Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Predictive Reliability and Calibration of Workload Autoscaling Models for Large Language Model Inference Clusters Under Sparse Telemetry and Runtime Anomalies

Mingxuan Tao

lip931122@gmail.com

Emporia State University

Research Article

Keywords: large language model serving, autoscaling, probability calibration, predictive reliability, distribution shift, cloud resource management

Posted Date: November 13th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11706225/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Predictive Reliability and Calibration of Workload Autoscaling Models for Large Language Model Inference Clusters Under Sparse Telemetry and Runtime Anomalies

***Abstract*—Autoscalers for large language model (LLM) inference clusters that add capacity when a predicted overload probability crosses a threshold depend on the probability scale, not only on the ranking. This simulation study evaluates five such predictors under sparse telemetry and runtime anomalies on two latent-state generators and a request-level serving simulator with queueing, service-level objectives (SLOs), startup delay, and modeled power. With 3-period instead of 50-period histories, models trained on long histories became overconfident (expected calibration error up to 0.099) or conservative, whereas a Markov filter stayed calibrated and retraining on short histories removed most of the error. Temperature scaling fitted on long histories did not transfer; refitted for short histories, it cut this error by up to 68.7%, removed all 290 wrong high-confidence predictions of one model by lowering their confidence, and cut the overconfident models' idealized unmitigated overload by 59.8–82.8% at the cost of more idle capacity. With a 20-s startup delay in the request-level simulator, learned predictors used 24.0–25.3% less modeled energy than always-on scale-out, with a 1.0–1.2 percentage-point increase in the SLO-violation rate, and 95.3% of scale-out-induced violations occurred on slow new replicas. Calibration should therefore be validated per telemetry regime and checked against request-level outcomes.**

Keywords—large language model serving, autoscaling, probability calibration, predictive reliability, distribution shift, cloud resource management

I. Introduction

Serving large language models (LLMs) requires provisioning graphics processing unit (GPU) replicas ahead of demand. Iteration-level scheduling [1], paged key–value (KV) cache management [2], chunked prefills [3], and prefill–decode splitting [4] have raised serving throughput, but KV-cache occupancy still grows with every decoded token, so a burst of long prompts can stall a batch, and tail latency turns such stalls into service-level objective (SLO) violations [5]. Late provisioning violates SLOs, whereas idle warm replicas consume energy in data centers whose electricity use is rising quickly [6].

Predictive autoscalers act on forecasts rather than on current utilization [7]–[9]. When a scale-out is triggered by a predicted overload probability crossing a threshold, the probability scale matters as much as the ranking: among periods predicted at 0.8, about 80% should overload. The telemetry behind such predictions is often degraded. Newly provisioned instances have short histories, monitoring records can be wrong, and stragglers and gray failures distort the relation between load and latency [10]; long-context requests, traffic spikes, and model hangs also undermine utilization-based autoscalers [11], while the NIST AI Risk Management Framework counts robustness across varied circumstances among the properties of trustworthy AI [12]. This paper addresses three research questions (RQs) in simulation.

RQ1: *How does limited telemetry history length affect overload discrimination, probability calibration, and downstream SLO compliance across workload prediction models?*

RQ2: *How do runtime straggler anomalies, prompt-length surges, and corrupted telemetry records affect high-confidence prediction errors and resource provisioning waste?*

RQ3: *Can simple post-hoc temperature scaling improve predictive reliability in cluster scheduling, and when do its benefits fail to transfer across workload regimes?*

The study is an evaluation of predictive reliability; it does not propose a new autoscaling policy. Its contributions are as follows.

1) A controlled evaluation of five overload predictors on two latent-state generators and a request-level simulator, with control predictors that test training–deployment mismatch, input channels, and initialization as explanations of short-history miscalibration.
2) A comparison of directly transferred and target-regime post-hoc calibration (temperature and Platt scaling), including how recalibration removes high-confidence errors and how it shifts idealized provisioning costs.
3) A request-level counterfactual analysis that separates idealized decision costs from realized SLO violations and modeled energy and shows how startup delay, backlog, and slow nodes limit the benefit of better probabilities.

II. Related Work

LLM serving and autoscaling. Autopilot sets resource limits and replica counts from historical usage [7], and SageServe combines traffic forecasts with routing and scaling of LLM deployments, noting that loading large models can block GPUs for seconds to minutes [8]. Uncertainty-aware hierarchical graph transformers have been applied to resource optimization in service meshes [9], anomaly-aware scheduling has targeted failure modes of LLM serving [11], and MAPS schedules requests with calibrated output-length bounds [13]. These studies evaluate forecast accuracy or end-to-end policy performance; they do not isolate how the calibration of an overload probability changes when telemetry is sparse or corrupted.

Calibration and dataset shift. Modern neural networks are often miscalibrated, and temperature scaling is a simple post-hoc remedy [14]; Platt scaling fits a slope and an intercept [15]; classical learners differ widely in probability quality [16]. Calibration fitted in distribution can degrade under dataset shift [17], and binned calibration estimates are biased and depend on the binning scheme [18], [19]. This paper applies these tools to autoscaling predictors, treats history truncation and marker corruption as controlled shifts, and links calibration changes to idealized and request-level decision outcomes.

III. Methodology

A. Prediction Task and Data Splits

Each generator simulates 2,000 serving instances in 200 pools (10 model-size and quantization groups) over 60 scheduling periods. Each period t of instance i emits six telemetry channels x_{it}: arrival rate, mean input and output token lengths, KV-cache occupancy, 95th-percentile latency, and a binary overload marker. The target $y_{it} \in \{0, 1\}$ indicates an overload or SLO violation in period t. A predictor receives the visible window $W_{it}(L) = (x_{i,t-L}, \ldots, x_{i,t-1})$ and the instance's pool and group, and returns p_{it}, the predicted probability that $y_{it} = 1$. The targets are periods $t = 50, \ldots, 59$, so windows of every length $L \in \{3, 5, 10, 20, 50\}$ end at the same periods and window length is not confounded with operating phase. Instances are split into 1,200 training, 300 validation, 200 calibration, and 300 test instances, giving 3,000 test and 2,000 calibration events per replicate. Each generator has five evaluation replicates (seeds 1–5) and one development replicate used only for hyperparameter selection.

B. Workload Generators

Synthetic-LLM-A and Synthetic-LLM-B (hereafter A and B) are abstract: labels are Bernoulli draws from a latent capacity state and telemetry is a noisy function of that state, so the information available to predictors is controlled, but requests are not simulated. In A, a latent state $z_{it} \in \{0, 1\}$ separates regular traffic from congestion and follows a Markov chain whose onset and recovery rates vary by group, pool, and instance. Overload follows

$$P(y_{it} = 1 \mid z_{it} = 1) = 1 - s_j,$$
$$P(y_{it} = 1 \mid z_{it} = 0) = g_j, \tag{1}$$

where the floor g_j of group j is the probability that a prompt-length spike overflows the cache during regular traffic and s_j the probability that batching drains the queue in time during congestion (defaults $g = 0.20$ and $s = 0.15$, around which group values vary). In B, a continuous load index θ_{it} combines an instance baseline, capacity drift, a first-order autoregressive (AR(1)) disturbance, and the offset of the hosting node; instances migrate with probability 1/30 per period, and each new node is a straggler with probability p_{str} (default 0.15) and a heavy-tailed offset. Overload follows

$$P(y_{it} = 1) = g_j + (1 - g_j - s_j)\,\sigma(a_j(\theta_{it} - b_j)), \tag{2}$$

where σ is the logistic function and a_j and b_j are the group slope and location.

Synthetic-LLM-C (hereafter C) is a request-level simulator in which labels arise from simulated requests. Each instance is one model replica with a paged KV cache of 24.8k–64k tokens (by group) and at most 64 running sequences, simulated in 1-s ticks within 60-s periods. Poisson arrivals follow a load level with AR(1) variation, bursts, and drift; during surges, 35% of prompts are about 6,000 tokens long. Requests are admitted first-come-first-served while KV blocks are free; the first token is emitted when prefill completes. A decode iteration takes a fixed time plus terms linear in the number of running sequences and in the resident KV tokens, scaled by the node's slowdown (stragglers occur with probability 0.15 and slow a node by 1.3–4×). KV overflow preempts the latest request, and requests still queued after 60 s are abandoned. A period violates its SLO when the nearest-rank 99th-percentile time to first token (TTFT) or time per output token (TPOT) exceeds group thresholds set at 2 and 1.5 times the 99th percentile of an anomaly-free pilot at 50% load. As in online monitoring, TTFT observations include requests still waiting at the period end (at their elapsed wait, a lower bound) and count abandoned and rejected requests as exceeding; TPOT covers the period's completions. Offered load is 0.42 of nominal throughput, giving a test base rate of 0.39 (0.47 in A and 0.41 in B). Engine and power parameters are illustrative and were not fitted to hardware measurements or production traces.

C. Predictors, Matched Training, and Controls

Five predictors are compared. (i) *Baseline* returns the smoothed pool rate $p_j = (c_j + \alpha\, p_0)/(n_j + \alpha)$, where c_j counts training overload periods in pool j, n_j is the number of its training instance-periods, p_0 is the overall training rate, and α is a pseudo-count; it ignores telemetry. (ii) *Markov-Queue* is a two-state hidden Markov model per group with a Bernoulli emission for the overload marker and a Gaussian emission for KV occupancy, fitted by Baum–Welch; its forward filter starts each window from the stationary distribution, and it reads two of the six channels. (iii) *Static-Logit* is a logistic regression on the window mean and last value of each channel plus group indicators, fitted by maximum a posteriori estimation with a Gaussian prior. (iv) *GRU-Scaler* passes embedded telemetry through a single-layer gated recurrent unit [20] from a zero state to a logistic output. (v) *Attn-Scaler* uses one attention head [21] that queries with the latest embedding and applies a learnable recency penalty to the attention logits before the softmax, followed by a two-layer head. The neural models were trained with Adam [22] and early stopping on validation negative log-likelihood (NLL); all hyperparameters were selected on the development replicate with clean 50-period windows.

Under *test-only shift*, all models were trained on clean full windows ($L = 50$, marker error rate $\eta = 0$) and evaluated on shorter or corrupted windows; marker errors invert each visible overload marker with probability $\eta \in \{0.05, 0.10, 0.20\}$ and leave the continuous channels and the labels clean. Under *matched training*, Static-Logit, GRU-Scaler, and Attn-Scaler were retrained on windows with the evaluated L or η. Three controls separate competing explanations: Static-Logit restricted to Markov-Queue's two channels (2-ch), Markov-Queue with all six channels as diagonal Gaussians (6-ch), and GRU-Scaler started from the mean training hidden state at the window start (warm start). Finally, the floors ($g \in \{0.10, 0.30\}$,

$s \in \{0.05, 0.25\}$) and the straggler probability of B ($p_{str} \in \{0, 0.30\}$) were varied with data regenerated and all models refitted.

D. Post-Hoc Calibration

Temperature scaling [14] and Platt scaling [15] map a prediction p to

$$p' = \sigma(\text{logit}(p)/T), \quad p' = \sigma(\kappa \, \text{logit}(p) + \beta), \tag{3}$$

with temperature $T > 0$, slope κ, and intercept β fitted by calibration-set NLL. *Direct transfer* fits the map once on clean 50-period calibration windows and applies it unchanged, as a map frozen at deployment would be; *target-regime recalibration* fits it on calibration windows with the test L and η. A positive temperature preserves the sign of every logit, so it never changes the predicted class $\hat{y} = I[p \geq 0.5]$, where $I[\cdot]$ is the indicator function.

E. Reliability Metrics and Statistics

Discrimination is measured by the area under the receiver operating characteristic curve (AUC; Mann–Whitney estimator). Probability quality is measured by the Brier score, NLL, calibration-in-the-large (CITL, mean prediction minus observed rate), and the expected calibration error

$$\text{ECE} = \Sigma_k \, (n_k/n) \, |\bar{y}_k - q_k|, \tag{4}$$

with $K = 15$ equal-width bins, where n_k, $\bar{y}_k$, and q_k are the count, observed overload frequency, and mean prediction in bin k and n is the number of events; four alternative binnings served as checks [18]. A prediction is high-confidence when $\max(p, 1 - p) \geq 0.90$; pooled over the 15,000 test events of the five replicates, the number of such predictions, the number that predicted the wrong class, and their coverage (number/15,000) are reported. Point estimates are means over replicates, and differences are computed before rounding. The 95% confidence intervals (CIs) in brackets come from 2,000 instance-level bootstrap draws [23] that resample test instances within each replicate and keep each instance's 10 target events together; they are bias-adjusted because binned ECE is biased upward under resampling [19].

F. Decision Evaluation

A scale-out is triggered when $p \geq \tau$. If a missed overload costs three times an idle replica-period ($c_{FN} = 3c_{FP}$), the Bayes threshold for calibrated probabilities is

$$\tau = c_{FP}/(c_{FP} + c_{FN}) = 0.25. \tag{5}$$

This threshold follows from the assumed cost ratio and is optimal only for calibrated probabilities under it; other cost ratios imply other thresholds. The *idealized proxy* assumes that a scale-out acts at once and absorbs the period's overload. It yields the unmitigated overload rate (UOR, the share of events with $y = 1$ and $p < \tau$), the over-provisioning rate (OPR, the share with $y = 0$ and $p \geq \tau$), and the decision cost OPR + 3UOR in units of c_{FP}; these are decision-model quantities, not simulated outcomes.

In C, each test target period is also re-simulated with the same arrivals after a scale-out at the period start launches one replica on a new node (a straggler with probability 0.15). The replica becomes ready after a startup delay $d \in \{0, 20, 40\}$ s, takes half of the waiting queue and a queue-length-weighted share of later arrivals, and is released at the period end. This single-period counterfactual yields the realized SLO-violation rate (RVR) and modeled energy; capacity does not persist across periods. Each replica draws $0.25 + 0.45u$ kW in a tick with busy share u (prefill work plus the union of decode intervals) and 0.35 kW while starting. Energy is reported in Wh per target period and instance; it is a model estimate, not a measurement. Because existing replicas had a mean busy share of 0.99 in the target periods, energy differences between policies come almost entirely from the added replica. All data are synthetic and are regenerated from fixed random seeds; the simulation and analysis code will be made publicly available upon publication.

IV. Experimental Results

A. Long Versus Short Histories (RQ1)

With $L = 50$ (Table I), the learned models had similar AUC in A (0.717–0.721) and B (0.721–0.725), against 0.841 and 0.756 for the true generating probabilities; neither sequence model outperformed Static-Logit (paired differences −0.004 to −0.001, all CIs within [−0.007, 0.002]). Violations in C were persistent (only 17.3% followed a period that met its SLO) and easier to rank. Telemetry-based models were calibrated on clean windows (ECE 0.018–0.033). Across the four alternative binnings, ECE orderings held in 97.9% (A) and 100.0% (B) of comparisons with 15-bin gaps of at least 0.02, against 78.1% and 77.4% for gaps below 0.01.

From $L = 50$ to $L = 3$, the learned models lost 0.038–0.042 AUC in A and 0.020–0.031 in B, and Markov-Queue 0.003 and 0.012, whereas no model lost more than 0.005 in C, where recent periods carry most of the signal. Calibration diverged by architecture (Fig. 1). Static-Logit and Attn-Scaler became overconfident: in A, their ECE rose to 0.099 and 0.092 (Static-Logit: +0.072 [0.063, 0.080]), periods they scored below 0.15 overloaded 29% of the time, and those scored above 0.85 only 72%. GRU-Scaler became conservative: its logit spread shrank by 38%, and its mean prediction exceeded the observed rate (CITL 0.017 [0.008, 0.027] in A and 0.029 [0.020, 0.038] in B). Markov-Queue stayed calibrated, and the shifts in C were smaller.

Two controls locate the overconfidence. Retraining on 3-period windows restored calibration (Static-Logit 0.031, 0.036, and 0.031 in A, B, and C; matched GRU-Scaler and Attn-Scaler 0.018–0.034), so the error came from applying a long-window mapping to short-window summaries. Static-Logit restricted to Markov-Queue's two channels was as overconfident as the full model (0.103 in A, 0.074 in B), whereas Markov-Queue with all six channels stayed calibrated (0.027, 0.034); the smaller input set therefore does not explain the filter's robustness, which is consistent with its structure, Bayesian updating from a stationary prior. A warm start reduced GRU-Scaler's mean bias (CITL 0.007 [−0.002, 0.017] in A) but left its ECE unchanged (0.041 in A, 0.068 in B), so the zero initial state contributes to the level shift but not to the compressed spread.

TABLE I. DISCRIMINATION (AUC) AND CALIBRATION (ECE) WITH LONG ($L = 50$) AND SHORT ($L = 3$) HISTORIES

Model	Synthetic-LLM-A				Synthetic-LLM-B				Synthetic-LLM-C			
	AUC		ECE		AUC		ECE		AUC		ECE	
	$L = 50$	$L = 3$	$L = 50$	$L = 3$	$L = 50$	$L = 3$	$L = 50$	$L = 3$	$L = 50$	$L = 3$	$L = 50$	$L = 3$
Baseline	0.559	0.559	0.024	0.024	0.574	0.574	0.028	0.028	0.690	0.690	0.050	0.050
Markov-Queue	0.680	0.677	0.025	0.025	0.706	0.695	0.031	0.035	0.894	0.894	0.033	0.033
Markov-Queue (6-ch)	0.666	0.666	0.027	0.027	0.703	0.698	0.033	0.034	0.887	0.887	0.031	0.031
Static-Logit	0.721	0.679	0.027	0.099	0.725	0.693	0.024	0.073	0.926	0.922	0.032	0.045
Static-Logit (2-ch)	0.721	0.678	0.025	0.103	0.724	0.693	0.027	0.074	0.923	0.915	0.028	0.048
Static-Logit (matched)	—	0.684	—	0.031	—	0.712	—	0.036	—	0.925	—	0.031
GRU-Scaler	0.717	0.678	0.031	0.042	0.723	0.704	0.025	0.067	0.934	0.929	0.019	0.025
GRU-Scaler (warm start)	—	0.676	—	0.041	—	0.701	—	0.068	—	0.929	—	0.023
GRU-Scaler (matched)	—	0.686	—	0.026	—	0.709	—	0.026	—	0.933	—	0.018
Attn-Scaler	0.719	0.681	0.026	0.092	0.721	0.700	0.024	0.051	0.932	0.929	0.018	0.022
Attn-Scaler (matched)	—	0.682	—	0.034	—	0.706	—	0.028	—	0.931	—	0.021

Means over five replicates (3,000 test events each); between-seed standard deviations were at most 0.040 (AUC) and 0.020 (ECE). Models without a suffix were trained on clean $L = 50$ windows (test-only shift); matched models were retrained on $L = 3$ windows; 2-ch, 6-ch, and warm start are the control predictors of Section III-C; —: identical to the test-only model by construction. ECE uses 15 equal-width bins. Oracle AUC: 0.841 (A), 0.756 (B).

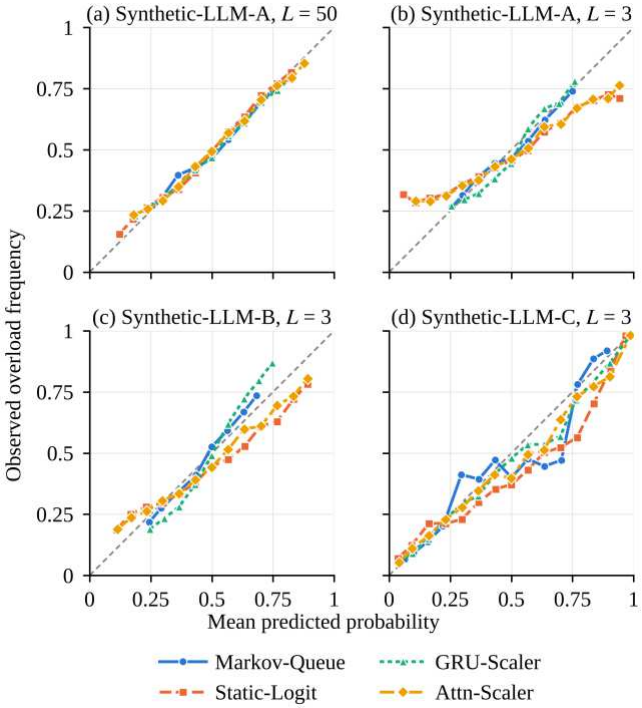


Fig. 1. Reliability diagrams of the test-only models before recalibration: (a) Synthetic-LLM-A at $L = 50$, (b) A at $L = 3$, (c) Synthetic-LLM-B at $L = 3$, and (d) Synthetic-LLM-C at $L = 3$. Each curve pools the five replicates (15,000 test events) in 15 equal-width bins with at least 30 events; the dashed diagonal marks perfect calibration.

Under the idealized proxy, the overconfident models skipped needed scale-outs: in A, UOR rose from 4.1% to 8.9% (Static-Logit) and from 3.6% to 8.1% (Attn-Scaler), raising decision costs to 0.591 and 0.584, above the always-on policy (0.532), whereas the conservative GRU-Scaler scaled out in 99.7% of periods (OPR 52.9%). In C, truncation to $L = 3$ moved realized violation rates by at most 0.3 percentage points at any startup delay.

B. Anomalies, Marker Errors, and Confident Errors (RQ2)

At $\eta = 0.20$, Markov-Queue lost 0.022 AUC in A and its six-channel version 0.021, against at most 0.012 for the learned models and 0.014 for two-channel Static-Logit, consistent with the filter's treatment of the marker as a direct emission of the latent state. The learned models' ECE rose by at most 0.007 in A, but Static-Logit's rose from 0.024 to 0.052 in B (CITL 0.026 [0.018, 0.035]), as expected when errors pull a window's marker frequency toward one half. Raising the floor g from 0.10 to 0.30 lowered the oracle AUC from 0.896 to 0.801 in A and from 0.805 to 0.715 in B; clean-window ECE stayed at or below 0.037, and the short-window ECE of Static-Logit and Attn-Scaler stayed inflated (0.083–0.102 in A, 0.042–0.092 in B). Raising the straggler probability of B from 0 to 0.30 raised the base rate from 0.383 to 0.441 and left the short-window inflation in place (0.051–0.089).

High-confidence predictions were nearly absent on clean long windows in A and B (at most 36 of 15,000). At $L = 3$, Static-Logit issued 1,041 in A (6.9% coverage), of which 290 (27.9% [25.0, 30.8]) were wrong, and Attn-Scaler 702 (208 wrong); in B, they issued 451 and 235 (86 and 39 wrong). Marker errors produced at most 12. In C, 4.9% of Static-Logit's high-confidence predictions were wrong on clean windows and 6.4% at $L = 3$.

C. Calibration Transfer and Recalibration (RQ3)

Temperatures fitted on clean long windows were close to one (0.94–1.06); under the evaluated short-history and marker-corruption shifts ($L = 3$ and $\eta = 0.20$), direct transfer changed ECE by at most 0.005 in A and B, and transferred Platt maps by at most 0.007 (Table II). Target-regime temperatures at $L = 3$ ranged from 0.66 to 1.86, above one for the overconfident models and below one for the conservative one, and changed ECE by −0.068 [−0.076, −0.058] (Static-Logit, A), −0.058 [−0.067, −0.048] (Attn-Scaler, A), and −0.036 [−0.043, −0.029] (GRU-Scaler, B). None of Static-Logit's 290 wrong high-confidence predictions in A or 86 in B kept a confidence of at least 0.90: recalibration removed these errors by lowering confidence and corrected none of them. Direct transfer left 274 of them in A and raised the count in B to 138.

A temperature has no intercept. In B at $L = 3$, temperature-scaled Static-Logit and Attn-Scaler kept ECE at 0.049 and 0.041 because softening pulled predictions toward 0.5 while the base rate was 0.41 (Static-Logit CITL 0.026 before and 0.045 after); Platt scaling reached 0.027 and 0.026 (CITL −0.003 and

−0.001), against 0.036 and 0.028 for the same models retrained on 3-period windows.

Recalibration changed decisions, but better calibration did not guarantee lower costs. Target temperatures cut the UOR of the two overconfident models by 59.8–82.8% across A and B; in A, from 8.9% to 1.8% (Static-Logit) and from 8.1% to 1.4% (Attn-Scaler), while OPR rose from 32.6% to 48.8% and from 34.1% to 50.2%, lowering decision costs from 0.591 to 0.543 and from 0.584 to 0.544, still above always-on (0.532); in B, the recalibrated costs (0.567–0.576) fell below always-on (0.588). Sharpening the conservative GRU-Scaler raised its cost in A from 0.531 to 0.541.

TABLE II. POST-HOC RECALIBRATION AT $L = 3$ IN SYNTHETIC-LLM-A AND -B

Model	T_{tgt}	ECE				Cost	
		Raw	TS dir.	TS tgt.	Platt tgt.	Raw	TS tgt.
Synthetic-LLM-A							
Markov-Queue	1.04	0.025	0.025	0.024	0.027	0.533	0.533
Static-Logit	1.86	0.099	0.098	0.031	0.032	0.591	0.543
GRU-Scaler	0.80	0.042	0.043	0.036	0.028	0.531	0.541
Attn-Scaler	1.76	0.092	0.086	0.034	0.029	0.584	0.544
Synthetic-LLM-B							
Markov-Queue	0.90	0.035	0.034	0.031	0.030	0.574	0.572
Static-Logit	1.56	0.073	0.078	0.049	0.027	0.586	0.570
GRU-Scaler	0.66	0.067	0.070	0.031	0.033	0.582	0.567
Attn-Scaler	1.37	0.051	0.051	0.041	0.026	0.576	0.576

Test-only models at $L = 3$; means over five replicates. TS: temperature scaling; dir.: fitted on clean $L = 50$ calibration windows and transferred; tgt.: fitted on $L = 3$ calibration windows; Platt tgt.: target-regime Platt scaling. Cost: idealized decision cost OPR + 3UOR at $\tau = 0.25$, in units of c_{FP}; always-on cost 0.532 (A) and 0.588 (B).

D. From Probability Quality to Scale-Out Benefit

Table III compares idealized and request-level outcomes in C. For the learned models at $L = 50$, the proxy counted 4.9–5.5% of periods as unmitigated, but the realized violation rate at $d = 20$ s was 37.8–37.9%, against 39.2% without scaling and 35.9% when scaling out exactly in violating periods.

Scaling out before every period averted 17.4%, 8.5%, and 3.4% of the 5,887 violations at $d = 0$, 20, and 40 s but induced 163, 129, and 96 violations in periods that had met their SLOs, so violations fell by a net 14.6%, 6.3%, and 1.7%. At $d = 20$ s, 95.3% of the induced violations occurred when the new node was a straggler (15.1% of scale-out nodes), and averted violations were concentrated in onset periods (32.3% averted) rather than in continuing ones (3.5%). At $d = 20$ s, modeled energy rose from 11.6 Wh per period without scaling to 20.8 Wh under always-on scale-out; with either history length, the learned models used 24.0–25.3% less than always-on scale-out, and 54.0–57.0% less for added replicas, at realized violation rates 1.0–1.2 percentage points higher. Because short histories changed calibration little in C (ECE at most 0.045 at $L = 3$; Table I), recalibration moved scale-out shares at $L = 3$ by at most 1.5 percentage points (non-baseline predictors) and realized violation rates by at most 0.7 percentage points in any condition or startup delay.

TABLE III. IDEALIZED AND REQUEST-LEVEL SCALE-OUT OUTCOMES IN SYNTHETIC-LLM-C

Policy	Scale-out (%)	UOR (%)	RVR (%)			Energy (Wh)
			$d = 0$ s	20 s	40 s	
Reference policies						
Never scale out	0.0	39.2	39.2	39.2	39.2	11.6
Always scale out	100.0	0.0	33.5	36.8	38.6	20.8
Oracle (iff $y = 1$)	39.2	0.0	32.4	35.9	37.9	15.4
Predictors, L = 50						
Baseline	86.5	2.9	34.3	37.2	38.6	19.6
Markov-Queue	37.8	7.4	37.1	38.4	39.0	15.2
Static-Logit	44.5	4.9	35.7	37.8	38.7	15.8
GRU-Scaler	41.6	5.4	35.8	37.8	38.7	15.6
Attn-Scaler	41.6	5.5	35.9	37.9	38.8	15.6
Predictors, L = 3						
Static-Logit, raw	42.7	5.7	36.1	37.9	38.8	15.7
Static-Logit, TS tgt.	44.2	5.3	35.8	37.8	38.8	15.8
Attn-Scaler, raw	42.7	5.3	35.8	37.8	38.8	15.7
Attn-Scaler, TS tgt.	43.9	5.1	35.7	37.8	38.8	15.8

15,000 target periods (five replicates); raw predictions at $\tau = 0.25$ unless marked TS tgt. (target-regime temperature). Scale-out: share of periods with a scale-out; UOR: idealized unmitigated overload rate; RVR: realized SLO-violation rate from single-period request-level re-simulation with one extra replica ready after startup delay d; Energy: modeled Wh per target period and instance at $d = 20$ s. Oracle: scale out exactly in violating periods.

V. DISCUSSION AND LIMITATIONS

Interpretation. Because the idealized proxy assumes that every scale-out absorbs the period's overload, it overstated both the value of prediction and the cost of miscalibration. Window summaries are noisier over three periods, attention is normalized over fewer positions, and a recurrent state stays near its initialization; these untested mechanisms may explain why mappings fitted on long windows mis-scale short ones. In C, gains shrank as the startup delay grew and came mostly from violations that had not yet begun, consistent with replicas arriving after backlog or a slow node had already fixed the period's tail latency; screening added replicas for slow nodes, in line with anomaly-aware scheduling [11], could reduce induced violations but was not tested. Operationally, reliability should be reported per telemetry regime, automation that acts on high confidence should be gated on regime-matched recalibration or retraining, and calibration should be evaluated with request-level outcomes. Temperature and Platt scaling need only labeled calibration windows from the target regime and leave the predictor and serving engine unchanged. The 0.25 threshold holds only for the assumed cost ratio.

Limitations. All data are synthetic and all outcomes are simulated. A and B are abstract, and C uses illustrative engine and power parameters that were not fitted to hardware or production traces, so its violation rates and energy values are model outputs; energy is a modeled estimate, not a power measurement. Scale-outs were evaluated one period at a time, without persistent capacity, scale-in, cooldown, or a closed control loop, so long-term deployment benefits are not established. Startup delays of at most 40 s are short relative to reported model-loading times [8], truncation does not model cold-start dynamics, corruption was limited to binary marker errors, hyperparameters were tuned only on clean long windows, and five replicates limit the precision of between-seed variation. The findings apply within the generators, parameters, and evaluation conditions studied.

VI. Conclusion

In this simulation study, short histories were the main threat to calibration (RQ1): models trained on long histories became overconfident or conservative, whereas a Markov filter stayed calibrated and matched retraining removed most of the error. Marker errors hurt the marker-dependent filter most, and neither noise floors nor stragglers removed the short-history miscalibration (RQ2). Target-regime recalibration cut ECE by up to 68.7% and the idealized UOR of the overconfident models by 59.8–82.8%, although idealized costs in A stayed above always-on provisioning, whereas transferred maps did little (RQ3). With a 20-s startup delay in the request-level simulator, the learned predictors used 24.0–25.3% less modeled energy than always-on scale-out, with a 1.0–1.2 percentage-point increase in the SLO-violation rate. Overload predictors should therefore be recalibrated per telemetry regime and checked at the request level before their probabilities gate provisioning.

Disclosure of AI Assistance

During the preparation of this work, the author used Claude (Anthropic) as an assistive tool for English language editing, manuscript formatting, and code debugging and verification. The experimental design, mathematical formulations, statistical analyses, and scientific conclusions are the original work of the author. The author reviewed and verified all text and code and takes full responsibility for the content of the publication.

References

[1] G.-I. Yu, J. S. Jeong, G.-W. Kim, S. Kim, and B.-G. Chun, "Orca: A distributed serving system for Transformer-based generative models," in *Proc. 16th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2022, pp. 521–538.

[2] W. Kwon *et al.*, "Efficient memory management for large language model serving with PagedAttention," in *Proc. 29th ACM Symp. Operating Syst. Principles (SOSP)*, 2023, pp. 611–626, doi: 10.1145/3600006.3613165.

[3] A. Agrawal *et al.*, "Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve," in *Proc. 18th USENIX Symp. Operating Syst. Design Implement. (OSDI)*, 2024, pp. 117–134.

[4] P. Patel *et al.*, "Splitwise: Efficient generative LLM inference using phase splitting," in *Proc. 51st ACM/IEEE Annu. Int. Symp. Comput. Archit. (ISCA)*, 2024, pp. 118–132, doi: 10.1109/ISCA59077.2024.00019.

[5] J. Dean and L. A. Barroso, "The tail at scale," *Commun. ACM*, vol. 56, no. 2, pp. 74–80, Feb. 2013, doi: 10.1145/2408776.2408794.

[6] A. Shehabi *et al.*, "2024 United States data center energy usage report," Lawrence Berkeley Nat. Lab., Berkeley, CA, USA, Rep. LBNL-2001637, 2024, doi: 10.71468/P1WC7Q.

[7] K. Rzadca *et al.*, "Autopilot: Workload autoscaling at Google," in *Proc. 15th Eur. Conf. Comput. Syst. (EuroSys)*, 2020, Art. no. 16, doi: 10.1145/3342195.3387524.

[8] S. Jaiswal *et al.*, "SageServe: Optimizing LLM serving on cloud data centers with forecast aware auto-scaling," *Proc. ACM Meas. Anal. Comput. Syst.*, vol. 9, no. 3, Art. no. 61, Dec. 2025, doi: 10.1145/3771576.

[9] Z. Gu, R. Chang, C. Li, Q. He, Z. Zhao, and M. Tao, "Uncertainty-aware hierarchical graph transformers for adaptive resource optimization in service meshes," in *Proc. 2026 Int. Conf. Commun. Netw. Mach. Learn. (CNML)*, pp. 1351–1354, doi: 10.1109/CNML68938.2026.11452354.

[10] P. Huang *et al.*, "Gray failure: The Achilles' heel of cloud-scale systems," in *Proc. 16th Workshop Hot Topics Operating Syst. (HotOS)*, 2017, pp. 150–155, doi: 10.1145/3102980.3103005.

[11] M. Tao, "Anomaly-aware resource scheduling in LLM inference & serving," in *Proc. 5th Int. Conf. Cyber Secur., Artif. Intell. Digit. Econ.*, 2026, pp. 376–382, doi: 10.1145/3819836.3819894.

[12] National Institute of Standards and Technology, "Artificial intelligence risk management framework (AI RMF 1.0)," Gaithersburg, MD, USA, Rep. NIST AI 100-1, Jan. 2023, doi: 10.6028/NIST.AI.100-1.

[13] T. Zhang, Y. Chen, Y. Zhao, S. Huang, C. Zhang, and X. Wang, "MAPS: Memory-aware predictive scheduling framework for large language model serving," in *Proc. 43rd Int. Conf. Mach. Learn. (ICML)*, 2026.

[14] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 1321–1330.

[15] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA, USA: MIT Press, 1999, pp. 61–74.

[16] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Proc. 22nd Int. Conf. Mach. Learn. (ICML)*, 2005, pp. 625–632, doi: 10.1145/1102351.1102430.

[17] Y. Ovadia *et al.*, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 13991–14002.

[18] J. Nixon, M. W. Dusenberry, L. Zhang, G. Jerfel, and D. Tran, "Measuring calibration in deep learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, 2019, pp. 38–41.

[19] A. Kumar, P. Liang, and T. Ma, "Verified uncertainty calibration," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 32, 2019, pp. 3792–3803.

[20] K. Cho *et al.*, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.

[21] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 30, 2017, pp. 5998–6008.

[22] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, 2015.

[23] B. Efron, "Bootstrap methods: Another look at the jackknife," *Ann. Statist.*, vol. 7, no. 1, pp. 1–26, 1979, doi: 10.1214/aos/1176344552.