOP where the rules of the main text do not, so we collect 32 fixed distribution-combination settings on the original seven-member pool on both benchmarks. The classical pooling rules are in Table 13 and the ten depth rules in Table 14. All methods consume the same nine-quantile forecasts, and the fixed depth grid is reported in full. The two benchmarks use their own score aggregation, so absolute scores are not compared across them.

Observations. No setting reaches a gain of half a percent over LOP on either benchmark. The best on GIFT is band weights at 0.200% and the best on TIME is band squared weights at 0.350%, while the quantile mean, the deepest half-region rule and the narrow angular rules lose one to two percent on GIFT. The five angular settings trail LOP by 0.16 to 1.49% CRPS on GIFT and 0.12 to 0.87% on TIME, and the eight radial settings by 0.47 to 1.13% and 0.23 to 0.71%.

Implication. On a curated pool the family of the rule does not matter once the rule bounds its members, which is why HuberLOP keeps the opinion pool and changes only its weights.