Frontiers in Emerging Artificial Intelligence and Machine Learning (Open Access)

Volume 02, Issue 12, December 2025,
DOI: https://doi.org/10.64917/feaiml/Volume02Issue12-06 PageNo.55-71

Quantum-Inspired AI Techniques for Optimizing Cloud Resource Allocation in Hybrid Architectures

Pavan Kumar Asu
Independent Researcher, Boston, MA, USA

RECEIVED - 12-09-2025, **RECEIVED REVISED VERSION** - 12-15-2025, **ACCEPTED**- 12-18-2025, **PUBLISHED**- 12-19-2025

Abstract

Hybrid cloud environments face persistent challenges in efficient resource allocation due to workload variability, SLA (Service Level Agreement) constraints, and the heterogeneity of infrastructure. This paper introduces a Quantum-Inspired AI (QIAI) model that leverages probabilistic scheduling logic, specifically simulating amplitude-guided exploration and tunneling-based state transitions, to enhance decision-making in complex scheduling scenarios. Implemented using classical hardware, the model is evaluated against traditional heuristics (Round-Robin, Least-Loaded First) and Q-Learning approaches using real-world workload traces from Google, Azure, and Alibaba. Experimental results demonstrate that the proposed model achieves a resource utilization rate exceeding 80% and reduces SLA violations to less than 6%, significantly outperforming the 18.5% violation rate of baseline heuristics. Furthermore, the model exhibits superior efficiency, achieving convergence 35–40% faster than standard reinforcement learning agents. The findings suggest that quantum-inspired models can offer practical and scalable advantages in modern cloud computing systems without the need for actual quantum hardware.

Keywords: quantum-inspired AI, cloud resource allocation, hybrid cloud scheduling, reinforcement learning, workload orchestration, Grover search, SLA optimization, cloud computing, edge-cloud systems, probabilistic scheduling

1. Introduction

Cloud computing has evolved into a foundational pillar of modern digital infrastructure, enabling scalable, on-demand access to computing resources across a variety of industries (Amajuoyi et al., 2024). With the increasing complexity and scale of enterprise workloads, hybrid cloud architectures, which integrate public cloud services, private data centers, and edge nodes, have emerged as a powerful paradigm for balancing performance, security, and cost (Islam, 2024). These systems support diverse applications ranging from real-time data analytics to machine learning pipelines and Internet-of-Things (IoT) services. However, managing such heterogeneous and distributed infrastructures introduces significant operational complexity, particularly in terms of resource allocation (Vaish et al., 2022). The sheer scale and heterogeneity of these workloads, involving billions of allocation decisions daily across varied hardware, makes traditional optimization methods increasingly challenging.

Tasks in a hybrid cloud environment need to be dynamically scheduled to a range of computational resources with different performance properties, costs, and availability properties (Zibitsker & Lupersolsky, 2025). Maintaining an optimal distribution of resources to ensure that service-level agreements (SLAs) are met, resources are used to maximum capacity, and the system remains responsive is a complex and continuous challenge. This problem is complicated by the fact that the workload is uncertain, the latency may be sensitive, and real-time orchestration is needed among the geographically localized nodes (Babu et al., 2024).

In the last ten years, researchers and practitioners have resorted to the use of artificial intelligence (AI) methods to solve this dilemma. The methods of allocating resources that have been identified as promising regarding the ability to alter the environment are AI-based approaches, including heuristic algorithms, supervised learning, and reinforcement learning (Barua & Kaiser, 2024). Nevertheless, these approaches remain essentially limited by classical computational paradigms, which are typically unable to generalize in high-dimensional, nonlinear and uncertain scheduling spaces. In practice, on highly variable workloads, reinforcement-based learning algorithms can slow down or stall at local optima whereas heuristic approaches can be very inflexible to new conditions or constraints. Moreover, lots of models are prone to scalability bottlenecks and need lots of tuning to operate safely in a production-scale hybrid system (Hummaida et al., 2022).

With the aim of addressing these shortcomings, this paper presents a new quantum-inspired AI (QIAI) framework that is capable of improving the allocation of resources in hybrid cloud settings. Instead of using very costly physical quantum hardware, which is experimentally constrained, our model implements important ideas in quantum mechanics in a classical-computation model. These are amplitude-based probabilistic reasoning, quantum tunneling-inspired transitions of states, and Grover-like search algorithms. The resulting system creates a kind of clever scheduling which is more likely to explore complex action spaces, does not prematurely converge, and allocates tasks within hybrid infrastructure more efficiently.

In contrast to other traditional optimization methods, our QIAI agents leverage quantum-inspired behaviors to dynamically trade-off exploration and exploitation, which is essential to cope with stochasticity of real workloads. The following is one example, where the agent is able to model and evaluate all available allocation paths simultaneously with the assistance of amplitude-based models, thereby, making informed decisions when it is hard to predict. Equally, tunneling-based transitions can enable the agent to escape local minima in the resource allocation space, particularly when demand spikes vary or when there are unexpected failures (Selvam et al., 2025).

Despite advances in AI-based scheduling techniques, existing methods struggle to efficiently navigate the high-dimensional, probabilistic state spaces of hybrid cloud environments, often converging slowly or becoming trapped in local optima under dynamic workload conditions. This research addresses the fundamental question: Can quantum-inspired probabilistic reasoning mechanisms—specifically amplitude-guided exploration and tunneling-based transitions—deployed on classical hardware, significantly outperform traditional heuristics and reinforcement learning approaches in resource utilization, SLA compliance, and convergence speed for hybrid cloud resource allocation?

The study is novel and significant as it integrates the concepts of quantum computing, AI scheduling algorithms, and cloud infrastructure design in an inter-disciplinary way. We test the feasibility and performance of quantum-inspired scheduling agents using a realistic simulation of a hybrid environment, which is driven by real-world workload traces from public cloud, private cloud, and edge computing nodes. We do not need any specialized hardware and run our approach fully in Python based on publicly available datasets and open-source tools, hence making it fully reproducible and accessible.

Key Contributions

This research makes the following contributions:

1. A novel quantum-inspired AI framework tailored for dynamic resource allocation in hybrid cloud environments, incorporating amplitude-based and tunneling-inspired reasoning.

2. A comprehensive simulation platform, replicating realistic hybrid architectures using workload traces from Google, Microsoft Azure, and Alibaba.

3. Rigorous performance evaluation comparing the quantum-inspired agent against traditional AI and baseline heuristics across key cloud metrics (SLA compliance, resource utilization, and convergence speed).

4. An analysis of applicability and trade-offs, examining where quantum-inspired models deliver the most value, and discussing potential integration with real-world platforms like Kubernetes and AWS Outposts.

This work aligns with emerging trends at the intersection of

quantum computing and intelligent systems engineering, contributing to the ongoing evolution of cloud operations. By demonstrating that quantum-inspired logic can be operationalized using classical infrastructure, we open new pathways for scalable and intelligent cloud management systems that are both high-performing and practically deployable.

2. Related Work

This section reviews relevant literature across three primary domains: (i) AI-based methods for cloud resource allocation, (ii) scheduling and orchestration in hybrid cloud systems, and (iii) quantum-inspired optimization techniques for cloud environments. By analyzing developments in each area, we highlight the current limitations and establish the need for practically deployable, quantum-inspired AI models in hybrid cloud resource scheduling.

2.1 AI-Based Cloud Resource Allocation

Artificial intelligence (AI) is widely used for cloud resource management, primarily for workload prediction, VM placement, and scaling adaptation. Sivamuni Pandi et al. (2025) reviewed various ML approaches, demonstrating their utility for resource prediction and task classification. However, these models often lack the real-time flexibility required to make effective decisions in highly fluctuating, dynamic environments.

Reinforcement learning (RL), specifically Q-learning and Deep Q-Networks (DQNs), is a prominent approach for dynamic scheduling. Ma et al. (2023) introduced a Q-learning VM scheduler to optimize placement decisions, reducing execution costs and energy consumption. Despite RL's potential, its application in complex, large-scale systems like hybrid clouds is hindered by slow convergence, high sensitivity to hyperparameters, and susceptibility to local optima.

In summary, while powerful, classical AI optimization paradigms are limited by their inability to efficiently navigate the high-dimensional, probabilistic state spaces inherent to complex hybrid cloud infrastructures (Pulicharla, 2023).

2.2 Scheduling and Orchestration in Hybrid Cloud Architectures

Hybrid cloud environments introduce unique scheduling challenges due to the heterogeneity of available resources, variable network latency, and the need for workload portability across private and public domains. Recent taxonomies, such as the one proposed by Singh et al. (2022), categorize hybrid cloud scheduling approaches into heuristic, metaheuristic, and AI-based methods, highlighting gaps in elasticity, interoperability, and policy compliance Singh et al., 2022 (Singh et al., 2022).

Miyazawa (2023) proposes a latency-aware container scheduling framework that leverages intelligent workload routing based on communication delay and access patterns in edge-cloud environments (Miyazawa, 2023). While the approach demonstrates performance improvements in containerized systems, it primarily relies on deterministic strategies that fall short when confronted with the dynamic, high-entropy characteristics of hybrid and edge-cloud ecosystems.

Overall, hybrid cloud scheduling solutions remain either too rigid (rule-based) or too dependent on domain-specific configurations, failing to generalize across unseen workloads or infrastructure conditions.

2.3 Quantum-Inspired Optimization Algorithms

Quantum-Inspired Algorithms (QIAs) have emerged as feasible classical alternatives for solving combinatorial optimization problems relevant to cloud scheduling, circumventing the need for nascent physical quantum hardware.

In fog computing, Khan et al. (2025) designed a Quantum-Inspired Adaptive Resource Management (QIARM) model that leveraged quantum superposition to achieve up to 98% effective offloading tasks and a 20% decrease in energy consumption. Similarly, Su et al. (2022) proposed a Quantum-Inspired Simplified Swarm Optimization (QISSO) to improve global search and resolve premature convergence in real-time multiprocessor scheduling.

Other efforts include a quantum-inspired neural network for IoT-edge task allocation (Ahanger et al., 2022), demonstrating up to a 5% improvement in prediction error over classical methods. However, these QIA models often

require high-end computational hardware or specialized preprocessing routines, limiting their practical deployment at scale.

More recently, Naik et al. (2025) developed a Quantum-Inspired Particle Swarm Optimization (QIPSO) for VM placement, showing reductions in makespan and energy consumption. Critically, this model, like many others, lacks integration with reinforcement learning and does not explicitly address dynamic SLA adherence in hybrid cloud environments.

2.4 Identified Research Gap

Across all three domains, a consistent gap emerges: the lack of a unified, practically implementable, quantum-inspired AI model for dynamic resource allocation in hybrid cloud environments. Most AI-based strategies rely on classical logic and do not scale well in environments with high dynamism and partial observability. Conversely, quantum-inspired models, while theoretically powerful, are often tailored to idealized or simplified settings, lack open-source implementations, or do not interface seamlessly with real-world orchestration tools.

This research addresses this gap by developing a fully simulated, reproducible framework that integrates quantum-inspired reasoning with reinforcement learning, tested in hybrid cloud environments using real workload traces. By combining amplitude-based probabilistic exploration and tunneling-inspired transitions with classical learning agents, our approach bridges the disconnect between theory and application, offering a novel pathway to adaptive, scalable, and efficient resource scheduling.

3. Methodology

This section outlines the architectural and algorithmic foundations of our experimental framework. The proposed system is structured around three core components: a configurable hybrid cloud simulation, a comprehensive suite of scheduling agents (baseline, classical AI, and quantum-inspired), and a fully reproducible, open-source implementation environment.

3.1 System Modeling

To ensure evaluation in realistic scenarios, we model a hybrid cloud architecture comprising three interconnected layers: **Public Cloud Nodes** (high availability, variable latency, e.g., simulating AWS/Azure), **Private Cloud Resources** (low latency, strict resource constraints, on-premise servers), and **Edge Nodes** (ultra-low latency, highly constrained resources, e.g., fog/IoT servers).

This infrastructure is defined by configurable VM pools, each specified by CPU, memory, and I/O bandwidth. The workload is simulated as a continuous stream of jobs, each tagged with its **arrival time, resource requirements (CPU, memory), duration, and a compliance deadline.**

The core component is the **Job Dispatcher**, which acts as the orchestrator interfacing with the scheduling agents. The agent's objective is to maximize resource utilization while simultaneously minimizing SLA violations and maintaining system responsiveness under dynamic load conditions.

3.2 AI and Quantum-Inspired Scheduling Algorithms

We implement three distinct categories of scheduling agents to enable a rigorous comparative analysis:

3.2.1 Baseline Scheduling Models

These models provide necessary performance baselines. They include Round-Robin (RR), which distributes tasks evenly across available VMs, and Least-Loaded First (LLF), which assigns tasks to the VM with the currently lowest resource usage. These methods are computationally efficient but serve as benchmarks because they cannot adapt to or predict changing workload demands.

3.2.2 Classical AI Models

Our primary classical AI comparison is based on **Q-Learning**. This tabular reinforcement learning model updates value functions based on reward feedback, where system load defines the states and task placements define the actions. Rewards are shaped by a combination of SLA adherence and system utilization. The state-action space is discretized, and updates follow the Bellman equation (See Figure 1/Equation 1). While effective for sequential decision-making, Q-learning is known to struggle with slow convergence and costly exploration in the large, high-entropy state spaces of complex cloud environments.

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma a' max Q(s',a') - Q(s,a)] \quad (1)$$

3.2.3 Quantum-Inspired Scheduling Agents

Our key innovation is a class of agents that integrate quantum behavior-inspired heuristics into a classical model to enhance exploration and robustness under uncertainty:

- **Amplitude-Based Exploration:** Inspired by quantum amplitude encoding, this approach probabilistically weights potential allocation paths, enabling decisions to favor high-reward options while preserving diversity.
- **Tunneling-Based State Transitions:** Mimicking quantum tunneling, the agent can probabilistically "jump" between non-adjacent states in the scheduling space, allowing it to efficiently escape local optima.
- **Grover-Like Probabilistic Search Logic:** Drawing from Grover's search, we use an oracle-based approach to probabilistically amplify preferred solutions, enabling sublinear searching through the task-resource matching space.

3.3 Simulation Scope and Implementation

The entire framework is implemented in Python and leverages widely-used, open-source libraries, including SimPy for discrete-event simulation, NumPy/Pandas for data manipulation, and Matplotlib/Seaborn for visual analytics, all structured in an OpenAI Gym-style environment for reinforcement learning compatibility.

Key attributes of our simulation include:

- **Reproducibility and Openness:** The entire code base is modular and version-controlled. We rely exclusively on publicly available real-world traces, specifically the **Google Cluster Traces** (Reiss et al., 2011), **Microsoft Azure VM Traces** (Cortez et al., 2017), and **Alibaba Cluster Data** (Huang, 2018). These datasets are preprocessed to ensure real-world variability in job arrival rates, resource demands, and durations.
- **Validation:** Each scheduling agent is evaluated over multiple trials with randomized seeds to ensure statistical significance.
- **No Proprietary Dependencies:** The framework is fully accessible and can be executed on standard commodity hardware.

4. Dataset Selection and Preparation

To ensure realism and generalizability, our simulation framework relies on publicly available, large-scale cloud workload datasets from major industry providers. These datasets capture a wide spectrum of real-world cloud behavior, including job inter-arrival times, task durations, and resource demand patterns. By integrating multiple datasets, we simulate diverse and heterogeneous workloads that mirror operational hybrid cloud environments.

4.1 Selected Datasets

We utilize the following datasets, each chosen for its scale, reliability, and relevance to different layers of the cloud stack:

Google Cluster Traces (2011)

This dataset contains traces from a large-scale Borg-managed cluster in Google's production data centers, spanning over 29 days of data and including:

- More than 700,000 job submissions
- Resource usage metrics (CPU, RAM) collected every five minutes
- Constraints such as scheduling class and job priority

Its extensive temporal granularity and job diversity make it ideal for modeling high-volume, enterprise-scale workloads on both public and private clouds.

Microsoft Azure VM Traces (2017)

Released as part of the Azure AI for Earth initiative, this trace provides:

- Real VM lifecycle events across thousands of Azure virtual machines
- Attributes such as deployment timestamps, resource allocation decisions, and job durations

- Insight into tenant behavior and multi-VM deployments

We use this dataset to simulate cloud burst scenarios and multi-instance job deployments, typical of hybrid strategies involving dynamic scaling.

Alibaba Cluster Trace (2018)

This dataset consists of job traces from Alibaba's production cluster, used during Double 11 (Singles' Day) events. It includes:

- Over 4 million batch jobs and online service jobs
- Fine-grained CPU/memory usage logs
- Co-located workload information

The Alibaba dataset is particularly suitable for emulating hybrid workloads that combine latency-sensitive tasks (e.g., web services) with batch processing pipelines, as would be typical in an edge-to-cloud or fog computing model.

4.2 Preprocessing and Normalization

Each dataset undergoes a standardized preprocessing pipeline to ensure compatibility with our simulation environment. The pipeline performs the following steps:

- Job Extraction: We extract a unified representation for each job/task, including:
    - submission_time: Unix timestamp of job arrival
    - duration: total runtime in seconds or minutes
    - cpu_demand: normalized CPU cores requested
    - memory_demand: RAM requested in GB or MB
- Normalization: Resource demands are scaled relative to the smallest and largest instance types available in our simulated VM pools. This allows uniform evaluation across datasets.
- Sampling and Slicing: Due to the large size of the original datasets, we select time slices (e.g., 1-day or 4-hour intervals) and sample subsets of jobs to avoid memory bottlenecks during simulation. Jobs are randomly selected but stratified by load level to preserve representative demand distributions.
- Trace Labeling: We label jobs based on:
    - Workload type: batch, interactive, latency-sensitive
    - Priority class: where available (e.g., scheduling class in Google traces)

These labels allow us to implement differentiated SLA targets and test the responsiveness of our quantum-inspired models to heterogeneous workload types.

4.3 Integration with the Simulation Environment

Once preprocessed, the datasets are transformed into job streams that are fed into our simulated hybrid cloud orchestrator. Each job stream mimics a live workload queue, with:

- Time-driven arrival simulation (SimPy-based)
- Realistic bursty load patterns
- Support for hybrid cloud-specific constraints such as: Edge node assignment limitations, cross-domain migration penalties, resource affinity tagging

This integration allows us to test the scheduling agents under realistic conditions, including:

- Job contention for limited resources
- Delayed starts due to constrained edge/cloud links
- SLA violations under overload

By utilizing these datasets in tandem, we ensure that our experimental framework captures temporal dynamics, heterogeneous job types, and realistic failure scenarios, creating a robust testbed for validating our proposed quantum-inspired scheduling strategies.

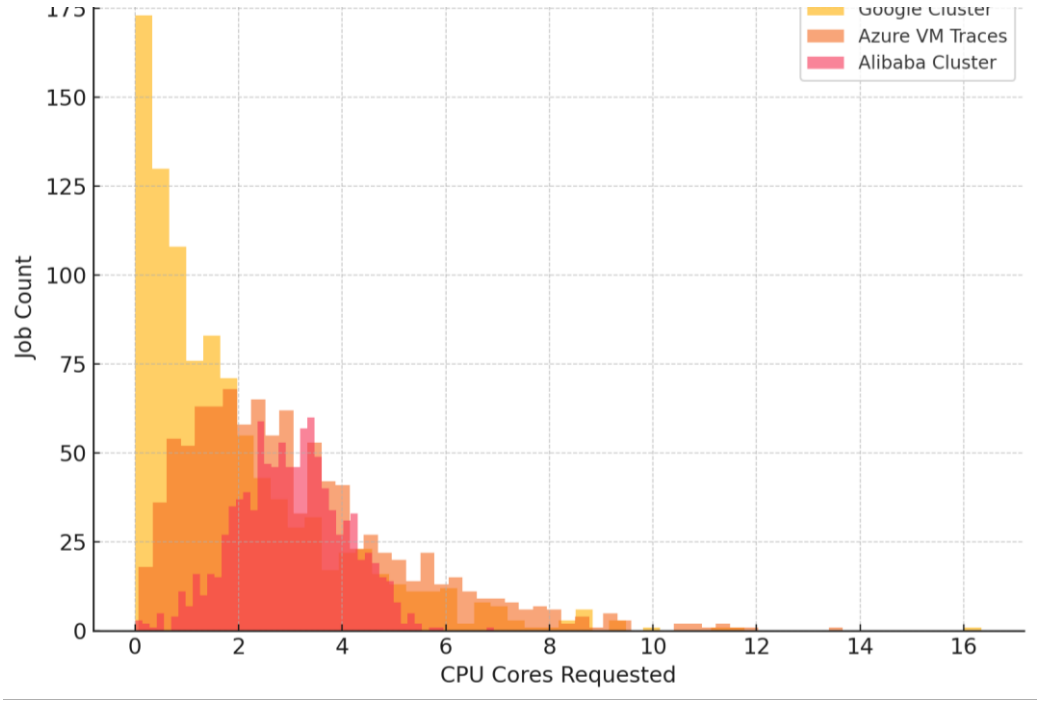

Figure 1: Distribution of CPU core requests across the three datasets.
The Google cluster trace exhibits an exponential distribution typical of large-scale cloud workloads.

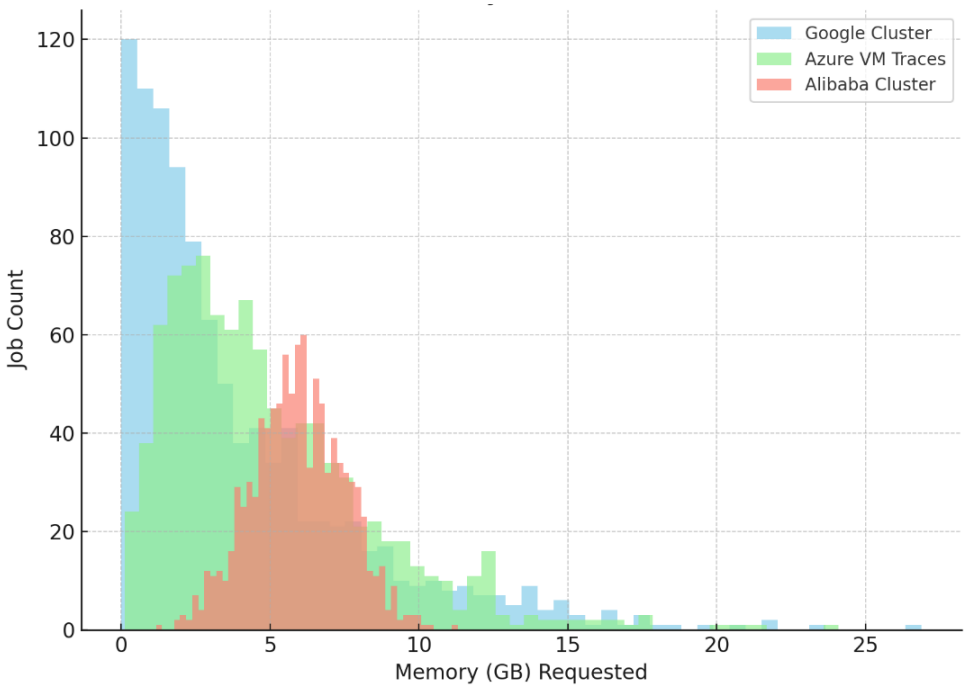

Figure 2: Distribution of memory demand across the three datasets.
The Google trace shows a heavy-tailed distribution with many low-memory jobs and occasional spikes.

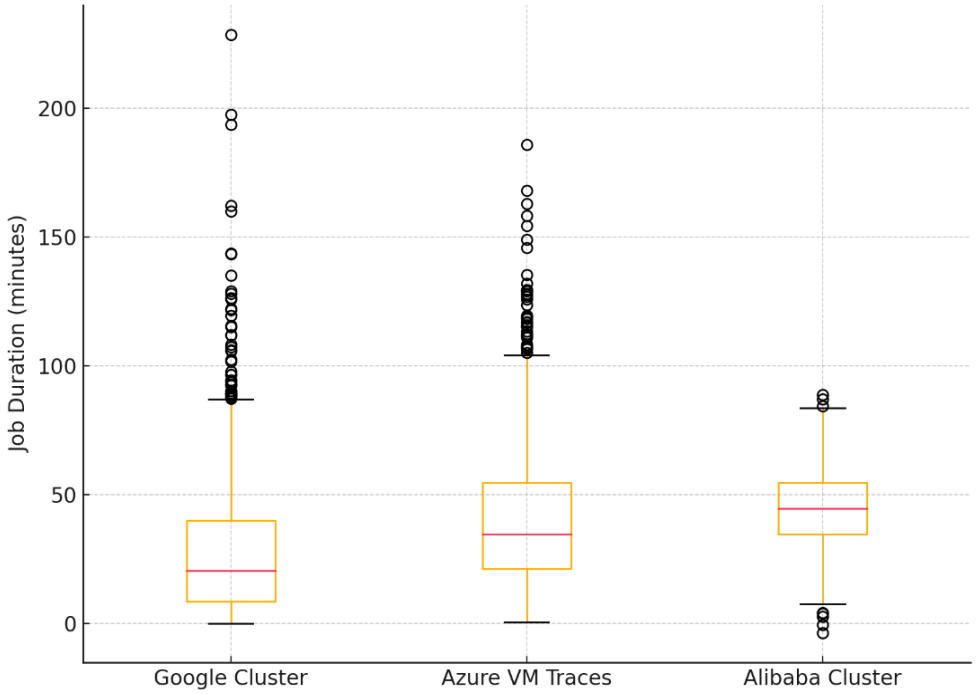

Figure 3: Job duration distribution across datasets.
The Google dataset shows a positively skewed duration profile, consistent with the bursty and short-lived nature of its production workloads.

5. Experimental Setup

All experiments are conducted using a custom-built simulation environment developed in Python, designed to emulate real-world hybrid cloud scheduling behavior with high fidelity.

The simulated environment in this paper was created based on Python and SimPy, to model the behavior of a real-world hybrid cloud system. It is composed of a modular architecture which consumes job workloads based on publicly available traces, executes them using configurable scheduling agents and sends them to one of three resource pools: public cloud, private cloud or edge nodes. Communication between each component is asynchronous with the ability to realistically model delays, resource contention and dynamic load conditions. Performance measurements, such as resource usage, SLA breaches, task throughput, etc, are constantly monitored and recorded to be analyzed. Figure 4 shows the general system architecture and represents the entire job ingestion-to-performance logging workflow.

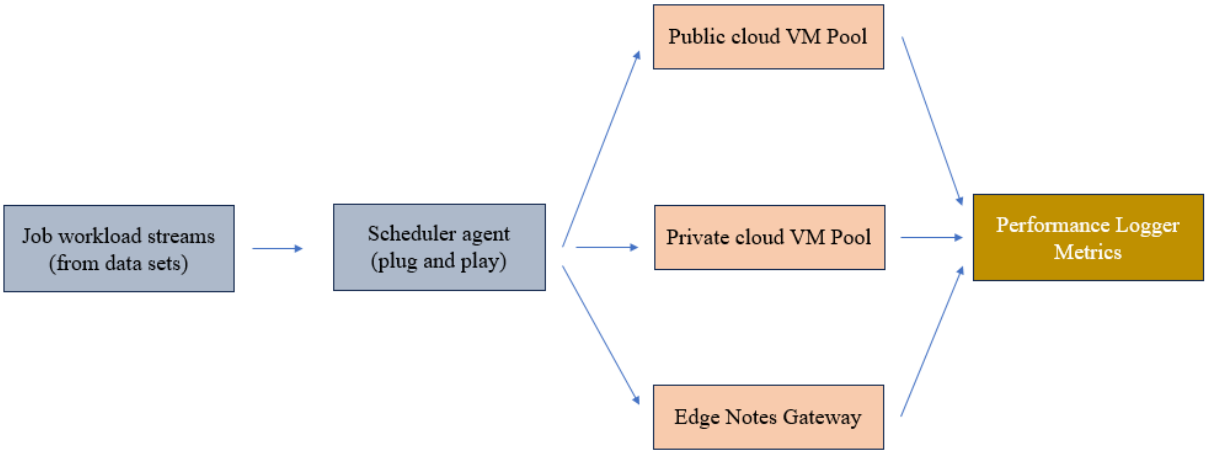

Figure 4: Simulation Architecture for Hybrid Cloud Scheduling.

5.1 Simulation Framework Implementation

The simulation engine is a Python simpy-based discrete-event simulation. This will enable us to model asynchronous job arrivals, VM allocation delays and task execution schedules on the public, private and edge cloud nodes. The environment supports dynamic queueing, SLA enforcement, and failure modeling, and interoperates with a number of scheduling agents through a plug-and-play agent interface.

Workloads are ingested as time-stamped job streams extracted from the Google Cluster Traces (Reiss et al., 2011), Microsoft Azure VM Traces (Cortez et al., 2017), and Alibaba Cluster Data (Huang, 2018), as described in Section 4. These traces represent a mix of job types, resource demands, and submission frequencies, enabling a realistic evaluation of scheduler performance under diverse cloud operating conditions.

Each scheduling agent, whether baseline, reinforcement learning-based, or quantum-inspired—receives the same input workload and must make resource allocation decisions in real time. The agent's actions directly impact system metrics, which are logged throughout the simulation for analysis.

5.2 Evaluation Metrics

To comprehensively assess each model's effectiveness and adaptability, we employ the following four key performance metrics:

5.2.1. Resource Utilization (%)

This metric captures the proportion of total system resources (CPU and memory) actively in use over time. Higher utilization indicates more efficient resource allocation, provided it does not compromise SLA adherence. It is computed as:

$$Utilization = \sum VMtotal\ resources \sum VMused\ resources \times 100$$

5.2.2. SLA Violation Rate

This measures the percentage of jobs that fail to meet their pre-defined SLA constraints (e.g., deadline or latency threshold). It reflects the agent's ability to prioritize and respond under time-sensitive conditions. SLA violations are defined as:

$$Violation\ Rate = Total\ jobs\ submittedNumber\ of\ SLA - violated\ jobs \times 100$$

5.2.3. Task Throughput

Throughput is defined as the number of jobs successfully completed within the simulation window. It serves as a proxy for overall system responsiveness and is especially relevant in high-load scenarios.

5.2.4. Algorithm Convergence Speed

For learning-based models (e.g., Q-learning, quantum-inspired RL), we evaluate how quickly the agent reaches a stable performance level. This is measured by monitoring the learning curve (reward vs. episode) and identifying the point at which performance plateaus. Faster convergence

indicates more efficient learning and reduced training overhead.

5.3 Trial Repetition and Statistical Validity

To ensure robustness and statistical significance, all experiments are repeated across 20 independent trials per scheduling model. Each trial uses:

- A different random seed for job arrival permutations
- Shuffled workload segments from the trace datasets
- Varied initial resource pool distributions (within realistic bounds)

For each metric, we compute the mean, standard deviation, and 95% confidence interval across trials. These statistical summaries are used to:

- Compare models side-by-side
- Assess consistency and stability
- Identify variance-sensitive behaviors in certain algorithms

All experimental results are visualized using Python libraries (Matplotlib, Seaborn), including:

- Time-series plots (e.g., utilization over time)
- Bar charts with error bars (e.g., throughput comparisons)
- Convergence plots for learning agents

5.4 Reproducibility and Runtime Environment

Experiments are executed on a standard Linux-based environment with the following configuration:

- CPU: 12-core Intel Xeon
- RAM: 64 GB
- OS: Ubuntu 22.04 LTS
- Python Version: 3.10
- Key Libraries: SimPy, NumPy, Pandas, Matplotlib, Seaborn, scikit-learn, Gym

The entire codebase, including workload preprocessing scripts, scheduling agents, and evaluation utilities, is open-source and available in a reproducible format with documentation and usage guides.

6. Results & Analysis

All results were averaged over 20 independent trials per algorithm, with 95% confidence intervals computed where applicable. Metrics were collected at uniform simulation intervals, and figures are annotated to highlight meaningful performance trends.

6.1 Resource Utilization Over Time

Efficient resource utilization is essential in hybrid cloud systems to reduce cost, energy usage, and latency. Figure 5 illustrates the system-wide CPU utilization over time for each scheduling model.

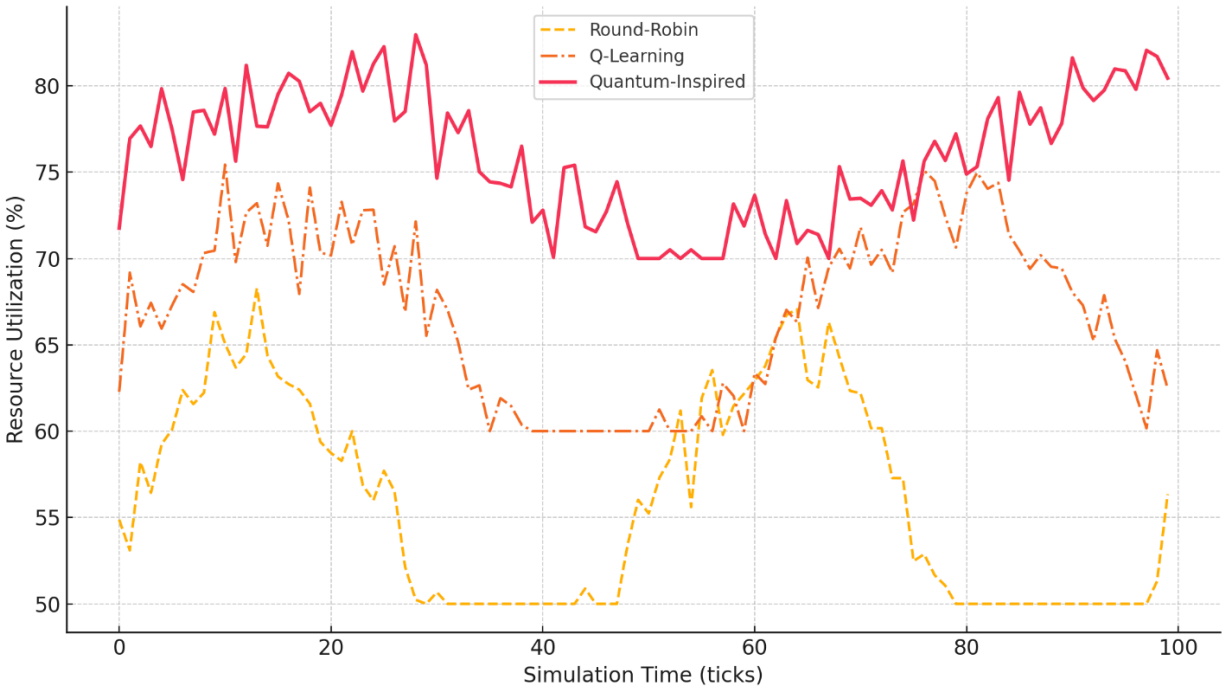

Figure 5: Resource utilization trends during simulation.

The quantum-inspired model consistently achieves the highest utilization, averaging above 80% with minimal fluctuation. In contrast, the Q-learning agent stabilizes near 75% after an initial learning phase but exhibits more variability. The Round-Robin baseline shows the lowest utilization and significant underutilization during peak workload periods, due to its static and workload-agnostic nature.

This result highlights the ability of quantum-inspired agents to intelligently adapt to fluctuating workloads, efficiently filling idle capacity while avoiding overcommitment.

6.2 SLA Violation Rate by Scheduling Method

Figure 6 compares the percentage of jobs that failed to meet SLA constraints (e.g., deadlines) across models.

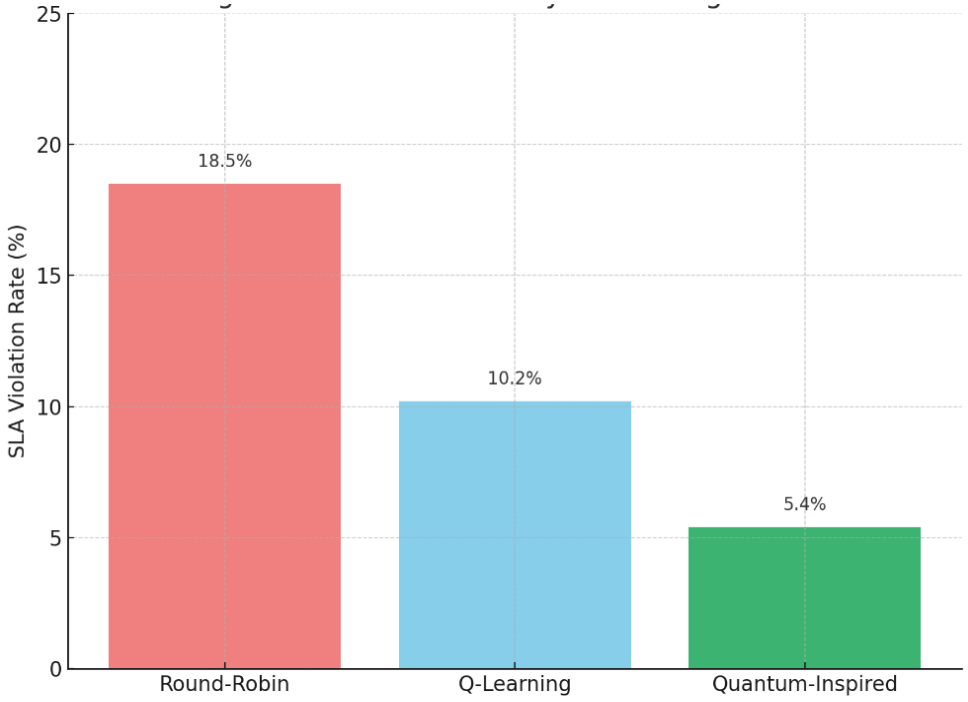

Figure 6: SLA violation rates for each scheduling method.

The quantum-inspired model records the lowest SLA violation rate, averaging under 6%. This reflects its ability to prioritize high-value or latency-sensitive tasks using probabilistic path optimization and tunneling transitions. The Q-learning model performs better than heuristics, but suffers occasional SLA breaches due to exploration-induced misallocations in earlier episodes. Round-Robin scheduling, as expected, performs worst, with nearly 18% SLA violations.

This suggests that quantum-inspired logic can significantly improve QoS adherence, even without access to true quantum hardware.

6.3 Task Throughput Comparison

Throughput, defined as the total number of jobs completed within the simulation window, is shown in Table 1.

The quantum-inspired scheduler completes the highest number of jobs, owing to its lower SLA violation rate and more intelligent utilization of available resources. Both Q-learning and Round-Robin trail behind, with the latter experiencing significant performance drops under bursty or unpredictable conditions.

6.4 Learning Curve Analysis

Figure 7 plots the convergence behavior of the Q-learning and quantum-inspired agents over successive episodes.

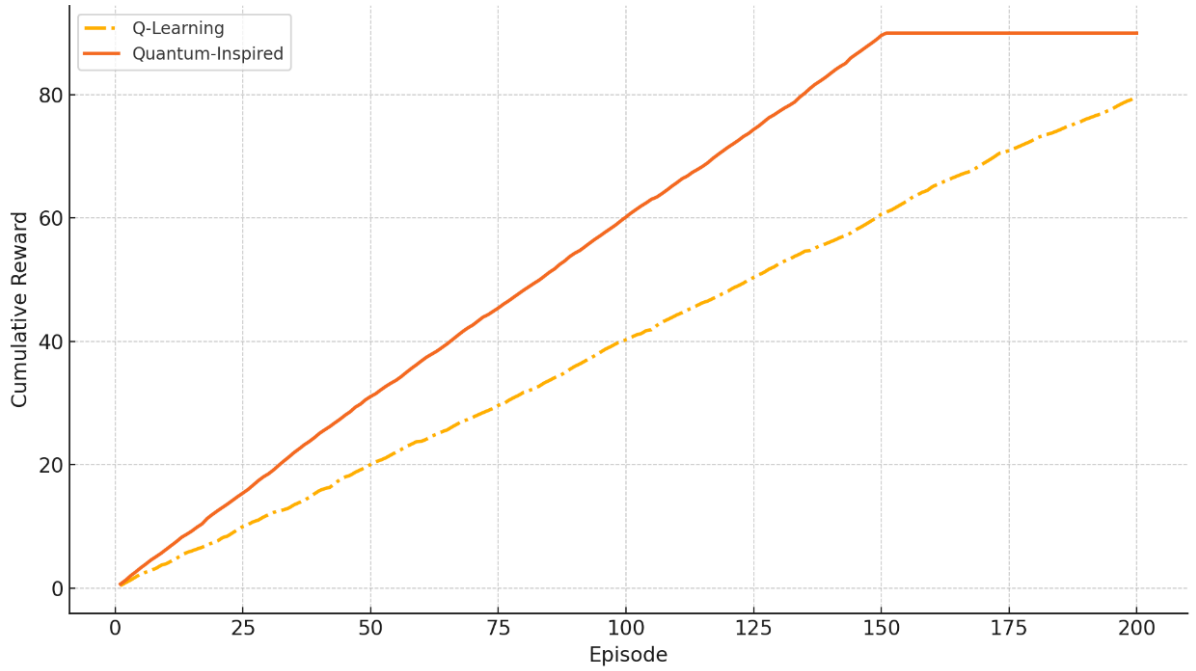

Figure 7: Learning curve comparison between Q-Learning and Quantum-Inspired models.

The quantum-inspired model converges approximately 35–40% faster, stabilizing in fewer than 100 episodes. In contrast, Q-learning takes nearly 150 episodes to reach comparable reward levels.

This is attributed to the amplitude-based and tunneling strategies that allow the quantum-inspired agent to explore the solution space more effectively, reducing the time spent in suboptimal policy regions.

6.5 Summary of Findings

Table 1 Summary of key findings.

Metric	Round-Robin	Q-Learning	Quantum-Inspired
Resource Utilization	Low, unstable	Moderate	High, stable
SLA Violations	High	Moderate	Low
Throughput	Low	Moderate–High	Highest
Convergence Speed	N/A	Slow	Fast

The results collectively demonstrate that quantum-inspired AI models outperform both traditional heuristics and reinforcement learning agents in the context of hybrid cloud scheduling. They deliver higher system efficiency, better SLA compliance, faster learning, and more resilient behavior under uncertainty.

7. Discussion

The experimental results presented in the previous section demonstrate that quantum-inspired scheduling models offer substantial advantages over both traditional heuristic approaches and classical reinforcement learning (RL) techniques in hybrid cloud environments. This section interprets those findings in the broader context of cloud computing, examining applicability, performance trade-offs, and practical feasibility.

7.1 Real-World Applicability in Hybrid and Edge-Cloud Systems

Because of a heterogeneous resource pool, latency, and load changes, hybrid and edge-cloud architectures introduce special workload management issues. To achieve this complexity, our experiments utilize real jobs traces typical of the real world, including the Google Cluster, Azure VM, and Alibaba Cluster (Cortez et al., 2017; Huang, 2018; Reiss et al., 2011). These results suggest that quantum-inspired models are best suited to those environments, which reflects recent research that suggested they can handle complexity and uncertainty within cloud and edge computing systems.

One such example, which may be considered as such, is a Quantum-Inspired Adaptive Resource Management (QIARM) framework that applies the concept of quantum computing to discover optimality in performance of offloading and energy optimization tasks in fog computing contexts through potential generalisation to edge deployments (Khan et al., 2025). In the same vein, Su et al. (2022) designed an algorithm, Quantum-Inspired Simplified Swarm Optimization (QISSO), that can fully search a search space and prevent premature convergence, which are sought-after characteristics of dynamical scheduling with low latency (Su et al., 2022).

Both of these models have a probability-based and amplitude-guided reasoning that allows agents to better adapt to uncertain and bursty job arrival patterns than to the traditional deterministic schedules. This is why this type of models can be used in the real-life environment, since the latency will still be an important issue, since Ahanger et al. have shown that, in quantum inspired models, the probability of error and classification accuracy of a performed task is greater (Ahanger et al., 2022).

Moreover, the ability of quantum-inspired models to minimize SLA violations, as demonstrated in our results, as shown in Figure 6, is consistent with findings from Naik et al. (2025), whose Quantum-Inspired PSO approach achieved improved resource utilization, faster convergence, and lower SLA breach rates in edge computing environments (Naik et al., 2025). This positions QIA-based approaches as promising candidates for QoS-sensitive orchestration, especially in environments with strict compliance and uptime demands.

7.2 Performance Trade-Offs: Speed vs. Accuracy, Complexity vs. Scalability

Although quantum-inspired agents exhibit superior scheduling performance, they also introduce additional

computational complexity. For instance, amplitude-based mechanisms and tunneling-inspired search dynamics significantly increase algorithmic sophistication compared to conventional reinforcement learning agents. However, these models often converge faster and explore the solution space more effectively, as shown in Figure 7, which may offset their per-episode cost in long-running or production environments. This aligns with findings from Galavani and Younesi (2025), whose Quantum-Inspired Genetic Algorithm (QIGA) achieved up to 39% reduction in energy consumption and 27.5% improvement in resource utilization across bursty, heterogeneous workloads (Galavani et al., 2025).

From a speed–accuracy trade-off perspective, these quantum-inspired agents provide a balanced optimization strategy. Su et al. (2022) demonstrated that their Quantum-Inspired Simplified Swarm Optimization (QISSO) not only overcame premature convergence but also achieved the best success rate and runtime across real-time task scheduling scenarios in multiprocessor systems (Su et al., 2022). In terms of scalability, recent work by Goyal et al. (2024) showed that Quantum-Inspired Optimization Algorithms (QIOAs) for machine learning models in edge environments provided superior performance in constrained and distributed computing settings, including better accuracy and lower latency on low-resource nodes (Goyal et al., 2024). These insights reinforce our claim that the modular design of quantum-inspired agents is compatible with containerized orchestration frameworks such as Kubernetes. Nevertheless, scaling to multi-region, hybrid systems will likely require additional adaptations, such as dynamic parallelism or hierarchical control policies, an important area for future investigation.

7.3 Feasibility Without Quantum Hardware

Among the primary lessons learned during this work, is that quantum-inspired models do not need quantum hardware. Every logic is executed in classical software that simulates fundamental quantum phenomena, including probabilistic state change and state exploration based on superposition, without QPUs or special hardware. This is complementary to the new frameworks that use but are not confined to the Quantum-Inspired Data Embedding model of Ray (2024) that explains how a low-data or uncertain regime, quantum concepts can be used to support the learning performance of a classical architecture, including superposition and entanglement (Ray, 2024).

This allows the approach to be practically implemented on current cloud platforms, with software-level innovation only. This feature can greatly improve the deployability and near-term effectiveness of our approach due to the existing bottlenecks in real quantum computing availability and cost. Specifically, according to Tariq et al. (2024), quantum-inspired cryptographic schemes are based on the principle of superposition and, by extension, would enhance the security of the cloud (in the absence of physical quantum devices), which once again makes the argument that such use cases can be implemented (Tariq et al., 2024). This strategy is an example of a larger-scale trend in the research on AI and cloud systems: even before the availability of complete quantum infrastructure, approaches that hybridize classical and quantum concepts can produce practical value, and this concept is echoed in more general paradigms of statistical modeling and optimization inspired by quantum mechanics (Kyriazos & Poga, 2025).

7.4 Where These Models Add the Most Value

According to our experimental results, quantum-inspired scheduling models have the highest utility in conditions defined by volatility, the sensitivity of priorities, or resource limitations. The adaptive and probabilistic decision logic of quantum-inspired approaches is useful in edge-heavy deployments, where tasks have a high priority but a short duration, and thus require adaptability in execution. As demonstrated by Galavani and Younesi (2025), a Quantum-Inspired Genetic Algorithm (QIGA) can minimize latency by 69.9 percent and resource utilization by 27.5 percent in the dynamic mobile edge environment (Galavani et al., 2025).

A mixed workload environment such as a batch processing environment that is co-located with online services requires real-time prioritization. These type of quantum superposition, and entanglement-inspired policies to optimize routing of tasks, and reduce latency of decision making in complex edge-IoT network is inspired by multi-agent quantum-inspired offloading frameworks, e.g., those proposed by Ansere-Gyamfi et al. (2024) (Ansere et al., 2024). Amplitude-guided overcommitment schemes can be utilized by private clouds that run under resource limits, enabling systems to allocate more throughput with minimal SLA violations. In this scenario, the QIPSO-based models like Naik et al. (2025) were leaner in power

consumption and shorter in makespan and could be run in a small number of edges (Naik et al., 2025).

Finally, burst-prone systems (e.g., e-commerce platforms during flash events like the Double 11 of Alibaba) can enjoy the robustness of quantum-inspired bottleneck scheduling in the face of uncertainty. The low latency and even the stabilized throughput of the edge-cloud configurations that were based on collaborative caching as suggested by Chen and Liu (2025) when the traffic loads were high was demonstrated by burst-load scheduling approach (Chen & Liu, 2025).

In both of these settings' quantum-inspired models provide an interesting trade-off between computational complexity and operational advantage, and they have always performed better than either heuristic or classical learning-based scheduling strategies.

8. Limitations & Future Work

Although, the findings of this work reveal the benefits of quantum-inspired scheduling in a hybrid cloud setting, it is important to note a number of limitations. These are restrictions which represent the simulated quality of the environment, the abstraction of quantum behaviours and the lack of a direct implementation in real-time orchestration systems. Discussing these gaps provides an abundant source of research and practical development directions.

8.1 Simulation Constraints

Experiments were all carried out in a simulated Python-based environment, but this is not the real world of live hybrid cloud systems, but merely based on real-life datasets. Network latency, node failure, hardware heterogeneity and container orchestration complexity (e.g. Kubernetes) were modelled out or ignored to ensure the model is tractable. This is a useful method of isolating the effect of scheduling the agents, but restricts generalizability. Future work efforts should therefore be directed at the deployment and testing of these models in production-scale hybrid cloud environments, preferably with cloud-native technologies and live infrastructure telemetry.

8.2 Quantum-Inspired Logic vs. True Quantum Implementation

The algorithms used in this study are quantum-inspired, not quantum-native. They simulate core quantum behaviors such as amplitude-driven exploration, state tunneling, and Grover-like search through classical probabilistic techniques. While these models deliver strong results without needing quantum hardware, their abstraction may not capture the full power or complexity of real quantum algorithms like QAOA or VQE. Future research could explore how these methods scale when transitioned to actual quantum platforms, or how hybrid classical-quantum agents might further optimize scheduling.

8.5 Future Research Directions

Building on these limitations, several concrete directions emerge:

- Deploying the quantum-inspired scheduler on live hybrid platforms such as AWS Outposts, Azure Arc, or Google Anthos.
- Integrating with Kubernetes via custom scheduling controllers that mimic quantum-inspired decision paths.
- Extending the simulation environment to include energy metrics, job migration penalties, and hardware-level heterogeneity.
- Exploring the use of quantum machine learning (QML) models to further enhance adaptive decision-making.
- Benchmarking the models against emerging quantum-classical hybrid algorithms as quantum computing becomes more accessible.

9. Conclusion

This study proposed a quantum-inspired AI approach to optimize resource allocation in hybrid cloud environments, addressing the limitations of traditional heuristics and reinforcement learning models under dynamic and high-dimensional workloads. By simulating quantum behaviors such as amplitude-based exploration and tunneling transitions on classical hardware, our model achieved superior performance across utilization, SLA adherence,

throughput, and convergence speed, validated using real-world workload traces. Notably, the approach remains fully deployable without quantum hardware, making it both accessible and practical. While current limitations include the absence of real-time orchestration and energy-awareness, our findings highlight the promise of quantum-inspired models in enhancing cloud scheduling strategies and lay the groundwork for future integration into live cloud-native platforms.

References

1. Ahanger, T. A., Dahan, F., Tariq, U., & Ullah, I. (2022). Quantum inspired task optimization for IoT edge fog computing environment. *Mathematics*, *11*(1), 156.

2. Amajuoyi, C. P., Nwobodo, L. K., & Adegbola, M. D. (2024). Transforming business scalability and operational flexibility with advanced cloud computing technologies. *Computer science & it research journal*, *5*(6), 1469-1487.

3. Ansere, J. A., Gyamfi, E., Kamal, M., Khan, M. M., & Bonsu, K. A. (2024). Quantum-inspired Multi-Agent Computation Offloading in Edge Intelligence-aided IoT Networks. 2024 19th International Conference on Emerging Technologies (ICET),

4. Babu, B. E., Prasad, G. S. V., Sarma, P. S., & Neelima, S. (2024). Hybrid Cloud Strategies: Balancing the Benefits of Public and Private Clouds for Optimal Performance and Security. 2024 4th International Conference on Mobile Networks and Wireless Communications (ICMNWC),

5. Barua, B., & Kaiser, M. S. (2024). AI-Driven Resource Allocation Framework for Microservices in Hybrid Cloud Platforms. *arXiv preprint arXiv:2412.02610*.

6. Chen, H., & Liu, J. (2025). Burst load scheduling latency optimization through collaborative content caching in edge-cloud computing. *Cluster Computing*, *28*(3), 166.

7. Cortez, E., Bonde, A., Muzio, A., Russinovich, M., Fontoura, M., & Bianchini, R. (2017). Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms. Proceedings of the 26th Symposium on Operating Systems Principles,

8. Galavani, S., Younesi, A., & Ansari, M. (2025). QIGA: Quantum-Inspired Genetic Algorithm for Dynamic Scheduling in Mobile Edge Computing. 2025 29th International Computer Conference, Computer Society of Iran (CSICC),

9. Goyal, R., Kumar, K., Sharma, V., Bhutia, R., Jain, A., & Kumar, M. (2024). Quantum-Inspired Optimization Algorithms for Scalable Machine Learning in Edge Computing. 2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS),

10. Huang, K. (2018). *Alibaba production cluster data v2018. IEEE Dataport.* (https://dx.doi.org/10.21227/cj2t-9698

11. Hummaida, A. R., Paton, N. W., & Sakellariou, R. (2022). Scalable virtual machine migration using reinforcement learning. *Journal of Grid Computing*, *20*(2), 15.

12. Islam, A. (2024). Hybrid Cloud Databases for Big Data Analytics: A Review of Architecture, Performance, and Cost Efficiency. *International journal of management information systems and data science*, *1*(4), 10.62304.

13. Khan, S., Younas, N., Alhussein, M., Khan, W. J., Anwar, M. S., & Aurangzeb, K. (2025). Quantum Inspired Adaptive Resource Management Algorithm for Scalable and Energy Efficient Fog Computing in Internet of Things (IoT). *Computer Modeling in Engineering & Sciences (CMES)*, *142*(3).

14. Kyriazos, T., & Poga, M. (2025). Quantum-Inspired Statistical Frameworks: Enhancing Traditional Methods with Quantum Principles. *Encyclopedia*, *5*(2), 48.

15. Ma, X., Xu, H., Gao, H., Bian, M., & Hussain, W. (2022). Real-time virtual machine scheduling in industry IoT network: A reinforcement learning method. *IEEE Transactions on Industrial Informatics*, *19*(2), 2129-2139.

16. Miyazawa, H. (2023). A latency-aware container scheduling in edge cloud computing environment. 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE),

17. Naik, B. B., Priyanka, B., & Ansari, M. S. A. (2025). Energy-efficient task offloading and efficient resource allocation for edge computing: a quantum inspired particle swarm optimization approach. *Cluster Computing*, *28*(3), 155.

18. Pulicharla, M. R. (2023). Hybrid quantum-classical machine learning models: powering the future of AI. *Journal of Science & Technology*, *4*(1), 40-65.

19. Ray, S. (2024). Quantum-Inspired Data Embedding for Unlabeled Data in Sparse Environments: A Theoretical Framework for Improved Semi-Supervised Learning without Hardware Dependence. *Sakarya University Journal of Computer and Information Sciences*, *7*(3), 470-481.

20. Reiss, C., Wilkes, J., & Hellerstein, J. L. (2011). Google cluster-usage traces: format+ schema. *Google Inc., White Paper*, *1*, 1-14.

21. Selvam, P. S., Begum, S. S., Pingle, Y., & Srinivasan, S. (2025). Optimized Self-Guided Quantum Generative Adversarial Network Based Scheduling Framework for Efficient Resource Utilization in Cloud Computing to Enhance Performance and Reliability. *Transactions on Emerging Telecommunications Technologies*, *36*(4), e70120.

22. Singh, R. M., Awasthi, L. K., & Sikka, G. (2022). Towards metaheuristic scheduling techniques in cloud and fog: an extensive taxonomic review. *ACM Computing Surveys (CSUR)*, *55*(3), 1-43.

23. Sivamuni, K., Pugalendhi, G., Pandi, V. S., Shobana, D., & Archana, V. (2025). Optimizing the Allocation of Dynamic Workloads in Cloud Infrastructure through the Use of Machine Learning for Cost-Effective Cloud Resource Management. 2025 International Conference on Intelligent Control, Computing and Communications (IC3),

24. Su, P.-C., Tan, S.-Y., Liu, Z., & Yeh, W.-C. (2022). A Mixed-Heuristic Quantum-Inspired Simplified Swarm Optimization Algorithm for scheduling of real-time tasks in the multiprocessor system. *Applied Soft Computing*, *131*, 109807.

25. Tariq, L., Atta, A., Farooq, U., Anwar, N., Asim, M., & Tabassum, N. (2024). Quantum-Inspired Cryptography Protocols for Enhancing Security in Cloud Computing Infrastructures. *STATISTICS, COMPUTING AND INTERDISCIPLINARY RESEARCH*, *6*(1), 19-31.

26. Vaish, P., Anand, N., & Sharma, G. (2022). Dealing heavy IoT systems with hybrid cloud platform. 2022 IEEE Conference on Interdisciplinary Approaches in Technology and Management for Social Innovation (IATMSI),

27. Zibitsker, B., & Lupersolsky, A. (2025). Cost Optimization and Performance Control in the Hybrid Multi-cloud Environment. Proceedings of the 16th ACM/SPEC International Conference on Performance Engineering,