International Journal of Advanced Engineering and Technology Research
ISSN: 3106-8677 (Print), ISSN: 3106-8685 (Online) | Vol. 3, No. 2, 2026

HARP: An Adaptive Hybrid Autoscaling Framework with Online Policy Arbitration for SLO-Aware Cloud-Native ML Inference

Yuchen Luo, Zihan Peng, Rui Zhang *

South China University of Technology, China

Abstract: Cloud-native machine learning inference services must hold tail-latency service level objectives (SLOs) while paying for as few replicas as possible. Two families of autoscalers compete for this job: reactive threshold controllers such as the Kubernetes horizontal pod autoscaler, and predictive controllers that provision ahead of demand. Neither family dominates. Which one wins is decided by operating conditions—replica activation delay, the price an operator places on a replica relative to an SLO violation, and the statistical regime of the arriving workload—that are unknown at design time and change during a deployment. We present HARP, an adaptive hybrid framework that treats the choice of control philosophy as an online decision rather than a design-time commitment. HARP maintains a pool of six heterogeneous demand hypotheses, scores every hypothesis at every epoch with a counterfactual shadow loss computed from telemetry a production deployment already collects, and arbitrates among them with a fixed-share Hedge rule that provably tracks the best hypothesis under regime change. Two further mechanisms make the arbitration actionable: an adaptive conformal margin whose coverage target is driven by an SLO error budget, and a risk-averse aggregation that maps the arbiter's belief to a replica count. On three real production traces (AWS ELB request counts, NYC taxi demand and tweet volume) driven by a measured CPU inference profile, HARP cuts SLO violations by 26.8–92.4% against reactive and forecasting baselines. Across 27 operating conditions its worst-case cost regret against the best fixed policy of each condition is 4.4%, versus 42.5% for a strong confidence-aware predictive controller and over 300% for threshold controllers.

Keywords: Autoscaling, machine learning inference, service level objectives, online learning, conformal prediction, Kubernetes, cloud-native systems.

1. Introduction

Machine learning inference has become a first-class cloud workload. Unlike batch training, inference is interactive: requests arrive continuously, and a service is judged by whether it keeps a tail-latency service level objective (SLO) while consuming as little capacity as possible. Because tail latency degrades sharply once utilisation approaches saturation [1], and because inference replicas are expensive, the elasticity controller that decides how many replicas to run is one of the most consequential components of an inference platform.

Two families of controllers are used in practice. Reactive controllers, exemplified by the Kubernetes horizontal pod autoscaler and by event-driven autoscalers, compare an observed signal against a target and correct the replica count afterwards. They are simple, robust and assumption-free, but they can only respond after demand has already moved, so they are structurally penalised by the replica activation delay. Predictive controllers instead forecast near-future demand and provision ahead of it, which removes the activation lag when the forecast is good and wastes capacity—or, worse, under-provisions confidently—when it is not.

The literature has not settled which family should be preferred, and there is direct evidence that the answer depends on the workload. CAPA, a prediction-driven autoscaler built specifically for inference services, couples a rolling ridge forecast over the startup horizon with an online one-sided residual quantile and an SLO-debt feedback signal; it lowers SLO violations by 17.8% on bursty traffic and by 11.4% under regime shifts relative to a reactive baseline, yet reports a 14.6% increase on smooth periodic traffic, where the short-term rate is already smooth and the reactive estimate is close to optimal [26]. That study is notably transparent in publishing the regression as a boundary condition rather than suppressing it, and it names an online policy gate as the natural remedy — which it leaves open as future work. The present paper takes up that question.

Our starting point is that the winning family is determined by operating conditions rather than by algorithmic merit. Three conditions matter. First, the replica activation delay: when a replica becomes ready in three seconds, anticipation buys almost nothing; when it takes thirty seconds, anticipation is the only way to avoid a deep backlog. Second, the operator's price of a replica relative to an SLO violation: a latency-critical service and a cost-sensitive batch-like service want opposite trade-offs from the same controller. Third, the arrival regime, which in production shifts within a single day. None of these is known when a controller is configured, and all of them move afterwards.

We therefore ask whether a controller can decide online, at every decision epoch, which control philosophy to trust, with a regret guarantee, using only telemetry that a real deployment already exports. We answer affirmatively with HARP (Hybrid Adaptive Regime-aware Provisioner). HARP does not select a controller; it maintains a pool of demand hypotheses, keeps a belief over them, and converts that belief into a replica count in a way that respects the asymmetry between over- and under-provisioning.

This paper makes the following contributions.

1) A counterfactual shadow evaluator that scores every candidate policy at every epoch against the demand that

actually materialised, using an Erlang-C model parameterised by measured service times. Because it consumes only arrival counts, backlog and latency histograms, it is computable in a production control loop and requires no counterfactual execution.

2) A fixed-share Hedge arbiter over six heterogeneous demand hypotheses, with volatility-adaptive mixing and request-volume-weighted losses. The arbiter inherits a tracking-regret bound against any sequence of hypothesis switches, which is the right guarantee for workloads whose regime changes.

3) An SLO-budget-driven adaptive conformal margin: the miscoverage level of the predictive upper bound is updated online in the style of adaptive conformal inference [30], but its target is modulated by an SLO error-budget debt signal rather than fixed a priori, so protection is bought only when the budget is actually being consumed.

4) A risk-averse aggregation rule that maps the arbiter's belief to an action through a weighted upper quantile of the hypothesis-induced replica counts, with the quantile level tightened by the same debt signal. We show empirically that aggregating in action space is essential: the seemingly more principled alternative of minimising expected cost in scenario space is 158% worse on average because a single-epoch risk model cannot see that a capacity shortfall persists for an entire activation delay.

5) An evaluation on three real production traces with a measured inference profile, 12 disjoint windows per workload, paired non-parametric tests with Holm correction and bootstrap confidence intervals, a 27-point operating-condition grid, a ten-way component ablation and a knob sensitivity study. All code, traces and result tables are released.

Across the default operating point HARP reduces SLO violations by 26.8–92.4% against reactive and forecasting baselines (9 of 9 comparisons, all significant after Holm correction over 12 tests). Its distinguishing property, however, is robustness: over 27 operating conditions its mean cost regret against the per-condition best fixed policy is 0.55% and its worst case is 4.44%, whereas the strongest fixed policy we tested degrades to 42.49% and threshold controllers exceed 300%.

2. Background and Related Work

2.1. Reactive and Predictive Cloud Autoscaling

Classical elasticity controllers keep a utilisation or queue signal at a target. AutoScale showed that maintaining a deliberate capacity headroom, rather than tracking demand exactly, is what makes threshold control robust to bursts and to server heterogeneity [14]; CloudScale added online resource prediction and conflict handling for multi-tenant deployments [15]. Surveys map the design space and note that no single policy dominates across workloads [13], [17]. Predictive controllers have since become mainstream in industry: Autopilot learns per-job resource limits from history [18], Resource Central predicts virtual-machine behaviour to guide placement [19], MagicScaler couples a probabilistic forecaster to a scaling decision under uncertainty [22], and burst-aware and reinforcement-learning controllers target specifically the bursty regime [21], [23], [24].

Hybrid proactive–reactive control is not new. Ali-Eldin et al. combined a proactive and a reactive component in an adaptive elasticity controller [16], and later work explored self-improving hybrid elasticity. What these designs share is a fixed combination rule: the blend is a design-time choice, tuned offline, and does not come with a guarantee when the environment changes. HARP differs in that the combination itself is the object of online learning and carries a tracking-regret bound.

2.2. SLO-aware ML Inference Serving

Inference-specific systems attack the same objective from the serving side. Clipper and TensorFlow-Serving introduced batching, caching and model-selection layers [2], [3]; Clockwork exploits the predictability of DNN execution to make latency controllable from the bottom up [4]; MArk, INFaaS, Cocktail and Shepherd optimise instance type, model variant and request batching under SLO constraints [5], [6], [8], [9]; InferLine provisions prediction pipelines end-to-end [7]; AlpaServe, Orca and vLLM address multiplexing and memory for large generative models [10-12]. Microservice resource managers such as FIRM learn QoS-aware allocations from traces [25]. Closest to our setting is CAPA [26], which sizes replicas from a rolling ridge forecast at the startup horizon, an online one-sided residual-quantile margin, a queue-drain term and an SLO-debt signal that lowers the target utilisation when recent requests approach the objective, with every service-rate and cold-start parameter measured rather than assumed. Its ablation is the clearest evidence we know of that uncertainty, not point accuracy, is what a serving autoscaler must act on: removing the one-sided margin costs 36% of the achieved violation reduction, far more than removing the debt loop. We adopt its measurement-driven evaluation discipline and reimplement its controller, which we denote CAPA-Style, as our strongest baseline.

2.3. Prediction with Expert Advice and Conformal Calibration

Combining several predictors with multiplicative weights has a long history: the weighted majority algorithm [27] and Hedge [28] guarantee regret against the best fixed expert, while the fixed-share variant of Herbster and Warmuth [29] guarantees regret against the best sequence of experts with a bounded number of switches, which is exactly the guarantee a regime-changing workload requires [31]. Conformal prediction supplies distribution-free predictive intervals [32], and adaptive conformal inference maintains validity under distribution shift by updating the miscoverage level online [30]. HARP is, to our knowledge, the first autoscaler to use fixed-share arbitration over control policies together with an SLO-budget-driven conformal margin.

3. System Model and Problem Statement

We consider a replicated stateless inference service behind a load balancer. Time is discrete at one-second resolution; control decisions are taken every Δ seconds, indexed by k. At epoch k the controller observes the arrival counts of the last interval, the queue backlog b_k, the number of active replicas, and an estimate of the fraction of requests that missed the objective. It emits a target replica count a_k. New replicas become ready after an activation delay d_s; removals are rate limited and observe a cooldown.

Each replica serves requests at rate $\mu = \eta \ell^{-1}$, where ℓ is the measured mean service time and $\eta < 1$ accounts for non-inference overhead. Backlog evolves as

$$b_t = \max(0,\ b_{t-1} + A_t - c_t\ \mu)$$

Where A_t is the number of arrivals in second t and c_t the number of active replicas. A request's response time is the deterministic drain delay b_{t−1}/(c_t μ), plus its service time drawn from the measured empirical distribution, plus an M/M/c waiting time; a violation occurs when the sum exceeds the objective L. Writing v_t for the resulting per-second violation probability, the request-weighted violation rate over a horizon T is V = Σ_t A_t v_t / Σ_t A_t.

The operator's cost combines quality and resources,

$$J = w_v\, V + w_r\, \bar{n} / n_{max}$$

With $\bar{n}$ the mean replica count and w_r/w_v the price of a replica relative to an SLO violation. Given a set of candidate policies Π and a sequence of operating conditions, the quantity we care about is the regret of a controller π against the best policy of each condition,

$$R(\pi) = J(\pi) - \min_{\{\pi' \in \Pi\}} J(\pi')$$

Reported as a percentage of the latter. A controller with small worst-case regret is one an operator can deploy without knowing in advance which condition it will meet. Internally, HARP additionally aims for small tracking regret against the best sequence of demand hypotheses within a single run.

4. The HARP Framework

4.1. Overview

Fig. 1 shows the control loop. Telemetry feeds a pool of demand hypotheses and an adaptive conformal margin. Every hypothesis induces a replica proposal. A fixed-share Hedge arbiter maintains a belief over hypotheses from counterfactual shadow losses, and a risk-averse aggregator turns that belief into the actuated replica count. A single scalar—the SLO error-budget debt—couples the quality feedback to both the conformal margin and the aggregator's risk level.

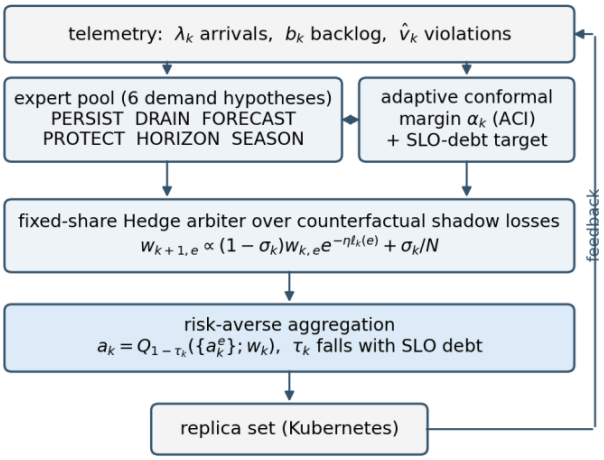

Figure 1. HARP control loop. All inputs are quantities a production deployment already exports; the arbiter and the aggregator add O(N) work per epoch for N = 6 hypotheses.

4.2. The Hypothesis Pool

HARP maintains six demand hypotheses, each of which is a compact statement about what the next activation window will look like, together with the utilisation target that a controller believing that statement would use. PERSIST assumes demand persists at the last observed rate (the assumption of a utilisation-threshold autoscaler). DRAIN adds the requirement that the current backlog be absorbed within a drain window T_d = max(5, 2d_s) seconds (the assumption of a queue-driven autoscaler). FORECAST uses the point prediction of a rolling ridge model at the activation horizon. PROTECT adds a one-sided conformal margin to that prediction and lowers the utilisation target in proportion to the margin width and to the SLO debt, reproducing the confidence-aware design discussed in Section II-B. HORIZON is protective over the whole activation window rather than at a single horizon, taking the worst case of the protected bound over horizons 1…h+1, where h = ⌈d_s/Δ⌉. SEASON is a seasonal-naive estimate whose period is detected online from the autocorrelation of the rate history.

Predictions come from a single ridge model fitted jointly for all horizons on a rolling window of 180 epochs with 12 features (four lags, two moving averages, a trend term, four harmonics and a bias), refitted every six epochs. The pool is deliberately heterogeneous rather than an ensemble of forecasters: its members disagree about the structure of the decision problem, not merely about a number.

4.3. Adaptive Conformal Margin Under an SLO Budget

Let r_j denote the one-sided residuals of the horizon-j predictor on the rolling window and Q_{1−α}(r_j) their empirical (1−α)-quantile. A fixed α (0.10 is the usual choice) is a design-time bet on how much protection the service needs. HARP instead treats α as a controlled variable. Following adaptive conformal inference [30] it updates

$$\alpha_{k+1} = \mathrm{clip}(\alpha_k + \gamma\,(\alpha^*_k - \mathrm{err}_k),\ 0.02,\ 0.30)$$

Where err_k is the empirical miscoverage of the bound issued h epochs earlier and γ = 0.02. The novelty is the target: instead of a constant, α_k is driven by an SLO error-budget debt D_k that measures how much of the budget V the service has recently consumed,

$$D_k = 0.75\, D_{k-1} + 0.25\, \mathrm{clip}((\bar{v}_k - V^*)/V^*,\ 0,\ 1),$$

$$\alpha^*_k = \mathrm{clip}(0.10 - \kappa \cdot 0.10 \cdot D_k,\ 0.02,\ 0.25),\ \kappa = 1.2.$$

When the budget is intact the target relaxes and the service stops paying for headroom it does not need; when violations appear the target tightens and the protected bound widens before the operator ever notices. Measured coverage of the resulting bound stayed within 89.6–90.6% across the three workloads, i.e. the loop tracks its target rather than drifting.

4.4. Counterfactual Shadow Evaluation

Only the executed action affects the system, so the losses of the non-executed hypotheses must be imputed. HARP does this with a shadow loss that is delayed by the activation horizon: a proposal made at epoch k is scored at epoch k+h against the demand that actually materialised by then. For a proposal of c replicas,

$$\ell_k(e) = [\, w_v(\hat{v} + p_h \cdot u) + w_r\, c/n_{max} + w_s |c - a_{k-1}|/n_{max}\,] / Z$$

Where $\hat{v}$ is the Erlang-C violation probability at the realised rate and backlog, u = max(0, λ_peak − cμ)/λ_peak is the fraction of the observed peak demand the proposal could not have served, and Z normalises ℓ to [0,1]. The shortfall term is priced by p_h = min(4, 1 + (h−1)/2) because a capacity shortfall persists for roughly h epochs before new replicas can arrive—a dynamic effect that a single-epoch violation estimate cannot express. Losses are finally weighted by the interval's request volume relative to its exponentially

weighted mean, so that quiet intervals, in which every hypothesis looks equally good, do not dilute the arbiter's evidence.

Every ingredient—arrival counts, backlog, measured service-time quantiles—is observable, so the shadow loss can be evaluated in a production control loop without any counterfactual execution.

4.5. Fixed-share Arbitration

Beliefs are updated with Hedge followed by a fixed-share mixing step:

$$w'_{k,e} \propto w_{k,e} \exp(-\eta\, \ell_k(e)),\quad w_{k+1,e} = (1-\sigma_k)\, w'_{k,e} + \sigma_k/N.$$

The mixing rate is volatility adaptive, $\sigma_k = \mathrm{clip}(0.002 + 0.10\cdot\hat{V}_k, 0.002, 0.05)$, with $\hat{V}_k$ an exponentially weighted estimate of the normalised one-step change of the arrival rate: the arbiter forgets faster exactly when the workload is moving. We use $\eta = 8$ and $N = 6$ throughout.

Proposition 1 (tracking regret). Let $\ell_k(e) \in [0,1]$ and let $\zeta = (e_1, \ldots, e_K)$ be any comparator sequence over K epochs with m switches. Fixed-share with parameters (η, σ) satisfies

$$\Sigma_k \langle w_k, \ell_k\rangle - \Sigma_k\, \ell_k(e_k) \le \eta K/8 + [\ln N + m\, \ln((N-1)/\sigma) + (K-1-m)\ln(1/(1-\sigma))\ ] / \eta$$

Which for $\sigma = m/(K-1)$ and $\eta = \Theta(\sqrt{((m\ \ln N + m\ \ln(K/m))/K))}$ is $O(\sqrt{(K(m\ \ln N + m\ \ln(K/m))))}$, i.e. sublinear in K for any $m = o(K)$. The bound follows from the standard Hedge analysis applied to the mixed weight sequence [29], [31]; the term in m is the price of allowing the best hypothesis to change. In our runs the arbiter changed leader between 7.9 and 24.9 times per two-hour run, so $m \ll K = 1440$ and the bound is informative.

4.6. Risk-averse Aggregation

A belief over hypotheses must still become one integer. Averaging the hypothesis-induced replica counts is the obvious choice, but it is symmetric, while the cost is not: one replica too few can cost far more than one replica too many. HARP therefore actuates the weighted upper quantile

$$a_k = \lceil \max(\, Q_{1-\tau_k}(\{a_k^e\};\, w_k),\ \ b_k/(T_d\, \mu))\ \rceil$$

$$\tau_k = \mathrm{clip}(\tau_0 - 0.30\ D_k,\ 0.05,\ \tau_0),\ \tau_0 = 0.20$$

So that the aggregator becomes more conservative through the same debt signal that widens the conformal margin, and never falls below the count needed to drain the current backlog. Section VI-D shows that this choice is worth 2.6% of cost against the mean rule and, more importantly, that the alternative of minimising expected cost over the hypothesis-induced demand scenarios is dramatically worse.

4.7. Cost and Implementability

Per epoch HARP performs one ridge prediction, $N = 6$ proposal evaluations, N shadow-loss evaluations and one quantile selection—$O(N)$ work plus a periodic $O(p^2 n)$ refit with $p = 12$ features. In our Python prototype a full two-hour run of 1440 decisions costs well under a second of controller time, i.e. under one millisecond per decision, which is negligible next to a five-second control interval. No component requires state beyond a rolling window of arrival rates and residuals.

5. Experimental Setup

5.1. Measured Inference Profile

Every service-time and capacity parameter in the evaluation is measured, not assumed. We profile a depthwise-separable convolutional network of MobileNet type [35-38] executing on CPU in a single thread, at batch size one and 160×160 input resolution, using NumPy 2.4.4 (OpenBLAS, 1 thread). After 30 warm-up passes we time 300 forward passes and, separately, 7 process-level cold starts (interpreter start, library import, weight construction and first inference). Table I reports the profile. The simulator draws service times from the 25-point empirical quantile grid of these measurements rather than from a parametric distribution.

Table I. Measured inference worker profile

Quantity	Value
Mean service time	3.75 ms
Median service time	4.15 ms
p95 / p99 service time	4.60 / 4.94 ms
Standard deviation	0.69 ms
Per-replica service rate	266.7 req/s
Effective rate μ ($\eta = 0.92$)	245.3 req/s
Process cold start (median / p95)	64.7 / 78.5 ms
Latency objective L	20 ms (5.3× mean service time)

The measured cold start is process-level and is a lower bound on the time for a replica to become ready: in Kubernetes, scheduling, image pull and readiness probing dominate. We therefore treat the activation delay d_s as a configured platform parameter, use 10 s at the default operating point, and sweep it over {3, 10, 30} s in Section VI-C.

5.2. Real Workloads

We drive the service with three real production traces redistributed by the Numenta Anomaly Benchmark [33], [34]: W1, request counts of a production AWS Elastic Load Balancer at five-minute resolution; W2, New York City taxi passenger pickups at thirty-minute resolution, a strongly diurnal demand signal; and W3, the volume of tweets mentioning a large-capitalisation ticker at five-minute resolution, a heavy-tailed social-media signal. Each trace is sliced into 12 disjoint windows; a window is mapped to a 1 Hz arrival-rate envelope by assigning $\tau = 30$ simulated seconds to each real bucket and linearly interpolating, and is amplitude-scaled so that the window mean is about 420 req/s. Per-second arrivals are then drawn from a Poisson process modulated by that envelope. Only the time and amplitude scaling and the sub-bucket Poisson noise are synthetic; the shape of every envelope is real. Table II and Fig. 2 summarise the resulting regimes, which differ sharply in burstiness.

Table II. Workload characteristics (mean over 12 windows)

Workload	Mean (req/s)	Peak/mean	CV	Disp. index
W1 AWS ELB	419.7	4.48	0.76	245.3
W2 NYC taxi	420.1	1.81	0.46	90.3
W3 tweets	392.7	5.15	0.79	256.8

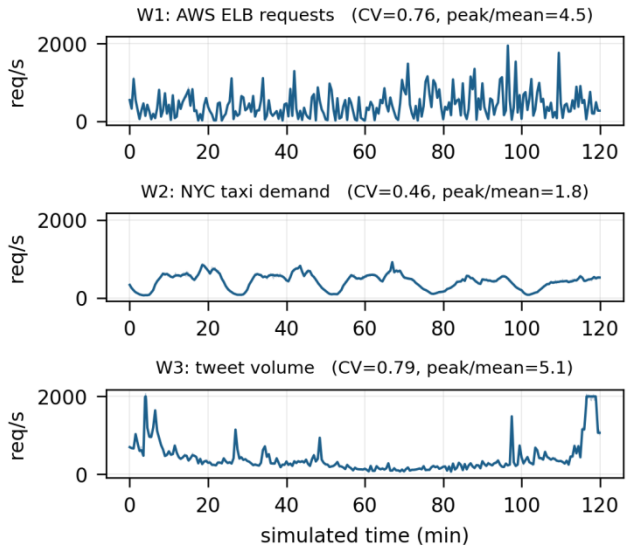

Figure 2. One window of each real workload: the interpolated rate envelope (dark) and the realised per-second Poisson arrivals (light). W2 is smooth and periodic; W1 and W3 are bursty and heavy tailed.

5.3. Simulator and Baselines

All controllers are compared inside one harness, so every comparison is paired on the same arrival realisation. The harness implements the model of Section III, with n_{max} = 20 replicas, Δ = 5 s, a 30 s scale-down cooldown and a 25% per-step removal limit applied identically to every controller. We compare against four baselines. Reactive-HPA targets 65% utilisation on the last interval's rate, the horizontal pod autoscaler design. Queue-KEDA targets 70% utilisation and adds a backlog drain term, the event-driven autoscaler design. DES-Predictive forecasts with Holt double exponential smoothing and adds the same drain term. CAPA-Style is our reimplementation of the prediction-driven controller described in Section II-B, with its published rule reproduced exactly: a 12-feature rolling ridge refitted every six decisions over a 180-window history, a fixed 0.90 one-sided residual quantile with a recent-peak guard, a queue-drain term with drain window max(5 s, 2d_s), and target utilisation clip(0.72 - 0.20z - 0.22D, 0.46, 0.72). It is by a wide margin the strongest baseline.

5.4. Metrics and Statistical Protocol

We report the request-weighted SLO violation rate, the mean replica count, an estimate of the response-time p95, the number of scaling actions and the composite cost J. The default operating point is L = 20 ms, d_s = 10 s, w_v = 1, w_r = 0.35. Each configuration is run on 12 disjoint windows per workload (180 runs in the main comparison). Because the windows are different slices of real traces, variation across them reflects genuine workload variation rather than only re-seeding. Paired comparisons use the one-sided Wilcoxon signed-rank test with Holm correction over the 24 tests of the main table, and effect sizes are reported as bootstrap 95% confidence intervals on the paired relative reduction (10 000 resamples). Hyperparameters (η, τ_0, γ, κ) were fixed once on windows 0–2 and never changed afterwards.

6. Results

6.1. Default Operating Point

Table III and Fig. 3 give the main comparison. On the two bursty workloads the gap between the two families is large: reactive and smoothing controllers sit at 5.7–23.7% violations, while the two uncertainty-aware controllers sit at 1.8–3.0%. On the smooth periodic workload W2 all controllers are close, and the reactive baseline is already better than the smoothing predictor—the regime in which prediction earns little, consistent with the negative result discussed in Section II-B.

Table III. Main comparison at the default operating point (mean over 12 real windows per workload; sd across windows)

Workload / controller	Viol. %	sd	Repl.	p95 (ms)	Actions	Cost J
W1 Reactive-HPA	15.97	2.10	3.78	64.0	329	0.2259
W1 Queue-KEDA	19.36	2.05	3.64	77.4	318	0.2572
W1 DES-Predictive	23.66	1.95	3.99	95.9	330	0.3065
W1 CAPA-Style	2.53	0.56	4.66	14.8	324	0.1068
W1 HARP	1.80	0.47	4.82	12.4	321	0.1024
W2 Reactive-HPA	0.74	0.17	3.10	11.0	91	0.0617
W2 Queue-KEDA	1.21	0.17	2.89	12.9	88	0.0626
W2 DES-Predictive	1.21	0.16	2.89	12.9	62	0.0627
W2 CAPA-Style	0.65	0.15	3.01	11.6	94	0.0593
W2 HARP	0.54	0.16	3.08	11.1	100	0.0594
W3 Reactive-HPA	5.69	2.32	2.99	27.6	157	0.1092
W3 Queue-KEDA	6.80	1.68	2.84	30.8	143	0.1177
W3 DES-Predictive	9.47	1.53	2.91	39.7	155	0.1456
W3 CAPA-Style	2.97	1.32	3.29	18.9	187	0.0873
W3 HARP	2.97	1.39	3.30	19.0	185	0.0874

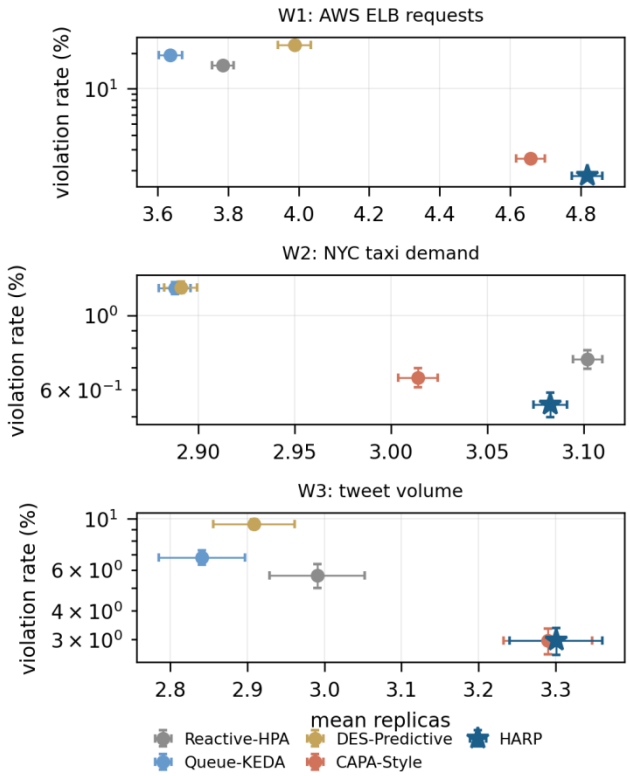

Figure 3. Cost of quality: mean replicas against SLO violation rate (log scale), with standard errors over 12 real windows. HARP (star) is on or below the achievable frontier on all three workloads.

Against the three conventional baselines HARP reduces violations by 26.8–92.4%, all significant at $p < 0.01$ after Holm correction. Against the reimplemented confidence-aware controller the picture is more nuanced and we report it as measured: HARP reduces violations by 28.7% on W1 (95% CI [21.6, 34.8], $p = 0.006$) and by 16.8% on W2 (CI [13.6, 21.1], $p = 0.006$), but the two are statistically indistinguishable on W3 (0.1%, CI [-4.5, 5.1], $p = 1.00$). At a single, favourable operating point a well-tuned predictive controller is hard to beat. The value of arbitration appears when the operating point is not known in advance, which is the subject of Section VI-C.

6.2. What the Arbiter Learns

Fig. 4 shows the belief trajectory of a representative run on each workload. The arbiter is not merely picking one hypothesis: it redistributes mass in a way that tracks the workload. On W1 protective mass (PROTECT and HORIZON) dominates during the volatile first half and cedes ground to FORECAST when demand settles. On W2 the belief is stable and periodic, with HORIZON and PROTECT sharing most of the mass. On W3 the arbiter concentrates almost entirely on HORIZON during the sustained quiet interval and reverts to a mixture within a few epochs of each flash crowd.

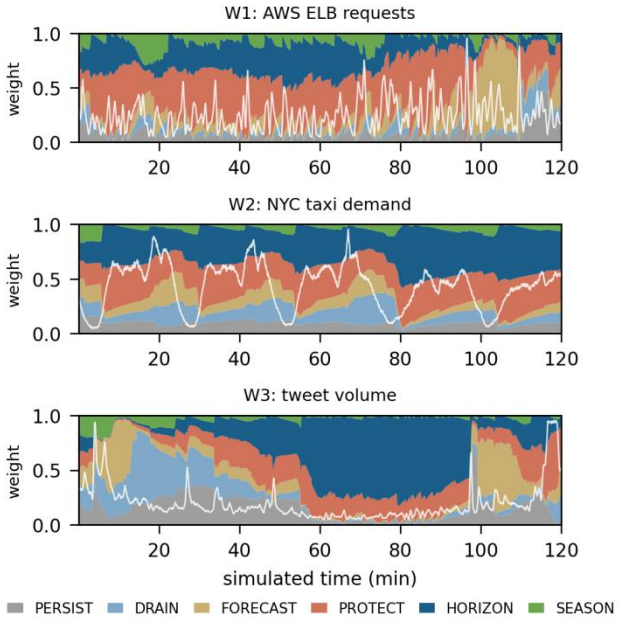

Figure 4. Arbitration weights over one run of each workload (stacked, summing to one); the white line is the arrival rate. Belief mass shifts with the regime rather than converging to a single hypothesis.

Table IV. Mean arbitration weight per hypothesis, adaptive margin level and aggregation risk level (12 windows)

Workload	PERS	DRAI	FORE	PROT	HORI	SEAS	$\bar{\alpha}$	$\bar{\tau}$
W1	0.079	0.058	0.145	0.406	0.236	0.076	0.082	0.192
W2	0.086	0.120	0.127	0.320	0.283	0.064	0.062	0.184
W3	0.170	0.161	0.127	0.249	0.257	0.036	0.078	0.181

Table IV quantifies this. Protective hypotheses hold 64% of the mass on W1 but only 51% on W3, where persistence-type hypotheses earn 33%. The adaptive margin settles near $\alpha \approx 0.06$–0.08, i.e. the SLO-budget loop chose slightly more protection than the fixed 0.90 quantile of the baseline on the bursty traces and slightly less on the periodic one. Fig. 5 shows the effect at the actuator: HARP raises capacity before the ramp that the reactive controller only answers afterwards.

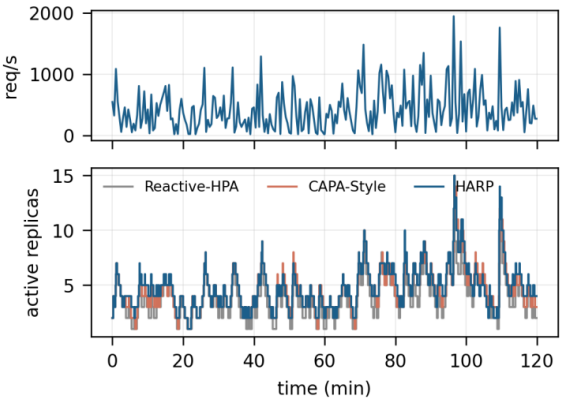

Figure 5. Arrival rate (top) and active replicas (bottom) for one W1 window. The reactive controller reacts after each ramp; HARP anticipates it and also releases capacity when the arbiter loses confidence in the protective hypotheses.

6.3. Robustness Across Operating Conditions

We now vary the two conditions an operator cannot fix at design time: the activation delay $d\_s \in \{3, 10, 30\}$ s and the replica price $w\_r \in \{0.15, 0.35, 0.80\}$. With three workloads this yields 27 conditions, each evaluated on 12 windows. Table V shows the aggregate. Two facts stand out. First, no fixed controller wins everywhere: the confidence-aware controller is best in 9 of 27 conditions and the queue-driven controller in 1, but each has conditions in which it is badly mismatched. Second, HARP is best in 17 conditions and, more importantly, never far from best: its worst-case cost regret is 4.44%, against 42.49% for the strongest fixed policy and above 300% for threshold control.

Table V. Cost regret against the best fixed policy of each of the 27 operating conditions

Controller	Mean %	Median %	Worst %	Wins
Reactive-HPA	52.73	25.09	307.56	0/27
Queue-KEDA	65.68	34.82	337.28	1/27
DES-Predictive	88.77	53.35	408.78	0/27
CAPA-Style	4.69	2.09	42.49	9/27
HARP	0.55	0.00	4.44	17/27

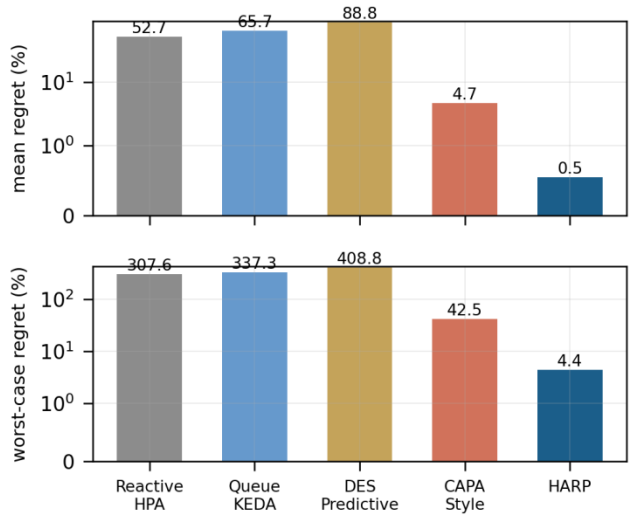

Figure 6. Mean and worst-case cost regret over 27 operating conditions (symmetric log scale). The adaptive controller is the only one whose worst case stays in single digits.

The mechanism is visible in the conditional means. As the activation delay grows from 3 s to 30 s the violation rate of the reactive controller rises from 2.14% to 18.19%, because anticipation becomes indispensable; HARP rises only from 0.72% to 4.98%, and its advantage over the confidence-aware controller widens from 0.09 to 1.42 percentage points, because the HORIZON hypothesis protects the whole activation window rather than a single point in it. In the opposite direction, as the replica price rises to $w\_r = 0.80$ the arbiter shifts mass towards cheaper hypotheses and the two uncertainty-aware controllers converge (0.1822 versus 0.1807), whereas at $w\_r = 0.15$ HARP is clearly ahead (0.0512 versus 0.0590).

6.4. Ablation

Fig. 7 (top) removes one component at a time at the default operating point. Three findings are worth separating.

Arbitration and aggregation matter. Replacing the weighted-quantile aggregation by hard selection of the leading hypothesis (argmax) costs 7.7% on average and 12.2% on W1: a belief is more useful than a decision. Replacing it by the mean action costs 2.6%, confirming that the asymmetry between over- and under-provisioning should be built into the aggregator.

Aggregating in scenario space fails. Choosing the replica count that minimises expected cost over the hypothesis-induced demand scenarios—the textbook decision-theoretic answer—is 158.1% worse on average and 307% worse on W3. The single-epoch risk model cannot represent the fact that a shortfall persists for a whole activation delay, so it systematically under-buys capacity. We report this negative result because it explains why the utilisation headroom carried by the individual policies is not redundant with an explicit risk calculation.

At a fixed operating point, a well-chosen single hypothesis is competitive. Using only HORIZON or only PROTECT is within 0.6–0.4% of the full framework, and disabling the adaptive margin or the adaptive mixing changes cost by less than 0.5%, which is inside the window-to-window spread. Using only PERSIST, in contrast, costs 49.0%. The honest reading is that these components do not buy much when the right hypothesis happens to be known; their value is that HARP does not need to know it, which is precisely what Table V measures.

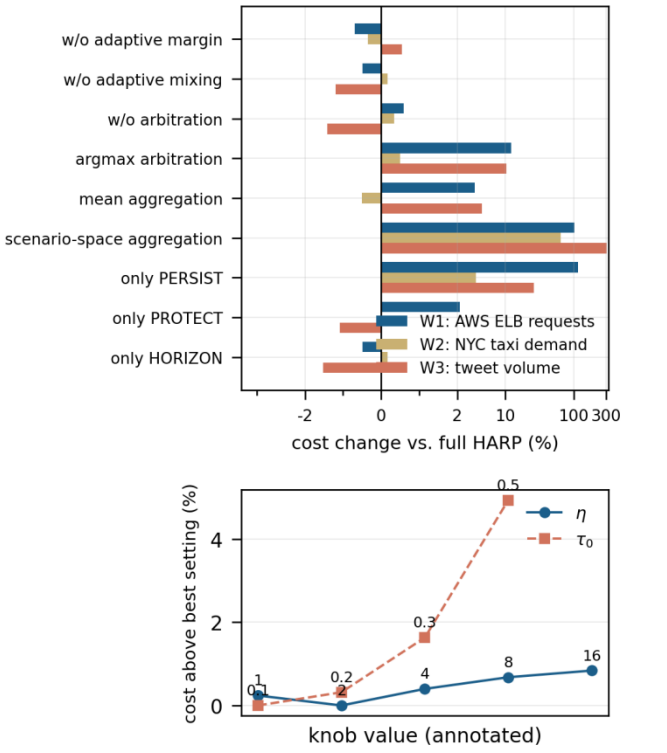

Figure 7. Top: component ablation, cost change relative to full HARP (symmetric log scale; positive is worse). Bottom: sensitivity of cost to the arbiter learning rate η and the aggregation risk level τ_0.

6.5. Sensitivity and Overhead

Fig. 7 (bottom) sweeps the two knobs a practitioner would touch. Cost varies by only 0.8% as the learning rate η ranges over a factor of sixteen (1 to 16), so the arbiter does not need careful tuning. The risk level τ_0 matters more, spanning 4.9% between 0.10 and 0.50, which is expected: τ_0 encodes how much the operator dislikes under-provisioning and is therefore a policy knob rather than a nuisance parameter. Controller overhead is under one millisecond per decision

against a five-second interval, and memory is bounded by a 180-epoch rolling window.

7. Discussion and Threats to Validity

Simulation. Our results come from a discrete-event harness rather than a live cluster. The harness is parameterised by measured service-time quantiles and a measured cold start, and every controller is evaluated on identical arrival realisations, so comparisons are internally valid; absolute violation rates would nonetheless differ on a real cluster, where scheduling jitter, imperfect load balancing and network effects add variance. We expect such variance to favour HARP rather than the baselines, because it increases the value of hedging, but this remains to be shown.

Trace scaling. The public traces are bucketed at five and thirty minutes, whereas an inference autoscaler acts on seconds. We compress time by a factor of ten (W1, W3) and sixty (W2) and add sub-bucket Poisson noise. Compression preserves the shape of demand but makes ramps steeper relative to the activation delay; the activation-delay sweep in Section VI-C is the control for this choice, and the ordering of controllers is stable across it.

Scope of the comparison. We compare replica-count control, holding the serving stack fixed. Orthogonal mechanisms—batching, model-variant selection, hardware heterogeneity [5], [6], [8]—would change absolute numbers and could be folded into the hypothesis pool as additional experts, which is a natural extension: the arbiter is agnostic to what an expert proposes as long as its proposal can be scored by a shadow loss.

Relation to prior findings. Our results corroborate rather than contradict the boundary condition documented in [26]: a predictive margin can be counterproductive when the workload is smooth and periodic. On W2, our own pure forecasting baseline is indeed worse than the reactive one, and the margin-based controller's advantage over reactive control is the smallest of the three workloads. What HARP adds is that the operator no longer has to know this in advance, nor to disable prediction by hand: the arbiter discovers from its own shadow losses that persistence-type hypotheses are adequate in that regime and reallocates belief accordingly, which is precisely the online policy gate that work called for. It is worth noting that this earlier study measured a 2.5 s process cold start for a PyTorch MobileNetV3-Small worker and evaluated at a 3 s activation delay; our activation-delay sweep shows the two families separate much further as that delay grows, so the arbitration question becomes more, not less, important on realistic container platforms.

Negative results we keep. Two of our own mechanisms did not pay off where one might expect: the adaptive conformal margin and the volatility-adaptive mixing are neutral at a fixed operating point, and scenario-space aggregation is actively harmful. We report both, because a framework paper that only reports the components that worked gives the reader no way to tell which parts are load-bearing.

8. Conclusion

Reactive and predictive autoscaling are usually presented as competing answers to the same question. We argued that they are better understood as hypotheses about demand whose relative merit is an empirical, time-varying quantity, and we built a controller that treats the choice as an online learning problem with a tracking-regret guarantee. On three real production traces with a measured inference profile, HARP reduces SLO violations by up to 92% against conventional autoscalers, and—its distinguishing property—keeps worst-case cost regret at 4.4% across 27 operating conditions in which every fixed policy we tested is at some point badly mismatched. We release the harness, the traces and the analysis so that the operating-condition grid can be reused as a robustness benchmark for future autoscalers.

References

[1] Zhang, H. (2024). Graph-aware multi-task learning for early prediction of timing violations and routing congestion in digital IC physical design. World Journal of Information Technology, 2(1), 13–20.

[2] Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic marketing uplift modeling: A symmetry-preserving framework integrating causal forests with deep reinforcement learning for personalized intervention strategies. Symmetry, 17(4), 610.

[3] Zhao, X., Liu, J., Wang, Y., & Wang, J. (2026). CryptoMamba-SSM: Linear complexity state space models for cryptocurrency volatility prediction. IEEE Open Journal of the Computer Society, 7, 226–243.

[4] Wang, J., Liu, J., Zheng, W., & Ge, Y. (2025). Temporal heterogeneous graph contrastive learning for fraud detection in credit card transactions. IEEE Access.

[5] Sun, T., Yang, J., Li, J., Chen, J., Liu, M., Fan, L., & Wang, X. (2024). Enhancing auto insurance risk evaluation with transformer and SHAP. IEEE Access, 12, 116546–116557.

[6] Zhang, H. (2025). Reinforcement learning approaches for layout optimization in electronic design automation with electromagnetic compatibility constraints. Frontiers in Robotics and Automation, 2(2), 77–93.

[7] Han, X., Yang, Y., Chen, J., Wang, M., & Zhou, M. (2025). Symmetry-aware credit risk modeling: A deep learning framework exploiting financial data balance and invariance. Symmetry, 17(3), 341.

[8] Chen, J., Cui, Y., Zhang, X., Yang, J., & Zhou, M. (2024). Temporal convolutional network for carbon tax projection: A data-driven approach. Applied Sciences, 14(20), 9213.

[9] Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. Asian Business Research Journal, 10(11), 1–13.

[10] Chen, J., Wang, M., & Sun, T. (2025). Intelligent tax systems and the role of natural language processing in regulatory interpretation. American Journal of Machine Learning, 6(4), 74–94.

[11] Chen, Z., Liu, J., & Chen, J. (2025). Machine learning methods for financial forecasting in enterprise planning: Transitioning from rule-based models to predictive analytics. Frontiers in Artificial Intelligence Research, 2(3), 541–564.

[12] Chen, J., Liu, J., Liang, Y., & Zhou, M. (2026). KE-MLLM: A knowledge-enhanced multi-sensor learning framework for explainable fake review detection. Applied Sciences, 16(6), 2909.

[13] Yue, X., Yang, J., & Liu, W. (2026). FraudDebate-Agent: A multi-agent LLM framework with an evidence-based debate mechanism for financial statement fraud detection. Mathematics, 14(15), 2695.

[14] Zhu, R., & Yue, X. (2024). An explainable machine learning framework for predicting dividend sustainability and financial risk of U.S. REITs under interest-rate shocks. Journal of Trends in Finance and Economics, 1(2), 48–57.

[15] Yue, X., & Zhu, R. (2024). An explainable hybrid machine learning framework for cash-flow-conditioned multi-horizon liquidity risk early warning: An SME-oriented benchmark study. World Journal of Management Science, 2(1), 63–72.

[16] Zi, B. (2024). Cloud-native distributed systems for real-time payment intelligence. AI and Data Science Journal, 1(1), 51–56.

[17] Zhang, S., & Qiu, L. (2023). An interpretable, uncertainty-aware framework for bridge deterioration prediction and risk-based maintenance prioritization. Academic Journal of Architecture and Civil Engineering, 1(2), 21–28.

[18] Jin, J., Xing, S., Ji, E., & Liu, W. (2025). Xgate: Explainable reinforcement learning for transparent and trustworthy API traffic management in IoT sensor networks. Sensors, 25(7), 2183.

[19] Xing, S., Wang, Y., & Liu, W. (2025). Multi-dimensional anomaly detection and fault localization in microservice architectures: A dual-channel deep learning approach with causal inference for intelligent sensing. Sensors, 25(11), 3396.

[20] Xing, S., & Wang, Y. (2025). Proactive data placement in heterogeneous storage systems via predictive multi-objective reinforcement learning. IEEE Access, 13, 117986–117998.

[21] Ji, E., Wang, Y., Xing, S., & Jin, J. (2025). Hierarchical reinforcement learning for energy-efficient API traffic optimization in large-scale advertising systems. IEEE Access.

[22] Xing, S., & Wang, Y. (2025). Cross-modal attention networks for multi-modal anomaly detection in system software. IEEE Open Journal of the Computer Society.

[23] Liu, Y., Ren, S., Wang, X., & Zhou, M. (2024). Temporal logical attention network for log-based anomaly detection in distributed systems. Sensors, 24(24), 7949.

[24] Ren, S., Jin, J., Niu, G., & Liu, Y. (2025). ARCS: Adaptive reinforcement learning framework for automated cybersecurity incident response strategy optimization. Applied Sciences, 15(2), 951.

[25] Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement learning with thinking model for dynamic supply chain network design. IEEE Access, 12, 195974–195985.

[26] Hu, X., Zhao, X., Wang, J., & Yang, Y. (2025). Information-theoretic multi-scale geometric pre-training for enhanced molecular property prediction. PLOS ONE, 20(10), e0332640.

[27] Wang, M., Zhang, X., Yang, Y., & Wang, J. (2025). Explainable machine learning in risk management: Balancing accuracy and interpretability. Journal of Financial Risk Management, 14(3), 185–198.

[28] Yang, J., Li, P., Cui, Y., Han, X., & Zhou, M. (2025). Multi-sensor temporal fusion transformer for stock performance prediction: An adaptive Sharpe ratio approach. Sensors, 25(3), 976.

[29] Zhao, X., Sun, T., Ren, S., Yang, J., & Liu, Y. (2025). RAG-Based AI Agents for Enterprise Software Development: Implementation Patterns and Production Deployment. Frontiers in Artificial Intelligence Research, 2(3), 501–520.

[30] Sun, T., Wang, M., & Chen, J. (2025). Leveraging machine learning for tax fraud detection and risk scoring in corporate filings. Asian Business Research Journal, 10(11), 1–13.

[31] Shang, W., Teng, D., Chen, T., Zhang, F., & Ding, J. (2026). Neuro-Elastic: A unified framework for hardware-aware adaptive quantization and dynamic sparsity in real-time edge intent prediction. IEEE Access.

[32] Teng, D. (2025). TEAS: Token-and energy-aware autoscaling for cost-efficient LLM serving. AI and Data Science Journal, 6(3), 1–10.

[33] Qin, Y., Rhee, M., Teng, D., Zhang, C., & Zi, B. (2026). Neuro-symbolic constraint verification for LLM-driven internet finance transaction execution. Scientific Reports.

[34] Wang, B., Zhao, W., & Wang, Z. (2024). DAS-GNN: A scalable disagreement-aware graph neural framework for anomaly detection in large-scale networks. AI and Data Science Journal, 1(1), 67–75.

[35] Wang, Z., & Shen, Z. (2024). Flex-Talent: An explainable and fair machine learning framework for high-stakes leadership-pipeline evaluation. World Journal of Management Science, 2(1), 73–82.

[36] Guo, Z., Chen, T., & Shang, W. (2022). Class-conditional intermediate-domain adversarial adaptation for cross-device acoustic scene classification. Journal of Computer Science and Electrical Engineering, 4(1), 26–35.

[37] Wang, Y., Bian, M., & Lin, Y. (2022). An explainable temporal graph neural network for disruption prediction and risk propagation in multi-tier supply chains. Journal of Computer Science and Electrical Engineering, 4(1), 17–25.

[38] Zhao, W., & Wang, B. (2022). CAPA: A prediction-driven autoscaling framework for SLO-aware cloud-native machine learning inference. Journal of Computer Science and Electrical Engineering, 4(1), 8–16.