PROBABILISTIC AGENTIC MEMORY: LEARNING DISTRIBUTIONAL EVIDENCE FOR SKILL REUSE

Anonymous authors
Paper under double-blind review

ABSTRACT

Repeated use of predictive skills gives an agent outcome feedback, but a numerical error alone does not specify how that experience should influence future reuse. We introduce *Probabilistic Agentic Memory* (PAM), which learns how outcomes become evidence for persistent skill support. PAM builds context-conditioned outcome distributions around black-box forecasts and uses their realized log scores in sequential generalized Bayesian updates. A memory-informed readout combines these same predictive distributions, linking the interpretation of past outcomes to future predictions. Training differentiates through chronological replay without gradients through the skills; at deployment, learned parameters are frozen and released outcomes update memory. Across three epidemic-forecasting tasks, PAM improves mean RMSE and weighted interval score over three agent selectors using the same skill library. It also improves both metrics over same-bank dynamic model averaging on Bench-A and B-FLU. On Bench-A, ablations support nonlinear, context-conditioned uncertainty modeling, and matched-component comparisons show that learned writes improve subsequent prediction at comparable coverage. Prospective diagnostics on both CDC tasks further show that accumulated support favors components with better subsequent predictions. Together, these results support learning the evidential meaning of experience as a mechanism for predictive-skill reuse.

1 INTRODUCTION

Repeated skill use gives an agent access to both skill descriptions and observed outcomes. Model descriptions support tool selection (Shen et al., 2023; Yao et al., 2023); realized outcomes provide task-specific evidence about future usefulness. Forecasting makes this feedback explicit because predicted quantities eventually become observable. The challenge is to turn such outcomes into persistent memory that improves later skill reuse.

Raw error is an incomplete memory signal. The same residual can be routine in a volatile setting but surprising in a stable one, while systematic bias and unpredictable variation call for different interpretations. Because each interpretation changes subsequent support, the central question is: *how can an agent learn what an outcome implies for the future reuse of a skill?*

Existing work addresses complementary pieces of this problem. Agent-memory systems organize and retrieve experience (Xu et al., 2025; Chhikara et al., 2025); neural postprocessing learns conditional predictive distributions (Rasp & Lerch, 2018); and sequential forecast combination adapts model support over time (Raftery et al., 2010; McAlinn & West, 2019). We focus on their interface: learning a prospective numerical model whose realized score becomes evidence for persistent skill memory. This differs from using a learned distribution only to calibrate the current output, or updating skill weights under a fixed interpretation of error.

We introduce **Probabilistic Agentic Memory (PAM)**. Its central principle is that *the same conditional distribution used to predict an outcome also determines how that outcome is written into memory*. Before observation, the distribution represents outcomes plausible under a skill and its context; after observation, its score updates persistent support over the selected skills. Future predictions read this support to combine the skills' conditional distributions. Learning the outcome model therefore changes how past performance guides future skill use.

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.

PAM combines neural evidence learning with sequential generalized Bayesian filtering (Bissiri et al., 2016). Skill-shared networks learn bounded location and scale corrections around black-box forecasts, allowing expected bias and variability to shape evidence. A normalized sequential update accumulates this evidence in the support state. Training replays historical forecast–outcome episodes without differentiating through the skills; at deployment, learned parameters are frozen and released outcomes update only memory. A bounded contextual proposal can affect the current readout without being written back, separating persistent evidence from transient context.

We evaluate PAM on three epidemic-forecasting tasks with a 15-skill bank retrieved from 31 candidates. The main comparison evaluates skill-selection and combination strategies, with individual forecast profiles reported separately. PAM improves mean RMSE and WIS over three agents using the same 31-skill library on all tasks and over same-bank DMA on Bench-A and B-FLU. On Bench-A, ablations connect feature-dependent uncertainty and contextual readout to predictive gains, while matched-component comparisons isolate the benefit of learned writes. Prospective analyses on both CDC tasks show that accumulated support favors skills with better subsequent predictions.

Our contributions are threefold:

- **An evidence-learning view of skill memory.** We formulate predictive-skill reuse as learning how observed outcomes become evidence for persistent support, separating interpretation of experience from its storage.
- **Probabilistic Agentic Memory.** PAM couples learned conditional outcome models with sequential generalized Bayesian support updates and memory-informed prediction, while treating the underlying skills as black boxes.
- **Empirical validation.** PAM improves mean RMSE and WIS over three 31-skill agent selectors across all tasks and over DMA on two tasks. Bench-A ablations and fixed-component tests support the evidence-learning design, while cross-task diagnostics characterize predictive skill memory.

2 Related Work

2.1 Agent memory and skill reuse

Agent-memory systems make experience available for later decisions: A-MEM and Mem0 organize persistent information, while Reflexion converts feedback into verbal reflections and Voyager accumulates executable skills (Xu et al., 2025; Chhikara et al., 2025; Shinn et al., 2023; Wang et al., 2024). Their stored notes, reflections, and programs guide later behavior. Numerical forecasting additionally requires a rule for how strongly an observed error should revise support for the skill that produced it. Uncertainty-aware memories come closer: BeliefMem retains competing propositions and Nous conditions belief updates on observation reliability (Liao et al., 2026; Singh, 2026). Yet neither formulation learns a prospective numerical outcome density from archived skill forecasts and realized outcomes; in Nous, observation reliability is estimated from epistemic language. PAM targets this missing interface: it learns a context-conditioned outcome model from measurable feedback and uses the score of each origin-committed prediction as a comparable write signal into persistent skill support.

2.2 Probabilistic forecasting and sequential combination

Probabilistic forecasting and adaptive combination provide the closest statistical ingredients. Neural postprocessing learns conditional predictive distributions, calibrated forecast mixtures estimate bias-corrected components and weights, and online expert methods update weights from supplied losses (Rasp & Lerch, 2018; Raftery et al., 2005; Freund & Schapire, 1997; Herbster & Warmuth, 1998); however, calibrating the current output does not by itself learn the evidence that persists in later support updates. DMA and dynamic BPS go further by adapting model probabilities, parameters, calibration, or dependence over time (Raftery et al., 2010; McAlinn & West, 2019), but do so through model-specific state/parameter tracking or a latent synthesis process. PAM instead learns one shared, context-conditioned observation interface for heterogeneous black-box skills: the same committed conditional distribution both scores released outcomes and contributes to future mixtures, while online memory updates only an explicit support vector. This avoids additional within-skill parameter

inference or latent synthesis inference at deployment and keeps each skill's accumulated evidence inspectable. Our identical-component, different-write controls directly test the resulting distinction between better predictive components and better memory updates.

3 Problem Formulation

We study *continual predictive-skill reuse*: using delayed outcome feedback to guide repeated forecasting. The inputs are a task description, the history $x_{\leq o}$ released by forecast origin o, and a library $\mathcal{S}$ of L executable predictive skills, with a budget of $K \leq L$ skills per task. The outputs are a predictive distribution P_o for each requested target and a nonnegative, sum-to-one support vector $W_t = (W_{t,1}, \ldots, W_{t,K})$ over a selected bank $\mathcal{S}_K$. Each forecast row e records its origin o_e and target component–horizon key r_e; its numerical outcome y_e becomes available later in a release batch $\mathcal{D}_t$. The objective is to use this feedback to improve the accuracy and calibration of future predictions while retaining an inspectable summary of relative skill support. Predictions and their input features use only origin-available information: later outcomes may revise support, but not the forecasts already issued. Support summarizes the skills' relative predictive usefulness for the task.

4 Probabilistic Agentic Memory

4.1 Overview

PAM learns how feedback should change future skill reuse. Its four operations are shown in fig. 1: *Record* commits skill predictions before outcomes arrive; *Interpret* constructs conditional outcome distributions; *Write* scores released outcomes and updates persistent support through sequential generalized Bayesian filtering; and *Read* combines current predictions under that support, optionally with a bounded contextual proposal. Memory $\mathcal{M}_t = (\mathcal{S}_K, \mathcal{A}_{\leq t}, W_t)$ comprises the bank, prediction archive, and accumulated support. Historical replay fits neural modules; deployment updates only support.

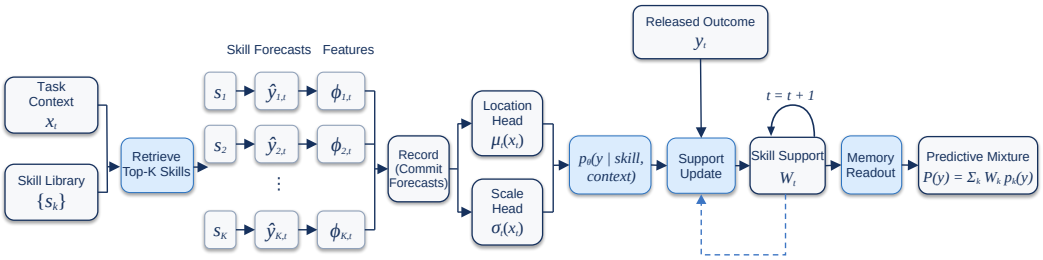


Figure 1: **PAM workflow.** Committed skill predictions provide evidence when outcomes arrive; accumulated support informs subsequent predictions. The dashed arrow carries support into the next release update. The skill bank stays fixed within each episode; the optional contextual readout is omitted.

4.2 Record: Select Skills and Commit Predictions

Retrieval identifies relevant skills; observed performance determines predictive support. PAM ranks skill descriptions by task–description embedding similarity plus a robustly standardized validation score $-\log(1 + \mathrm{RMSE})$, where RMSE is the validation root mean squared error. The top K skills form a fixed episode bank $\mathcal{S}_K$. Retrieval selects the bank but does not determine its memory weights. A separately learned prior initializes support:

$$W_{0,k} = \frac{\exp \psi_k}{\sum_{j=1}^{K} \exp \psi_j}, \qquad |\psi_k| \leq 0.5. \tag{1}$$

Here k, j index selected skills and ψ contains their prior logits; the bound limits initial weight ratios to $\exp(1)$.

For each forecast row e and skill k, PAM records forecast mean and variance summaries, features $\xi_{e,k} \in \mathbb{R}^D$, component parameters, and the issued mixture at origin o_e. Features comprise forecast summaries, target metadata, and history available at origin o_e. We use the outcome coordinate $z_e = \log(1 + y_e)$, with uncorrected transformed mean $\mu^0_{e,k}$ and variance summary $v^0_{e,k} \geq 0$. The superscript 0 denotes uncorrected skill output and D is the feature dimension. Skills without native uncertainty use released-history first-difference variance. These records remain fixed when feedback arrives; section A.2 specifies the transformations.

4.3 INTERPRET: LEARN DISTRIBUTIONAL EVIDENCE

A residual's meaning depends on predictable bias and expected variability. Two independent, skill-shared multilayer perceptrons learn these separately:

$$\delta_{e,k} = F^{\mathrm{loc}}_\theta(\xi_{e,k}), \qquad a_{e,k} = F^{\mathrm{sc}}_\theta(\xi_{e,k}). \tag{2}$$

Here θ collects the networks' disjoint parameters, and the scalar outputs $\delta_{e,k}$ and $a_{e,k}$ control location and residual scale. Each network has two tanh hidden layers of width $H = 32$; inputs have $D = 19$ features. Corrections remain anchored to the skill forecast:

$$\begin{aligned} \widetilde{m}_{e,k} &= \mu^0_{e,k} + B_{r_e} \tanh \delta_{e,k}, \\ S^2_{e,k} &= \sigma^2_{0,r_e} \exp(2c_s \tanh a_{e,k}) + \gamma_{r_e} v^0_{e,k} + s^2_{\min}. \end{aligned} \tag{3}$$

For component–horizon key r, $B_r \geq 0$ bounds the location shift, $\sigma_{0,r} > 0$ is the frozen reference residual scale, and $\gamma_r \geq 0$ scales the forecast uncertainty contribution. The constant $c_s > 0$ bounds the residual-scale multiplier to $[e^{-c_s}, e^{c_s}]$; $s^2_{\min} > 0$ is a numerical floor. Thus $\widetilde{m}$ is the bounded pre-bridge location and $S > 0$ is the kernel scale. Zero-initialized output layers recover the reference corrections; settings are in sections A.1 and A.2.

We censor a Student-t distribution to the nonnegative target range and adjust its center to match the corrected raw-scale mean. The underlying kernel is

$$q_{\theta,e,k}(z) = \mathrm{StudentT}(z; m_{e,k}, S_{e,k}, \nu_{r_e}), \qquad z \in \mathbb{R}, \tag{4}$$

where $\nu_r > 2$ is the degrees of freedom and $m_{e,k}$ is the final kernel center. The bridge places tail probability at 0 and U_r, a fixed calibration bound. Mean alignment determines the final center $m_{e,k}$ from the pre-bridge correction $\widetilde{m}_{e,k}$. Before censoring, the kernel variance is $\nu_{r_e} S^2_{e,k}/(\nu_{r_e} - 2)$. The issued component $p_{\theta,e,k}$ combines an interior density with boundary probabilities; q is the unbounded kernel used in the diagnostic plots. Section A.2 gives censoring and mean alignment.

4.4 WRITE: SEQUENTIAL BAYESIAN SUPPORT UPDATES

New outcomes revise earlier support according to relative predictive fit. Active skills are scored on the same nonempty release batch $\mathcal{D}_t$ using their committed components:

$$\ell_{t,k} = \frac{\sum_{e \in \mathcal{D}_t} \lambda_{r_e} \log p_{\theta,e,k}(z_e)}{\sum_{e \in \mathcal{D}_t} \lambda_{r_e}}, \tag{5}$$

where $\lambda_r > 0$ weights target keys and $p_{\theta,e,k}(z_e)$ is the appropriate density or boundary probability. With all skills active,

$$W_{t,k} = \frac{W_{t-1,k} \exp(\rho \ell_{t,k})}{Z_t}, \qquad Z_t = \sum_{j=1}^{K} W_{t-1,j} \exp(\rho \ell_{t,j}). \tag{6}$$

The validation-selected $\rho > 0$ controls evidence strength and $Z_t > 0$ normalizes support. Empty batches leave W_t unchanged. Skills missing any batch forecast retain their absolute mass; active skills redistribute their previous combined mass (section A.3). Computation uses log weights.

The recursion carries forward prior support and corrects it with new evidence. Batch averaging and tempering yield generalized Bayesian updating over skills (Bissiri et al., 2016). Single-observation updates with $\rho = 1$ recover finite-model Bayes under conditional-likelihood factorization. The networks learn the observation evidence, and W_t is the updated state; section B analyzes its log odds and computational cost.

4.5 READ: MEMORY-INFORMED SKILL REUSE

Current context can guide prediction without being treated as observed evidence. At origin o, renormalize the latest support over available skills to $\bar{W}_o$ and construct their new components $p_{\theta,o,k}$. The readout is

$$P_o(z) = \sum_k \widehat{W}_{o,k} p_{\theta,o,k}(z), \qquad \widehat{W}_{o,k} = (1-\alpha)\bar{W}_{o,k} + \alpha \pi^{\mathrm{ctx}}_{o,k}, \tag{7}$$

where the sum covers available skills and validation-selected $\alpha \in [0, 1)$ controls the contextual share; $\alpha = 0$ gives pure memory. The prior-centered proposal is

$$\pi^{\mathrm{ctx}}_{o,k} = \frac{W_{0,k} \exp(r_{o,k})}{\sum_j W_{0,j} \exp(r_{o,j})}, \qquad r_{o,k} = 0.5 \tanh g_{\phi,o,k}. \tag{8}$$

Here $r_{o,k}$ is a bounded scalar residual, and $g_{\phi,o,k}$ is skill k's output from a permutation-equivariant DeepSets network with parameters ϕ (Zaheer et al., 2017). It reads $\xi_{o,k}$, $\log W_{0,k}$, and $\log \bar{W}_{o,k}$, combining local and mean-pooled encodings. Architecture dimensions are in table 3. The proposal affects only the current readout; persistent memory retains W_t. Predictions are committed before observation, with means and quantiles computed from the mixture distribution.

4.6 LEARNING AND DEPLOYMENT

To fit evidence in the context of subsequent reuse, PAM replays chronological forecast–feedback episodes and alternates four parameter blocks. The location head and contextual proposal minimize issued-mixture-mean RMSE; the scale head minimizes the negative log likelihood (NLL) of issued censored mixtures. The two correction blocks are regularized, while the prior optimizes cumulative evidence with auxiliary terms (section A.4). Gradients pass through the networks and support recursion, not retrieval or black-box skills. Validation selects the checkpoint and (ρ, α) using a weighted combination of standardized RMSE, five-quantile WIS, and NLL (eq. (19)); test reporting uses decile WIS. After selection, θ, ψ, ϕ, ρ, and α are frozen, and released outcomes update only W_t. Section A.4 and algorithm 1 provide training settings and the deployment procedure.

5 EXPERIMENTS

We test whether learning how outcomes become evidence improves skill reuse. Strategy comparisons assess the complete system; the core design tests in table 2 separate module learning (A), matched write evidence (B), and non-learned error-driven adaptation (C). Prospective diagnostics examine the resulting memory.

5.1 EXPERIMENTAL SETUP

Bench-A forecasts daily COVID-19 cases for 349 European regions at direct 1-/3-day and recursive 7-day endpoints (Panagopoulos et al., 2021). B-COVID and B-FLU forecast finalized weekly hospital-admission rates for 51 U.S. jurisdictions at direct 1-/2-week and recursive 4-week endpoints (Centers for Disease Control and Prevention, Division of Healthcare Quality Promotion, National Healthcare Safety Network, 2026). Each task retrieves 15 of 31 archived skills. Chronological release-time replay permits feedback only after its release and only for later predictions; CDC outcomes have a one-week lag. Splits, covariates, and fitting details are in section C.

Metrics and reporting. We report RMSE and decile WIS (lower is better) (Bracher et al., 2021), Cov90 (target 0.90), and Width90 (smaller at comparable coverage). RMSE, WIS, and Width90 are normalized by training-target SD. Fitted methods report mean $\pm$ sample SD over three independent fits; deterministic evaluations of frozen archives report single values. Training-run SD and time-block confidence intervals describe different sources of variation (sections C.2 and C.3).

Strategy-level baselines. We compare **Agent: Top-1** (description retrieval), **Agent: Rediscovery** (selection at each origin), and **Agent: ReAct** (reasoning–action prompting adapted to skill selection (Yao et al., 2023)) over the same 31-skill library used for PAM's retrieval. Their selected forecasts are evaluated on the frozen test events. **DMA** (Raftery et al., 2010) combines PAM's Top-15 bank using anchored, zero-head components. These strategy-level comparisons complement the fixed-component write tests; implementations and individual forecast profiles are in sections C.2 and E.

Table 1: **Skill-selection and combination strategies.** PAM reports mean ± sample SD over three independent fits; agent and DMA rows give single scores for fixed selection or forecast archives. RMSE, WIS, and Width90 are training-SD normalized. Bold marks the lowest RMSE/WIS mean within each task.

Method	RMSE ↓	WIS ↓	Cov90	Width90
Bench-A				
PAM (ours)	**0.3347**± 0.0162	**0.0914**± 0.0008	0.9230± 0.0137	0.5314± 0.0406
Agent: Top-1†	0.4729	0.1162	0.7409	0.5729
Agent: Rediscovery†	0.3811	0.1106	0.7235	0.5053
Agent: ReAct†	0.3905	0.1083	0.7359	0.5457
DMA	0.6690	0.1132	0.9094	1.0597
B-COVID				
PAM (ours)	0.2984± 0.0111	0.1513± 0.0050	0.9643± 0.0053	1.0905± 0.0287
Agent: Top-1†	0.3756	0.2403	0.8660	1.3435
Agent: Rediscovery†	0.3870	0.2248	0.9091	1.2261
Agent: ReAct†	0.4147	0.2462	0.8717	1.1847
DMA	**0.2967**	**0.1505**	0.9783	1.1722
B-FLU				
PAM (ours)	**1.0447**± 0.1773	**0.4449**± 0.0268	0.9557± 0.0300	2.9089± 0.6524
Agent: Top-1†	1.3888	0.4898	0.8362	2.5642
Agent: Rediscovery†	1.2852	0.5901	0.5551	1.9986
Agent: ReAct†	1.4309	0.6430	0.5176	1.7593
DMA	1.4408	0.5519	0.8147	2.0533

Cov90 targets 0.90; smaller Width90 is preferable only at comparable coverage. †Agent forecasts use the selected skill's frozen moments with the fixed zero-head censored moment-t bridge. All three agents select from the same 31-skill library used to retrieve PAM's Top-15 bank; DMA combines that bank with anchored components.

5.2 Comparing skill-reuse strategies

Consistent gains over agent selection. PAM lowers mean RMSE and WIS relative to all three agents using the same 31-skill library (table 1). Its WIS is 0.0914, 0.1513, and 0.4449 on Bench-A, B-COVID, and B-FLU, versus the lowest agent WIS of 0.1083 (ReAct), 0.2248 (Rediscovery), and 0.4898 (Top-1), respectively. The strongest selector changes across tasks, whereas PAM improves both error metrics throughout. On Bench-A, PAM's coverage is closer to 0.90 than every agent's; on B-COVID, its intervals are narrower than every agent's. The matched-write comparison below separately tests whether learned evidence improves reuse with component predictions fixed.

Comparison with dynamic combination. On the same bank, PAM improves Bench-A RMSE/WIS from DMA's 0.6690/0.1132 to 0.3347/0.0914, and B-FLU from 1.4408/0.5519 to 1.0447/0.4449 (table 1). On B-COVID, the mean errors are close: 0.2984/0.1513 for PAM and 0.2967/0.1505 for DMA. On Bench-A, PAM narrows Width90 from 1.0597 to 0.5314 while retaining coverage near 0.90. On B-COVID, it reduces width from 1.1722 to 1.0905 and brings coverage closer to the target, from 0.9783 to 0.9643. On B-FLU, the WIS improvement accompanies broader intervals and coverage closer to nominal, 0.9557 versus 0.8147.

5.3 Ablation study: which design choices matter?

Panel A of table 2 retrains one change at a time with matched data and tuning: removing location or scale correction, replacing the heads with constants or linear maps, or disabling contextual readout. Panel B fixes learned readout components. Panel C compares Hedge (Freund & Schapire, 1997), inverse recent error, and Fixed Share (Herbster & Warmuth, 1998) as non-learned error-driven rules on the same bank.

Table 2: **Bench-A evidence-learning and write-path controls.** A: independently retrained PAM restrictions with matched data and tuning. B: identical learned readout components within each checkpoint at $\alpha = 0$; only write evidence changes, evaluated on subsequent forecasts. C: error-driven rules with anchored components on the same frozen Top-15 bank and release schedule. A/C use the complete Bench-A test set. Mean $\pm$ sample SD over three fits where applicable; single values denote deterministic evaluation. Metrics follow table 1; bold marks the lowest RMSE/WIS mean per panel.

Method	RMSE ↓	WIS ↓	Cov90	Width90
A. Independently retrained modules				
PAM (full)	$\mathbf{0.3347} \pm 0.0162$	0.0914 ± 0.0008	0.9230 ± 0.0137	0.5314 ± 0.0406
w/o location correction	0.3495 ± 0.0041	$\mathbf{0.0881} \pm 0.0008$	0.9318 ± 0.0034	0.6223 ± 0.0213
w/o scale learning	0.4611 ± 0.0704	0.1058 ± 0.0034	0.9191 ± 0.0188	0.7918 ± 0.1054
w/o context-dependent heads	0.3423 ± 0.0026	0.1007 ± 0.0005	0.8834 ± 0.0030	0.5591 ± 0.0118
Linear evidence heads	0.5148 ± 0.0000	0.1079 ± 0.0000	0.9281 ± 0.0000	0.9297 ± 0.0000
w/o contextual readout ($\alpha = 0$)	0.4478 ± 0.0432	0.1051 ± 0.0035	0.8947 ± 0.0537	0.7672 ± 0.1801
B. Matched components; write evidence only				
PAM: learned writes	$\mathbf{0.3308} \pm 0.0098$	$\mathbf{0.0917} \pm 0.0011$	0.9245 ± 0.0111	0.5665 ± 0.0398
Anchored writes	0.3872 ± 0.0210	0.1006 ± 0.0016	0.9245 ± 0.0072	0.6528 ± 0.0161
C. Non-learned error-driven support updates				
Hedge (absolute error)	0.3554	**0.1073**	0.8144	0.5590
Inverse recent error	**0.3471**	0.1077	0.8032	0.5401
Fixed Share (absolute error)	0.3545	0.1074	0.8137	0.5565

Context-conditioned uncertainty. Removing scale learning raises RMSE/WIS from 0.3347/0.0914 to 0.4611/0.1058 and Width90 from 0.5314 to 0.7918 at similar coverage (table 2A). Constant and linear heads yield WIS 0.1007 and 0.1079. These results support nonlinear, context-dependent evidence and learned scales that improve both scores and interval sharpness.

Location correction and current context. Location correction reduces RMSE (0.3495 to 0.3347) and width (0.6223 to 0.5314), although its removal lowers WIS to 0.0881 (table 2A). Removing contextual readout raises RMSE/WIS to 0.4478/0.1051, supporting its complementary role at prediction time.

5.4 Does memory improve future skill reuse?

On Bench-A, Panel B of table 2 fixes the learned readout components, bank, and release schedule within each checkpoint at $\alpha = 0$. It compares fitted and anchored zero-head write distributions on the same later-origin subset.

Learned writes reduce RMSE/WIS from 0.3872/0.1006 to 0.3308/0.0917 and width from 0.6528 to 0.5665 at the same displayed coverage, 0.9245 (table 2B). The paired WIS difference is -0.00889 $[-0.00919, -0.00864]$ over this short Bench-A window. With readout components unchanged, the improvement comes through the memory weights.

Why not update support from raw error directly? Hedge, inverse recent error, and Fixed Share yield WIS 0.1073, 0.1077, and 0.1074 and Cov90 0.8032–0.8144; full PAM yields 0.0914 and 0.9230 (table 2A,C). Panel C compares complete strategies with anchored components on the full test set. Panel B fixes learned readout components to isolate write evidence.

Does support anticipate future usefulness? Figure 2 examines the state accumulated on B-COVID. For skill k, let $L_{o,k}$ be the WIS of its committed component at origin o, evaluated when that target becomes available. Alignment compares the initial prior with issue-time support:

$$\Delta_{\text{align}} = \text{Avg}_o \left[\sum_k (W_{0,k} - \bar{W}_{o,k}) L_{o,k} \right]. \tag{9}$$

Here Avg_o uses the fixed hierarchical averaging over eligible targets and origins, followed by equal weighting of the fitted runs. Positive alignment means that updated weights favor components with lower subsequent WIS. This is a support-weighted component comparison. Both CDC tasks show positive alignment: **0.0508** $[0.0207, 0.0825]$ on B-COVID and **0.1799** $[0.0632, 0.2980]$ on B-FLU (table 5). These results connect accumulated support to future component quality. Section D gives the complete-task estimates, time-block intervals, and case-selection rules.

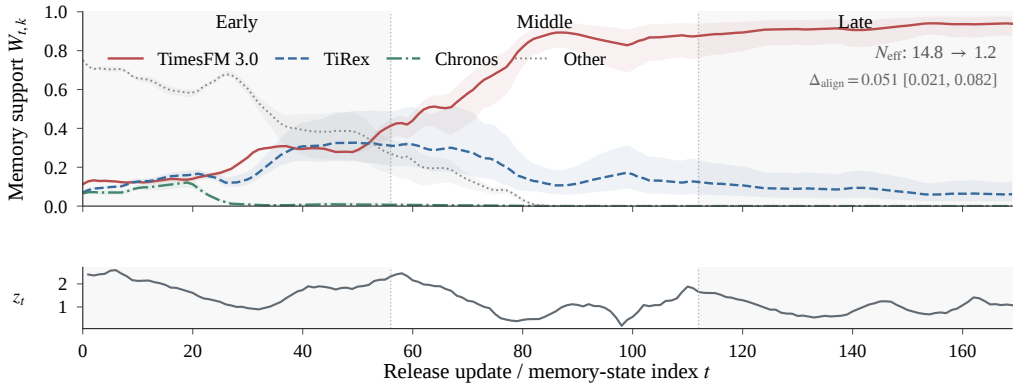


Figure 2: **B-COVID skill-support dynamics.** Lines and ribbons show the mean and range across three fitted runs. Named skills are fixed by initial support; "Other" retains the remaining 12. The lower strip is batch-median $\log(1 + y)$. Early, middle, and late denote three equal-length portions of the 169-update sequence.

5.5 How do probabilistic writes and readouts behave?

Same residual, different evidence. Figure 3 isolates the role of conditional scale. Center, outcome, base variance, prior memory, release batch, and competing scores remain fixed; only the focal fitted residual-scale multiplier is replaced by one. For the B-COVID TiRex forecast, fitted/reference latent densities at the same outcome are 1.200/1.033, giving support 0.033585/0.033569. The identical residual therefore carries different evidence, while full-batch normalization keeps the resulting weight change small. This selected case illustrates the write mechanism; table 2A separately tests the predictive value of context-dependent heads on Bench-A, and section D specifies case selection.

Same components, different memory. Figure 4 fixes all 15 current kernels and changes only the support from W_0 to $\bar{W}_o$. Effective support $N_{\text{eff}}(W) = \exp\{-\sum_k W_k \log W_k\}$ falls from 14.74 to 1.21, while TimesFM-3 support increases from 0.117 to 0.952. Accumulated evidence therefore changes how the same predictive components are reused. Both panels use the same current components: the left applies initial memory, and the right uses B-COVID's actual issue-time weights ($\alpha = 0$). The following complete-test comparison evaluates the predictive value of retaining alternative components.

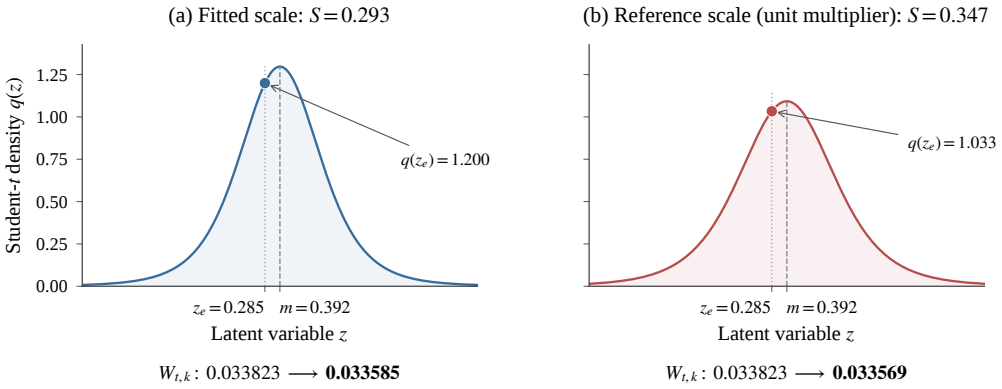


Figure 3: **Matched-scale evidence.** TiRex forecast for Oregon, issued 7 February 2026 for 7 March. Only residual scale changes. Curves show the unbounded latent kernels q; predictive scores use the censored distributions p. Weights use full-batch normalization.

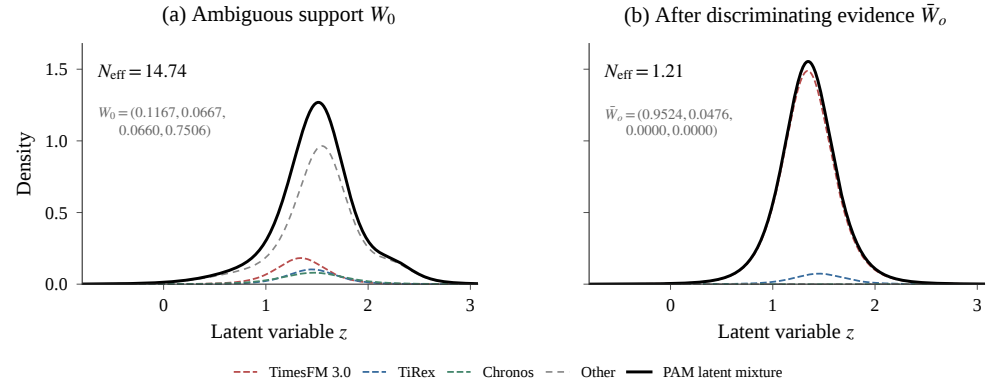


Figure 4: **Memory-only readout.** B-COVID forecast for South Carolina, issued 20 September 2025 at a one-week horizon. Only support changes; kernels and axes are fixed. Dashed curves are weighted latent contributions; black is their sum including "Other." N_{eff} summarizes the spread of skill weights.

When is a distribution preferable to hard selection? A hard readout selects the censored component with largest issue-time weight. Its complete-test WIS difference from the mixture is

$$\Delta_{\text{soft}} = \text{Avg}_o[\text{WIS}(P_o^{\text{hard}}, y_o) - \text{WIS}(P_o, y_o)]. \tag{10}$$

Memory-only readout ($\alpha = 0$) improves WIS over hard selection by **0.01797** $[0.01639, 0.01955]$ on Bench-A and **0.00806** $[0.00266, 0.01379]$ on B-FLU (table 5). Bench-A's validation-selected issued mixture is also better; the B-COVID and contextual B-FLU intervals include zero. These comparisons identify predictive gains from retaining alternatives under the specified readouts, complementing the fixed-component visualization in fig. 4.

6 Discussion and Conclusion

PAM couples learned conditional outcome models with sequential generalized Bayesian updating to turn feedback into persistent skill memory. It improves mean RMSE and WIS over three 31-skill agent selectors across all tasks and over same-bank DMA on Bench-A and B-FLU. On Bench-A, ablations support context-conditioned uncertainty and contextual readout, matched-component comparisons isolate gains from learned writes, and error-driven strategies do not reproduce the full-system performance. Positive prospective alignment on both CDC tasks links stored support to subsequent predictive usefulness. The present implementation uses an episode-fixed bank; extending its evidence–memory interface to growing skill libraries is a natural direction for broader skill reuse.

Reproducibility Statement

The appendices describe the architecture, blockwise training setup, optimization settings, data splits, release-time evaluation, and scoring rules. The source package contains vector figures and the numerical summaries used in the tables. Performance is summarized across three fitted runs where available; sections C.2 and C.3 define training-run SD and time-block confidence intervals. Section D specifies diagnostic case selection. The full executable benchmark and per-event forecast archives are not included in this package.

AI Use Statement

An AI assistant helped organize and edit the manuscript, prepare code, and implement, score, and tabulate experiments from archived predictions. The authors are responsible for verifying all technical content, references, code, and reported results.

Ethics Statement

We evaluate PAM on epidemic forecasting tasks. The study uses publicly released aggregated surveillance data, without individual-level records.

References

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. URL https://openreview.net/forum?id=gerNCVqqtR.

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. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-1929.

Pier Giovanni Bissiri, Chris C. Holmes, and Stephen G. Walker. A general framework for updating belief distributions. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 78 (5):1103–1130, 2016. doi: 10.1111/rssb.12158.

Johannes Bracher, Evan L. Ray, Tilmann Gneiting, and Nicholas G. Reich. Evaluating epidemic forecasts in an interval format. *PLOS Computational Biology*, 17(2):e1008618, 2021. doi: 10.1371/journal.pcbi.1008618.

Centers for Disease Control and Prevention. CDC NWSS influenza A wastewater viral activity level by state/territory. https://www.cdc.gov/wcms/vizdata/NCEZID_DIDRI/FluA/nwssfluastateactivitylevelDL.csv, 2026. Accessed: 2026-04-08.

Centers for Disease Control and Prevention, Division of Healthcare Quality Promotion, National Healthcare Safety Network. Weekly United States hospital respiratory data (HRD) metrics by jurisdiction, National Healthcare Safety Network (NHSN). https://data.cdc.gov/d/ua7e-t2fy, 2026. Accessed: 2026-04-08.

Centers for Disease Control and Prevention, Respiratory Virus Response. NSSP emergency department visits—COVID-19, flu, RSV, combined—by demographic category. https://data.cdc.gov/d/7xva-uux8, 2024. Accessed: 2026-04-08.

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

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*, 2025. URL https://arxiv.org/abs/2504.19413.

Abhimanyu Das, Weihao Kong, Rajat Sen, and Yichen Zhou. A decoder-only foundation model for time-series forecasting. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10148–10167. PMLR, 2024. URL https://proceedings.mlr.press/v235/das24c.html.

Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997. doi: 10.1006/jcss.1997.1504.

Tilmann Gneiting and Adrian E. Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007. doi: 10.1198/016214506000001437.

Mark Herbster and Manfred K. Warmuth. Tracking the best expert. *Machine Learning*, 32(2): 151–178, 1998. doi: 10.1023/A:1007424614876.

Rob J. Hyndman and George Athanasopoulos. *Forecasting: Principles and Practice*. OTexts, 2021. ISBN 9780987507136. URL https://books.google.dk/books?id=gZB-zgEACAAJ.

Junfeng Liao, Qizhou Wang, Jianing Zhu, Bo Du, Rui Yan, and Xiuying Chen. Belief memory: Agent memory under partial observability. *arXiv preprint arXiv:2605.05583*, 2026. URL https://arxiv.org/abs/2605.05583v2. Version 2.

Kenichiro McAlinn and Mike West. Dynamic Bayesian predictive synthesis in time series forecasting. *Journal of Econometrics*, 210(1):155–169, 2019. doi: 10.1016/j.jeconom.2018.11.010. URL https://arxiv.org/abs/1601.07463.

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://arxiv.org/abs/2211.14730.

George Panagopoulos, Giannis Nikolentzos, and Michalis Vazirgiannis. Transfer graph neural networks for pandemic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4838–4845, 2021. doi: 10.1609/aaai.v35i6.16616.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. URL https://arxiv.org/abs/2412.15115v1. Version 1.

Adrian E. Raftery, Tilmann Gneiting, Fadoua Balabdaoui, and Michael Polakowski. Using Bayesian model averaging to calibrate forecast ensembles. *Monthly Weather Review*, 133(5):1155–1174, 2005. doi: 10.1175/MWR2906.1. URL https://sites.stat.washington.edu/raftery/Research/PDF/fadoua2005.pdf.

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.

Stephan Rasp and Sebastian Lerch. Neural networks for postprocessing ensemble weather forecasts. *Monthly Weather Review*, 146(11):3885–3900, 2018. doi: 10.1175/MWR-D-18-0187.1.

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

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-1657.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, 2023. doi: 10.52202/075280-0377.

Pranav Singh. When does belief-based agent memory help? Reliability-conditional updating and provenance-capped poisoning defense. *arXiv preprint arXiv:2606.22030*, 2026. URL https://arxiv.org/abs/2606.22030v2. Version 2.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=ehfRiF0R3a.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0593.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabás Póczos, Ruslan Salakhutdinov, and Alexander J. Smola. Deep Sets. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/f22e4747da1aa27e363d86d40ff442fe-Abstract.html.

A Implementation and Sequential Updating

A.1 PAM neural architecture and initialization

PAM uses separate location and scale networks, a contextual router, and a learned task prior (table 3). Each head is shared across skills and receives the 19-dimensional origin-available feature vector $\xi_{e,k}$. Its scalar output enters the bounded corrections in eq. (3).

Table 3: **PAM neural architecture.** Dimensions refer to the fitted Top-15 configuration with feature dimension $D = 19$ and hidden width $H = 32$. Black-box skills and the retrieval mechanism are excluded from the parameter count.

Component	Input	Architecture and output	Params.
Location correction	$\xi_{e,k} \in \mathbb{R}^{19}$	Independent skill-shared MLP, 19–32–32–1; tanh hidden units; scalar $\delta_{e,k}$	1,729
Scale correction	$\xi_{e,k} \in \mathbb{R}^{19}$	Independent skill-shared MLP, 19–32–32–1; tanh hidden units; scalar $a_{e,k}$	1,729
Contextual residual router	$\xi_{o,k}$, $\log W_{0,k}$, $\log \bar{W}_{o,k}$ (21 dimensions per skill)	DeepSets encoder 21–32–32; mean pool over skills; concatenate local and pooled states; decoder 64–32–1; tanh hidden units	3,873
Learned prior	Fixed Top-15 skill bank	One bounded residual logit per skill followed by softmax to form W_0	15
Total			**7,346**

All three neural output layers are initialized to zero, and the networks use no dropout. The router combines each skill's encoding with the mean encoding over the bank, giving a permutation-equivariant map; its output residual is bounded by $0.5\tanh(\cdot)$. A head with input dimension D and hidden width H has $H(D+1) + H(H+1) + (H+1)$ parameters, giving 1,729 for $D = 19, H = 32$.

A.2 Transformed summaries, censored outcomes, and mean alignment

For a raw mean μ and variance summary v, define

$$\mu^0 = \log(1 + \max\{\mu, 0\}), \qquad v^0 = \log\left(1 + \frac{\max\{v, 0\}}{(1 + \max\{\mu, 0\})^2}\right). \tag{11}$$

A skill without native uncertainty receives the variance of first differences of the history released at its forecast origin. The settings in eqs. (3) and (4) are $B_r = 2\sigma_{0,r}$, $c_s = \log 4$, $\gamma_r = 1$, and $\nu_r = 5$. The floor $s^2_{\min} = 10^{-6}$ is included in the squared kernel scale. For component–horizon key r, the raw upper support bound is

$$U_r = 1.25 \max\left\{1, \max_e y_e, \max_{e,k}(\mu_{e,k} + 4\sqrt{v_{e,k}})\right\}, \tag{12}$$

using nonnegative means and variances from training events in the calibration bank. Reference scales $\sigma_{0,r}$ are calibrated on its selected Top-10 bank. Both U_r and $\sigma_{0,r}$ remain fixed for the 31-skill library; feature and validation-score normalizers are estimated for the retrieved Top-15 bank.

A safeguarded Newton–bisection solver chooses the Student-t center so that the censored distribution matches the corrected raw-scale mean. Lower- and upper-tail probabilities are assigned to 0 and U_r, respectively:

$$Z \sim \mathrm{StudentT}(m, S, \nu), \qquad Y = \min\{U_r, \max\{0, e^Z - 1\}\}, \qquad \mathbb{E}[Y] = \mu^{\mathrm{req}}. \tag{13}$$

Here μ^{req} is the raw-scale mean after location correction. The bounded correction yields $\widetilde{m}$ in eq. (3); mean alignment determines the final center m. The bound U_r stays fixed during alignment.

For $0 < y < U_r$, the raw density is $q(\log(1+y))/(1+y)$. At 0 and U_r, the component assigns the corresponding tail probabilities. Its reference measure is

$$\xi_r = \delta_0 + \mathrm{Leb}_{(0,U_r)} + \delta_{U_r}. \tag{14}$$

For interior observations, the Jacobian factor $1/(1+y)$ is identical across skills and cancels in support normalization. Training and evidence scores use the density in z; reported raw-scale NLL adds $\log(1+y)$, and training-normalized NLL additionally subtracts $\log s_{\text{train}}$. Boundary probabilities are unchanged. Quantiles are obtained by inverting the weighted censored CDF. Diagnostic plots show the unbounded kernel q; the transformation in the preceding equation maps negative latent values to the boundary at $Y = 0$.

A.3 Active-set updates and the Bayesian interpretation

The target-key weights in eq. (5) are $\lambda_r = 1$ in the experiments. Let A_t contain skills with a forecast for every row of the released batch. If $M_t = \sum_{j \in A_t} W_{t-1,j}$, the implemented update is

$$W_{t,k} = \begin{cases} M_t \dfrac{W_{t-1,k} e^{\rho \ell_{t,k}}}{\sum_{j \in A_t} W_{t-1,j} e^{\rho \ell_{t,j}}}, & k \in A_t, \\ W_{t-1,k}, & k \notin A_t. \end{cases} \tag{15}$$

Inactive skills retain their weights, and evidence redistributes only the active set's existing mass M_t. Empty batches or zero active mass leave the state unchanged. With all skills active, eq. (15) reduces to eq. (6). Log-space evaluation adds $\rho\ell$, subtracts log-sum-exp, and restores M_t.

The exponentiated average log score is a loss-based evidence factor, yielding generalized Bayesian updating (Bissiri et al., 2016). A single modeled observation per step with $\rho = 1$ recovers finite-model Bayesian updating under conditional factorization. The state is a distribution over skill support; the network parameters remain fixed during deployment.

Proposition 1 (Anchored initialization)**.** *Fix the bank, archive, prior, bridge settings, and readout rule. If both evidence heads output zero, the pre-bridge location is $\mu^0_{e,k}$ and squared scale is $\sigma^2_{0,r_e} + \gamma_{r_e} v^0_{e,k} + s^2_{\min}$. Applying the same censoring and mean alignment yields the anchored components. Their support trajectory and the mixtures under the same readout rule are identical to those computed directly from that bridge.*

Proof. Substitution in eq. (3) gives the same bridge inputs and therefore the same component distributions and scores. Equal initial support and equal scores imply equal next support under eq. (15); induction establishes the whole path. Applying the same readout rule to equal states and components gives equal mixtures. □

Cumulative evidence identity. For a fixed bank active at every step, let $L_{t,k} = e^{\rho \ell_{t,k}}$. Expanding the normalized recursion gives

$$\prod_{t=1}^{T} Z_t = \sum_k W_{0,k} \prod_{t=1}^{T} L_{t,k}. \tag{16}$$

Successive normalizers cancel. For fixed evidence factors, this expression is order-invariant, while intermediate states and issued forecasts can differ. The identity describes cumulative evidence for a fixed, fully active bank. The four-block training objective includes additional terms; changing active sets require eq. (15).

A.4 Blockwise fitting and release-time deployment

The correction penalty over training row–skill pairs I is

$$\Omega(\theta) = \frac{1}{|I|} \sum_{(e,k) \in I} \left[\left(\frac{B_{r_e}}{\sigma_{0,r_e}} \tanh \delta_{e,k} \right)^2 + (c_s \tanh a_{e,k})^2 \right]. \tag{17}$$

The location block optimizes mixture-mean RMSE, the scale block optimizes censored-mixture NLL, and both use $\beta\Omega$ with $\beta = 10^{-3}$. The prior block minimizes

$$\mathcal{L}_{\text{prior}}(\psi) = -\sum_{t \in \mathcal{T}} \log Z_t(\psi) + R(\psi), \tag{18}$$

where $\mathcal{T}$ is the set of training release steps, Z_t is the evidence normalizer from support replay, and R combines posterior-distillation, Dirichlet, and macro-averaged squared-error terms. The contextual block optimizes mixture-mean RMSE.

Adam uses learning rate 10^{-3}, gradient-norm clipping at 1, at most 200 epochs, and early stopping after 20 unimproved epochs. Each pair of $\rho \in \{0.2, 0.3, 0.4, 0.5, 0.6\}$ for Bench-A or $\{0.1, 0.15, 0.2, 0.35, 0.5\}$ for CDC and $\alpha \in \{0, 0.25, 0.5, 0.75\}$ is trained separately, followed by refinement of ρ at midpoints between neighboring candidates. Validation uses 360 endpoints in ten rolling-origin folds replayed from the prior. The checkpoint and (ρ, α) minimize the standardized validation risk

$$0.80 z_{\mathrm{RMSE}} + 0.15 z_{\mathrm{WIS}_5} + 0.05 z_{\mathrm{NLL}}. \tag{19}$$

Each z_{metric} is standardized using the frozen bank's single-skill validation scores. Validation WIS uses quantiles $0.05, 0.25, 0.5, 0.75, 0.95$; reported WIS uses deciles (section C.3).

Selected (ρ, α) are $(0.3, 0.25)$ for two Bench-A seeds and $(0.3, 0.5)$ for one, $(0.5, 0)$ for every B-COVID seed, and $(0.5, 0.75)$ for B-FLU. All fitted parameters are frozen during validation and testing, while released outcomes update support.

Algorithm 1 PAM training and release-time prediction

Require: Forecast archive; selected skill bank; fixed distribution settings
Initialize evidence heads and contextual network with zero output layers
Fit location, scale, prior, and proposal in alternating replay blocks
Select checkpoint and (ρ, α) on validation; freeze fitted parameters
Initialize W_0; commit the initial components and predictive mixture
for each release step t **do**
    Read newly eligible outcomes and their origin-committed components
    Score active skills and update support by eq. (15)
    if a forecast is requested after the information cutoff **then**
        Execute available skills on released history; construct new components
        Read the latest support and form the mixture by eq. (7)
        Commit components, features, mixture, checkpoint, and target provenance
    end if
end for

Each forecast is stored with its origin, target, horizon, target version, and component parameters. The prescribed target version is scored once after release. Missing outcomes contribute no evidence; later revisions can enter future histories while issued forecasts remain unchanged.

A.5 Hardware and execution

Restricted controls ran on AMD MI250X GPUs. Runtime was measured on one MI250X graphics compute die: a neural-head pass over 360 events took 38–42 ms, support replay over 26k–42k released rows took 26–90 ms, and the end-to-end pass took 314–397 ms. Each value is a batch-timing median over three fitted runs and three synchronized repetitions. The experiment campaign used 408.027 allocated GPU-hours, including unsuccessful and verification runs.

B Analysis of Sequential Evidence

For two skills i, j active at every release step, the common normalizer cancels:

$$\log \frac{W_{t,i}}{W_{t,j}} = \log \frac{W_{0,i}}{W_{0,j}} + \rho \sum_{s=1}^{t} (\ell_{s,i} - \ell_{s,j}). \tag{20}$$

The log odds accumulate pairwise score differences. A change in one skill's evidence affects its support relative to the evidence for competing skills.

In the Gaussian limit, the interior log score is $-\log S - (z-m)^2/(2S^2) - \frac{1}{2}\log(2\pi)$. Location determines the residual, while scale balances fit against the concentration penalty $-\log S$. This explains how equal residuals can produce different writes. Predictive performance is assessed with

held-out proper scores (Gneiting & Raftery, 2007). Evaluating the two heads and contextual network costs $O(K(DH + H^2))$ per target, and support updating costs $O(K)$, excluding skill execution. The support vector requires $O(K)$ storage; the archive grows with issued predictions.

C Evaluation Protocol

C.1 Data splits and release-time replay

Table 4: **Forecast-origin splits.** Embargo outcomes can update memory after release but are not scored. CDC dates apply to B-COVID and B-FLU.

Split	Bench-A	CDC
Training	15 March–9 April 2020	26 November 2022–8 March 2025
Validation	10–23 April 2020	15 March–16 August 2025
Embargo	24–30 April 2020	23 August–13 September 2025
Test	1–5 May 2020	20 September 2025–28 February 2026

Bench-A contains daily COVID-19 case counts for 349 regions in England, France, Italy, and Spain from PANDEMIC-TGNN (Panagopoulos et al., 2021). B-COVID and B-FLU use finalized weekly NHSN hospital-admission rates per 100,000 for 51 U.S. jurisdictions (Centers for Disease Control and Prevention, Division of Healthcare Quality Promotion, National Healthcare Safety Network, 2026). Bench-A uses direct 1- and 3-day forecasts and a recursive 7-day endpoint; the CDC tasks use direct 1- and 2-week forecasts and a recursive 4-week endpoint. CDC outcomes become available after a one-week lag. Both CDC tasks use release-lagged emergency-department covariates (Centers for Disease Control and Prevention, Respiratory Virus Response, 2024); B-FLU additionally uses influenza-A wastewater measurements with a 28-day lag (Centers for Disease Control and Prevention, 2026). Two Bench-A skills use mobility-graph summaries. Evaluation applies the specified release lags to finalized targets; historical surveillance revisions are not reconstructed.

The library contains 31 statistical, state-space, compartmental, renewal, neural, and pretrained skills. Retrieval uses Qwen2.5-7B-Instruct embeddings (Qwen Team, 2024) to select 15 skills per task. The same archived base forecasts are used across PAM fits. PAM and its retrained restrictions share targets, splits, seeds, and selection budgets. Pretraining exposure to the evaluation series is unknown. Section A.5 describes the hardware and runtime measurements.

C.2 Strategy baselines and repeated-fit reporting

All three agents select from the same 31-skill library used for PAM's retrieval. Top-1 selects by description retrieval; Rediscovery repeats selection at each origin; ReAct adapts reasoning-and-action prompting to skill selection (Yao et al., 2023). The selected skills' archived forecast moments are mapped to predictive distributions using the fixed zero-head censored moment-t bridge. This comparison evaluates the resulting selection strategies on the common test events.

DMA combines PAM's Top-15 bank using anchored components. Its output-level weight updates are applied to archived forecasts, with settings selected from a 20-value grid using PAM's validation risk. The underlying skill parameters are held fixed. Section E reports individual forecast performance under the common bridge.

Error-driven support controls. Panel C of table 2 compares Hedge (absolute error), inverse recent error, and Fixed Share (absolute error). These rules update support from observed errors on the same Top-15 bank and release schedule, with anchored predictive components. Their complete-test scores are compared with full PAM in Panel A. Panel B instead holds the learned readout components fixed and evaluates subsequent forecasts.

Repeated fits. PAM and the module restrictions use three independent fits with seeds 42, 43, and 44. Each run is scored separately; tables report the arithmetic mean and sample SD (ddof=1). Panel B replays each write rule with each of the three PAM checkpoints. The independently trained reference architectures in table 6 use the same three-fit summary.

Agent, DMA, and Panel C results are single deterministic evaluations of fixed forecast or selection archives. Their training-run SD is unavailable. Cov90 and Width90 describe predictive intervals; SD measures score variation across fitted runs, conditional on the shared base forecasts. Figure 2 and the aggregate diagnostics use all three fits. The matched case studies use seed 42 with the selection rules in section D. Strategy-level comparisons report mean differences; paired confidence intervals are provided for the matched-write and mechanism analyses.

C.3 Metrics, aggregation, and matched comparisons

For central interval endpoints l, u and miscoverage a, the interval score is

$$\mathrm{IS}_a(l, u; y) = (u - l) + \frac{2}{a}(l - y)_+ + \frac{2}{a}(y - u)_+, \tag{21}$$

where $(x)_+ = \max\{x, 0\}$. Reported decile WIS is

$$\mathrm{WIS}_9 = \frac{0.5|y - q_{.5}| + \sum_{a\in\{.2,.4,.6,.8\}}(a/2)\mathrm{IS}_a(q_{a/2}, q_{1-a/2}; y)}{4.5}, \tag{22}$$

where q_p is the forecast's p-quantile. Validation uses $a \in \{.1, .5\}$ with denominator 2.5, giving the five-quantile WIS in eq. (19). Cov90 and Width90 use $[q_{.05}, q_{.95}]$ (Bracher et al., 2021). Cov90 is the empirical coverage, with target 0.90; Width90 is the average interval length, interpreted jointly with coverage.

Scores are averaged within countries for Bench-A and jurisdictions for CDC, then across reporting units, horizons, and strategies. Direct and recursive forecasts each receive half the total weight, giving endpoint weights $1/4, 1/4, 1/2$. Regional squared errors are pooled before taking the square root. RMSE, WIS, and Width90 are divided by the training-target SD; coverage is unscaled. The same hierarchical aggregation, followed by equal weighting of fitted runs, defines Avg_o in eqs. (9) and (10).

Paired 95% confidence intervals use 2,000 circular time-block bootstrap replicates, sharing each sampled block across the compared methods. Blocks contain one origin for Bench-A and four origins for CDC, with resampling seed 20260923. These intervals describe variation over evaluation periods conditional on the fitted forecasts. Bench-A has five test origins and CDC has 24; Bench-A intervals are interpreted descriptively.

Panel A of table 2 independently retrains each module restriction. Panel B uses the same learned readout components with $\alpha = 0$ and changes the scoring distribution for support updates: fitted evidence or anchored zero-head evidence. Its paired comparisons use the same later-origin subset. On Bench-A, the learned-minus-anchored WIS difference is -0.00889, with interval $[-0.00919, -0.00864]$. Panels A and C use the complete test set; Panel B evaluates only forecasts issued after the eligible updates.

D Empirical Mechanism Analyses Across Tasks

These analyses use the selected checkpoints, committed forecasts, released outcomes, and support states from the forecasting experiments. The diagnostic protocols were specified after the primary evaluation, with checkpoints kept fixed. Complete-test statistics assess predictive performance; selected cases illustrate the numerical behavior of the write and read operations.

D.1 Prospective alignment of the support state

Support trajectories include the initial state and every release update for the three fits in section C.2. The three named skills have the largest mean initial support; "Other" contains the remaining 12 skills' combined mass. Lines show means and ribbons show ranges across fits. Early, middle, and late divide each update sequence into three equal-length portions. Lower strips show the released-batch median of $z = \log(1 + y)$.

Bench-A contains 51 release updates and 5,235 test forecast–outcome rows per fit. Each CDC task has 169 updates and 3,672 test rows. The test periods contain five origins for Bench-A and 24

for CDC. Equation (9) compares initial and issue-time support using the subsequent WIS of each component on common eligible targets.

Table 5: **Complete-test memory diagnostics.** Entries are estimates with paired 95% time-block confidence intervals. All effects use the training scale. Positive alignment favors updated support; positive soft-over-hard effects favor the mixture. Memory-only readout sets $\alpha = 0$; issued readout uses the selected share. Bench-A intervals are descriptive.

Effect	Bench-A	B-COVID	B-FLU
Δ_{align}	-0.00020 $[-0.00102, 0.00077]$	0.0508 $[0.0207, 0.0825]$	0.1799 $[0.0632, 0.2980]$
Δ_{soft}, memory	0.01797 $[0.01639, 0.01955]$	0.000050 $[-0.000574, 0.000624]$	0.00806 $[0.00266, 0.01379]$
Δ_{soft}, issued	0.01997 $[0.01835, 0.02168]$	0.000050 $[-0.000574, 0.000624]$	0.00265 $[-0.04246, 0.04486]$

Both CDC tasks have positive prospective alignment. The Bench-A interval includes zero. Alignment measures whether updated weights favor subsequently useful components; the soft-readout comparisons separately evaluate mixture performance.

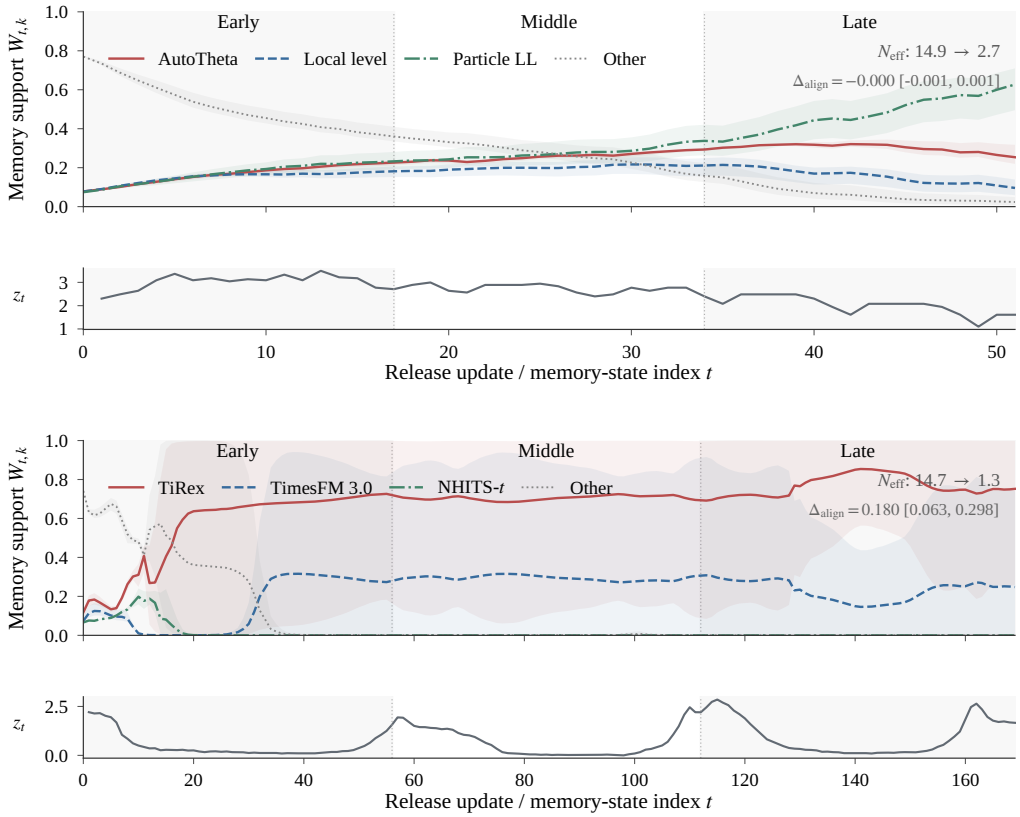


Figure 5: **Additional empirical support trajectories.** Top: Bench-A regional daily COVID-19 cases. Bottom: B-FLU weekly admission rates across 51 U.S. jurisdictions. All formal release updates are shown, with seed means/ranges and the full undisplayed support retained in "Other." Lower strips show log outcomes. Period labels identify equal-length portions of the update sequence.

D.2 Matched residual-scale interventions

For each task, we select the seed-42 test event with the largest absolute fitted-to-reference log-scale ratio among events with interior outcomes; event and skill order break ties. The intervention replaces

the residual-scale multiplier by one, holding the kernel center, base variance, degrees of freedom, outcome, prior support, release batch, and competing scores fixed. This within-event comparison measures score sensitivity to residual scale.

The B-COVID case is a TiRex forecast for Oregon, issued on 7 February 2026 for 7 March, with observed admissions of 0.33 per 100,000 (fig. 3). The fitted and reference kernels assign densities 1.200 and 1.033 to the same outcome. Full-batch normalization yields only a small difference in updated support.

Figure 6 gives the additional cases: TiRex for Torino, issued on 3 May 2020 for 10 May, and NHITS-t for Colorado, issued on 20 September 2025 for 18 October. For Torino, the fitted density is lower than the reference (0.173 versus 0.300); the displayed updated weights coincide at the reported precision. For Colorado, the fitted density is higher (1.208 versus 0.765), but the pre-update support is approximately 10^{-33}. These cases show how competing scores and existing support determine the magnitude of a write.

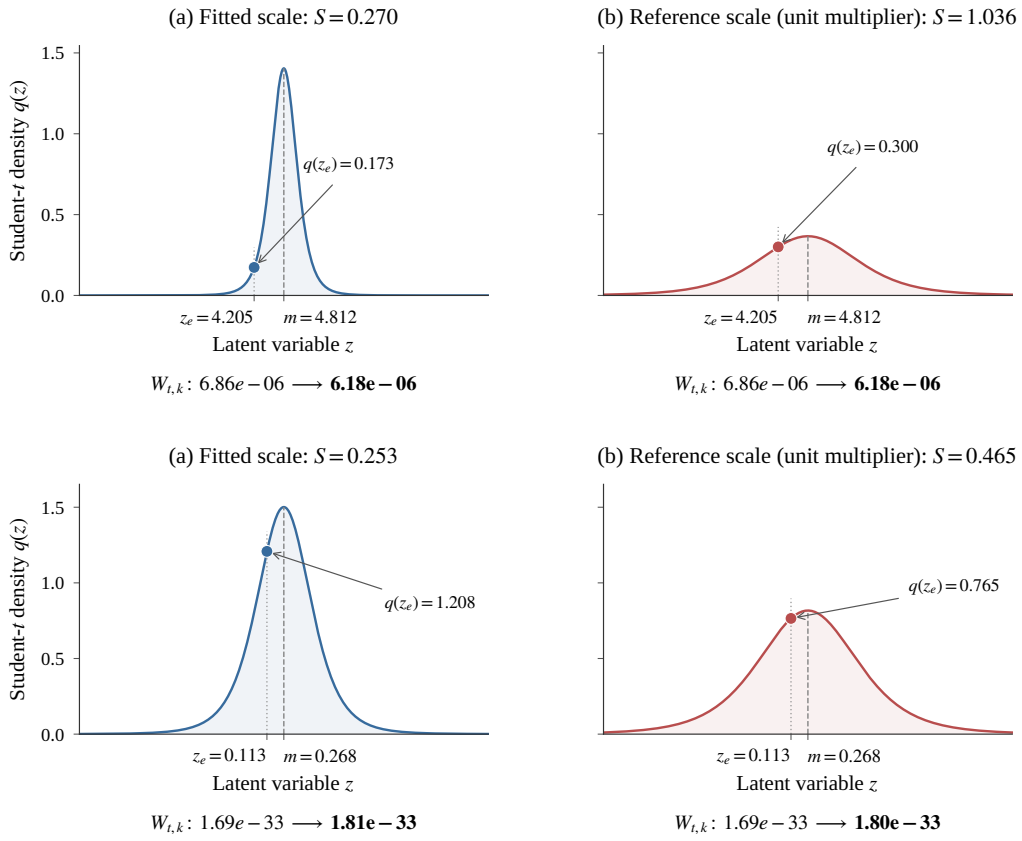


Figure 6: **Matched scale interventions on Bench-A and B-FLU.** Top: TiRex for Torino. Bottom: NHITS-t for Colorado. Only residual scale changes in each pair; centers, outcomes, competing evidence, and axes are fixed. Curves show the unbounded latent Student-t kernels q. Formal predictive scores use the censored distributions.

D.3 Fixed-component memory readouts

Each seed-42 case uses the earliest test origin and shortest direct horizon, then selects the target with the greatest prediction-only component disagreement. All 15 component parameter triples are held fixed. The two panels compare W_0 with $\bar{W}_o$ on the same current components. Black curves sum all weighted kernels, including the 12 contributions grouped as "Other." The figures display the unbounded latent kernels; complete-test WIS uses the censored distributions.

On B-COVID, the South Carolina forecast issued on 20 September 2025 for 27 September has effective support N_{eff} decreasing from 14.74 to 1.21 and TimesFM-3 weight increasing from 0.1167 to 0.9524 (fig. 4). Its selected share is $\alpha = 0$, so the updated weights are also those used at prediction time. The complete-test soft-over-hard interval includes zero.

Figure 7 shows Essonne's one-day-ahead forecast issued on 1 May 2020 and the District of Columbia's one-week-ahead forecast issued on 20 September 2025. Their effective supports decrease from 14.94 to 2.46 and from 14.59 to 1.01, respectively. The selected contextual shares are 0.25 and 0.75, so the updated panels isolate memory-only readouts. Table 5 reports memory-only and issued-mixture effects separately: both favor soft readout on Bench-A; on B-FLU, the memory-only interval is positive and the issued-mixture interval includes zero.

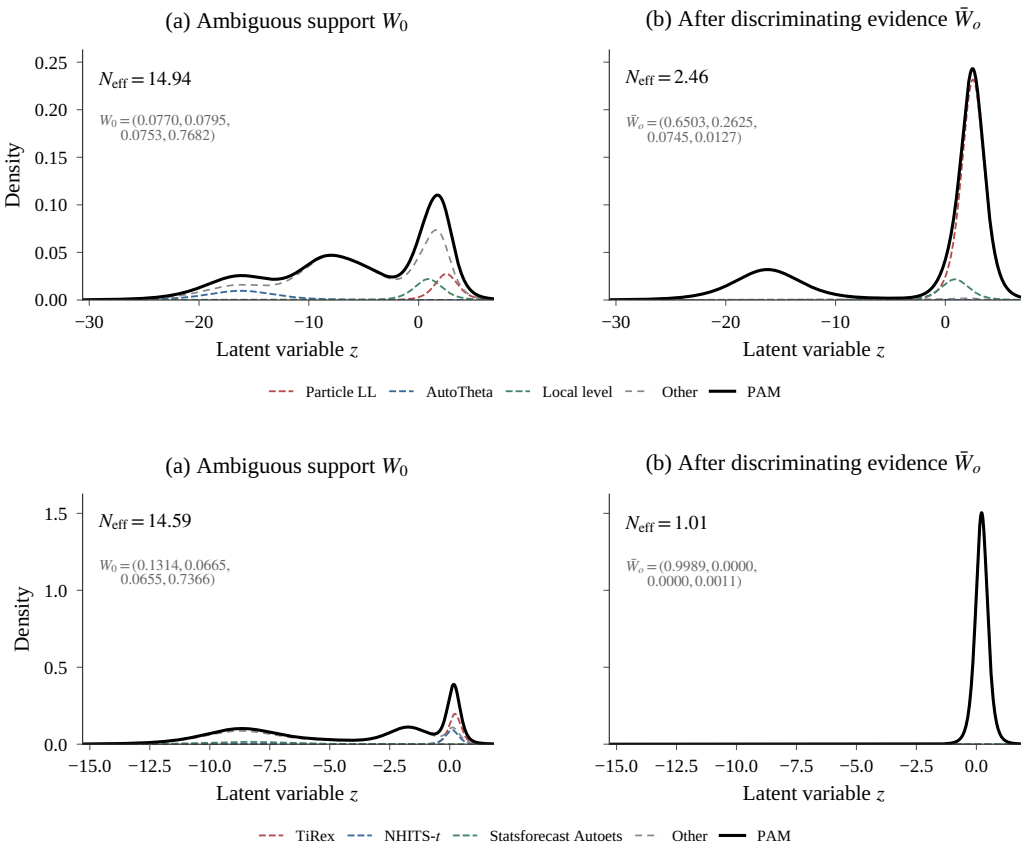


Figure 7: **Additional fixed-component readouts.** Top: Bench-A, Essonne one-day forecast issued 1 May 2020. Bottom: B-FLU, District of Columbia one-week forecast issued 20 September 2025. Each pair uses the same 15 unbounded kernels and identical axes; only the state changes. The updated panels use memory-only weights; the deployed readouts additionally use contextual shares of 0.25 and 0.75, respectively.

The trajectory, scale, and readout analyses characterize accumulated support, evidence sensitivity, and predictive combination. Table 5 summarizes their complete-test predictive results.

E INDIVIDUAL FORECAST DIAGNOSTICS

Table 6 reports individual forecast performance under the common fixed zero-head bridge. Representative models in these diagnostics include the statistical forecasters Seasonal naive, AutoARIMA, and AutoETS (Hyndman & Athanasopoulos, 2021), as well as pretrained time-series models such as Chronos (Ansari et al., 2024), TimesFM (Das et al., 2024), and TiRex (Auer et al., 2025). The independently retrained neural references (Challu et al., 2023; Salinas et al., 2020; Nie et al., 2023), marked [R], use three fits with at most 1,000 optimization steps and early stopping after five unimproved checks, evaluated every 20 steps. For these references, the bridge's variance input is the variance of ten archived uncertainty centers, used as a predictive summary.

Table 6: **Individual forecasts under a common probability mapping.** [B]: a skill in the task's Top-15 bank; [L]: a library skill outside that bank; [R]: an independently retrained architecture reference. Mean ± sample SD summarizes three retrained fits; frozen skills have single scores. Metrics follow table 1. All rows use the fixed zero-head censored moment-t bridge.

Forecast / archive role	RMSE ↓	WIS ↓	Cov90	Width90
Bench-A				
Seasonal naive [B]	0.4722	0.1166	0.6985	0.6083
AutoARIMA [L]	0.4037	0.1085	0.7772	0.6204
AutoETS [L]	0.3511	0.1087	0.7490	0.5195
Chronos-Bolt [L]	0.5394	0.1168	0.7709	0.6685
TimesFM-2 [L]	1.7585	0.1656	0.6969	1.0196
TimesFM-3 [B]	0.3464	0.1058	0.8117	0.5160
TiRex [B]	0.3726	0.1060	0.8073	0.5405
NHITS-t [R]	0.4204 ± 0.0876	0.1167 ± 0.0054	0.7346 ± 0.0886	0.4940 ± 0.1352
DeepAR-t [R]	0.4092 ± 0.0418	0.1111 ± 0.0038	0.7740 ± 0.0546	0.5242 ± 0.0571
PatchTST +Cal [R]	0.3996 ± 0.0044	0.1068 ± 0.0007	0.7446 ± 0.0086	0.5720 ± 0.0082
B-COVID				
Seasonal naive [L]	0.5067	0.2977	0.8562	1.3044
AutoARIMA [B]	0.3384	0.1978	0.9616	1.2740
AutoETS [L]	0.3810	0.2010	0.9718	1.4031
Chronos-Bolt [B]	0.4293	0.2017	0.9489	1.3750
TimesFM-2 [B]	0.3281	0.1736	0.9761	1.4003
TimesFM-3 [B]	0.2912	0.1490	0.9771	1.1606
TiRex [B]	0.3133	0.1549	0.9741	1.2089
NHITS-t [R]	0.3182 ± 0.0033	0.1776 ± 0.0019	0.9126 ± 0.0126	1.0423 ± 0.0152
DeepAR-t [R]	0.3175 ± 0.0098	0.1711 ± 0.0046	0.9125 ± 0.0118	0.9960 ± 0.0187
PatchTST +Cal [R]	0.3427 ± 0.0288	0.2256 ± 0.0103	0.9221 ± 0.0150	1.3811 ± 0.0181
B-FLU				
Seasonal naive [L]	1.5012	0.8820	0.3098	1.2085
AutoARIMA [B]	1.1327	0.5842	0.5233	1.3808
AutoETS [B]	1.3158	0.6333	0.5374	1.7020
Chronos-Bolt [L]	1.6319	0.5271	0.8143	3.1122
TimesFM-2 [B]	1.3888	0.4898	0.8362	2.5642
TimesFM-3 [B]	0.9245	0.4296	0.8366	1.7750
TiRex [B]	0.9477	0.4474	0.8562	1.7164
NHITS-t [R]	1.8048 ± 0.3030	0.6095 ± 0.0466	0.8234 ± 0.0106	2.9065 ± 0.3739
DeepAR-t [R]	1.1451 ± 0.0178	0.6564 ± 0.0245	0.6741 ± 0.0215	0.9976 ± 0.0993
PatchTST +Cal [R]	1.6253 ± 0.4128	0.6822 ± 0.0776	0.5362 ± 0.0156	1.9973 ± 0.2788

The fixed bridge maps archived forecast moments to a censored distribution; WIS and interval metrics are computed from that distribution. Its mean matches the nonnegative archived point forecast, and its variance input uses frozen forecast summaries.

TimesFM-3 has the lowest mean WIS among the bridged profiles on all three tasks and a lower mean than PAM on the CDC tasks. These are descriptive comparisons under the common probability

mapping. Figure 2 and table 5 separately examine how PAM's learned components receive and use memory support.