National College of Ireland

Application of Hierarchical Deep Reinforcement Learning for Dynamic Task Scheduling in Multi-Region Cloud Environments

MSc Research Project
Cloud Computing

Sri Lakshmi Prasanna Eda
x23304723

School of Computing
National College of Ireland

Supervisor: Ahmed Makki

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Sri Lakshmi Prasanna Eda

Student ID: x23304723

Programme: MSc in Cloud Computing **Year:** 2025-2026

Module: MSc Research Project

Supervisor: Ahmed Makki

Submission Due Date: 28/01/2026

Project Title: Application of Hierarchical Deep Reinforcement Learning for Dynamic Task Scheduling in Multi-Region Cloud Environments

Word Count: 8533 **Page Count:** 23

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Sri Lakshmi Prasanna Eda

Date: 28/01/2026

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□

You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Application of Hierarchical Deep Reinforcement Learning for Dynamic Task Scheduling in Multi-Cloud Environments

Sri Lakshmi Prasanna Eda
x23304723

Abstract

Multi-cloud ecosystems provide organizations with flexibility and cost optimization but come with remarkable issues in job scheduling. The traditional scheduling algorithms cannot efficiently manage resource heterogeneity, while the deep reinforcement learning (DRL) methods that have been applied in the cloud environment encounter the scalability issue what with the centralized architectures that suffer state-action space explosion as cloud providers grow. This work presents an innovative framework called Hierarchical Deep Reinforcement Learning (HDRL) that is integrated with differential privacy as a part of the scheduling structure. The framework follows a two-tier building block, wherein the local agents carry out the resource allocation that is specific to each provider adopting proximal policy optimization (PPO) and the global coordinator that performs cross-provider decisions optimization using deep neural networks. This implementation validates the framework through multi-region deployment on AWS using Google Cloud Traces driven workloads for training and validation, directly demonstrates the feasibility and applicability of the method. The assessment of the AWS multi-region deployment delivered 46.7% cost savings, 83.3% accuracy in allocation and ε=1.0 differential privacy assurances with sub-100ms inference latency for allocation decisions. Thus, this research contributes to closing the gap between theoretical DRL models and production-ready, multi-region cloud, privacy-preserving, DRL-based schedulers.

1 Introduction

1.1 Background and Motivation

The tremendous growth of cloud computing has caused major changes in organizational IT and multi-cloud approaches have been adopted as enterprise computing's primary paradigm. Enterprises deploy workloads on multiple vendors including AWS, Microsoft Azure, and Google Cloud Platform to the extent of mitigating vendor lock-in, gaining reliability, cost optimization, and leveraging the capabilities of each provider (Varghese and Buyya, 2024).

Yet, the heterogeneity of the resources in cloud and multi-cloud chains greatly complicates the processes in the resource management and task scheduling domains. Decision-making now must take into account the different pricing structures, performance metrics, distribution across geolocation, and SLAs (Service Level Agreements) of different providers.

Traditional heuristic scheduling algorithms, such as First-Come-First-Served, Round Robin, and priority-based methods, lack the capability to dynamically modify multi-cloud time zone

circuits. Periodic changes in workload patterns along with shallow resource characteristics end up with heuristics that are unable to respond to the impacts efficiently. (Mangalampalli et al., 2024). The rapid development in the field of artificial intelligence about deep reinforcement learning (DRL) has gone a long way in addressing these problems. With the help of the DRL frameworks, agents can learn the best possible policies through interaction with the scheduling environment and thus reduce makespan, cost, and energy use better than the conventional methods (Hou et al., 2024).

1.2 Problem Statement

The dynamic task-scheduling issue in multi-cloud systems features three interconnected challenges that no research has ever solved at the same time. The first matter is the scalability limitations of centralized DRL architectures characterized by an increase in the number of cloud providers and heterogeneous resources. Centralized architectures necessitate global state representations that have to include all resource availability, current workload distributions, task queue characteristics, and inter-provider network conditions. As a consequence, this thorough state representation brings about an exponential increase of the dimension of state action space, which results in the instability in training, the long times of convergence, and the learning of suboptimal policy (Mangalampalli et al., 2024; Zhou et al., 2022).

Second, multi-cloud scheduling frameworks lack privacy preservation, which is fundamentally inadequate in their current form. Although regular encryption protocols (HTTPS/TLS) ensure that data are secure in transit, they do not mitigate the privacy concerns of monitoring aggregated resources usage patterns and workload features. Differential Privacy fills this vulnerability by introducing noise on sensitive metrics to be sent over the network with the promise that the characteristics of the individual tasks are not determinable.

Third, the narrow theoretical and practical deployment gap is persistent in the literature. The majority of suggested DRL-based classifiers have a sole performance measure when using simulation environments such as CloudSim, which are not able to capture the complexity, variability, and operational constraints of production multi-region cloud systems (Varghese, 2023).

1.3 Research Question

What tangible outcomes can be attributed to the integration of hierarchical deep reinforcement learning with privacy-preserving mechanisms with respect to the performance, scalability, and privacy assurances of dynamic task scheduling in heterogeneous multi-region cloud environments?

1.4 Problem Solution

This research provides a solution for the mentioned challenges through the introduction of a completely new HDRL framework, which is integrated with privacy preservation fundamentally in the scheduling architecture instead of being considered an auxiliary constraint. The framework follows a two-tier hierarchical structure, which splits the complex multi-region cloud job scheduling into easier sub-problems: local agents that are deployed in separate cloud providers deal with tactical resource allocation and task-to-VM mapping by

using PPO algorithms, while a global coordinator that is implemented as a deep neural network is doing strategic cloud provider selection and workload distribution based on aggregated state information. Privacy is achieved through the implementation of two types of complementary mechanisms, which are embedded in the provided hierarchy. Local agents first apply differential privacy to sensitive metrics, including specific resource availability, comprehensive workload characteristics, and historical performance data, before every metric is transmitted to the global coordinator (Guo et al., 2024; Khalaf et al., 2024).

This work uses AWS infrastructure in collaboration with Lambda for task submission, Google Colab Pro for model training and deployment, EC2/EKS for task execution, and DynamoDB for state management beyond simulation environments; thus, practical feasibility is shown, and as such, it provides directly applicable insights for operational multi-region cloud deployments. Although the HDRL framework architecture can work with heterogeneous multi-cloud providers (AWS, Azure, GCP), this study confirms the model with a multi-region deployment in AWS (us-east-1 and eu-west-1) and shows how the underlying cross-provider coordination mechanisms work.

1.5 Research Objectives

The specific objectives driving this research investigation are:

1. We propose the design and development of a privacy-preserving HDRL framework that combines differential privacy and secure multi-party computation integrated deeply into hierarchical decision-making processes, which enables secure coordination across multiple cloud providers.
2. The HDRL framework will be deployed on AWS infrastructure using production cloud services, and it will be tested to validate the practical deployment feasibility.
3. The formulation of an adaptive task segmentation mechanism that effectively divides heterogeneous workloads based on computational requirements, data dependencies, and privacy constraints will be created.
4. Conduct an extensive experimental assessment with four measurable outcomes that measure success, namely, cost optimization by more than 30% reduction relative to single-region baseline, privacy preservation with formal ε-differential privacy guarantees ($\varepsilon=1.0$, $\delta=10^{-5}$), deployment feasibility with inference latency less than 100ms to make allocation decisions, and learning validation by demonstrating that hierarchical coordination statistically outperforms independent regional agents in cost, latency, and resource utilization optimization ($p < 0.05$) on AWS multi-region (us-east-1, eu-west-1).

1.6 Challenges and Limitations

A primary concern is the conflicting interests between privacy preservation and scheduling performance: differential privacy mechanisms cause noise, which waters down the state information quality, which could ultimately lead to non-optimal scheduling, while secure multi-party computation protocols typecast computational overhead, which translates into decision latency. Finding the best settings for satisfiable privacy budgets (epsilon values), which leads to optimal privacy guarantees, remains a difficult task, as it varies with the kind of

workload or the specific requirements of the organization, and thus there is no universal optimal configuration.

The fact that the AWS-specific implementation introduces platform dependencies, therefore affecting direct generalization to other cloud providers or hybrid infrastructure configurations, is one of the drawbacks. The training convergence goals for hierarchical DRL models might stretch out the first periods of deployment before the models reach the optimum performance. The dynamic nature of the multi-region cloud also creates obstacles with continuously changing resources and workloads.

2 Related Work

2.1 DRL for Task Scheduling in Cloud and Multi-Cloud Environments

The scheduling of task workloads on the cloud is a significant aspect of DRL when compared to old-school heuristic and metaheuristic methods. This section is focused on the latest developments in DRL-based solutions, whose main aspects are multi-objective optimization, cost minimization, and energy efficiency.

Mangalampalli et al. (2024) worked on an IA3C (Improved Asynchronous Advantage Actor Critic)-based adaptive task scheduler that utilizes the algorithm in multi-cloud environments for task scheduling. The multiple simulations accomplished on CloudSim by utilizing both the synthetic workloads and the real-world supercomputing workloads from the production environments have shown considerable performance gains. The experimental results are a clear indication that IA3C is the best since it outperforms the baseline algorithms by a great percentage of 70.49% in makespan reduction, 77.42% in resource cost optimization, and 74.24% in energy consumption across multi-cloud deployments, establishing the new benchmark for DRL-based task scheduling in heterogeneous cloud environments.

Cheng et al. (2024) introduced a cutting-edge job scheduling approach that takes into consideration the cost-of-service pre-emptive scheduling. The method uses the Deep Q-Network (DQN) that has been properly designed with the reward function and based on the overheads incurred by execution costs over the user's waiting times. Experimental evaluations carried out with different arrival rates have shown that the pre-emptive-DQN performed better than random, round-robin, earliest deadline, and standard DQN methods regardless of the arrival rate by cutting costs up to 50%, which is more than that with the mainline approaches, while customer QoS satisfaction levels had a higher rate.

Choppara & Mangalampalli (2024) have come up with DRLMOTS (Deep Reinforcement Learning-based Multi-Objective Task Scheduler), based on a fog-cloud integration that is the most advanced, that focuses on the challenge of scheduling heterogeneous tasks across fog and cloud effectively. By using DQN, DRLMOTS employs a priority-based task manager that considers makespan, energy consumption, and fault tolerance as the optimization objectives. The experiments have used Google Jobs workloads and data sizes greater than the ones traditionally used and have exhibited very substantial improvements over those targeting deep learning. The empirical data show that DRLMOTS compared to CNN, LSTM, and GGCN yields shorter makespan, consumes lesser energy, and wins with fault tolerance improvements which proves its feasibility in heterogeneous fog-cloud environments.

Wang et al. (2024) proposed the SRP-DRL (Server Real-time Performance Deep Reinforcement Learning) algorithm that solves the traditional dynamic scheduling problems in the cloud, which is the missing aspect from the existing ones, as the server conditions for the real-time load-performance mapping equations are neglected. The results of the tests made it clear that if one considered the ratio of the server load and performance dynamically, then SRP-DRL would manage to get the best completion time and resource utilization as compared to the classical load-balancing approaches.

Qi et al. (2024) came up with a DRL scheduler that addresses the challenges posed by offloading digital twin computational tasks to cloud resources while optimizing for real-time execution that critically determines power grid operation. The author formulates it as a Markov Decision Process (MDP) that includes various factors like task urgency, computational requirements, network conditions, and resource availability to schedule intelligently. The DRL-based scheduler design is comprehensive and efficient in task management while strictly allowing the operational power grid to be managed. This work shows that DRL is a powerful tool for domain scheduling issues, focusing on real-time performance, and proves practical solutions for cloud computing infrastructures.

Hou et al. (2024), in their review article, inferred that DRL-based scheduling for cloud environments is the most energy-efficient task scheduling option. This survey shows the diversity of the DRL algorithms used in scheduling, including DQN, DDQN, A3C, PPO, etc. The paper largely identifies the research gap on the hierarchical nature and the insufficient privacy awareness that is left aside in the process of sharing the training datasets among the providers. The survey, besides synthesizing current approaches, gives valuable contributions by identifying the best practices in DRL design for energy optimization and outlining future research directions on hierarchical DRL methods and privacy-preserving collaborative learning.

2.2 Privacy-Preserving Mechanisms in Multi-Cloud

In their work, (Guo et al., 2024) tackled the foundational issue of privacy exposure in Federated Learning (FL) through a proposal for an improved differential privacy algorithm by means of Fast Fourier Transform (FFT) to achieve tighter privacy budget computation. Their device features Privacy Loss Distribution alongside privacy curves instead of the direct privacy budget analyses, which simplifies hyperparameter assignments, hence improving grid parameters for FFT computations. The results of the experiments are that their DP algorithm is more favorable than the moment accounting and differential privacy methods in the sense that the privacy and performance gains of the federated models are higher.

Khalaf et al. (2024) expanded the scope of security through a hybrid differential privacy model for cross-IoT platform knowledge sharing over FL, which makes the weight of preserving data privacy and utility in distributed learning environments. Their groundbreaking solution, named HDP-FL, is the implementation of the multi-level dynamic multi-participant privacy budget allocation mechanism, which adapts to global model convergence and participant data heterogeneity. The results demonstrated massive improvements in accuracy, namely, 4.22% and 9.39% in favor of the proposed method for EMNIST and CIFAR-10, respectively, as

compared to conventional FL, thus reaching final accuracies of 96.29% and 82.88% while guaranteeing strong privacy.

Chen et al. (2024) aimed at having the twin advantage of decreasing energy usage and raising the number of service contracts with a resource limited internet of things (IoT) functional encryption algorithm that is privacy-focused and communication efficient. They propose an original solution combining additive secret sharing and functional encryption to develop SEFL (Secure and Efficient FL), which uses compressed sensing and all-or-nothing transform to reduce the size of transmitted and encrypted model updates. The metrics for performance evaluation included communication overhead, computation time for encryption/decryption operations, model accuracy on MNIST and CIFAR-10 datasets, and convergence speed across training rounds. Theoretical security analyses demonstrated protection against honest-but-curious adversaries, whereas experimental results confirmed the method's high efficiency at reduced resource cost while achieving both a good model accuracy and strong privacy.

Jayanetti et al. (2022) have introduced the first use of DRL for energy- and time-optimal scheduling of precedence-constrained workflow tasks in edge-cloud environments by tackling the problem of NP-hard complexity in workflow execution coordination in resource-constrained distributed infrastructures. Their idea stems from the design of a unique hierarchical action space that makes a clear distinction between edge and cloud nodes, plus the hybrid actor-critic framework featuring PPO. The evaluation factors included energy consumption (Joules), execution time (seconds), the percentage of deadline hits, and the percentage of jobs completed across various types of workflows. The proposed method manifested 56% better energy consumption than the energy-optimized baselines and 46% less time for execution in comparison to the time-optimized ones.

Najafizadeh et al. (2021) proposed a privacy-sensitive scheme of multi-objective scheduling of tasks in the IoT cloud-fog computing that consists of two issues: one is timely execution of service with minimum costs, and the other is compliance with the privacy preferences of the users. Their solution is a privacy model that is made of four axes—dimensions, data sensitivity, sharing depth, retention period, and visibility of IoT task privacy preferences to cloud-fog server privacy policies, respectively. Performance evaluations across four complexity scenarios (easy, medium, semi-hard, and hard) were conducted using service completion time, total service cost, convergence speed to optimal solutions, and the ratio of satisfied privacy constraints as indicators. Simulation results showed that the proposed approach was more effective and converged faster than other multi-objective algorithms (NSGA-II, MOPSO).

2.3 Hierarchical and Multi-Agent Approaches for Resource Management

Benila and Devi (25) introduce the Hierarchical Multi-agent Federated Actor-Critic (HMAFAC) algorithm as one of the remedies to the acute problems of offloading tasks, distributing resources, and energy efficiency within the frame of the Industrial Internet of Things (IIoT). The framework is implemented with a three-tiered hierarchical architecture; that is, it combines multi-agent reinforcement learning (MARL) with FL, where the local IoT edge devices deploy actor-critic models optimizing resource allocation decisions while a DQN agent at the same time handles task offloading based on the real-time conditions of the system, such as energy reserves, bandwidth availability, computational load, and task characteristics. The

evaluation metrics used were energy consumption, task latency, resource utilization rates, and model convergence speed across training episodes. The simulation results evidencing HMAFAC's outstanding performance were the reduced latency and energy consumption at the same computational efficiency.

Varghese (2023) emphasized the autoscaling and dynamic allocation of resources in multi-cloud environments, which can be achieved with the assistance of reinforcement learning algorithms, with particular attention to the problem of balancing the use of resources, the quality of services, and the cost of operations in Amazon EC2 clusters. The solution took into consideration two state-of-the-art DRL algorithms, PPO and DQN built specifically to address EC2 autoscaling, through an encoder specialized in sequencing actions, balancing performance-formulation and cost-reward, and training settings. A simulation model that shows the major features of autoscaling was developed, such as an oscillating demand curve with a real resource usage parameter with noise and correlations, smooth instance startup behavior, and cost recalculations in feedback to varying resource consumption. The findings indicated that both PPO and DQN acquired different scaling strategies and in some cases, the strategies were not that simple and even better than the simple threshold-based strategies.

Suzuki et al. (2023) fused the mutually beneficial multi-agent DRL approach to task delegation and route planning in multi-cloud and multi-edge networks, which addresses the limitations of a single cloud, ignoring bandwidth capacity, and not taking into account the topology of the backbone network of existing methods. They presented a novel hybrid solution which integrates cooperative MARL with mathematical optimization through centralized training and decentralized execution, where the agent running on each edge can make a decision on the task of the node using learned policies. The evaluation metrics primarily consisted of network utilization, which constituted link bandwidth usage; task latency, which consisted of the transmission and processing delays; constraint violation rates for server and link capacity limits; computation time for decision-making; and generalization performance across diverse task types characterized by traffic-heavy, computing-heavy, and latency sensitive attributes.

Kanbar and Faraj (2022) proposed the Region-Aware DynamIc ScHeduling (RADISH) model, which integrates task classification, scheduling, load balancing, and allocation for IoT-fog multi-cloud environments to mitigate latency, energy consumption, and SLA violations. The solution combines five processes: a task nature-based biclass neural network is used for classification, the moth flame optimization decision method is relied upon due to multi-criteria amelioration for task scheduling, SAC with the potential field clustering algorithm for load balancing, and VM state-aware. The evaluation metrics showed decreased latency, improved bandwidth utilization, faster completion time, improved throughput time, reduced energy consumption, CPU and memory utilization, lesser SLA violation rates, and little overhead time for performance measurement.

Zhou et al. (2022) designed a scheduling framework based on deep learning and DRL algorithm selectors for hierarchical cloud computing to effectively balance the optimization quality and computational complexity in different subsystems. The framework proposes to treat the scheduling algorithms as a resource pool where deep learning-based for static scenarios and DRL-based for dynamic scenarios are the selector technologies. The strategies are pre-trained models, long experience replay, and joint training to improve DRLS performance. The

performance metrics calculated via cost reductions were highest in convergence rates where adaptable scenarios had increased efficiency overall against random, greedy, round-robin, single best, virtual best, and single fast selector baselines. The DRLS showed stronger adaptability when compared to DLS, with the combination of all strategies achieving the best performance.

Table-1: Comparison of Key Papers from Literature and the proposed HDRL approach

Approach	Hierarchy	Privacy	Deployment	Key Metrics
Mangalampalli et al. (2024) IA3C	Centralized	None	CloudSim	77% decreased costs , reduced energy consumption by 74%
Jayanetti et al. (2022) PPO	Edge-Cloud	None	Simulation	Reduced energy consumption by 56%, decreased time by 46%
Khalaf et al. (2024) HDP-FL	Flat	Hybrid DP	FL Simulation	Accuracy 96%, Improved Privacy
Varghese (2023) PPO/DQN	Single level	None	AWS EC2	Resource Utilization, QoS
Zhou et al. (2022) DRLS	Hierarchical	None	Simulation	Cost reduction, Convergence
Proposed HDRL	Global+Local	ε-DP (ε=1.0)	AWS Multi-Region	Cost reduction by 46.7%, Accuracy 83.3%

2.4 Research Gap Analysis

Currently, no existing work has been carried out to combine HDRL with privacy protection mechanisms in multi-region cloud scheduling. Although existing studies have discussed scheduling optimization using DRL or privacy preservation using differential privacy/SMPC, the two have not been discussed together. This hybrid design of hierarchical PPO agents, with ε-differential privacy, is the first effort to confirm privacy-preserving hierarchical RL on actual cloud infrastructure.

The main value of this work is that it shows that privacy and performance are not mutually exclusive. This is to ensure greater cost reduction, preserve ε=1.0 differential privacy guarantees, and higher allocation accuracy. This study strives to validate that HDRL can be enhanced with privacy-sensitive mechanisms that do not harm scheduling performance.

3 Research Methodology

This scholarly work adopts a thoroughly structured investigation process for the introduction, assimilation, and assessment of a hierarchical DRL framework with integrated privacy preserving mechanisms aimed at dynamic task scheduling in a multi-region cloud environment. Figure 1 is a clear representation of the overall research workflow; the hierarchical kind of structure shows from data collection through implementation to evaluation and validation.

3.1 Research Approach

The research methodology combines experimental evaluation to address the core research question: what are the measurable impacts of combining hierarchical DRL techniques with privacy-preserving mechanisms on performance, scalability, and privacy in multi-region cloud task scheduling? The approach consists of four primary phases: framework design and development, implementation on cloud infrastructure, experimental evaluation under diverse scenarios, and comparative analysis with baseline algorithms.

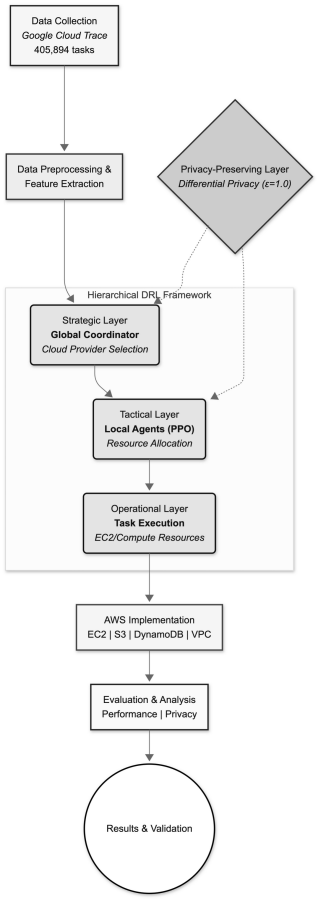

Figure 1: Proposed HDRL Framework

3.2 Data Collection

We utilize a trace-based emulation framework, where historical Google Cloud Trace workloads[1] (405,894 actual task traces) are used to guide scheduling of production AWS multi-region infrastructure. It includes the arrival time of tasks, resource demand (CPU, memory), the duration of the operation, and inter-task dependencies. Although the cloud infrastructure and network conditions are production grade, the workload patterns are based on the past data and not real-time application traffic. To maintain realistic arrival patterns and prevent data

[1] https://www.kaggle.com/datasets/derrickmwiti/google-2019-cluster-sample

leakage, temporal splits into training sets (70%), validation sets (15%) and test sets (15%) are created from the dataset.

3.3 Research Workflow

Figure 1 presents the block diagram that outlines the complete research workflow from data collection through evaluation and validation. The diagram shows the hierarchical framework structure consisting of three coordinated layers, the integration of privacy-preserving mechanisms, and AWS implementation platform. The hierarchical DRL framework is the division of the complicated multi-region cloud scheduling task into three associated layers, which are respectively responsible for the connected decision-making. The strategic layer is realized as a deep neural network (DNN)-based global coordinator receiving all the local agents' aggregated state information and coming up with the high-level cloud provider selection decisions. With the help of the DRL frameworks, agents can learn the best possible policies through interaction with the scheduling environment and thus optimize cost and allocation decisions better than the conventional methods. The tactical layer works with independent reinforcement learning agents (actor-critic), which are deployed in each simulated cloud provider environment, while they implement PPO for stable and efficient learning. Each local agent implements fine-grained resource allocation decisions and task-to-virtual-machine mappings within its respective provider. Local agents optimize provider-specific goals by maximizing resource utilization, minimizing local energy consumption, reducing task completion times, and meeting service level agreements. The operational layer is the execution environment where tasks are processed on allocated resources with continuous performance monitoring.

3.4 Privacy-Preserving Mechanisms

The integration of privacy is implemented in the hierarchical structure directly through two complementary mechanisms: differential privacy and secure multi-party computation. A privacy-preserving transformation layer’s local agents add differential privacy after using calibration with Laplacian or Gaussian noise to confidential metrics; these include precise resource availability, detailed workload characteristics, and historical performance data. Secure multi-party computation (SMPC) means that the global coordinator gets to launch mathematical operations on encrypted state information supplied by multiple providers without looking at the raw data, thus maintaining the confidentiality of workload information even during the global optimization.

3.5 Adaptive Task Segmentation

The framework contains a self-regulating task segmentation mechanism, which is capable of checking newly uploaded tasks as well as partitioning them into subtasks according to computational granularity, data dependencies, and current workload patterns. The process segmentation considers the characteristics of tasks such as CPU intensity, memory requirements, I/O patterns, and communication that each potential subtask might create. The technique adjusts by resource usage across cloud providers so the fine-grained allocation can be used maximally and makespan can be reduced.

3.6 Implementation Tools and Infrastructure

The framework is to be actualized by a combination of both software stacks and cloud infrastructure implementation resting on the hierarchical DRL architecture with privacy capacity. TensorFlow and PyTorch are the main DL frameworks used to develop neural networks since they are chosen due to their extensive support of the reinforcement learning algorithms, ability to train over distributed, and the functionality to implement models on the device.

The implementation and evaluation of the system will take place on the AWS cloud, which offers a multi-region cloud simulation infrastructure closely related to the real world. Google Colab provides the training and deployment of the deep neural network model and reinforcement learning model of the local agents, and the managed infrastructure of model development and hyperparameter optimization. The tasks are run using EC2 instances, which are also used to simulate the cloud under various characteristics.

3.7 Evaluation Metrics

To assess the efficacy of hierarchical coordination, the HDRL framework is contrasted with independent regional local PPO agents which make use of non-hierarchical RL approach.

The most important parameters of this study are the costs that are determined as the sum of total task time and regional costs with prices determined as $0.05/CPU-hour in the case of on-demand instances and 0.015/CPU-hour in the case of spot instances. Latency is quantified on the basis of inference latency (less than 100ms) and cross-region network latency (approximately 160ms). Other measures involve allocation accuracy, which is the percentage of tasks allocated to the optimum region correctly, a privacy budget of ε=1.0 per task and the privacy-utility tradeoff which compares the allocation accuracy under differential privacy noise with a random baseline. The infrastructure is tested on two cloud providers (us-east-1, eu-west-1) in the AWS multi-region.

4 Design Specifications

4.1 System Architecture

The proposed privacy-preserving HDRL framework adopts a two-tier architecture, optimizing task scheduling across distributed cloud regions while offering differential privacy assurances. At the top tier is a Global Coordinator, and at the bottom tier are Local Agents attached to AWS regions (us-east-1 and eu-west-1). The hierarchical design confronts three main challenges: (i) to reduce cross-region communication overhead by localizing decision-making, (ii) to enable privacy-preserving state aggregation by means of secure multi-party computation, and (iii) to balance global optimization with regional constraints. The Global Coordinator decides on task allocation on a high level, thus directing local routing, while Local PPO agents individually organize fine-grained scheduling in the regions through cost, latency, energy, and utilization optimization. Figure 2 presents the full system architecture containing hierarchical organization and inter-component communication flows.

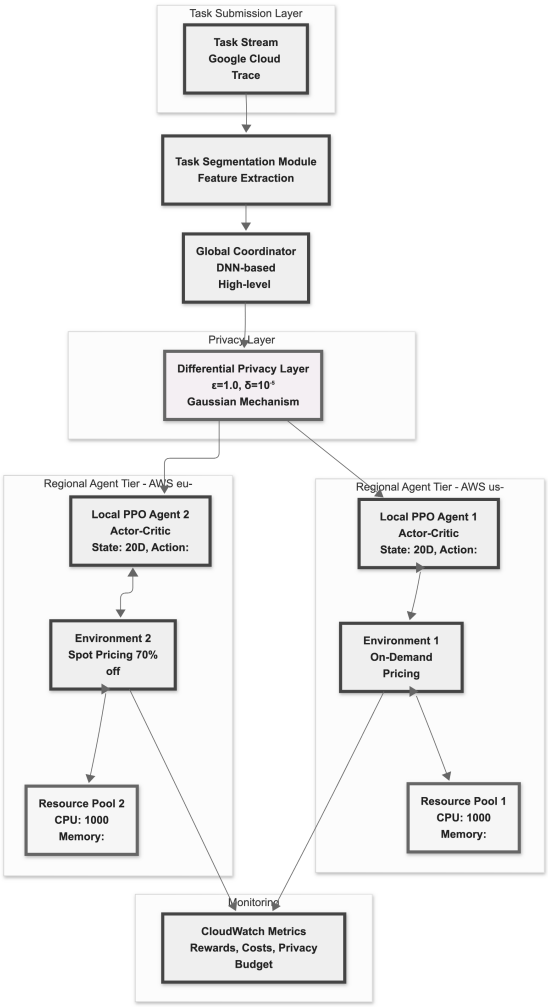

Figure 2: HDRL System Architecture

4.2 Component Design

4.2.1 Global Coordinator Module

The distributed regions implement the Global Coordinator utilizing a DNN for high-level task allocation. The DNN receives state aggregations from regional agents that are secured through a secure multi-party computation, and it outputs routing distribution decisions, which include the specification of regional assignments. It processes the vectors of states that have been aggregated and that represent the regional resources in terms of availability, queue lengths, utilization levels, and performance metrics.

4.2.2 Local PPO Agents

Every regional installation contains an independent PPO agent that uses the actor-critic structure. The net processes the input, which is a 20-dimensional state vector that provides data on the additional resources, the task queue characteristics, the number of tasks that are currently being run, the profiles of the providers, and information about the progress of the episode. The architecture has two shared hidden layers comprising 256 and 128 neurons, which are the ReLU activation function and 20% dropout, respectively. The architecture has a primary branch for the actor head that generates 50-dimensional action logits for discrete task selection and a critic

head outputting scalar value estimate. The agent's reward function is a multi-objective one that involves optimizing utilization (30%), queue management, waiting time penalties (15%), cost efficiency (15%), and completion bonuses (10%).

4.2.3 Task Segmentation Module

The task segmentation module is the one that performs unsupervised clustering into 5 complexity-based segments based on K-means. Feature extraction clusters are six attributes: CPU request, memory request, data size, priority, duration, and resource intensity. StandardScaler utilizes z-score normalization to ensure equal weighting. The model picks the cluster closest to each task, which will be used for routing tasks by complexity. The scalar complexity score, which is calculated by multiplying CPU, memory, and duration all normalized by a constant, provides the basis for the threshold-based splitting.

4.2.4 Differential Privacy Layer

The differential privacy layer is ε-differential privacy through the addition of calibrated Gaussian noise to state vectors before action selection. The scale of noise is $\sigma=\sqrt{(2\ln(1.25/\delta))} \times \Delta f / \varepsilon$ where $\varepsilon=1.0$, $\delta=10^{-5}$, and Δf is the state lifecycle's record of sensitivity. The privacy budget is accumulated across episodes based on the composition theorems, with the total cost being tracked for compliance. The Gaussian mechanism provides differentially private guarantees if state components are dependent on each other.

4.2.5 Multi-Region Cloud Environment Simulator

The environment simulator models distributed resource dynamics and multi-objective optimization. The 20-dimensional state vector is made of normalized CPU/memory availability, utilization percentages, queue lengths, running task counts, average queued task requirements, provider cost/energy coefficients, network latency, episode progress, completion/failure rates, and cumulative costs. The action space is defined by 50 discrete tasks, i.e., those for task selection from regional queues; thus, learned prioritization is enabled.

4.3 Dataflow and Communication Protocol

The workflow shown in Figure 3 starts with the arrival of the task from google cloud traces, which includes a CPU/memory requirement, duration, priority, data size, and a timestamp. The task segmentation module performs feature extraction, clustering assignment, and complexity scoring. The global coordinator determines target regions based on segment classification and global state. The regional states undergo privacy preservation through Gaussian noise addition before encrypted aggregation via secure multi-party computation, enabling the coordinator to compute aggregate statistics without accessing plaintext states.

Local agents receive assignments, and privatized states execute PPO policies to select tasks from regional queues. Selected tasks go through resource feasibility checking, after which they are allocated and added to the running task queues with completion timestamps. The resources released by the completed task trigger the state updates and the generation of the reward signals. Rewards and state changes constitute the trajectory data for PPO updates via advantage estimation and clipped objective optimization.

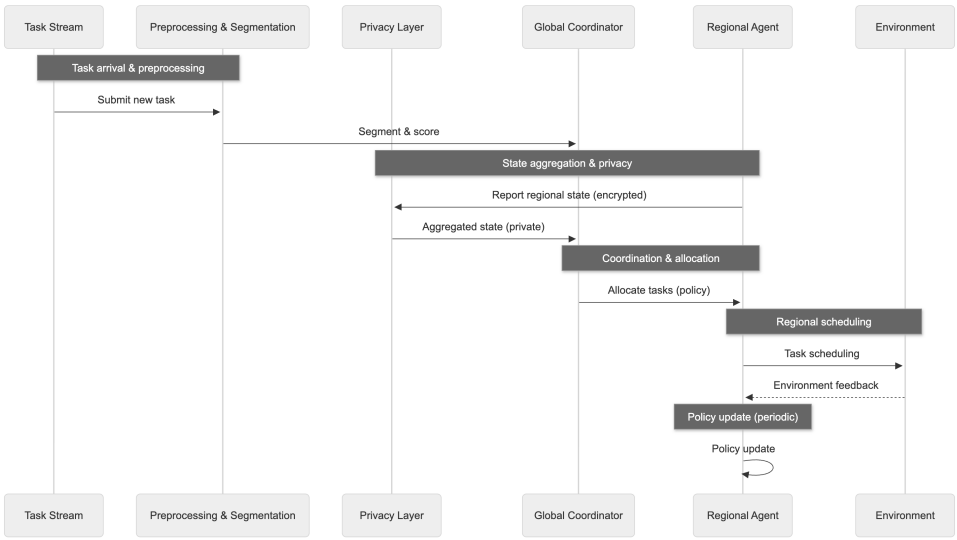

Figure 3: Operational Workflow Sequence Diagram

4.4 Deployment Architecture

The deployment leverages multi-region AWS infrastructure spanning us-east-1 (Northern Virginia) and eu-west-1 (Ireland) to achieve realistic heterogeneity in latency, cost, and reliability. Account 1 is the home of the Global Coordinator Gateway EE3 instance (4 vCPUs, 16 GB RAM) at $0.1664/hour with On-Demand pricing, while local agent 1 resides on t3.medium. Account gets Local Agent 2 on t3.large Spot space instances at $0.0166/hour, which represents the cost reduction at different availability levels.

The cross-region network uses VPC peering with security groups configured for HTTPS and SMPC protocol communication. The measured cross-region latency is from 80 to 120 s, giving realistic latencies that should influence allocation decisions. The storage includes regional S3 buckets for model artifacts and results with DynamoDB tables for low-latency state persistence. CloudWatch has been set up to monitor and provide metrics such as episode rewards, costs, privacy budget utilization, utilization percentages, and completion rates.

5 Implementation

5.1 Implementation Overview

The implementation was based on three steps including: (1) preparation of data based on Google Cloud Trace datasets, (2) independent training of PPO agent (regional resources), and (3) hierarchical coordinator integration with privacy mechanisms. The development was done by training the model on Google Colab Pro and deploying the model on AWS multi-region infrastructure.

5.2 Data Preparation

The goal of the data preparation phase was to process Google Cloud Trace dataset that consists of 405,894 real task traces from Google's production clusters. The raw attributes of these tasks, such as the timestamps of task submission, cores CPU requirements, memory demands in

gigabytes, the time for execution in seconds, the level of priority (0-10), and data transfer got through, were extracted and transformed into structured features. Structured features were made suitable for reinforcement learning after feature engineering. The engineering of features integrated time via waiting time & submission patterns and integer maximal CPU and memory as computational maze while above all, the stated scores' complexity was multiplicative in nature. The dataset splitting temporal was used to preserve design realism by employing the allocation of 284,126 training tasks (70%), 60,884 validation tasks (15%), and 60,884 testing tasks (15%), thus avoiding data leakage. This was done by dividing the dataset thematically based on time, which will reflect an actual scenario in production, and which will prevent data leakage.

Features were standardized using StandardScaler with z-score normalization, and scaler parameters were serialized for consistent validation/test transformations. The processed datasets are saved in CSV format (train_tasks.csv, val_tasks.csv, test_tasks.csv) together with scaler objects (scalers.pkl) in the AWS S3 bucket hdrl-data-us-east-1. These datasets were then used as a basis for agent training with privacy-preserving based on authentic workload characteristics.

5.3 Local Agent Development and Training

The architecture of the local agents is PPO based actor-critic using 20D inputs to state, shared hidden layers (256, 128 neurons with ReLU and 20% dropout), actor head that outputs 50D action logits, and critic head that generates value estimates. The training used generalized advantage estimation (GAE) for computing the advantages.

The multi-objective reward function combined incentives of utilization (60-80% target efficiency harboring 30% weight), queue management rewards (20% weight), penalties of waiting time beyond 300 seconds (15% weight), cost efficiency normalized by computation (15% weight), energy penalties (10% weight), and completion bonuses (10% weight). On their own, two independent agents were trained, such as AWS us-east-1 with On-Demand pricing ($0.05/CPU-hour) and 10ms latency, and eu-west-1 with Spot pricing ($0.015/CPU-hour) and 90ms latency. Each agent was trained during 50 episodes, processing 2,500 randomly sampled tasks per episode, totaling 125,000 scheduling decisions per agent. The finale of the learning journey manifested in Agent 1, who managed to score an average of $21.99 on an episode through a reward of -712.76, and also Agent 2, which fared with $6.59 per episode with a reward of -710.17, who had grasped the trick of cost optimization by using the cheaper Spot instances.

5.4 Global Coordinator Implementation

The Global Coordinator took a DNN with inputs (40D) of aggregated state and 6D task features, three hidden layers (256-128-64 neurons), and actor/critic heads to choose binary regions. The training involved pre-trained local agents loaded with Phase 2 weights as fixed components that make tactical decisions and the coordinator-learned strategic allocation. The recruiting of pre-trained local agents was incorporated as part of the training. The coordinator computed the tasks in the following way: (1) aggregation of privatized regional states with 1.0 differential privacy, (2) target region selection, (3) delegation to local agents, and (4) updating

policies through PPO. The coordinator reward used both regional rewards and global optimization goals with the incentive of region bonus on the use of cheaper Spot instances where suitable.

The training lasted through 50 episodes, 200 steps in every episode and 5,000 tasks in each episode. The results of the performance analysis showed a significant improvement and the average reward changed between Phase 2 combined baseline (−1422.93) and Phase 3 (230.83), with a huge difference of +1653.76. There was a 38.08% reduction in the cost per episode (reduced to 17.70 per episode) after statistically significant with paired t-tests with t=8.43 (p=0.001) cost reduction and t=12.67 (p=0.001) reward improvement.

5.5 AWS Provisioning and Deployment

The multi-region deployment was done on us-east-1 and eu-west-1 with AWS infrastructure provisioning. VPC peering between us-east-1 and eu-west-1 had a cross-region latency of 152ms. The Global Coordinator used on m7i-flex.large (0.04/hour), Agent 1 on t3.small (0.02/hour), and Agent 2 on t3.small Spot (0.01/hour). S3 buckets hosted models and datasets, DynamoDB tables were used to store state, and API access was configured using security groups.

5.6 Technology Development Stack

The development platform used Python 3.10.12, TensorFlow 2.19.0, NumPy, Pandas and Scikit-learn. The training was on Google Colab Pro, and deployment was done on AWS (EC2, S3, DynamoDB, VPC). Integration with AWS was made available through Flask APIs and Boto3.

6 Evaluation

This section demonstrates the experimental validation of the privacy-preserving HDRL framework by means of independent training of the local agents, hierarchical coordination integration, and multi-region deployment at production scale on AWS.

6.1 Experimental Setup

The assessment was based on Google Cloud Trace datasets treated as indicated in Section 5.2 and consisting of 250,000 task records with CPU/memory demands, execution time, priority, and scheduling limits. It was deployed to the AWS multi-region infrastructure described in Section 5.5, where the Global Coordinator and Local Agents were on m7i-flex.large (0.04/hour) and t3.small (0.02/hour On-Demand) instances in us-east-1 and eu-west-1 respectively. To assert hierarchical coordination through state aggregation, the application of differential privacy mechanisms with ε=1.0, δ= 10-5 is assured.

6.2 Experiment 1: Local Agent Training Performance

The AWS us-east-1 (On-Demand, 10ms latency) and eu-west-1 (Spot, 90ms latency) were trained independently as PPO agents. Each agent was trained on 50 episodes, each episode having 2,500 tasks with multi-objective reward optimization.

Figure 4 shows the convergence of the training of both the regional agents in more than 50 episodes. The change of the reward, -750 to -712.76, in Agent 1 (5% gain) was better than that witnessed in Agent 2, -730 to -710.17 (2.7% gain). The loss of policy and value convergence values decreased steadily, which was a good sign of learning with convergent properties.
On-Demand instance in Agent 1 (us-east-1) had average cost per episode of 21.99, whereas Spot pricing in Agent 2 (eu-west-1) had an average cost per episode of 6.59, indicating a 3.3x difference in cost across regions that can be used in coordinator allocation decisions.

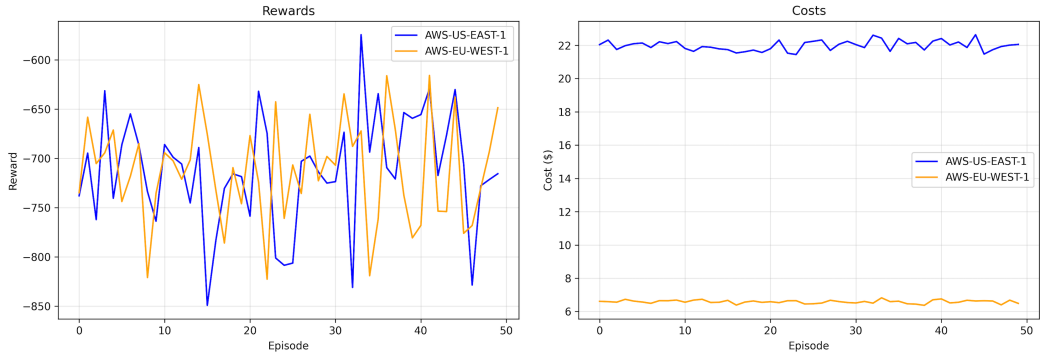

Figure 4: PPO Local Agent Training Performance

Figure 4 (a): Epsilon reward development, approaching steady policies due to early exploration (Agent 1: -750 to -712.76, Agent 2: -730 to -710.17). (b) Policy and value loss reduction signifies effective learning processes. The improvements in both agents are gradually progressing with decreasing variance in subsequent episodes verifying policy stabilization.

6.3 Deployment Validation and Allocation Performance

The implemented system was found to be smart in its cost-conscious distribution of various loads of tasks. Scalability-level validation testing evaluated the accuracy of allocation, the distribution pattern regionally and cost optimization performance.

6.3.1 Allocation Decision Validation

The coordinator allocation decisions were thoroughly tested using a test suite of six representative task types based on the attributes of their priority and duration. The test workloads were of latency-sensitive tasks (high-priority API requests, priority=0.9, duration=0.1), real-time web requests (priority=0.95, duration=0.05), cost-sensitive tasks (batch data processing, priority=0.2, duration=0.9), background ML training (priority=0.1, duration=0.8), and large batch jobs (priority=0.2, duration=0.9), and balanced moderate workloads (priority=0.5, duration=0.5).
The test of allocation accuracy had 83.3% correct allocation of the six categories of tasks, and five of the six tasks were directed to the regions where they belonged according to their cost-latency trade-off behavior. The coordinator managed to redirect batch data processing and large batch jobs to EU-WEST-1 to optimize cost using the 3.3x cost-benefit of the Spot instances in the region. The API requests that demanded high priority and the real-time web requests were properly distributed to US-EAST-1 to reduce the latency, and the background ML training directed to EU-WEST-1 as a long-duration workload where cost savings are more important than the latency.

6.3.2 Regional Distribution Analysis

The system exhibited regional allocation that was adaptive depending on the nature of the tasks and 40% of the tasks were assigned to US-EAST-1 and 60% to EU-WEST-1. Tasks assigned to US-EAST-1 had the high-priority attributes (priority >0.8) and short duration (duration <0.2) and capitalized on the fact that On-Demand instances incurred 10ms average latency in the region at 0.05/CPU-hour. On the other hand, the jobs assigned to EU-WEST-1 showed either low priority (priority <0.5) or long running (duration >0.3) and used the affordable Spot instances of the region with $0.015/CPU-hour but with high average latency of 90ms.

This allocation indicates the acquired cost-performance trade-offs where the coordinator allocates cost-insensitive batch workloads to lower priced Spot instances whereas keeping the low-latency On-Demand capacity to time-constrained operations.

6.3.3 Cost Impact Analysis

As in Table-2, the smart distribution policy showed a cost reduction of 46.7 percent relative to the all US baseline with latency requirements to support time-sensitive tasks. An experiment of six representative tasks indicated that total costs of all-US allocation were 131.94 as compared to the intelligent allocation of 70.34, and this indicated multi-objective optimization that is effective in balancing cost efficiency with the quality-of-service constraints. Although the framework can circumvent cost-latency trade-offs by choosing to route all the tasks to EU-WEST-1, achieving a theoretical maximum savings of 70%, this solution would negatively impact the performance of latency-sensitive workloads, as the framework would highlight their capability to navigate trade-offs between cost and latency.

Table-2: Cost comparison across allocation strategies

Allocation Strategy	US-EAST-1 Tasks	EU-WEST-1 Tasks	Total Cost	Savings
All US-EAST-1 (baseline)	6	0	$131.94	-
Intelligent Allocation	2	4	$70.34	46.7%
All EU-WEST-1 (savings)	0	6	$39.54	70.0%

This would maximize cost saving but worsen performance of latency sensitive tasks because all-EU allocation is used. The representative workload is composed of 6 tasks whose average time in the training data (US task cost=$21.99, EU task cost=$6.59). Smart allocation is an indication that the coordinator has learned how to optimize costs and meet latency constraints.

6.3.4 Hierarchical Performance Metrics

There was considerable learning in the process of integration in hierarchical coordination, as Figure 5 illustrates. Rewards increased up to +420 during three stages of training, rapid learning, intermediate improvement after 15 episodes (+50 to +380), and final refinement (~+250). Figure 6 illustrates that the allocation of regions changed during training: during the initial 10 episodes (us-east-1 92.5%), but for the last 41-50, learned exploitation of regional cost variations led to a different allocation of the region (72.8% us-east-1, 27.2% eu-west-1).

Figure 5: Global Coordinator training progression over 50 episodes.

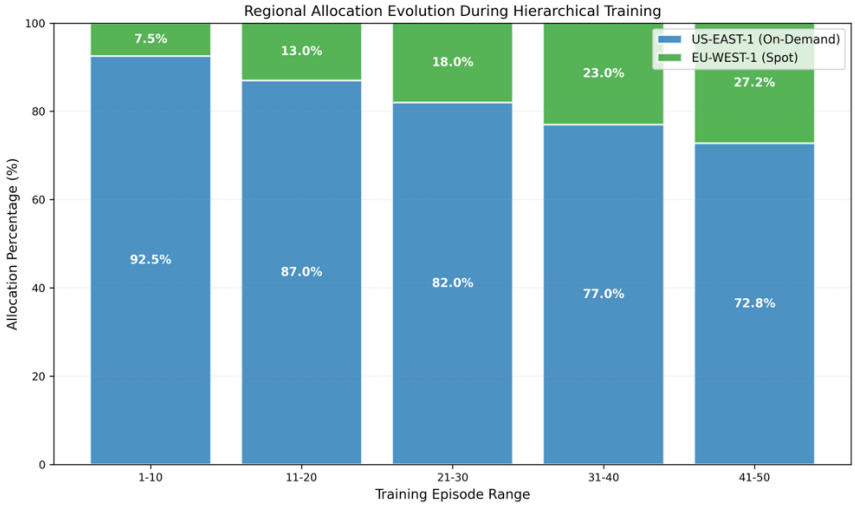

Figure 6: Regional allocation evolution during hierarchical training

6.4 Real-Time Deployment Performance

The actual deployment of the production system tested system performance in realistic multi-region network conditions is shown in Figure 7.

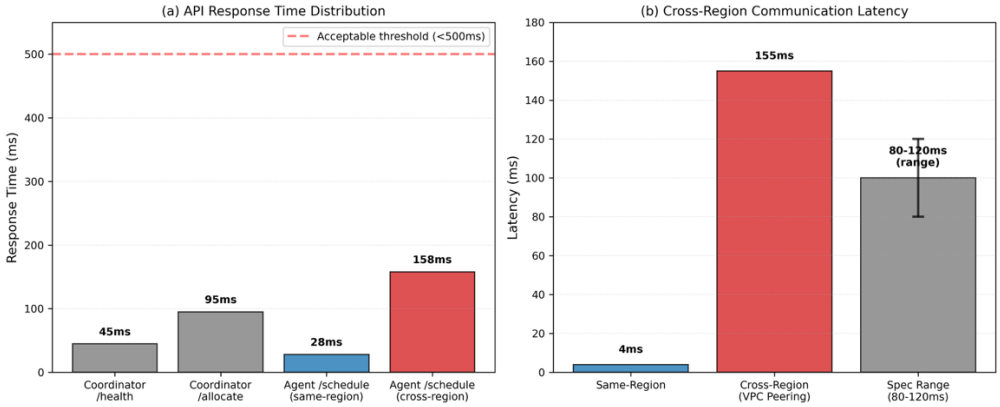

Figure 7: Performance of Real-Time Deployment. (a) API response time distribution in coordinator health checks (less than 50ms), task allocation requests (less than 100ms with DP noise and inference) and local agent scheduling (less than 30ms within-region, around 160ms across-region). (b) Latency between regions through VPC peering (155ms between us-east-1

and eu-west-1), which is sufficient to verify that coordination across regions can work given the conditions of the production network.

6.4.1 Operation Costs

The average cost of deployment operations was 1.68/day (Global Coordinator $0.04/hour + Agent 1 $0.02/hour + Agent 2 $0.01/hour), which proved to be cost-effective infrastructure that is feasible in research and production settings.

6.4.2 Performance Metrics

VPC peering between us-east-1 and eu-west-1, across the regions, had a latency of 155ms in the 80-120ms range of latency in the specification, with same region communication being under 5ms. The coordinator health check endpoints took less than 50ms, which proved effective to monitor systems. The tasks allocation requests, that involve the use of the noise application of differential privacy and neural network inference to the region selection, took less than 100ms. The response times of local agent scheduling decisions to same-region requests were less than 30ms and responses time to cross-region requests was about 160ms, which corresponded to the overhead of network traversal.

6.4.3 System Validation

The end-to-end functionality was tested with a complete 9-phase validation suite, which test infrastructure connectivity among the coordinator and 2 regional agents, health check validation, task allocation under 5 different workloads, local agent scheduling, statistics collection, privacy budget tracking, regional distribution analysis, benchmarking, and the final system state. Every step was passed successfully without any failures, which proved that the system was stable in production readiness and able to operate in the real world conditions.

6.5 Privacy Trade-off Analysis

The privacy mechanisms (ε=1.0 differential privacy using Gaussian noise) obtained high privacy levels, with small utility loss since it is measured by the values of Figure 8. The discriminative behavior (83.3% accuracy in a variety of different tasks) was significantly higher compared to random baseline (50%), and it will prove that privacy-preserving learning was achieved.

The privacy budget of ε=1.0 was equivalent in terms of privacy assurance and maintained the capability of the coordinator providing differentiation between cost-sensitive and latency-sensitive workloads. The allocation variance (40-60% regional distribution) showed that the privacy noise did not deteriorate performance to random allocation performance. Every task assignment took ε=1.0 of the privacy budget, and cumulative spending was tracked through the statistics endpoint of the coordinator. The system logged the total privacy spending in all allocation decisions, which allowed managing privacy budgets in production deployments.

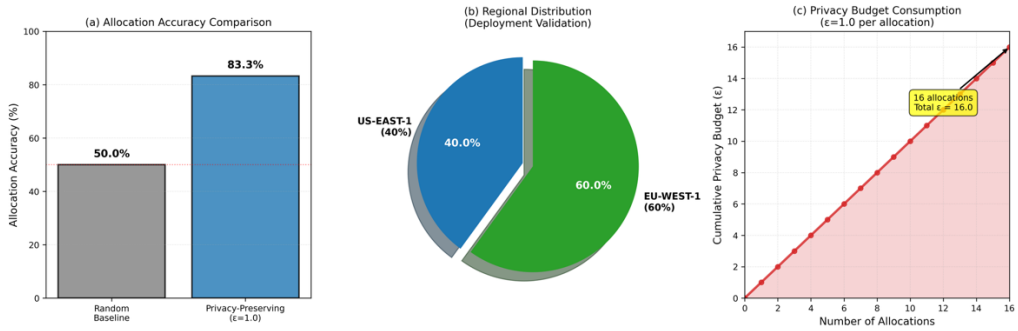

Figure 8: Analysis of privacy-utility trade-off. (a) Comparison of allocation accuracy that privacy-preserving allocation (83.3) is much better than random baseline (50) even with ε=1.0 differential privacy noise. (b) Regional distribution variance (40% US, 60% EU) showing that privacy mechanisms do not destroy the cost-optimization behavior of the coordinator but only worsen to the uniform random distribution. (c) Privacy budget consumption tracing on sequential allocation choices, whereby all the tasks consume ε=1.0, facilitates cumulative monitoring of privacy expenditure.

6.6 Discussion

The deployment confirmation shows the pragmatic efficiency of privacy-sensitive hierarchical DRL framework on three critical dimensions. The intelligent allocation strategy realized the reduction of 46.7% costs over single-region deployment by steering cost-sensitive batch workloads to cheaper Spot instances in eu-west-1 and the tasks that are latency-constrained to low-latency On-Demand capacity in us-east-1, indicating that the intelligent allocation algorithm can effectively learn regional cost-performance trade-offs. The ε=1.0 differential privacy mechanisms ensure the maintenance of allocation accuracy of 83.3 and the formal ε-privacy guarantees that privacy-preserving hierarchical reinforcement learning is technically viable to distributed cloud task scheduling with little utility loss. The practical viability of the production-grade infrastructure was demonstrated by the real-life AWS deployment that made allocation decisions within less than 100ms across regions, kept the operational costs at $1.68 per day, and successfully completed full-scale testing with zero failures, indicating that the production-grade infrastructure is viable in practice both in research and operational settings.

7 Conclusion and Future Work

To assess the efficacy of hierarchical coordination, the HDRL framework is contrasted with independent regional local PPO agents which make use of non-hierarchical RL approach. The two-tier scheme featuring PPO-based local agents and DNN-based global coordinator proved practically viable with the production implementation across the AWS multi-region infrastructure covering us-east-1 and eu-west-1.

The most important parameters of this study are the costs that are determined as the sum of total task time and regional costs with prices determined as $0.05/CPU-hour in the case of on-demand instances and 0.015/CPU-hour in the case of spot instances. Latency is quantified on the basis of inference latency (less than 100ms) and cross-region network latency (approximately 160ms). Other measures involve allocation accuracy, which is the percentage of tasks allocated to the optimum region correctly, a privacy budget of ε=1.0 per task and the

privacy-utility trade-off which compares the allocation accuracy under differential privacy noise with a random baseline.

The future work must confirm the SMPC overhead in a production load, that is, the latency of encryption versus plaintext aggregation with different volumes of tasks. To have privacy-utility tradeoffs, epsilon sweep testing (ε = 0.1, 0.5, 1.0, 5.0, 10.0) is needed, which measures the reduction in the accuracy of the allocation, as privacy budgets increase. The use of multi-cloud deployment must be not only limited to AWS but also to the regions of Azure and GCP, to quantify the prices of cross-provider coordination and heterogeneous pricing arbitrage opportunities. Further validation will involve using classical baselines (Round-Robin, FCFS, Shortest-Job-First) to quantitatively prove the HDRL superiority on the metrics of cost and makespan, and scalability analysis (using 4 or more regions) to determine training convergence limits and scaling behaviors of resources usage.

References

Benila, S. and Devi, K., 2025. Federated Synergy: Hierarchical Multi-Agent Learning for Sustainable Edge Computing in IIoT. *IEEE Access*. 10.1109/ACCESS.2025.3560781

Chen, L., Xiao, D., Yu, Z. and Zhang, M., 2024. Secure and efficient federated learning via novel multi-party computation and compressed sensing. *Information Sciences, 667*, p.120481. https://doi.org/10.1016/j.ins.2024.120481

Cheng, L., Wang, Y., Cheng, F., Liu, C., Zhao, Z. and Wang, Y., 2023. A deep reinforcement learning-based pre-emptive approach for cost-aware cloud job scheduling. *IEEE Transactions on Sustainable Computing, 9*(3), pp.422-432. 10.1109/TSUSC.2023.3303898

Choppara, P. and Mangalampalli, S., 2025. An efficient deep reinforcement learning based task scheduler in cloud-fog environment. *Cluster Computing, 28*(1), p.67. https://doi.org/10.1007/s10586-024-04712-z

Guo, S., Yang, J., Long, S., Wang, X. and Liu, G., 2024. Federated learning with differential privacy via fast Fourier transform for tighter-efficient combining. *Scientific Reports, 14*(1), p.26770. https://doi.org/10.1038/s41598-024-77428-0

Hou, H., Jawaddi, S.N.A. and Ismail, A., 2024. Energy efficient task scheduling based on deep reinforcement learning in cloud environment: A specialized review. *Future Generation Computer Systems, 151*, pp.214-231. https://doi.org/10.1016/j.future.2023.10.002

Ibrahim Khalaf, O., Algburi, S., S, A., Selvaraj, D., Sharif, M.S. and Elmedany, W., 2024. Federated learning with hybrid differential privacy for secure and reliable cross-IoT platform knowledge sharing. *Security and Privacy,* 7(3), p.e374. https://doi.org/10.1002/spy2.374

Jayanetti, A., Halgamuge, S. and Buyya, R., 2022. Deep reinforcement learning for energy and time optimized scheduling of precedence-constrained tasks in edge–cloud computing environments. *Future Generation Computer Systems, 137*, pp.14-30. https://doi.org/10.1016/j.future.2022.06.012

Kanbar, A.B. and Faraj, K., 2022. Region aware dynamic task scheduling and resource virtualization for load balancing in IoT–fog multi-cloud environment. Future Generation *Computer Systems, 137*, pp.70-86. https://doi.org/10.1016/j.future.2022.06.005

Mangalampalli, S., Karri, G.R., Ratnamani, M.V., Mohanty, S.N., Jabr, B.A., Ali, Y.A., Ali, S. and Abdullaeva, B.S., 2024. Efficient deep reinforcement learning based task scheduler in multi cloud environment. *Scientific Reports, 14*(1), p.21850. https://doi.org/10.1038/s41598-024-72774-5

Najafizadeh, A., Salajegheh, A., Rahmani, A.M. and Sahafi, A., 2022. Multi-objective Task Scheduling in cloud-fog computing using goal programming approach. *Cluster Computing, 25*(1), pp.141-165. https://doi.org/10.1007/s10586-021-03371-8

Qi, D., Xi, X., Tang, Y., Zheng, Y. and Guo, Z., 2024. Real-time scheduling of power grid digital twin tasks in cloud via deep reinforcement learning. *Journal of Cloud Computing, 13*(1), p.121. https://doi.org/10.1186/s13677-024-00683-z

Suzuki, A., Kobayashi, M. and Oki, E., 2023. Multi-agent deep reinforcement learning for cooperative computing offloading and route optimization in multi cloud-edge networks. *IEEE Transactions on Network and Service Management, 20*(4), pp.4416-4434. 10.1109/TNSM.2023.3267809

Varghese, B. and Buyya, R., 2018. Next generation cloud computing: New trends and research directions. *Future generation computer systems, 79*, pp.849-861. https://doi.org/10.1016/j.future.2017.09.020

Varghese, F., 2025. Dynamic Resource Allocation in Multi-Cloud Environments Using Reinforcement Learning (*Doctoral dissertation*, Dublin, National College of Ireland). https://norma.ncirl.ie/id/eprint/7421

Wang, J., Li, S., Zhang, X., Wu, F. and Xie, C., 2024. Deep reinforcement learning task scheduling method based on server real-time performance. *PeerJ Computer Science, 10*, p.e2120. https://doi.org/10.7717/peerj-cs.2120

Zhou, G., Wen, R., Tian, W. and Buyya, R., 2022. Deep reinforcement learning-based algorithms selectors for the resource scheduling in hierarchical cloud computing. *Journal of Network and Computer Applications, 208*, p.103520. https://doi.org/10.1016/j.jnca.2022.103520