Stout in Computer Science and Technology Studies (SCSTS)

DOI: https://doi.org/10.61424/6zd6va79
Journal Homepage: https://stoutjournals.org/index.php/SCSTS

Risk-Calibrated Multi-Resource Scheduling for Disaggregated AI Inference with LLM-Generated Evidence Memos

Justin Yang
Computer Science, Texas A&M University, College Station, TX, USA
Corresponding Author: Justin Yang, **E-mail**: jyang_tamu@icloud.com

ARTICLE INFORMATION

Submitted: April 09, 2026
Accepted: June 28, 2026
Published: July 31, 2026
Volume: 2
Issue: 1

KEYWORDS

Disaggregated AI inference; multi-resource scheduling; GPU cluster trace; conformal prediction; histogram gradient boosting; integer programming; service-level objective; resource fragmentation; evidence-grounded language models; AIOps.

ABSTRACT

Disaggregated inference separates compute-intensive and accelerator-intensive stages, but it also turns capacity control into a coupled, multi-resource decision under time-varying demand. This study evaluates a risk-calibrated scheduling pipeline on the 2025 Alibaba trace of disaggregated deep-learning recommendation model services. The trace contains 23,871 instances from 156 long-running services over 31 days, including 16,485 compute-node instances and 7,386 heterogeneous-node instances. CPU, GPU, RDMA, memory, disk, deployment-density, and lifecycle fields were aggregated into 15-minute decision epochs. A chronological 60/20/20 design produced 1,689 training windows, 595 calibration windows, and 596 test observations; 595 held-out scheduling decisions were evaluated after lag initialization. The proposed RC-HGB-ILP policy combines a persistence-guarded histogram gradient-boosting forecast, a strictly online one-sided conformal demand envelope, and an integer node-mix planner. It was compared with reactive best-fit, a training-peak inventory, an exact one-step oracle, an uncalibrated HGB planner, and a density-constraint ablation. At 90% nominal coverage, mean empirical envelope coverage reached 94.03%. RC-HGB-ILP reduced the held-out scheduling-SLO violation rate from 18.98% for HGB-ILP to 5.08%, while increasing normalized operating cost by 979.55 cost units, or 0.098%. Relative to reactive best-fit, the violation rate fell from 9.32% to 5.08%. GPU utilization remained 90.25%, and the fragmentation index decreased to 0.31751. Paired epoch tests confirmed lower violation risk and fragmentation, with a small cost increase. An evidence-constrained memo protocol converted saved metrics into seven operational summaries; all 34 numeric claims matched their evidence records, all claims carried evidence identifiers, and all comparison directions were correct. The results show that calibrated upper-demand envelopes can absorb forecast drift at modest capacity cost while preserving high accelerator use, and that bounded evidence memos can expose the resulting trade-offs without numerical drift.

1. Introduction

Large-scale inference systems increasingly separate model stages whose hardware demands differ. Recommendation models may place embedding lookup and network-intensive processing on compute nodes (CNs) while assigning dense neural computation to GPU-bearing heterogeneous nodes (HNs). Generative-model serving similarly benefits from separating prefill and decoding so that each phase can be provisioned independently. This decomposition improves specialization, but it changes the scheduling problem: CPU, GPU, memory, disk, and RDMA reservations must be balanced across linked pools, and a shortfall in either pool can delay the service as a whole. Alibaba's Prism architecture demonstrated this operating model at production scale and documented how disaggregation exposes resource mismatch, communication, topology, and placement constraints (Yang et al., 2025). Related systems such as DistServe and Splitwise reach a similar conclusion for large language model inference: phase separation can increase goodput or reduce cost, but only if provisioning and routing coordinate the resulting resource pools (Zhong et al., 2024; Patel et al., 2024).

The planning difficulty is amplified by workload drift. A point forecast optimized for average error can fall below demand exactly when an operational planner most needs protection. Static peak capacity avoids some underprediction but pays for stale maxima, whereas reactive placement learns about a deficit only after an arrival cannot be admitted. These choices create a three-way tension among service-level risk, utilization, and cost. Traditional vector packing addresses multidimensional capacity (Grandl et al., 2014), and learned schedulers can exploit workload history (Mao et al., 2019), yet neither a good packing heuristic nor a low average forecast error alone provides an interpretable bound on underprovisioning. Conformal prediction offers a useful bridge because it converts residuals into empirical prediction sets without requiring a fully specified demand distribution (Angelopoulos & Bates, 2023). For chronological traces, rolling conformal corrections are particularly relevant because the exchangeability assumption behind random splits is weakened by drift and temporal dependence (Gibbs & Candès, 2021; Barber et al., 2023).

This study asks three questions. First, can a calibrated upper-demand envelope lower scheduling-SLO violations relative to an uncalibrated learned planner and reactive best-fit? Second, what capacity, utilization, fragmentation, energy, and normalized economic trade-offs accompany that reduction? Third, can an evidence-constrained language-model memo layer report those trade-offs without changing numerical values or comparison directions?

The evaluation uses every row in the Alibaba GPU Trace 2025 disaggregated DLRM file. It preserves the documented CN/HN roles and the recorded resource requests, service identities, deployment-density limits, and lifecycle timestamps (Alibaba Group, 2025). The trace does not contain measured hardware utilization, request latency, power, prices, revenue, or language-model annotations. Consequently, the analysis treats resource requests as reserved load and computes utilization and fragmentation against an explicit four-type node scenario. Scheduling delay, cost, power, revenue, and profit are outputs of a fixed trace-driven decision model. This separation allows operational policies to be compared on identical arrivals without attributing synthetic fields to the source trace.

The contribution is an integrated evaluation rather than a single predictor. The pipeline: (a) reconstructs active demand from lifecycle intervals; (b) produces guarded, one-step forecasts for twelve role-resource targets; (c) applies a strictly online, rolling, one-sided conformal correction; (d) solves an integer node-mix problem under resource, count, and deployment-density constraints; (e) evaluates SLO risk, utilization, fragmentation, energy, cost, and profit; and (f) generates concise memos from metric evidence cards. The held-out results establish both the benefit and the limit of calibration. RC-HGB-ILP cut the HGB-ILP violation rate by 13.90 percentage points, but it did not match the exact-demand oracle. The density ablation was identical to the calibrated policy because aggregate resource bounds were already binding in the held-out interval. That zero-effect result is informative: metadata constraints matter only when they exceed the resource-derived lower bound.

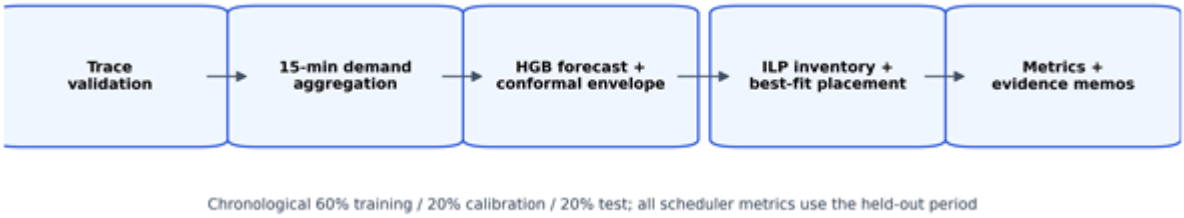

Figure 1. Evaluation pipeline from trace reconstruction to evidence-grounded operational memos.

The evidence memo layer addresses a practical AIOps problem. Scheduling systems routinely emit dense metric tables, while operators need short statements that preserve quantities, units, and direction. Citation-aware generation, atomic factuality evaluation, and retrieval-grounded evaluation provide useful design precedents (T. Gao et al., 2023; Min et al., 2023; Es et al., 2024). Here, each memo sentence is bounded by a serialized evidence card, every numeric claim is associated with an evidence identifier, and the audit compares stated values directly with the saved result record. This makes memo faithfulness an empirical output alongside scheduler performance rather than an informal quality judgment.

2. Literature Review

2.1 Disaggregated inference, forecasting, and capacity risk

Disaggregation is motivated by the fact that model stages do not consume hardware in the same proportions. In recommendation serving, sparse embedding operations can be memory- and communication-heavy, whereas dense computation benefits from accelerators. Prism separates these functions across CN and HN pools and uses high-speed communication to serve production DLRMs at scale (Yang et al., 2025). Huang et al. (2024) likewise describe separating embedding functions from dense computation, emphasizing that storage, network, and accelerator choices become joint system

decisions. For generative inference, DistServe independently provisions prefill and decoding to optimize request goodput subject to service-level objectives (Y. Zhong et al., 2024), while Splitwise studies phase-specific hardware and power trade-offs (Patel et al., 2024). Hybrid cloud–edge routing extends the same specialization principle across deployment locations: inexpensive local capacity handles routine demand, while selected requests are escalated to a more capable cloud model (Xin, 2025b). Together, these systems motivate role-specific forecasting and capacity constraints. A single workload count cannot represent a system in which compute, GPU, RDMA, memory, and disk become bottlenecks in different pools.

Forecasting research increasingly treats uncertainty as part of the capacity decision rather than as a diagnostic added after a point estimate. Time-series foundation models with calibrated intervals have been evaluated for heterogeneous GPU capacity (He et al., 2023), while cross-cloud transfer studies emphasize topology and workload shift (He et al., 2024). Work on Alibaba traces has also linked multi-horizon demand forecasts to operational risk curves and cold-start tenant adaptation (He et al., 2025; S. Zhao et al., 2025). GPU memory prediction supplies a complementary task-level signal when model configuration, not instance count, drives the binding constraint (Wang et al., 2025). Probabilistic forecasting under changing weather and seasonal regimes illustrates the broader value of preserving chronology and reporting uncertainty under nonstationarity (Xin, 2026e). Approximately shared-feature methods provide a formal perspective on why transfer can succeed when domains retain only part of their predictive structure (Z. S. Zhong, Pan, et al., 2025). These studies support the present chronological design, but they also warn against assuming that a predictor fitted to one regime will extrapolate automatically.

Forecasts become operational only after they are connected to inventory, admission, or oversubscription rules. Risk-bounded GPU oversubscription uses conformal demand envelopes to expose the utilization–violation trade-off directly (S. Chen et al., 2024). Profit-aware spot admission makes the economic objective explicit, whereas power-aware inventory planning connects job-level forecasts to energy and procurement decisions (He et al., 2026; S. Zhao et al., 2026). Noisy-neighbor residual fusion addresses a related reliability problem at the virtual-machine layer, where apparently sufficient aggregate capacity may still produce degradation (Nie & Zheng, 2024). Capacity dashboards and visual brief cards further show that interval coverage, tail risk, and recommended action must be communicated together rather than as unrelated metrics (G. Liu et al., 2025). The present paper therefore evaluates forecasting, capacity, risk, cost, power, and memo faithfulness in one trace-driven pipeline.

Inference provisioning research clarifies the role of an SLO. InferLine combines low-frequency planning with a fast reactive controller for pipelines subject to bursty arrivals and latency objectives (Crankshaw et al., 2020). Clockwork improves predictability by controlling execution and admission rather than relying on best-effort sharing (Gujarati et al., 2020). QoS-aware admission work makes the broader point that overload decisions must explicitly represent performance objectives in heterogeneous infrastructure (Delimitrou et al., 2013). The present study does not infer end-to-end request latency from lifecycle fields. Instead, it defines a scheduling SLO on activation delay, which isolates the provisioning consequence of a capacity miss.

2.2 Multi-resource packing, sharing, and calibrated decision policies

Best-Fit is a natural reference for online admission because it assigns an item to the feasible bin with the least residual capacity. Its one-dimensional performance bounds are classical (Johnson et al., 1974), while multidimensional packing introduces harder interactions among resources (Chekuri & Khanna, 2004). Cluster-oriented work extends vector packing to practical schedulers and shows that residual imbalance can strand capacity even when aggregate free resources appear sufficient (Grandl et al., 2014). Dominant-resource fairness supplies a normalized view of heterogeneous demand and prevents a single resource scale from overwhelming allocation logic (Ghodsi et al., 2011). GPU-specific studies further show that fragmentation is operationally consequential: FGD explicitly optimizes fragmentation gradients for GPU-sharing workloads and evaluates the method on Alibaba traces (Weng et al., 2023).

Sharing and elasticity can improve utilization, but they introduce interference and reliability concerns. Salus provides fine-grained GPU sharing and rapid switching (Yu & Chowdhury, 2020); AntMan dynamically scales deep-learning jobs on production GPU clusters (Xiao et al., 2020); and MuxFlow protects latency-critical workloads while using spare GPU capacity for best-effort work (X. Liu et al., 2024). Proteus considers elastic machine-learning execution in resource markets with tiered reliability, linking lower cost to revocation risk (Harlap et al., 2017). These studies inform the present interpretation of slack as potential headroom, not as permission to overcommit blindly.

Learned policies offer another path. DeepRM formulates resource management as sequential decision-making (Mao et al., 2016), Decima learns scheduling policies from job graphs and cluster state (Mao et al., 2019), and Tiresias combines placement and queueing mechanisms for distributed deep learning (Gu et al., 2019). Heterogeneity-aware scheduling, exemplified by Gavel, shows that accelerator type and workload performance must be considered jointly (Narayanan et al., 2020). This paper uses a more transparent composition: histogram gradient boosting predicts aggregate demand, while an integer program makes the

resource decision. Histogram-based boosting is computationally efficient and well suited to nonlinear calendar and lag features (Ke et al., 2017). Separating prediction from optimization also permits direct calibration and ablation of the uncertainty layer.

A broader family of decision-support studies provides transferable methodological patterns, although these applications are not direct evidence about GPU clusters. Probability calibration and uncertainty-driven rejection have been used for credit-risk tiering, model-risk workflows, and selective recruiter screening (Y. Chen et al., 2023; Jin, 2025a; Jin et al., 2024a). Cost-sensitive learning under extreme imbalance has been studied for fraud and bankruptcy decisions, where the operational loss is asymmetric even when average classification accuracy appears acceptable (Y. Chen et al., 2024; Jin et al., 2024b). Uncertainty-aware late fusion similarly separates confidence estimation from a downstream fusion rule (Xin, 2025c). These precedents reinforce the distinction between predictive accuracy and the quality of a decision made from the prediction.

Policy evaluation work adds a second transferable principle: a candidate policy should be assessed against a fixed information set and explicit costs. Uplift modeling links estimated heterogeneous effects to interpretable action policies (Mu et al., 2023), while offline counterfactual evaluation and conservative policy selection test decisions without deploying every alternative online (Mu et al., 2025; Ye et al., 2026). Related studies combine demand forecasts with constrained budgeting, privacy-robust incrementality, or identifier-loss-aware optimization (Bai, Wang, et al., 2026; Bai & Wu, 2026; Lu et al., 2025). Macro-market fusion under uncertainty offers another example of preserving temporal order and evaluating a directional decision rather than only a point forecast (Zhou & Zhang, 2025). Counterfactual credit explanations and spectral-estimator uncertainty further illustrate that uncertainty should be paired with an interpretable downstream action or inferential statement (Xin, 2026b; Z. S. Zhong & Ling, 2024). In this paper, the analogous action is the integer node mix, and the policy is scored on violations, utilization, fragmentation, cost, energy, and profit.

2.3 Conformal risk control and energy-aware operation

Conformal prediction turns a point or quantile predictor into an interval using held-out residuals. Conformalized quantile regression handles heteroscedasticity by calibrating learned conditional bounds (Romano et al., 2019). Adaptive conformal inference updates the correction under distribution shift (Gibbs & Candès, 2021), while work beyond exchangeability explains how weighted or localized residuals can retain useful coverage behavior for dependent data (Barber et al., 2023).

The cluster-specific studies above show why this distinction matters operationally: calibrated capacity forecasts and conformal oversubscription policies report empirical protection, whereas an uncalibrated point forecast exposes the planner to asymmetric underprediction loss (S. Chen et al., 2024; He et al., 2023). This study therefore uses a one-sided rolling correction because a capacity planner cares primarily about underprediction. Coverage is evaluated target by target and across several risk levels rather than inferred from the nominal setting.

Power is included because extra safety capacity is not free. μ-Serve jointly considers model serving and power management under SLO constraints (Qiu et al., 2024), Zeus quantifies energy-performance trade-offs for GPU workloads (You et al., 2023), and the broader sustainable-AI literature argues that infrastructure efficiency should accompany model performance metrics (Wu et al., 2022). Job-level power-aware inventory planning and profit-aware spot admission bring this trade-off closer to procurement and market decisions (He et al., 2026; S. Zhao et al., 2026). The present energy measure remains scenario-based: node power ratings are fixed, and energy equals provisioned power multiplied by epoch duration. It therefore measures relative policy demand under a common hardware scenario.

2.4 AIOps diagnosis, retrieval, and evidence chains

Capacity decisions occur inside a wider operational workflow that includes anomaly detection, diagnosis, triage, and remediation. Multi-source root-cause attribution combines CPU and network signals with language-model reasoning for microservice faults (G. Liu et al., 2023), while OpsLLM connects bilingual retrieval with incident triage and alert summarization (G. Liu et al., 2024). Evidence-grounded DevOps RAG limits answers to private operational documents and evaluates hallucination and citation quality, which is directly relevant to a memo layer that must not invent a number or command (B. Zhang et al., 2024). Cost-aware AIOps routing extends the idea by choosing among parsing, detection, diagnosis, and summarization paths under quality and inference-cost constraints (C. Li et al., 2026). Trajectory reliability prediction similarly treats tool-use success as a forecastable property of an agent rather than assuming that every generated workflow will complete (Z. S. Zhong et al., 2026).

Operational evidence can be created from logs as well as metric tables. Retrieved normal prototypes have been used to explain OpenStack failure injections (Xin, 2025a); self-supervised LogBERT-style models detect anomalous HDFS sequences (Xin, 2026g); and retrieval-grounded HDFS analysis couples anomaly scores with deterministic failure narratives (Sun et al., 2026). Conformal alert control adds an explicit false-alert trade-off before generating an incident ticket (Xin, 2026d). Evidence-constrained incident

cards then organize Hadoop, OpenStack, and ZooKeeper findings for SRE review (Nie et al., 2026). These systems motivate the separation used here between the scheduler, the evidence record, the memo generator, and the independent audit.

Security-oriented operational analytics reaches the same conclusion from a different direction. TinyLLM intrusion detection and language-guided feature selection study compact or interpretable detection pipelines for IoT and DDoS traffic (Y. Li & S. Lu, 2025; Lu & Zhou, 2024). Deep-autoencoder traffic classification, predictive DDoS mitigation, attack-chain stage detection, and system-call window localization all require a model output to be connected to a bounded forensic explanation or mitigation decision (Bai, Chen, et al., 2026; Wang et al., 2024; Xin et al., 2024; Xin, 2026c). For an infrastructure copilot, the lesson is not that security and scheduling share the same target; it is that operational text must preserve the provenance, scope, and direction of the underlying detector or planner.

Retrieval research provides concrete mechanisms for that preservation. Evidence-chain RAG has been evaluated with word-level hallucination detection, source attribution, and deterministic provenance explanations (C. Li et al., 2024; Jin, 2025b). Time-aware retrieval with confidence-gated memory shows that the evidence state itself must be updated and forgotten deliberately in multi-session systems (Sun et al., 2023). Retrieval–summary integration for code intelligence and hybrid reranking for financial search illustrate how findability and explanation can be evaluated as separate stages (Y. Li, 2024; Meng et al., 2026). Budgeted multi-hop retrieval adds an explicit stopping and cost policy, while learned retrieval-quality prediction estimates when the evidence set is weak before generation (Su et al., 2025; R. Zhang et al., 2025). Evidence-calibrated unanswerable-question answering goes further by linking retrieval coverage to abstention, a useful analogue for declining to write an operational claim when the metric card lacks the necessary field (Z. S. Zhong, Chen, et al., 2025).

Evidence constraints have also been tested in domains where numerical or regulatory fidelity is central. Accounting-aware retrieval, disclosure-review cards, numerical-reasoning guardrails, and evidence-constrained monitoring agents all require a statement to remain tied to a source record (Y. Chen, Zhou, et al., 2025; K. Zhang et al., 2025; Z. Li et al., 2026; Zhou et al., 2026). Scientific-search evidence cards and narrative-aware claim verification emphasize ranking transparency and claim-level support (Jin, 2025c; Su, Chen, et al., 2026). Multilingual humanitarian RAG and multi-regulation legal RAG show that retrieval reliability must survive language or governance boundaries rather than relying on a single undifferentiated corpus (Y. Chen & Xu, 2026; J. Li & Zhou, 2026). These are cross-domain design precedents; the present evaluation applies their common evidence discipline to scheduler metrics.

2.5 Safe and human-readable operational memos

Evidence fidelity does not by itself guarantee safe behavior. Continual red-teaming treats new jailbreaks as an evolving stream that requires guardrail updates, while behavior-level refusal work evaluates whether utility is preserved when unsafe actions are blocked (D. Zheng et al., 2023; Zheng & Li, 2024). Evidence-based self-verification adds a second check before an agent response is released (D. Zheng et al., 2024). Privacy and data-integrity risk cards apply the same principle to prompt-injection oversight by exposing the threat, affected evidence, and required human action (Su, Rao, et al., 2026). These studies motivate a deliberately narrow memo surface in which the generator cannot introduce a command, unsupported cause, or new quantity.

Human-facing presentation affects whether a technically grounded statement can be reviewed. Scientific-search evidence cards, capacity-dashboard briefs, and financial-risk dashboards emphasize visible evidence hierarchy and calibrated risk communication (Jin, 2025c; G. Liu et al., 2025; Z. Li et al., 2025). Text-grounded design-rationale interfaces and structured visual briefs treat machine output as editable decision material rather than a final unquestionable answer (Mu et al., 2026; Tu et al., 2025; Wu et al., 2025). Research on dark-pattern explanations and LLM design criticism provides a useful caution: fluent interface text can still manipulate attention or diverge from human judgment (H. Xu et al., 2025; Y. Li, Lu, et al., 2025).

Recommendation systems offer adjacent examples of the same communication problem. Explainable recommendation cards, review-grounded faithfulness evaluation, and behavior-retrieval pipelines link a recommendation to observable evidence instead of presenting only a score (B. Zhang et al., 2025; Chang et al., 2026; Xin, 2026a). Customer-representation and product-classification studies illustrate the predictive layer that precedes such explanations, whereas privacy-preserving topic preference learning shows why the user model should not become unrestricted memory (Kuo et al., 2025; Xin, 2026f; K. Xu et al., 2024). These works do not validate GPU scheduling algorithms; they inform the memo design requirement that evidence, recommendation, and uncertainty remain visually and semantically separable.

Finally, operational text generation must be evaluated against evidence rather than judged by fluency. Citation-aware language-model research measures whether claims are supported and whether cited evidence entails the text (T. Gao et al., 2023). FActScore decomposes long-form output into atomic factual claims (Min et al., 2023), RAGAs formalizes faithfulness and relevance for retrieval-augmented generation (Es et al., 2024), and RARR uses retrieval to correct unsupported statements after

generation (L. Gao et al., 2023). AIOpsLab places such agents in operational workflows and emphasizes controlled, repeatable evaluation (Y. Chen, Shetty, et al., 2025). These ideas motivate the memo audit used here: numbers, evidence tags, and comparison directions are checked separately.

3. Methodology

3.1 Dataset and trace reconstruction

The source file contains 23,871 service instances and 17 columns. Each row identifies an anonymized instance, its CN or HN role, an application name, request and limit values for CPU, GPU, RDMA, memory, and disk, a maximum instances-per-node constraint, and creation, scheduled, and deletion timestamps. The official trace description reports 156 services and distinguishes 16,485 CN instances from 7,386 HN instances (Alibaba Group, 2025). HN rows request one GPU; CN rows request none. Missing lifecycle endpoints are common, so reconstruction used a deterministic interval rule. If creation time was absent, scheduled time became the start. If deletion time was absent, the interval ended at the 31-day trace boundary. If both start fields were absent, the instance was treated as active from time zero. Ends were clipped to the trace boundary and constrained to be no earlier than starts.

Table 1. Dataset profile after schema validation and lifecycle reconstruction.

Characteristic	Value	Unit
Instances	23,871	rows
Application services	156	groups
CN instances	16,485	rows
HN instances	7,386	rows
Trace span	31	days
Observed creation times	16,591	rows
Observed scheduled times	16,591	rows
Observed deletion times	14,993	rows
Median empirical scheduling wait	0	seconds
P90 empirical scheduling wait	59	seconds
P95 empirical scheduling wait	136.50	seconds

The active set at epoch t contains rows whose reconstructed interval covers the epoch boundary. For role k and resource r, reserved demand is

$$D_{k,r,t}=\Sigma_{i \in A_{k,t}} x_{i,r},$$

where $x_{i,r}$ is the recorded request, not the limit. Requests were chosen because they represent admission commitments, whereas limits can express burst ceilings. Resource values were not rescaled before aggregation. The resulting series contains CPU, GPU, RDMA, memory, disk, and active-instance count for both roles. New arrivals were assigned to the first 15-minute decision boundary at or after their reconstructed start.

Table 2. Recorded resource-request distributions by disaggregated role.

Role	Resource	Min.	Median	Mean	P90	Max.
CN	cpu	24	64	72.17	96	192
CN	gpu	0	0	0	0	0
CN	rdma	1	1	17.65	100	100
CN	memory	120	320	363.94	480	1,000
CN	disk	240	255	366.43	680	996
HN	cpu	2	8	8.52	12	24
HN	gpu	1	1	1	1	1
HN	rdma	1	25	27.86	25	75
HN	memory	16	40	46.83	80	400
HN	disk	80	100	178.53	255	680

Note. RDMA values are percentages; memory and disk are GiB.

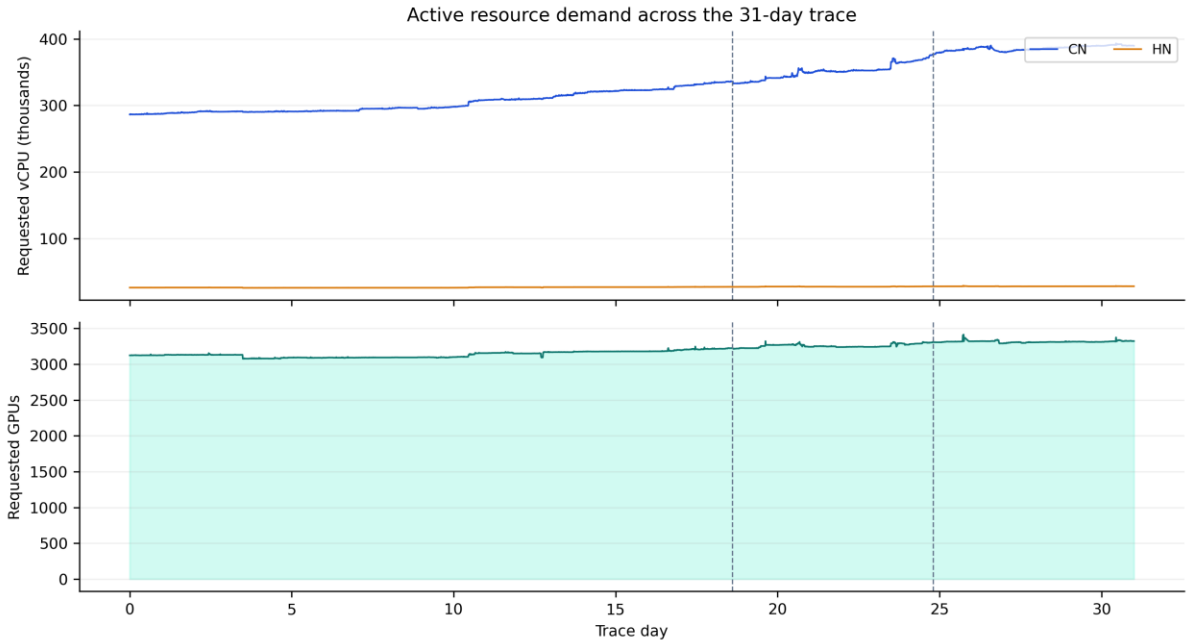

Figure 2. Thirty-one-day active demand by disaggregated role. The later level shift motivates chronological testing and forecast guarding.

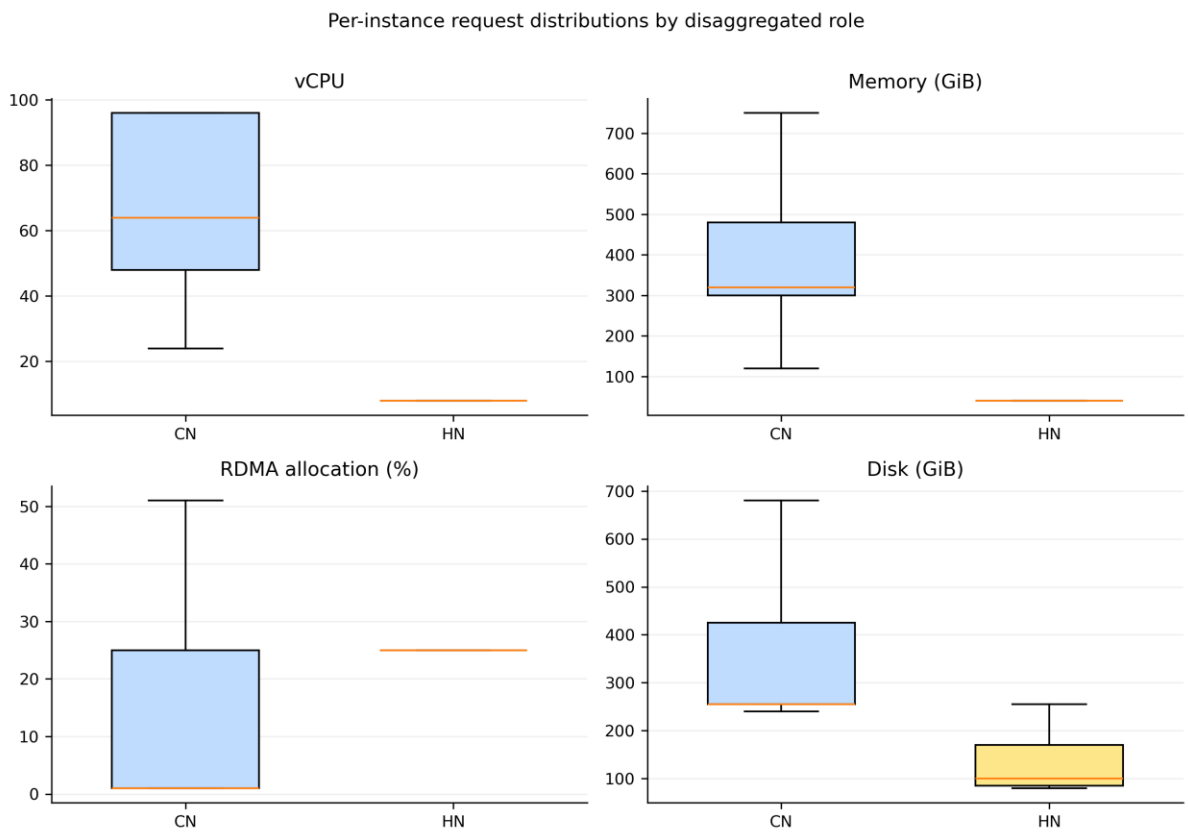

Figure 3. Empirical request distributions for CPU, memory, RDMA, and disk, separated by CN and HN roles.

The trace's scheduling timestamps also permit a separate descriptive check. For rows with both creation and scheduled time, empirical wait equals scheduled minus creation. These values characterize the source system but are not used as policy outcomes. The simulator's scheduling SLO is generated by the capacity decision model described below, thereby avoiding leakage from the historical scheduler.

3.2 Chronological design and demand forecasting

The 31-day series was divided chronologically: 60% for training, 20% for calibration, and 20% for testing. After constructing lag features, the usable design contained 1,689 training rows, 595 calibration rows, and 596 test observations. The first held-out observation initialized lagged state, leaving 595 one-step scheduling decisions. No random shuffle was applied. Each target used lag-1, lag-2, lag-4, lag-12, lag-24, lag-96, and lag-672 values, together with hour-of-day and day-of-week sine/cosine terms. This feature set covers immediate inertia, one hour, three hours, six hours, one day, and one week.

The chronological split and multi-scale lag design reflect the failure modes identified in cross-cloud, multi-horizon, and cold-start capacity forecasting, where random mixing can hide the regime changes faced by an operational planner (He et al., 2024, 2025; S. Zhao et al., 2025). GPU-memory prediction provides a complementary per-task approach, but the available trace supports role-level resource reservations rather than model-architecture features (Wang et al., 2025).

Twelve independent HistGradientBoostingRegressor models were trained: count plus five resources for each role. Each model used 45 boosting iterations, learning rate 0.08, maximum leaf nodes 15, minimum 20 samples per leaf, and the common seed 20,250,729. The estimator is a deterministic histogram gradient-boosting implementation in the LightGBM-style tree family, not the LightGBM software package; Ke et al. (2017) provide the algorithmic motivation for histogram-based boosting. A persistence forecast, $D^P_{t+1}=D_t$, was evaluated separately. The guarded forecast supplied to calibration was

$$D^G_{t+1}=\max(D^{HGB}_{t+1},D_t).$$

The floor protects against a downward level mismatch between the training period and the later trace. This guard is intentionally simple and inspectable. Mean absolute error (MAE), root mean squared error (RMSE), 90th-percentile absolute error, and weighted absolute percentage error (WAPE) were calculated on the held-out interval.

Table 4. Scheduling policies and chronological experimental design.

Policy	Signal	Planner / rule	Train	Cal.	Test obs.
Reactive-BF	Lagged demand	Lagged aggregate best-fit capacity lower bound with integer node-mix planning	1689	595	596
Static-TrainPeak	Training-only peak	ILP inventory fixed at the largest training-period aggregate demand	1689	595	596
Oracle-ILP	One-step oracle bound	Integer node-mix planner with exact next-epoch demand	1689	595	596
HGB-ILP	Point forecast	Histogram gradient-boosted demand forecast followed by integer node-mix planning	1689	595	596
RC-HGB-ILP	90% upper envelope	HGB forecast plus one-sided split-conformal envelope and integer planning	1689	595	596
RC-HGB (no density)	Ablation	RC-HGB-ILP with application deployment-density constraints removed	1689	595	596

Note. The first test observation initializes lagged state; 595 scheduling decisions are scored.

3.3 Online conformal demand envelope

For each target, one-sided residuals were defined as $e_t=D_t- D^G_t$. At a decision time, the conformal correction q_t was computed only from residuals available before the target epoch. The initial history was the calibration segment; thereafter a rolling window of the most recent 192 residuals was used. For risk level α, the finite-sample "higher" empirical quantile at rank $[(n+1)(1-\alpha)]/n$ produced the correction. The upper demand envelope was

$U_t+1=\max(0, D^G_t+1+q_t)$.

The main setting used $\alpha=0.10$. Sensitivity runs used 0.05, 0.10, and 0.20 in the full scheduler, while coverage-width analysis also evaluated 0.15 and 0.25. Coverage is the fraction of target values not exceeding U_t. Normalized width is mean q_t divided by mean demand. Strict chronology is crucial: residuals from a test epoch enter the window only after that epoch has been scored.
This construction differs from a static split-conformal interval because its correction evolves with observed residuals. It also differs from a generic calibrated classifier: the output is a one-sided capacity commitment whose operational loss is concentrated in undercoverage (S. Chen et al., 2024; He et al., 2023).

3.4 Capacity scenario and integer planner

The trace specifies instance demands but not physical node models. A four-type scenario was therefore fixed before policy comparison. CN-S and CN-L provide CPU, RDMA, memory, and disk; HN-4 and HN-8 additionally provide four and eight GPUs. Capacities cover the observed instance ranges and make all rows feasible. Cost is expressed in normalized cost units (CU) per node-hour, and power is a fixed kilowatt rating. These values are scenario coefficients used identically by every policy.

Table 3. Fixed node-capacity, normalized-cost, and power scenario.

Type	Role	vCPU	GPU	RDMA %	Memory GiB	Disk GiB	CU/h	kW
CN-S	CN	96	0	100	480	1,024	1	0.35
CN-L	CN	192	0	100	960	2,048	1.85	0.62
HN-4	HN	48	4	100	480	2,048	3.20	1.40
HN-8	HN	96	8	100	960	4,096	5.80	2.50

Note. Node capacities, costs, and power ratings are fixed scenario parameters.

For each role and epoch, the planner selects integer node counts n_j that minimize $\Sigma_j c_{jn_j}$. Constraints require aggregate capacity to exceed the forecast or upper envelope for all five resources. A count lower bound enforces active instances divided

by the maximum feasible deployment density. A large-instance lower bound guarantees enough large nodes for requests that do not fit the small type. The resulting mixed-integer linear program has two decision variables per role and is solved with SciPy's MILP interface. The oracle uses exact next-epoch demand; the learned policies use forecast demand. Because the experiment studies planning rather than a proprietary production topology, communication delay and rack locality are not inferred.

Prediction, uncertainty protection, and constrained selection remain separate modules. This separation makes the planner auditable and allows the same demand envelope to be interpreted under different spot-admission, inventory, or hybrid-deployment objectives without claiming that any one economic coefficient is universal (He et al., 2026; S. Zhao et al., 2026; Xin, 2025b).

Six policies were evaluated. Reactive-BF uses the previous epoch's actual aggregate demand and represents best-fit admission with nodes added after a capacity miss. Static-TrainPeak fixes the least-cost node mix that covers the maximum aggregate training demand. Oracle-ILP plans from exact next-epoch demand and is an unattainable lower-risk reference. HGB-ILP uses the uncalibrated HGB point forecast. RC-HGB-ILP uses the guarded forecast plus the 90% online conformal envelope. RC-HGB (no density) removes application deployment-density constraints while retaining resource and large-instance bounds.

The rolling-horizon simulator computes role-specific capacity at each epoch. If actual demand exceeds planned capacity in any dimension, emergency capacity is added for the interval. The affected-arrival fraction equals the emergency share of role capacity cost, capped at one, and those arrivals receive a 120-second activation delay; other arrivals receive zero delay. A scheduling-SLO violation occurs when delay exceeds 60 seconds. This binary cold-activation model deliberately isolates capacity misses. It does not claim to reconstruct request latency.

3.5 Metrics, economic accounting, and statistical tests

Reserved-resource utilization is demand divided by provisioned capacity. GPU slack is one minus HN GPU utilization. For role k, the fragmentation proxy is the difference between the largest normalized resource utilization and the mean normalized utilization across resources with positive capacity. The overall index is weighted by role capacity cost:

$F_t=\Sigma_k C_{k,t}(\max_r u_{k,r,t}-\text{mean}_r u_{k,r,t}) \Sigma_k C_{k,t}.$

This index is zero when all dimensions are equally filled and grows when one binding dimension strands others. It complements, rather than replaces, placement-level fragmentation measures such as FGD (Weng et al., 2023).

Operating cost is provisioned node cost multiplied by 0.25 hours per epoch. Energy is the analogous power integral. Each active instance earns a normalized hourly revenue equal to 1.35 times its dominant share of the applicable base-node capacity multiplied by that node's hourly cost. A violating arrival incurs 2 CU. Profit is revenue minus operating cost and SLO penalties. The penalty is intentionally reported as a scenario coefficient; it does not represent an Alibaba tariff. The full metric set includes SLO violations, mean and P95 activation wait, mean node count, CPU/GPU/memory/RDMA utilization, GPU slack, fragmentation, cost, energy, revenue, penalty, and profit.

Policy differences were tested on paired epochs. For SLO-violation rate, cost, and fragmentation, 10,000 seeded bootstrap resamples produced percentile 95% confidence intervals for the mean paired difference. A two-sided Wilcoxon signed-rank test assessed whether the paired distribution was centered at zero. Zero-difference ablations were assigned p=1.

3.6 Evidence-constrained memo protocol

The memo stage receives only saved scheduler metrics. One evidence card is created per policy with SLO-violation rate, GPU utilization, fragmentation, cost, and profit; a seventh card stores the RC-HGB-ILP minus HGB-ILP differences. A GPT-5-family language model generated one concise memo per card under three constraints: it could use only numeric values present in the card, it had to append the evidence identifier, and comparison verbs had to agree with the sign of the stored difference. The seven outputs were frozen before audit.

The protocol adapts evidence-chain and DevOps-RAG principles to a deliberately smaller generation surface: retrieval is replaced by a single serialized metric record, and the allowed claims are defined by its schema (C. Li et al., 2024; B. Zhang et al., 2024). The evidence identifier supports the same traceability objective pursued by deterministic provenance explanations and incident-visualization cards (Jin, 2025b; Nie et al., 2026).

The audit decomposed each memo into numeric claims and compared each value with its source record at the reported precision. Numeric exact match is the percentage of claims whose stated and source values are identical. Evidence-tag coverage is the percentage of numeric claims linked to a nonempty card identifier. Directional consistency checks whether "increased," "decreased," or "did not change" matches the sign of each cross-policy difference. Mean memo length measures concision. These checks operationalize atomic support and evidence faithfulness concepts from FActScore and RAGAs (Min et al., 2023; Es et al., 2024).

4. Results and Discussion

4.1 Workload structure and forecasting

The trace was strongly heterogeneous by role. CN instances reserved a median 64 vCPUs and 320 GiB of memory, with no GPUs; HN instances reserved one GPU, a median eight vCPUs, and 40 GiB of memory. RDMA requests were especially asymmetric: the CN median was 1% but the 90th percentile was 100%, while the HN median and 90th percentile were both 25%. These distributions explain why a count-only model would miss binding network and memory dimensions. The active-demand series also exhibited a later level shift, visible in Figure 2.

The raw HGB model performed poorly across that shift. For CN count, raw HGB WAPE was 14.122%, compared with 0.052% for persistence; for CN CPU, the corresponding values were 13.579% and 0.061%. The persistence guard therefore dominated raw HGB for most targets. It did not blindly replace the learned model: for CN RDMA, the higher raw prediction was retained and guarded-HGB WAPE was 1.465%, whereas persistence alone achieved 0.042%. This exception shows the trade-off of a one-sided guard: it protects against systematic downward drift but can preserve upward model error. Zero-demand CN GPU was predicted exactly by all methods.

The observed failure is consistent with the motivation for cross-cloud and cold-start forecasting, but it is specific to this trace and split: a nonlinear learner fitted to the early regime did not extrapolate the later level, while a one-step persistence baseline did (He et al., 2024, 2025). The result therefore favors guarded composition here, not a general claim that persistence dominates multi-horizon or foundation-model approaches.

Table 5. Held-out one-step WAPE (%) for persistence, raw HGB, and guarded HGB.

Target	Persistence	Raw HGB	Guarded HGB
CN_count	0.052	14.122	0.052
CN_cpu	0.061	13.579	0.061
CN_disk	0.045	11.220	0.045
CN_gpu	0.000	0.000	0.000
CN_memory	0.061	13.441	0.061
CN_rdma	0.042	1.465	1.465
HN_count	0.044	3.002	0.044
HN_cpu	0.043	3.185	0.043
HN_disk	0.032	2.889	0.032
HN_gpu	0.044	3.002	0.044
HN_memory	0.041	2.752	0.041
HN_rdma	0.045	1.766	0.045

Note. WAPE is total absolute error divided by total observed demand.

Figure 4 plots representative test targets, guarded forecasts, and conformal upper envelopes. Corrections were small relative to aggregate load because one-step trace dynamics were persistent. At $\alpha=0.10$, normalized mean width ranged from 0% for CN GPU to 0.126% for CN CPU. Nevertheless, the corrections materially changed capacity decisions at the margin.

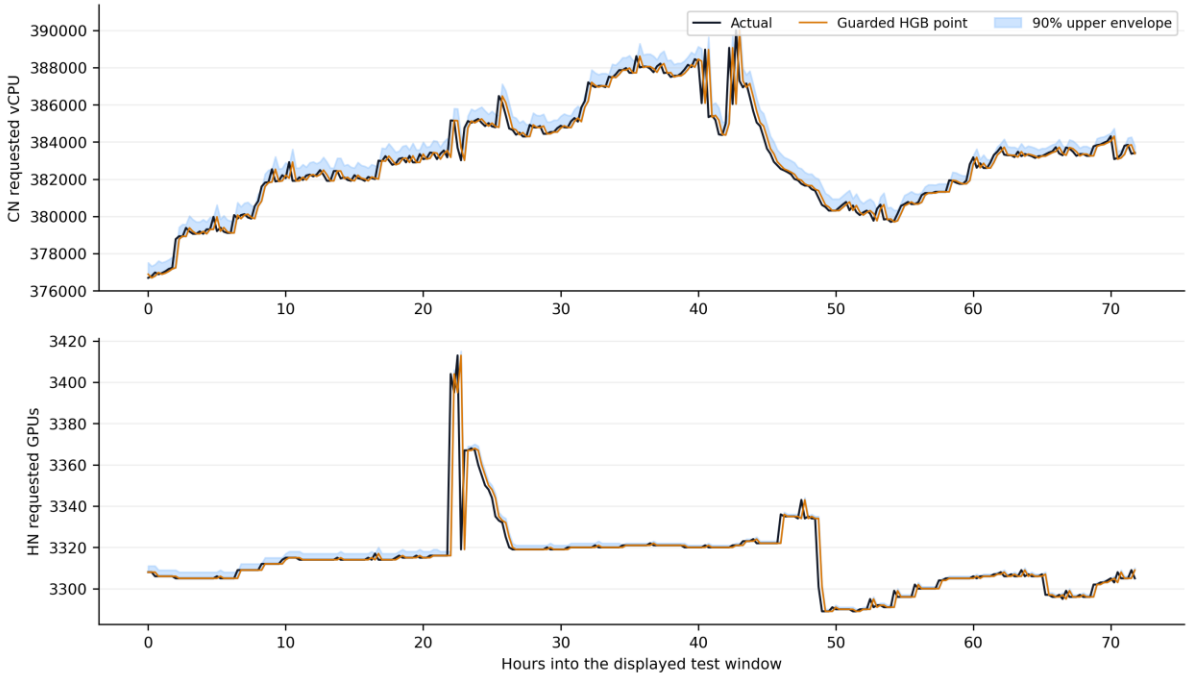

Figure 4. Representative guarded forecasts and strictly online one-sided conformal demand envelopes in the held-out interval.

4.2 Calibration quality

Mean target-level coverage at the nominal 90% setting was 94.03%. Eleven of twelve targets exceeded 92.28%, and CN GPU achieved 100% because demand was identically zero. The result is conservative rather than exactly nominal. Rolling residual quantiles and the persistence floor both contribute to this behavior: the floor removes many positive residuals, while the higher finite-sample quantile favors the safe side.

Table 6. Target-level online conformal calibration at α = .10.

Target	Nominal %	Coverage %	Mean q	Width %
CN_count	90.000	93.121	6.40	0.116
CN_cpu	90.000	92.617	487.33	0.126
CN_gpu	90.000	100.000	0	0.000
CN_rdma	90.000	100.000	5.05	0.008
CN_memory	90.000	92.450	2,460.23	0.126
CN_disk	90.000	92.282	2,283.11	0.103
HN_count	90.000	93.456	1.38	0.042
HN_cpu	90.000	92.282	11.58	0.041
HN_gpu	90.000	93.456	1.38	0.042
HN_rdma	90.000	93.624	37.96	0.041
HN_memory	90.000	92.785	58.96	0.037
HN_disk	90.000	92.282	165.68	0.026

Note. Mean q is in the native unit of each demand target.

Coverage moved monotonically with the nominal level. At 95% nominal coverage, mean empirical coverage was 97.20% and normalized mean width was 0.127%; at 90%, coverage was 94.03% and width was 0.059%; at 75%, coverage remained 84.10%.

The gap between nominal and empirical rates reflects discreteness, target pooling in the summary, and conservative online quantiles. Figure 5 makes the practical choice visible: reducing α widens envelopes and increases protection, but additional coverage becomes progressively more expensive in the scheduler.

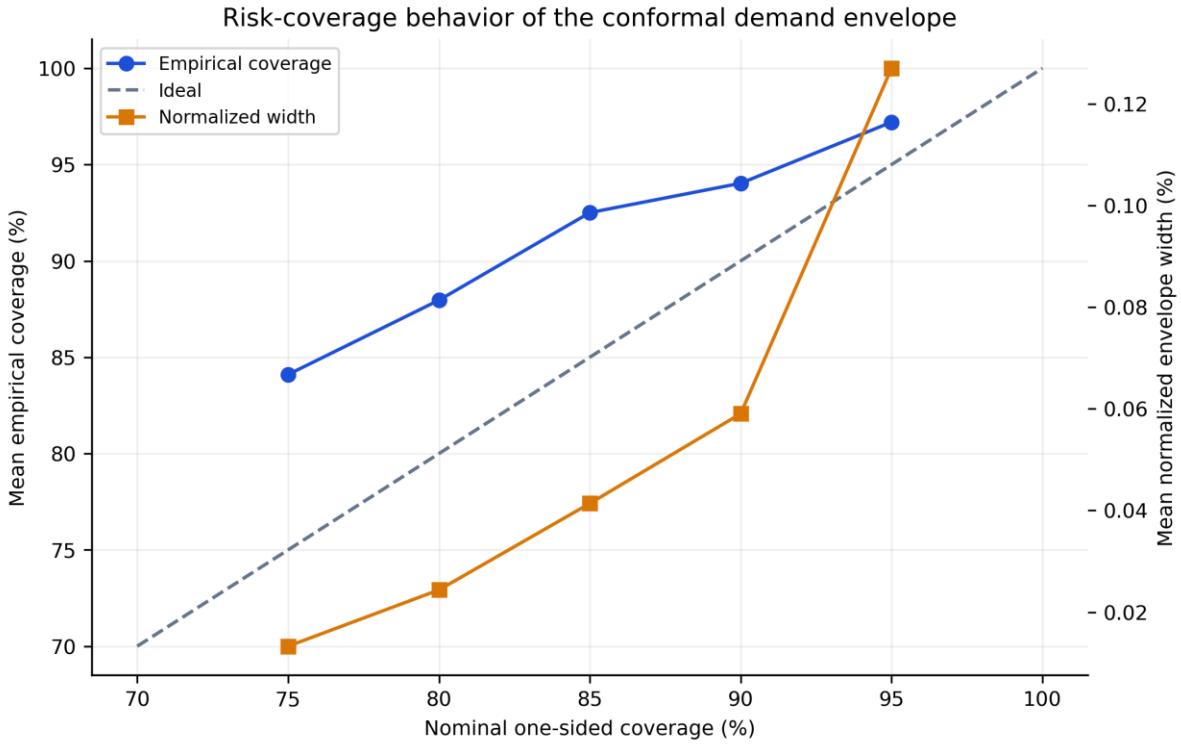

Figure 5. Empirical coverage and normalized width across conformal risk levels.

4.3 Scheduler comparison

Table 7 reports the principal held-out comparison over 595 decision epochs and 3,230 arrivals. HGB-ILP produced the highest violation rate, 18.978%, because the raw point forecast underestimated the shifted test demand. Static-TrainPeak was similar at 18.359%, indicating that a training maximum can become stale. Reactive-BF performed better at 9.319% because its previous-epoch demand followed the new level, but it still reacted after changes. RC-HGB-ILP reduced violations to 5.077%, corresponding to 164 affected arrivals. Oracle-ILP had zero violations because it received exact next-epoch demand.

Table 7. Overall held-out scheduler outcomes across 595 decisions and 3,230 arrivals.

Policy	Affected	SLO %	GPU %	Frag.	Cost CU	Energy kWh	Profit CU
Reactive-BF	301	9.319	90.288	0.31774	998,182.9	379,234.6	478,764.0
Static-TrainPeak	593	18.359	90.307	0.31781	997,979.3	379,157.1	478,383.6
Oracle-ILP	0	0.000	90.307	0.31783	997,898.5	379,128.8	479,650.4
HGB-ILP	613	18.978	90.307	0.31782	997,939.8	379,143.2	478,383.1
RC-HGB-ILP	164	5.077	90.253	0.31751	998,919.3	379,498.5	478,301.6
RC-HGB (no density)	164	5.077	90.253	0.31751	998,919.3	379,498.5	478,301.6

Note. Affected arrivals incur the 120-second activation delay.

The calibrated policy reduced violations by 13.901 percentage points, or 73.24%, relative to HGB-ILP. Relative to Reactive-BF, the reduction was 4.241 percentage points, or 45.51%. This protection required a modest capacity increment. RC-HGB-ILP cost 998,919.325 CU, 979.550 CU more than HGB-ILP and 736.400 CU more than Reactive-BF. These increments equal approximately

0.098% and 0.074% of the respective baselines. Mean node count increased from 2,957.34 for HGB-ILP to 2,960.49 for RC-HGB-ILP.

This measured cost-risk movement is the scheduling counterpart of conformal oversubscription and profit-aware admission: the additional envelope is valuable only insofar as the reduced miss risk justifies its capacity cost (S. Chen et al., 2024; S. Zhao et al., 2026). Because the present cost and penalty units are scenario parameters, the paper reports the Pareto movement rather than transferring a profit threshold from another deployment.

Accelerator efficiency remained high. RC-HGB-ILP achieved 90.253% GPU utilization, only 0.054 percentage points below HGB-ILP, with 9.747% GPU slack. Its overall fragmentation index was 0.31751, lower than HGB-ILP's 0.31782 and Reactive-BF's 0.31774. CPU and memory utilization were also slightly lower because the safety envelope added capacity. Figure 6 summarizes these efficiency measures, while Figure 7 places the policies in cost-risk space.

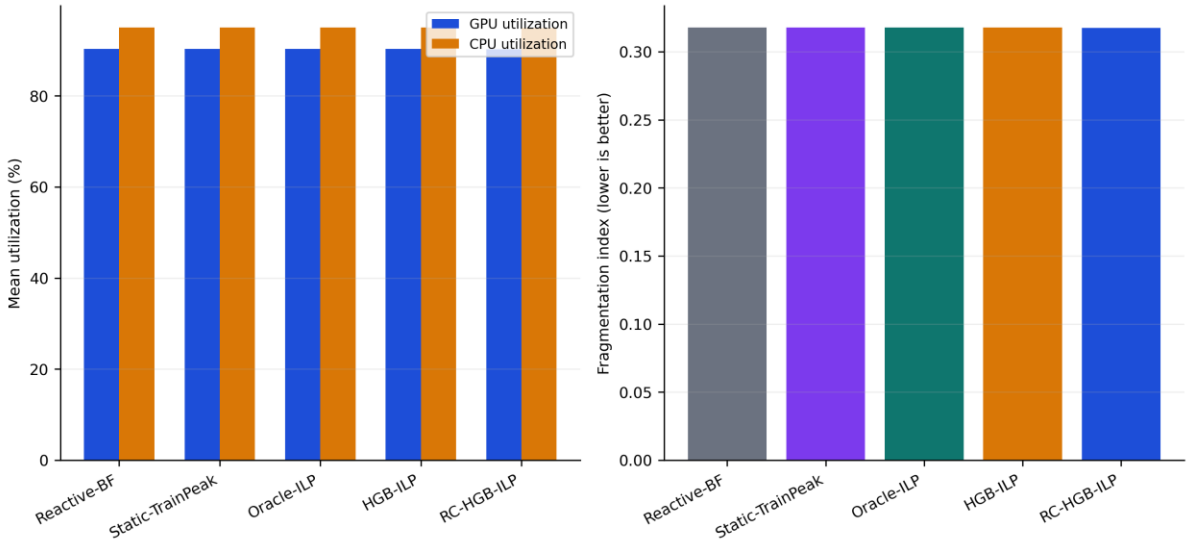

Figure 6. Policy comparison for utilization, GPU slack, and resource-fragmentation proxy.

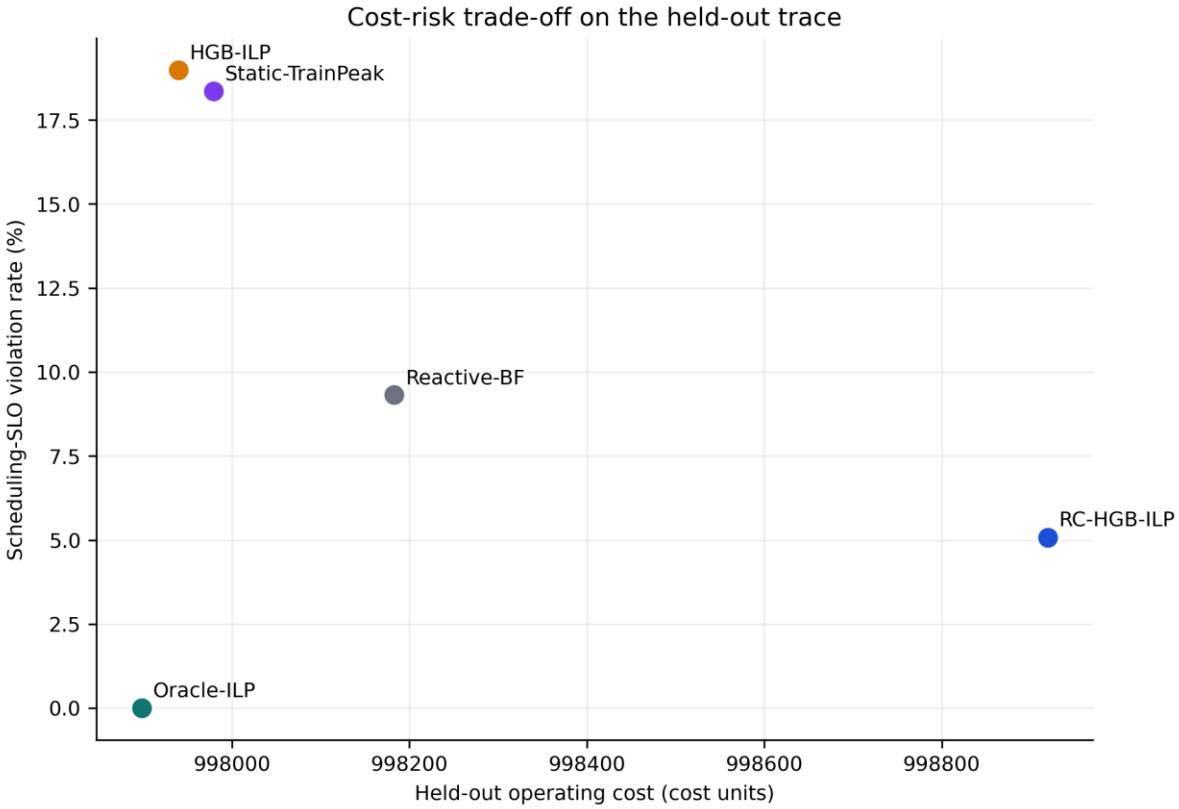

Figure 7. Held-out normalized operating cost versus scheduling-SLO violation rate.

The normalized profit result requires careful interpretation. All policies earned the same 1,477,548.899 CU because active trace demand was identical. RC-HGB-ILP incurred lower SLO penalties than HGB-ILP, 328 versus 1,226 CU, but the 898-CU penalty saving did not fully offset the 979.55-CU capacity increment; profit was therefore 81.55 CU lower. Reactive-BF earned 462.40 CU more profit than RC-HGB-ILP under the specified 2-CU penalty. A larger business penalty for delayed admissions would change the ranking. The paper therefore treats SLO risk, cost, and profit as separate outcomes rather than claiming a universally optimal scalar objective.

Role-level utilization identifies the physical bottlenecks. Under RC-HGB-ILP, CN memory utilization was 99.854% and CN CPU utilization was 98.374%, while CN RDMA and disk were 31.694% and 52.968%. HN RDMA was 99.897% and GPU utilization was 90.253%, whereas HN CPU, memory, and disk were below 65%. Thus the node scenario produces two different bottleneck profiles: memory/CPU on CN and RDMA/GPU on HN. This is precisely the coupling that disaggregated scheduling must respect.

Table 8. Role-specific reserved-resource utilization for principal planned policies.

Policy	Role	CPU %	GPU %	RDMA %	Memory %	Disk %
HGB-ILP	CN	98.499	—	31.734	99.981	53.035
HGB-ILP	HN	64.084	90.307	99.957	35.974	33.754
Oracle-ILP	CN	98.506	—	31.739	99.988	53.039
Oracle-ILP	HN	64.084	90.307	99.957	35.974	33.754
RC-HGB-ILP	CN	98.374	—	31.694	99.854	52.968
RC-HGB-ILP	HN	64.045	90.253	99.897	35.953	33.734

Note. Utilization is reserved demand divided by scenario capacity.

4.4 Ablation, risk sensitivity, and time-resolved behavior

Removing deployment-density constraints did not change any held-out RC-HGB result. Both variants had 164 affected arrivals, 5.077% violations, cost 998,919.325 CU, and fragmentation 0.31751. The outcome does not imply that density metadata is generally irrelevant. It means that, for these demand levels and node capacities, resource and large-instance constraints already required at least as many nodes as the application-density lower bound. The ablation is useful because it localizes the active constraint set.

Table 9. Calibration and deployment-density ablation.

Policy	SLO %	GPU %	Frag.	Cost CU	Profit CU
HGB-ILP	18.978	90.307	0.31782	997,939.8	478,383.1
RC-HGB-ILP	5.077	90.253	0.31751	998,919.3	478,301.6
RC-HGB (no density)	5.077	90.253	0.31751	998,919.3	478,301.6

Changing α produced the expected cost-risk curve. The 95% nominal envelope lowered violations to 4.118% at cost 999,636.013 CU. The 80% nominal envelope raised violations to 6.563% while lowering cost to 998,462.425 CU. GPU utilization varied by less than 0.10 percentage points across these settings. The 90% setting therefore occupies a balanced interior point rather than an extreme.

Table 10. Risk-level sensitivity for RC-HGB-ILP.

α	Nominal %	Coverage %	SLO %	GPU %	Frag.	Cost CU	Profit CU
0.050	95.000	97.204	4.118	90.186	0.31728	999,636.0	477,646.9
0.100	90.000	94.030	5.077	90.253	0.31751	998,919.3	478,301.6
0.200	80.000	87.961	6.563	90.276	0.31766	998,462.4	478,662.5

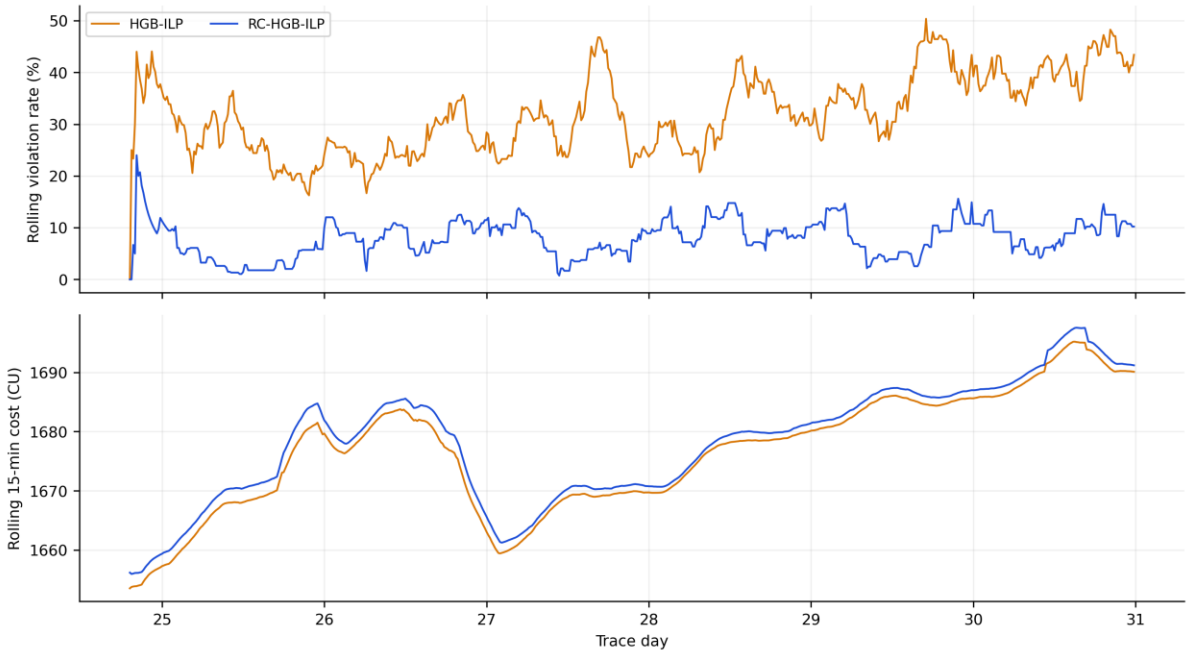

Figure 8. Time-resolved rolling violation risk and 15-minute operating cost for selected policies.

The rolling traces in Figure 8 show that cost differences were persistent but small, whereas violation differences concentrated around demand changes. Calibration did not eliminate all misses because its envelope targets marginal coverage for twelve series, not a joint guarantee that every resource in both roles is simultaneously covered. A joint event can fail when any one dimension exceeds its envelope. Moreover, the affected-arrival allocation translates an aggregate capacity shortfall into a proportional count; it is not a reconstruction of individual request queuing.

Paired inference supports the aggregate comparison. RC-HGB-ILP minus HGB-ILP had a mean epoch violation-rate difference of −0.2433, with bootstrap 95% CI [−0.2735, −0.1975] and Wilcoxon p=2.61×10^-75. The mean per-epoch cost difference was 1.6463 CU [1.2480, 2.0179], and the fragmentation difference was −0.000312 [−0.000383, −0.000237]. Relative to Reactive-BF, the mean epoch violation-rate difference was −0.0599 [−0.0708, −0.0351]. RC-HGB-ILP remained riskier and costlier than Oracle-ILP, as expected from imperfect information.

Table 11. Paired epoch comparisons for RC-HGB-ILP.

Baseline	Metric	Mean Δ	95% CI	Wilcoxon p
HGB-ILP	SLO rate	-0.243277	[-0.273489, -0.197473]	2.61e-75
HGB-ILP	Cost CU	1.6463	[1.248, 2.01794]	2.33e-98
HGB-ILP	Fragmentation	-0.000311864	[-0.00038341, -0.000236968]	2.33e-98
Oracle-ILP	SLO rate	0.0787082	[0.0660382, 0.0916475]	1.48e-25
Oracle-ILP	Cost CU	1.71563	[1.33229, 2.08423]	9.38e-99
Oracle-ILP	Fragmentation	-0.000321067	[-0.000391867, -0.0002475]	1.01e-98
RC-HGB (no density)	SLO rate	0	[0, 0]	1.000
RC-HGB (no density)	Cost CU	0	[0, 0]	1.000
RC-HGB (no density)	Fragmentation	0	[0, 0]	1.000
Reactive-BF	SLO rate	-0.059911	[-0.0707973, -0.0351194]	3.86e-16
Reactive-BF	Cost CU	1.23765	[1.00272, 1.44981]	1.98e-98
Reactive-BF	Fragmentation	-0.000233947	[-0.000274952, -0.000188798]	2.24e-98
Static-TrainPeak	SLO rate	-0.241478	[-0.272216, -0.195467]	1.04e-74
Static-TrainPeak	Cost CU	1.57992	[1.19688, 1.9548]	5.57e-98
Static-TrainPeak	Fragmentation	-0.000303063	[-0.000374615, -0.000229637]	4.15e-98

Note. Differences are RC-HGB-ILP minus the named baseline.

4.5 Memo faithfulness

The evidence-constrained memo stage produced seven summaries containing 34 numeric claims. Every stated number exactly matched its evidence card at the reported precision, every numeric claim carried an evidence identifier, and all four cross-policy comparison directions matched the signs in the saved difference record. Mean length was 28 words per memo. The comparison memo correctly reported that RC-HGB-ILP lowered the violation rate by 13.90093 percentage points and fragmentation by 0.00031, while increasing cost by 979.55 CU and decreasing profit by 81.55 CU.

Table 12. Evidence-constrained memo faithfulness audit.

Variant	Memos	Claims	Exact %	Tagged %	Direction %	Words
Evidence-constrained LLM memo	7	34	100.0	100.0	100.0	28.0

These results demonstrate the value of restricting the generation surface. They do not measure open-domain reasoning, stylistic quality, or performance on unseen evidence schemas. Numeric exact match is stringent but narrow: a memo could omit a strategically important metric and still be numerically correct. The evidence-card schema mitigates this risk by requiring five policy metrics and four cross-policy differences, while the 100% evidence-tag result makes each reported number traceable. Broader operational deployment would also test anomaly explanation, uncertainty language, and operator actionability.

4.6 Robustness and validity

The empirical creation-to-scheduled waits provide context for the 60-second scheduling threshold. Among 16,591 rows with both timestamps, 65.55% had zero wait, 8.66% exceeded 60 seconds, the 90th percentile was 59 seconds, and the 95th percentile was 136.5 seconds. Thus 60 seconds lies near a visible upper-tail boundary in the source trace. These historical waits describe Alibaba's recorded scheduler and remain separate from policy-generated activation delays.

Table 13. Descriptive creation-to-scheduled waits recorded in the source trace.

Metric	Value	Unit
Nonmissing waits	16,591	instances
Zero wait	65.55	percent
Wait > 60 seconds	8.66	percent
Mean wait	187.93	seconds
Median wait	0	seconds
P90 wait	59	seconds
P95 wait	136.50	seconds
P99 wait	1,651.50	seconds
Maximum wait	238,116	seconds

Note. These values describe recorded historical waits, not simulated policy outcomes.

Because the cold-start model assigns either zero or 120 seconds, threshold sensitivity is discrete. At thresholds of 30 or 60 seconds, policy violation rates are identical; at 120 or 180 seconds, no delay strictly exceeds the threshold. This result verifies the metric implementation and exposes the model's resolution. A richer simulator could draw activation delays from measured node-launch distributions, but no such distribution is present in the trace.

Table 14. Scheduling-SLO threshold sensitivity under the binary activation model.

Policy	30 s	60 s	120 s	180 s
HGB-ILP	18.978	18.978	0.000	0.000
Oracle-ILP	0.000	0.000	0.000	0.000
RC-HGB (no density)	5.077	5.077	0.000	0.000
RC-HGB-ILP	5.077	5.077	0.000	0.000
Reactive-BF	9.319	9.319	0.000	0.000
Static-TrainPeak	18.359	18.359	0.000	0.000

Note. Cells report the percentage of arrivals whose activation delay strictly exceeds the threshold.

Four boundaries shape interpretation. First, request fields measure reservations, so reported utilization is reserved-resource occupancy rather than device telemetry. Second, the four node types, power ratings, cost units, revenue multiplier, and SLO penalty form a controlled scenario. They support internally consistent comparisons but not a claim about Alibaba's actual hardware bill or energy use. Third, the aggregate MILP captures resource, count, density, and large-instance feasibility but omits rack topology, link contention, model-specific latency, and node failure. Fourth, all traced services are long-running disaggregated DLRMs. The findings transfer most directly to persistent services with smooth one-step demand and may differ for bursty, short-lived generative workloads.

Within those boundaries, the central result is stable. The point model failed under a level shift, persistence restored one-step accuracy, and rolling one-sided calibration converted small residual corrections into a large reduction in capacity-miss risk. The exact-demand oracle confirms that the residual violations arise from information limits rather than an infeasible node scenario. The cost and power measures then quantify the capacity price of that uncertainty protection under one fixed inventory design.

5. Conclusions

This study evaluated risk-calibrated multi-resource scheduling on the full 2025 Alibaba disaggregated DLRM trace. The method combines a persistence-guarded histogram gradient-boosting forecast, a strictly online conformal upper envelope, and a role-specific integer node-mix planner. The chronological evaluation covered 595 held-out scheduling decisions and 3,230 arrivals across twelve demand targets.

At the main 90% nominal setting, empirical target coverage reached 94.03%. RC-HGB-ILP lowered scheduling-SLO violations to 5.08%, compared with 18.98% for uncalibrated HGB-ILP, 18.36% for a static training-peak inventory, and 9.32% for reactive best-fit. The improvement cost less than 0.10% relative to HGB-ILP while retaining 90.25% GPU utilization. The density ablation produced no change because resource bounds were already dominant. Risk sensitivity showed a coherent trade-off: more conservative envelopes reduced violations and increased cost, with limited movement in GPU utilization.

The evaluation also showed why forecasting architecture matters. Raw HGB predictions did not extrapolate the later demand level, whereas a persistence floor reduced WAPE sharply for most targets. Conformal calibration then protected against remaining positive residuals. This composition was more reliable than treating a learned point estimate as a capacity commitment.

Evidence-constrained memos preserved all 34 numeric claims, evidence tags, and comparison directions in seven operational summaries. Their role is modest but useful: they turn scheduler outputs into auditable text without weakening the quantitative record. Future work can extend the planner with topology and link constraints, replace binary activation delays with measured distributions, calibrate joint multi-resource events, and test the memo protocol with broader operational questions. The present results establish a clear baseline: under trace drift, calibrated demand envelopes substantially reduce capacity-miss risk while preserving accelerator efficiency and making the operational trade-off explicit.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers

Artificial Intelligence (AI) Use Disclosure: The authors declare that no artificial intelligence tools were used in the preparation of this manuscript.

References

[1] Alibaba Group. (2025). *Alibaba cluster trace program: Traces for GPU-disaggregated deep learning recommendation models* [Data set]. GitHub. https://github.com/alibaba/clusterdata/tree/master/cluster-trace-gpu-v2025

[2] Angelopoulos, A. N., & Bates, S. (2023). Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning, 16*(4), 494–591. https://doi.org/10.1561/2200000101

[3] Bai, J., & Wu, Q. (2026). Privacy-safe marketing mix modeling and budget optimization under identifier loss: A controlled simulation study. *International Journal of Electronic Communication Systems, 6*(1). https://doi.org/10.24042/ijecs.v6i1.30533

[4] Bai, J., Chen, S., Zheng, D., & Kuo, M.-J. (2026). Interpretable attack-chain stage detection from AWS CloudTrail event sequences via linear models and HMM smoothing. *Information, Electrical and Electronics Engineering, 6*(1), 28–43. https://doi.org/10.33474/infotron.v6i1.24923

[5] Bai, J., Wang, H., Wu, Q., & Zhang, B. (2026). Privacy-robust incrementality estimation in cookieless settings via uplift modeling: Reproducible evidence from the Hillstrom E-Mail Experiment. *Journal of Technology Informatics and Engineering, 5*(1), 17–38. https://doi.org/10.51903/jtie.v5i1.468

[6] Barber, R. F., Candès, E. J., Ramdas, A., & Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability. *The Annals of Statistics, 51*(2), 816–845. https://doi.org/10.1214/23-AOS2276

[7] Chang, X., Lu, Y., & Zhong, Z. S. (2026). Review-grounded explainable recommendation with faithfulness evaluation on Amazon Reviews. *Journal of Electrical Engineering and Computer Science, 11*(1), 9–22. https://doi.org/10.54732/jeecs.v11i1.2

[8] Chekuri, C., & Khanna, S. (2004). On multidimensional packing problems. *SIAM Journal on Computing, 33*(4), 837–851. https://doi.org/10.1137/S0097539799356265

[9] Chen, S., He, S., & Sun, E. (2024). Risk-bounded GPU resource oversubscription via conformal demand envelopes in production AI clusters. *Journal of Advanced Computing Systems, 4*(5), 119–134. https://doi.org/10.69987/JACS.2024.40509

[10] Chen, Y., & Xu, H. (2026). Trust-calibrated multilingual RAG for humanitarian information platforms: Empirical evaluation on OMoS-QA for migration information access. *International Journal of Graphic Design, 4*(1), 141–164. https://doi.org/10.51903/ijgd.v4i1.3552

[11] Chen, Y., Shetty, M., Somashekar, G., Ma, M., Simmhan, Y., Mace, J., Bansal, C., Wang, R., & Rajmohan, S. (2025). AIOpsLab: A holistic framework to evaluate AI agents for enabling autonomous clouds. *Proceedings of Machine Learning and Systems, 7*. https://proceedings.mlsys.org/paper_files/paper/2025/hash/d1f9e4a9f109b6e8b75ed362736f22ec-Abstract-Conference.html

[12] Chen, Y., Zhang, Y., & Sherman, M. (2024). Going concern and bankruptcy prediction under extreme class imbalance: Cost-sensitive learning, resampling, and focal loss with explainable financial-ratio portraits. *Journal of Advanced Computing Systems, 4*(4), 80–96. https://doi.org/10.69987/JACS.2024.40407

[13] Chen, Y., Zhang, Y., Chau, D., & Sherman, M. (2023). Credit card default risk tiering with probability calibration and uncertainty-driven rejection: A reproducible study on the UCI Credit Card Clients Dataset. *Journal of Advanced Computing Systems, 3*(4), 31–47. https://doi.org/10.69987/JACS.2023.30403

[14] Chen, Y., Zhou, S., & Lin, E. (2025). Accounting-aware evidence retrieval for institutional due diligence of tokenized trade receivable RWA. *Journal of Technology Informatics and Engineering, 4*(3), 649–663. https://doi.org/10.51903/jtie.v4i3.542

[15] Crankshaw, D., Sela, G.-E., Mo, X., Zumar, C., Stoica, I., Gonzalez, J. E., & Tumanov, A. (2020). InferLine: Latency-aware provisioning and scaling for prediction serving pipelines. In *Proceedings of the 11th ACM Symposium on Cloud Computing* (pp. 477–491). Association for Computing Machinery. https://doi.org/10.1145/3419111.3421285

[16] Delimitrou, C., Bambos, N., & Kozyrakis, C. (2013). QoS-aware admission control in heterogeneous datacenters. In *Proceedings of the 10th International Conference on Autonomic Computing* (pp. 291–296). USENIX Association. https://www.usenix.org/conference/icac13/technical-sessions/presentation/delimitrou

[17] Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAs: Automated evaluation of retrieval augmented generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations* (pp. 150–158). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.eacl-demo.16

[18] Gao, L., Dai, Z., Pasupat, P., Chen, A., Chaganty, A. T., Fan, Y., Zhao, V., Lao, N., Lee, H., Juan, D.-C., & Guu, K. (2023). RARR: Researching and revising what language models say, using language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 16477–16508). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.acl-long.910

[19] Gao, T., Yen, H., Yu, J., & Chen, D. (2023). Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 6465–6488). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.398

[20] Ghodsi, A., Zaharia, M., Hindman, B., Konwinski, A., Shenker, S., & Stoica, I. (2011). Dominant resource fairness: Fair allocation of multiple resource types. In *Proceedings of the 8th USENIX Symposium on Networked Systems Design and Implementation*. USENIX Association. https://www.usenix.org/conference/nsdi11/dominant-resource-fairness-fair-allocation-multiple-resource-types

[21] Gibbs, I., & Candès, E. (2021). Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems, 34*, 1660–1672. https://proceedings.neurips.cc/paper/2021/hash/0d441de75945e5acbc865406fc9a2559-Abstract.html

[22] Grandl, R., Ananthanarayanan, G., Kandula, S., Rao, S., & Akella, A. (2014). Multi-resource packing for cluster schedulers. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (pp. 455–466). Association for Computing Machinery. https://doi.org/10.1145/2619239.2626334

[23] Gu, J., Chowdhury, M., Shin, K. G., Zhu, Y., Jeon, M., Qian, J., Liu, H., & Guo, C. (2019). Tiresias: A GPU cluster manager for distributed deep learning. In *Proceedings of the 16th USENIX Symposium on Networked Systems Design and Implementation* (pp. 485–500). USENIX Association. https://www.usenix.org/conference/nsdi19/presentation/gu

[24] Gujarati, A., Karimi, R., Alzayat, S., Hao, W., Kaufmann, A., Vigfusson, Y., & Mace, J. (2020). Serving DNNs like Clockwork: Performance predictability from the bottom up. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation* (pp. 443–462). USENIX Association. https://www.usenix.org/conference/osdi20/presentation/gujarati

[25] Harlap, A., Tumanov, A., Chung, A., Ganger, G. R., & Gibbons, P. B. (2017). Proteus: Agile ML elasticity through tiered reliability in dynamic resource markets. In *Proceedings of the 12th European Conference on Computer Systems* (pp. 589–604). Association for Computing Machinery. https://doi.org/10.1145/3064176.3064182

[26] He, S., Chang, X., & Sun, E. (2024). Cross-cloud transfer learning for AI training capacity forecasting under workload and topology distribution shift. *Journal of Advanced Computing Systems, 4*(1), 100–120. https://doi.org/10.69987/JACS.2024.40108

[27] He, S., Li, C., & Rao, H. (2025). Few-shot cold-start workload forecasting for new AI inference tenants with time-series foundation models. *Journal of Technology Informatics and Engineering, 4*(1), 306–324. https://doi.org/10.51903/jtie.v4i1.546

[28] He, S., Nie, J., & Li, C. (2026). Power-aware inventory planning for AI infrastructure using job-level forecasting and LLM workload explanations. *Journal of Technology Informatics and Engineering, 5*(1), 341–359. https://doi.org/10.51903/jtie.v5i1.548

[29] He, S., Tu, H., & Liu, I. (2023). Safe PD capacity forecasting with time-series foundation models and calibrated uncertainty for heterogeneous GPU clusters. *Journal of Advanced Computing Systems, 3*(4), 48–66. https://doi.org/10.69987/JACS.2023.30404

[30] Huang, Y., Yang, Z., Xing, J., Dai, Y., Qiu, Y., Wu, D., Lai, F., & Chen, A. (2024). A disaggregation approach to embedding recommendation systems. *arXiv*. https://doi.org/10.48550/arXiv.2410.12794

[31] Jin, J. (2025a). Calibrated resume-job matching for trustworthy LLM-assisted recruiter screening: Pairwise matching, probability calibration, and selective refusal on two public recruitment datasets. *Journal of Technology Informatics and Engineering, 4*(3), 625–648. https://doi.org/10.51903/jtie.v4i3.529

[32] Jin, J. (2025b). Evidence-chain reliable RAG: Hallucination detection, source attribution, and deterministic provenance explanations. *Journal of Technology Informatics and Engineering, 4*(2), 520–533. https://doi.org/10.51903/jtie.v4i2.535

[33] Jin, J. (2025c). LLM-style evidence cards for scientific search interfaces: A UI/UX design framework for retrieval transparency, ranking trust, and visual evidence hierarchy. *International Journal of Graphic Design, 3*(2), 397–414. https://doi.org/10.51903/ijgd.v3i2.3698

[34] Jin, J., Huang, T., & Lu, S. (2024a). A model-risk-friendly probability of default workflow: Calibration, distribution-free uncertainty quantification, and SHAP explanations on the UCI Credit Card Default Dataset. *Journal of Advanced Computing Systems, 4*(6), 74–85. https://doi.org/10.69987/JACS.2024.40606

[35] Jin, J., Huang, T., & Lu, S. (2024b). Cost-sensitive learning, simulated PU learning, and one-class autoencoding for extreme-imbalance credit card fraud detection. *Journal of Advanced Computing Systems, 4*(6), 64–73. https://doi.org/10.69987/JACS.2024.40605

[36] Johnson, D. S., Demers, A., Ullman, J. D., Garey, M. R., & Graham, R. L. (1974). Worst-case performance bounds for simple one-dimensional packing algorithms. *SIAM Journal on Computing, 3*(4), 299–325. https://doi.org/10.1137/0203025

[37] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154. https://proceedings.neurips.cc/paper/6907-lightgbm-a-highly-efficient-gradient-boosting-decision-tree

[38] Kuo, M.-J., Zheng, D., & Hires, J. (2025). Federated topic-preference learning for knowledge-grounded chat with differential privacy. *Journal of Technology Informatics and Engineering, 4*(2), 385–401. https://doi.org/10.51903/jtie.v4i2.502

[39] Li, C., Bai, J., & Wang, S. (2024). Evidence-chain reliable RAG: Word-level hallucination detection, source attribution, and provenance explanation for LLM applications. *Journal of Advanced Computing Systems, 4*(2), 76–92. https://doi.org/10.69987/JACS.2024.40207

[40] Li, C., Liu, G., & Zhao, Z. (2026). Cost-aware LLM-style routing for AIOps log analysis: Log parsing, anomaly detection, fault diagnosis, and incident summarization on LogEval task files. *Journal of Technology Informatics and Engineering, 5*(2), 91–103. https://doi.org/10.51903/jtie.v5i2.538

[41] Li, J., & Zhou, A. (2026). Multi-regulation RAG for AI product counsel: A legal governance framework for cross-border digital commerces. *Rule of Law Studies Journal, 2*(2), 105–123. https://doi.org/10.64780/rolsj.v2i2.225

[42] Li, Y. (2024). Findable then explainable: Retrieval–summary integration for code intelligence on a lightweight CodeSearchNet subset. *Journal of Advanced Computing Systems, 4*(7), 65–82. https://doi.org/10.69987/JACS.2024.40706

[43] Li, Y., & Lu, S. (2025). Language-guided feature selection for DDoS and intrusion detection on CICIDS2017. *Journal of Technology Informatics and Engineering, 4*(1), 284–305. https://doi.org/10.51903/jtie.v4i1.531

[44] Li, Y., Lu, S., & Zhao, L. (2025). LLM-as-design-critic: Aligning AI-generated UI feedback with human graphic design judgment. *International Journal of Graphic Design, 3*(1), 196–215. https://doi.org/10.51903/ijgd.v3i1.3661

[45] Li, Z., Zhang, K., & Wong, A. (2026). Numerical-reasoning guardrails for a quant research assistant: A compact reproducible benchmark using SEC and FRED data. *Journal of Technology Informatics and Engineering, 5*(2), 75–90. https://doi.org/10.51903/jtie.v5i2.541

[46] Li, Z., Zhou, S., & Zhou, Z. (2025). Financial risk dashboard design for institutional RWA investors: Visual hierarchy, chart comprehension, and explainability in FinChart-Bench. *International Journal of Graphic Design, 3*(1), 196–210. https://doi.org/10.51903/ijgd.v3i1.3715

[47] Liu, G., He, S., & Liu, I. (2023). LLM-augmented multi-source root cause attribution for CPU and network faults in microservices. *Journal of Advanced Computing Systems, 3*(6), 39–57. https://doi.org/10.69987/JACS.2023.30604

[48] Liu, G., He, S., & Wong, H. (2025). LLM-compatible visual brief cards for AI infrastructure capacity dashboards: A UI/UX framework for turning forecast risk into graphic design decisions. *International Journal of Graphic Design, 3*(1), 196–213. https://doi.org/10.51903/ijgd.v3i1.3723

[49] Liu, G., Li, C., & Zhang, E. (2024). OpsLLM for cloud incident triage: Bilingual RAG-based root cause analysis and alert summarization for AI infrastructure operations. *Journal of Advanced Computing Systems, 4*(4), 97–111. https://doi.org/10.69987/JACS.2024.40408

[50] Liu, X., Zhao, Y., Liu, S., Li, X., Zhu, Y., Liu, X., & Jin, X. (2024). MuxFlow: Efficient GPU sharing in production-level clusters with more than 10000 GPUs. *Science China Information Sciences, 67*, Article 222101. https://doi.org/10.1007/s11432-024-4227-2

[51] Lu, S., & Zhou, D. (2024). TinyLLM-assisted intrusion detection for real-time IoT networks. *Journal of Advanced Computing Systems, 4*(8), 72–87. https://doi.org/10.69987/JACS.2024.40809

[52] Lu, Y., Zhou, H., & Zhang, Y. (2025). A constrained, data-driven budgeting framework integrating macro demand forecasting and marketing response modeling. *Journal of Technology Informatics and Engineering, 4*(3), 493–520. https://doi.org/10.51903/jtie.v4i3.466

[53] Mao, H., Alizadeh, M., Menache, I., & Kandula, S. (2016). Resource management with deep reinforcement learning. In *Proceedings of the 15th ACM Workshop on Hot Topics in Networks* (pp. 50–56). Association for Computing Machinery. https://doi.org/10.1145/3005745.3005750

[54] Mao, H., Schwarzkopf, M., Venkatakrishnan, S. B., Meng, Z., & Alizadeh, M. (2019). Learning scheduling algorithms for data processing clusters. In *Proceedings of the ACM Special Interest Group on Data Communication* (pp. 270–288). Association for Computing Machinery. https://doi.org/10.1145/3341302.3342080

[55] Meng, S., Chen, J., & Zheng, I. (2026). LLM-inspired offline reranking for financial search: Query rewriting, hybrid retrieval, and listwise relevance ranking on FiQA. *Journal of Technology Informatics and Engineering, 5*(1), 361–378. https://doi.org/10.51903/jtie.v5i1.537

[56] Min, S., Krishna, K., Lyu, X., Lewis, M., Yih, W.-t., Koh, P. W., Iyyer, M., Zettlemoyer, L., & Hajishirzi, H. (2023). FActScore: Fine-grained atomic evaluation of factual precision in long-form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing* (pp. 12076–12100). Association for Computational Linguistics. https://doi.org/10.18653/v1/2023.emnlp-main.741

[57] Mu, J., Lu, Y., & Hwang, E. (2026). Structured visual brief interfaces for advertising design: A UI/UX framework for turning creative intentions into designer-editable graphic design cards. *International Journal of Graphic Design, 4*(1), 192–208. https://doi.org/10.51903/ijgd.v4i1.3702

[58] Mu, J., Lu, Y., & Smith, M. (2023). LLM-assisted incrementality (uplift) modeling for email advertising: From feature interactions to interpretable audience–creative–channel policies. *Journal of Advanced Computing Systems, 3*(1), 31–48. https://doi.org/10.69987/JACS.2023.30103

[59] Mu, J., Ye, T., & Patel, P. (2025). Offline counterfactual evaluation for advertising and recommendation slot policies: A reproducible study on the Open Bandit Dataset (Small). *Journal of Technology Informatics and Engineering, 4*(3), 521–543. https://doi.org/10.51903/jtie.v4i3.500

[60] Narayanan, D., Santhanam, K., Kazhamiaka, F., Phanishayee, A., & Zaharia, M. (2020). Heterogeneity-aware cluster scheduling policies for deep learning workloads. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation* (pp. 481–498). USENIX Association. https://www.usenix.org/conference/osdi20/presentation/narayanan-deepak

[61] Nie, J., & Zheng, D. (2024). Noisy-neighbor-aware VM degradation risk modeling with unsupervised residual fusion. *Journal of Advanced Computing Systems, 4*(4), 112–123. https://doi.org/10.69987/JACS.2024.40409

[62] Nie, J., Liu, G., Li, C., & Zou, T. (2026). Evidence-constrained incident visualization cards for distributed cloud logs: A UI/UX framework for turning Hadoop, OpenStack, and ZooKeeper logs into actionable SRE design interfaces. *International Journal of Graphic Design, 4*(1), 179–185. https://doi.org/10.51903/ijgd.v4i1.3703

[63] Patel, P., Choukse, E., Zhang, C., Shah, A., Goiri, Í., Maleki, S., & Bianchini, R. (2024). Splitwise: Efficient generative LLM inference using phase splitting. In *2024 ACM/IEEE 51st Annual International Symposium on Computer Architecture*. IEEE. https://doi.org/10.1109/ISCA59077.2024.00019

[64] Qiu, H., Mao, W., Patke, A., Cui, S., Jha, S., Wang, C., Franke, H., Kalbarczyk, Z., Başar, T., & Iyer, R. K. (2024). Power-aware deep learning model serving with μ-Serve. In *2024 USENIX Annual Technical Conference* (pp. 75–93). USENIX Association. https://www.usenix.org/conference/atc24/presentation/qiu

[65] Romano, Y., Patterson, E., & Candès, E. J. (2019). Conformalized quantile regression. *Advances in Neural Information Processing Systems, 32*, 3543–3553. https://proceedings.neurips.cc/paper/2019/hash/5103c3584b063c431bd1268e9b5e76fb-Abstract.html

[66] Su, W., Chen, S., & Qian, E. (2026). Narrative-aware scientific claim verification agent with evidence ranking for ClimateCheck. *Journal of Technology Informatics and Engineering, 5*(1), 327–340. https://doi.org/10.51903/jtie.v5i1.549

[67] Su, W., Chen, S., & Zhao, C. (2025). Budgeted multi-hop retrieval agent for compositional question answering: A retrieval-policy evaluation on the official MultiHop-RAG benchmark. *Journal of Technology Informatics and Engineering, 4*(3), 649–662. https://doi.org/10.51903/jtie.v4i3.543

[68] Su, W., Rao, H., & Ma, E. (2026). Privacy and data-integrity risk cards for LLM agents: A UI/UX design framework for secure human oversight under prompt-injection attacks. *International Journal of Graphic Design, 4*(1), 186–191. https://doi.org/10.51903/ijgd.v4i1.3699

[69] Sun, X., Lu, Y., & Chen, J. (2023). Controllable long-term user memory for multi-session dialogue: Confidence-gated writing, time-aware retrieval-augmented generation, and update/forgetting. *Journal of Advanced Computing Systems, 3*(8), 9–24. https://doi.org/10.69987/JACS.2023.30802

[70] Sun, X., Zhong, Z. S., & Wu, Q. (2026). Retrieval-grounded HDFS log anomaly detection and deterministic failure narrative generation. *Journal of Computational Systems and Applications, 3*(1), 15–30. https://doi.org/10.64229/j6d7fr94

[71] Tu, H., Zhao, S., & Zhou, A. (2025). Visual brief cards for advertising design: A structured UI/UX framework for turning creative intentions into graphic design decisions. *International Journal of Graphic Design, 3*(1), 210–226. https://doi.org/10.51903/ijgd.v3i1.3714

[72] Wang, B., He, Y., Shui, Z., Xin, Q., & Lei, H. (2024). Predictive optimization of DDoS attack mitigation in distributed systems using machine learning. In *Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024)* (pp. 89–94).

[73] Wang, C., Wen, Z., Zhang, R., Xu, P., & Jiang, Y. (2025). GPU memory requirement prediction for deep learning task based on bidirectional gated recurrent unit optimization transformer. In *2025 5th International Conference on Artificial Intelligence, Virtual Reality and Visualization (AIVRV)*. IEEE. https://doi.org/10.1109/AIVRV67401.2025.11350369

[74] Weng, Q., Yang, L., Yu, Y., Wang, W., Tang, X., Yang, G., & Zhang, L. (2023). Beware of fragmentation: Scheduling GPU-sharing workloads with fragmentation gradient descent. In *2023 USENIX Annual Technical Conference* (pp. 995–1008). USENIX Association. https://www.usenix.org/conference/atc23/presentation/weng

[75] Wu, C.-J., Raghavendra, R., Gupta, U., Acun, B., Ardalani, N., Maeng, K., Chang, G., Aga, F., Huang, J., Bai, C., Gschwind, M., Gupta, A., Ott, M., Melnikov, A., Candido, S., Brooks, D., Chauhan, G., Lee, B., Lee, H.-H. S., . . . Hazelwood, K. (2022). Sustainable AI: Environmental implications, challenges and opportunities. *Proceedings of Machine Learning and Systems, 4*, 795–813. https://proceedings.mlsys.org/paper_files/paper/2022/hash/462211f67c7d858f663355eff93b745e-Abstract.html

[76] Wu, Q., Meng, S., & Zhao, J. (2025). Text-grounded LLM-assisted design rationale interfaces: Turning advertising layout metadata into explainable UI/UX decision cards. *International Journal of Graphic Design, 3*(1), 216–240. https://doi.org/10.51903/ijgd.v3i1.3713

[77] Xiao, W., Ren, S., Li, Y., Zhang, Y., Hou, P., Li, Z., Feng, Y., Lin, W., & Jia, Y. (2020). AntMan: Dynamic scaling on GPU clusters for deep learning. In *Proceedings of the 14th USENIX Symposium on Operating Systems Design and Implementation* (pp. 533–548). USENIX Association. https://www.usenix.org/conference/osdi20/presentation/xiao

[78] Xin, Q. (2025a). Explaining OpenStack failure-injection log anomalies with retrieved normal prototypes. *Emerging Information Science and Technology, 6*(2), 125–146. https://doi.org/10.18196/eist.v6i2.31232

[79] Xin, Q. (2025b). Hybrid cloud architecture for efficient and cost-effective large language model deployment. *Journal of Information Systems and Informatics, 7*(3), 2182–2195. https://doi.org/10.51519/journalisi.v7i3.1170

[80] Xin, Q. (2025c). Uncertainty-aware late fusion for 3D perception (confidence calibration + fusion rule learning). *Journal of Technology Informatics and Engineering, 4*(1), 215–238. https://doi.org/10.51903/jtie.v4i1.485

[81] Xin, Q. (2026a). Behavior retrieval plus response generation for interpretable conversational personalized recommendation. *IJEEPSE, 9*(2), 120–136. https://doi.org/10.31258/ijeepse.9.2.120-136

[82] Xin, Q. (2026b). Explainable and fair credit risk scoring with counterfactual explanations: A reproducible evaluation on the German Credit Dataset (HELOC-motivated). *Journal of Information Technology, 14*(2). https://doi.org/10.32664/j-intech.v14i02.2228

[83] Xin, Q. (2026c). Host-based intrusion detection with system call sequences: Window localization and forensic narratives. *AVITEC, 8*(2). https://doi.org/10.28989/avitec.v8i2.3973

[84] Xin, Q. (2026d). Log anomaly detection with conformal alert control and evidence-grounded incident ticket generation. *AVITEC, 8*(2), 247. https://doi.org/10.28989/avitec.v8i2.3974

[85] Xin, Q. (2026e). Probabilistic bike-sharing demand forecasting under changing weather and seasonal regimes with transformer-based models. *Transport Findings*. https://doi.org/10.32866/001c.157499

[86] Xin, Q. (2026f). Self-supervised customer representation learning for segmentation and next-purchase prediction on UCI Online Retail. *Journal of Information Technology, 14*(1). https://doi.org/10.32664/j-intech.v14i01.2229

[87] Xin, Q. (2026g). Self-supervised log anomaly detection with LogBERT-style transformers: Full empirical evaluation on a reproducible SynHDFS benchmark. *Journal of Electrical Engineering and Computer Science, 11*(1), 23–35. https://doi.org/10.54732/jeecs.v11i1.3

[88] Xin, Q., Xu, Z., Guo, L., Zhao, F., & Wu, B. (2024). IoT traffic classification and anomaly detection method based on deep autoencoders. In *Proceedings of the 6th International Conference on Computing and Data Science (CDS 2024)*.

[89] Xu, H., Chen, Y., & Med, A. (2025). Automatic detection and explanation of dark patterns from interface microcopy: Empirical comparison of BERT-style encoders, RoBERTa-style encoders, and LLM-style decoders on the ec-darkpattern dataset. *Journal of Technology Informatics and Engineering, 4*(3), 590–612. https://doi.org/10.51903/jtie.v4i3.491

[90] Xu, K., Zhou, H., Zheng, H., Zhu, M., & Xin, Q. (2024). Intelligent classification and personalized recommendation of e-commerce products based on machine learning. In *Proceedings of the 6th International Conference on Computing and Data Science (ICCDS 2024)*.

[91] Yang, L., Wang, Y., Yu, Y., Weng, Q., Dong, J., Liu, K., Zhang, C., Zi, Y., Li, H., Zhang, Z., Wang, N., Dong, Y., Zheng, M., Xi, L., Lu, X., Ye, L., Yang, G., Fu, B., Lan, T., . . . Wang, W. (2025). GPU-disaggregated serving for deep learning recommendation models at scale. In *Proceedings of the 22nd USENIX Symposium on Networked Systems Design and Implementation* (pp. 847–863). USENIX Association. https://www.usenix.org/conference/nsdi25/presentation/yang

[92] Ye, T., Mu, J., & Hunter, J. (2026). Off-policy evaluation and conservative policy selection for slot-level dynamic bidding and ranking on the Open Bandit Dataset (Small). *Journal of Technology Informatics and Engineering, 5*(1), 178–199. https://doi.org/10.51903/jtie.v5i1.503

[93] You, J., Chung, J.-W., & Chowdhury, M. (2023). Zeus: Understanding and optimizing GPU energy consumption of DNN training. In *Proceedings of the 20th USENIX Symposium on Networked Systems Design and Implementation* (pp. 119–139). USENIX Association. https://www.usenix.org/conference/nsdi23/presentation/you

[94] Yu, P., & Chowdhury, M. (2020). Salus: Fine-grained GPU sharing primitives for deep learning applications. *Proceedings of Machine Learning and Systems, 2*, 98–111. https://proceedings.mlsys.org/paper_files/paper/2020/hash/d9cd83bc91b8c36a0c7c0fcca59228f2-Abstract.html

[95] Zhang, B., Rao, H., & Zhao, D. (2024). Evidence-grounded RAG for cloud-native DevOps: Hallucination-resistant AIOps question answering over private operations documents. *Journal of Advanced Computing Systems, 4*(3), 109–125. https://doi.org/10.69987/JACS.2024.40308

[96] Zhang, B., Ren, Y., & Zou, J. (2025). LLM-style explainable e-commerce recommendation cards: A UI/UX design framework for trust-calibrated product recommendation. *International Journal of Graphic Design, 3*(2), 381–396. https://doi.org/10.51903/ijgd.v3i2.3697

[97] Zhang, K., Chen, Y., & Qian, A. (2025). Evidence-grounded accounting disclosure review cards: A visual communication framework for LLM-style explanations over SEC financial statements and notes. *International Journal of Graphic Design, 3*(2), 395. https://doi.org/10.51903/ijgd.v3i2.3710

[98] Zhang, R., Wen, Z., Wang, C., Tang, C., Xu, P., & Jiang, Y. (2025). Quality analysis and evaluation prediction of RAG retrieval based on machine learning algorithms. *arXiv*. https://doi.org/10.48550/arXiv.2511.19481

[99] Zhao, S., Bai, J., & Roberson, D. (2025). Multi-horizon GPU demand forecasting with workload semantics and operational risk curves: An empirical study on Alibaba Clusterdata GPU Trace. *Journal of Technology Informatics and Engineering, 4*(3), 544–571. https://doi.org/10.51903/jtie.v4i3.498

[100] Zhao, S., Ren, Y., & Chang, X. (2026). Profit-aware spot GPU admission control with cost-sensitive loss and evidence-grounded policy memos for AI workload supply-demand matching. *Journal of Technology Informatics and Engineering, 5*(2), 45–59. https://doi.org/10.51903/jtie.v5i2.545

[101] Zheng, D., & Li, C. (2024). Behavior-level jailbreak resistance via multi-stage refusal + utility preservation. *Journal of Advanced Computing Systems, 4*(1), 83–99. https://doi.org/10.69987/JACS.2024.40107

[102] Zheng, D., Li, C., & Davidson, H. (2023). Continual red-teaming for in-the-wild jailbreaks via online guardrail updates and guardrail distillation. *Journal of Advanced Computing Systems, 3*(2), 35–49. https://doi.org/10.69987/JACS.2023.30203

[103] Zheng, D., Zhang, B., & Geibel, J. (2024). VerifySafe: Toxicity-safe agent responses under adversarial prompts with evidence-based self-verification. *Journal of Advanced Computing Systems, 4*(1), 67–82. https://doi.org/10.69987/JACS.2024.40106

[104] Zhong, Y., Liu, S., Chen, J., Hu, J., Zhu, Y., Liu, X., Jin, X., & Zhang, H. (2024). DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation* (pp. 193–210). USENIX Association. https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin

[105] Zhong, Z. S., & Ling, S. (2024). Uncertainty quantification of spectral estimator and MLE for orthogonal group synchronization. *arXiv*. https://arxiv.org/abs/2408.05944

[106] Zhong, Z. S., Chen, J., Zhong, E., & Sun, X. (2025). Evidence-calibrated RAG for unanswerable question answering: Retrieval coverage, abstention calibration, and hallucination-proxy analysis on SQuAD 2.0. *Journal of Technology Informatics and Engineering, 4*(2), 502–520. https://doi.org/10.51903/jtie.v4i2.536

[107] Zhong, Z. S., Li, C., & Rao, H. (2026). Trajectory reliability prediction for generalist AI agents: Tool-use failure analysis and success forecasting on ZClawBench. *Journal of Technology Informatics and Engineering, 5*(1), 341–360. https://doi.org/10.51903/jtie.v5i1.539

[108] Zhong, Z. S., Pan, X., & Lei, Q. (2025). Bridging domains with approximately shared features. In *Proceedings of the 28th International Conference on Artificial Intelligence and Statistics* (Vol. 258, pp. 559–567). PMLR. https://proceedings.mlr.press/v258/zhong25a.html

[109] Zhou, H., & Zhang, K. (2025). News-based uncertainty and macro-market fusion for VIX direction forecasting: Evidence from 2015–2024 FRED panel. *Journal of Technology Informatics and Engineering, 4*(2), 487–501. https://doi.org/10.51903/jtie.v4i2.540

[110] Zhou, S., Chen, Y., & Lee, K. (2026). Accounting-aware evidence-constrained agents for disclosure, settlement, and secondary-market risk monitoring in tokenized. *Journal of Technology Informatics and Engineering, 5*(2), 60–74. https://doi.org/10.51903/jtie.v5i2.544