# HetRL: Efficient Reinforcement Learning for LLMs in Heterogeneous Environments

**Yongjun He**[1] (yongjun.he@inf.ethz.ch), Shuai Zhang[2], Jiading Gai[2], Xiyuan Zhang[2], Boran Han[2], Bernie Wang[2], Huzefa Rangwala[2], George Karypis[2]

1 Systems@ETH zürich

2 aws

# The Rise of RL for LLMs

RL has become a key component of post-training pipelines for LLMs.

- Human Alighment with RLHF
- Reasoning Tasks (e.g., math and coding)
- Agentic Tasks (multi-turn, tool-use, environment interactions)


*Gemini-3.1*


*Qwen-3.6*


*DeepSeek-V4*


*Opus-4.7*

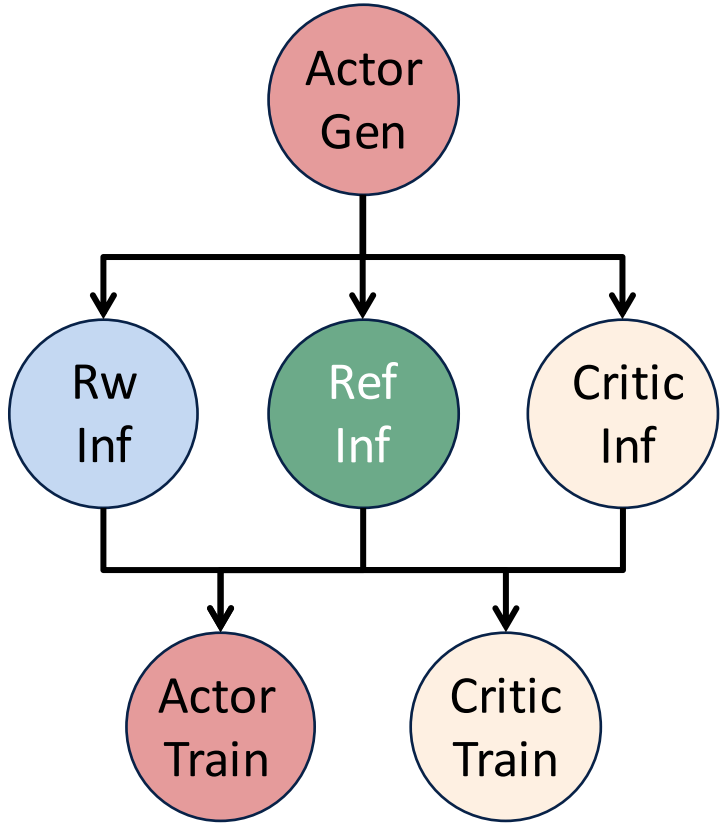

*GPT-5.5*

…

# Heterogeneity in RL Workflows

## Observations

- Multiple models and tasks with complex computational and data dependencies
  - RL models: actor model, reference model, reward model, and critic model
  - RL tasks: actor generation, reference inference, reward inference, critic inference, actor training, and critic training.

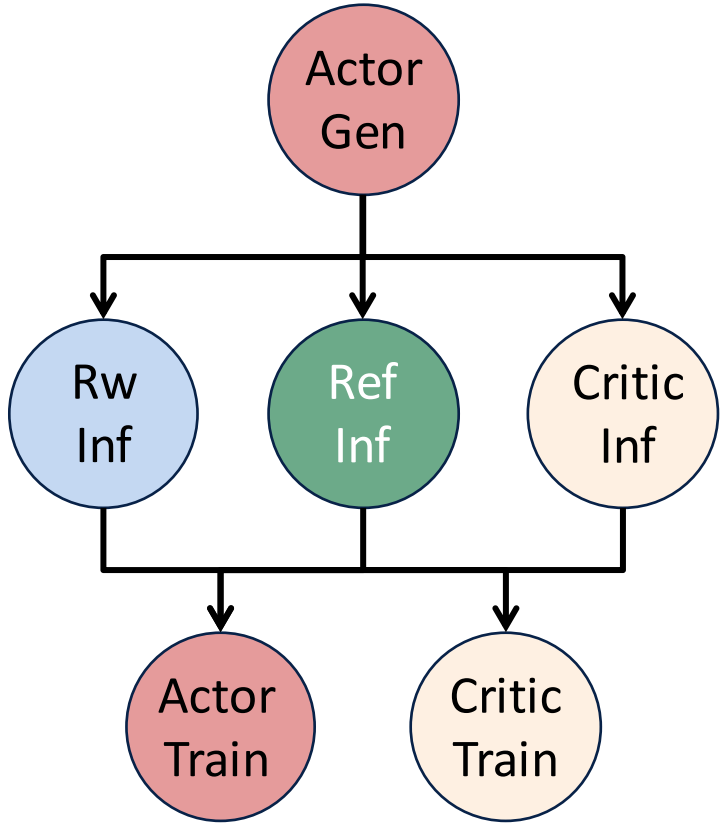



Proximal policy optimization (PPO)

# Heterogeneity in RL Workflows

## Observations

- Multiple models and tasks with complex computational and data dependencies
- Heterogeneous computational characteristics across tasks
  - Generation task: memory-bound
  - Inference task: compute-bound
  - Training task: compute-bound & inter-node communication heavy

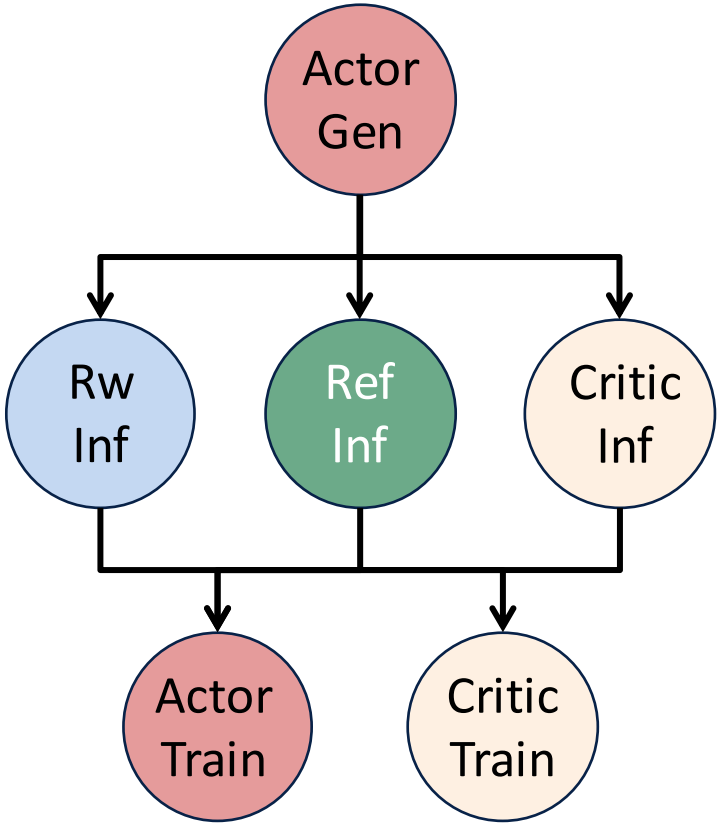



Proximal policy optimization (PPO)

# Heterogeneity in RL Workflows

## Observations

- Multiple models and tasks with complex computational and data dependencies
- Heterogeneous computational characteristics across tasks
- Heterogeneous parallelization strategies across tasks




Proximal policy optimization (PPO)

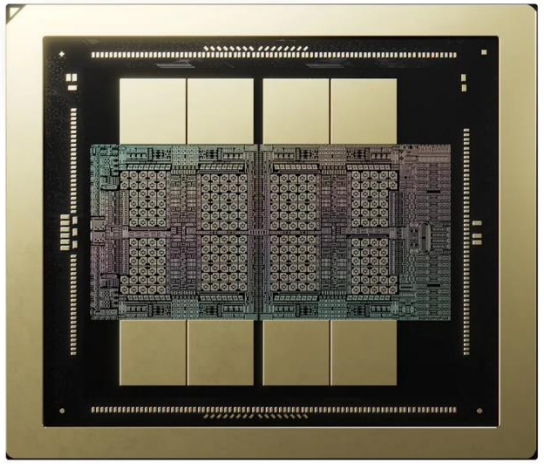
# Heterogeneity in Computing Environments

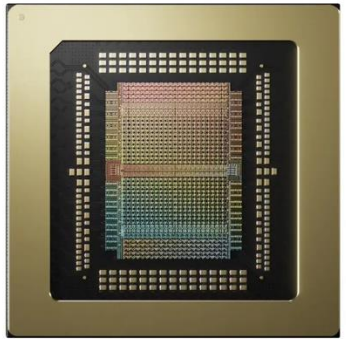
## Observations

- Heterogeneous device capabilities (e.g., compute, memory capacity, and memory bandwidth).
- Heterogeneous interconnects (e.g., network bandwidth and latency).

**Uniting Processors of Extreme FLOPS and Bandwidth**

NVIDIA Rubin GPU + Groq 3 LPU





| | |
|---|---|
| **288 GB** HBM4 | 500 MB SRAM |
| 22 TB/s | **150 TB/s** SRAM Bandwidth |
| **50 PFLOPs** (NVFP4) | 1.2 PFLOPs (FP8) |

# Heterogeneity in Computing Environments

## Observations

- Heterogeneous device capabilities (e.g., compute, memory capacity, and memory bandwidth).
- Heterogeneous interconnects (e.g., network bandwidth and latency).

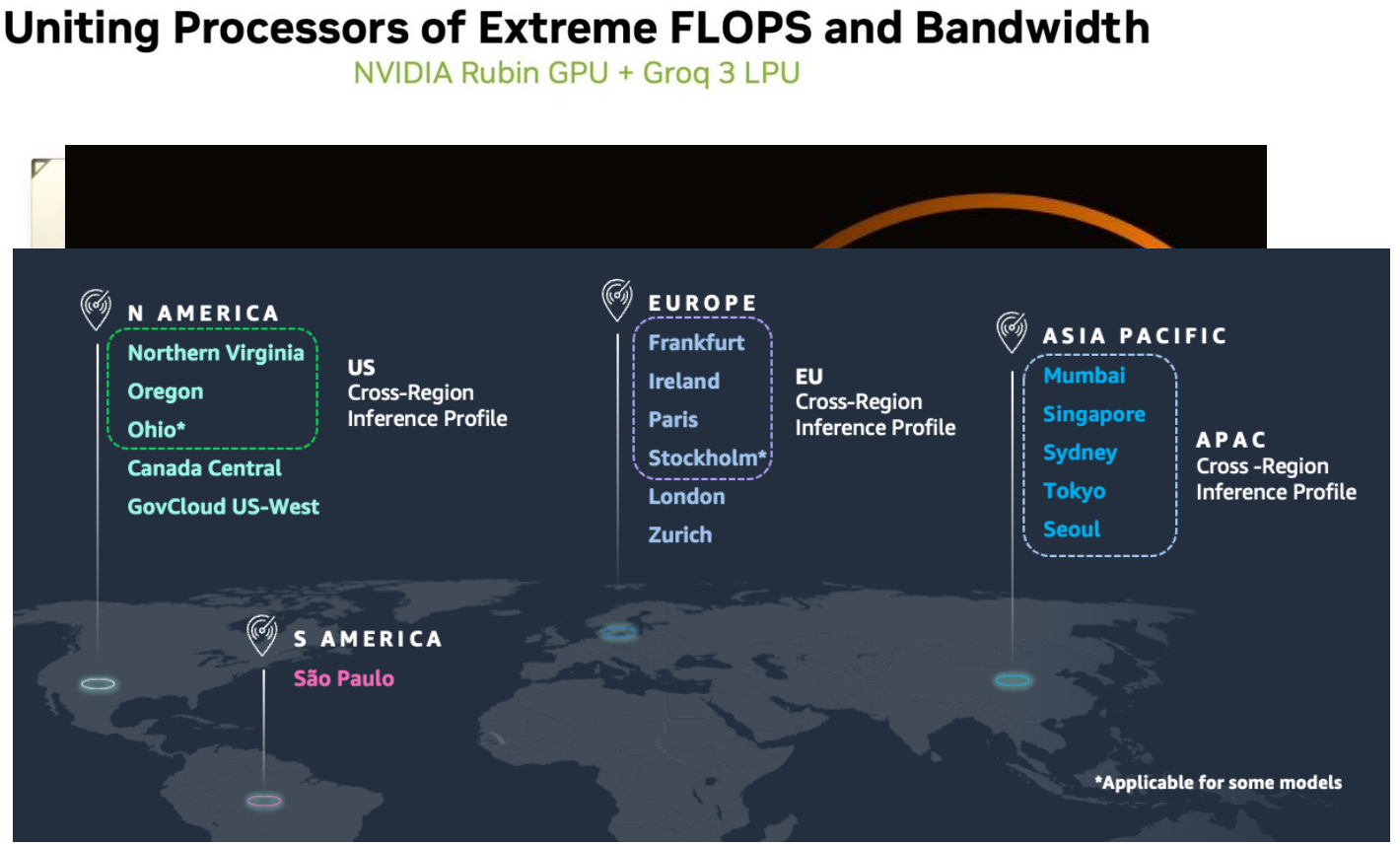
**Uniting Processors of Extreme FLOPS and Bandwidth**

NVIDIA Rubin GPU + Groq 3 LPU

# Heterogeneity in Computing Environments

## Observations

- Heterogeneous device capabilities (e.g., compute, memory capacity, and memory bandwidth).
- Heterogeneous interconnects (e.g., network bandwidth and latency).
- Heterogeneous resource availability across regions (e.g., device type and quantity).



Can we deploy RL training across a set of heterogeneous GPUs connected via heterogeneous networks?

# Can we deploy RL training across a set of heterogeneous GPUs connected via heterogeneous networks?

Existing solutions

- Lack support for heterogeneity-aware deployment
- Limited search space
- Time-consuming scheduling algorithms

ETH zürich aws

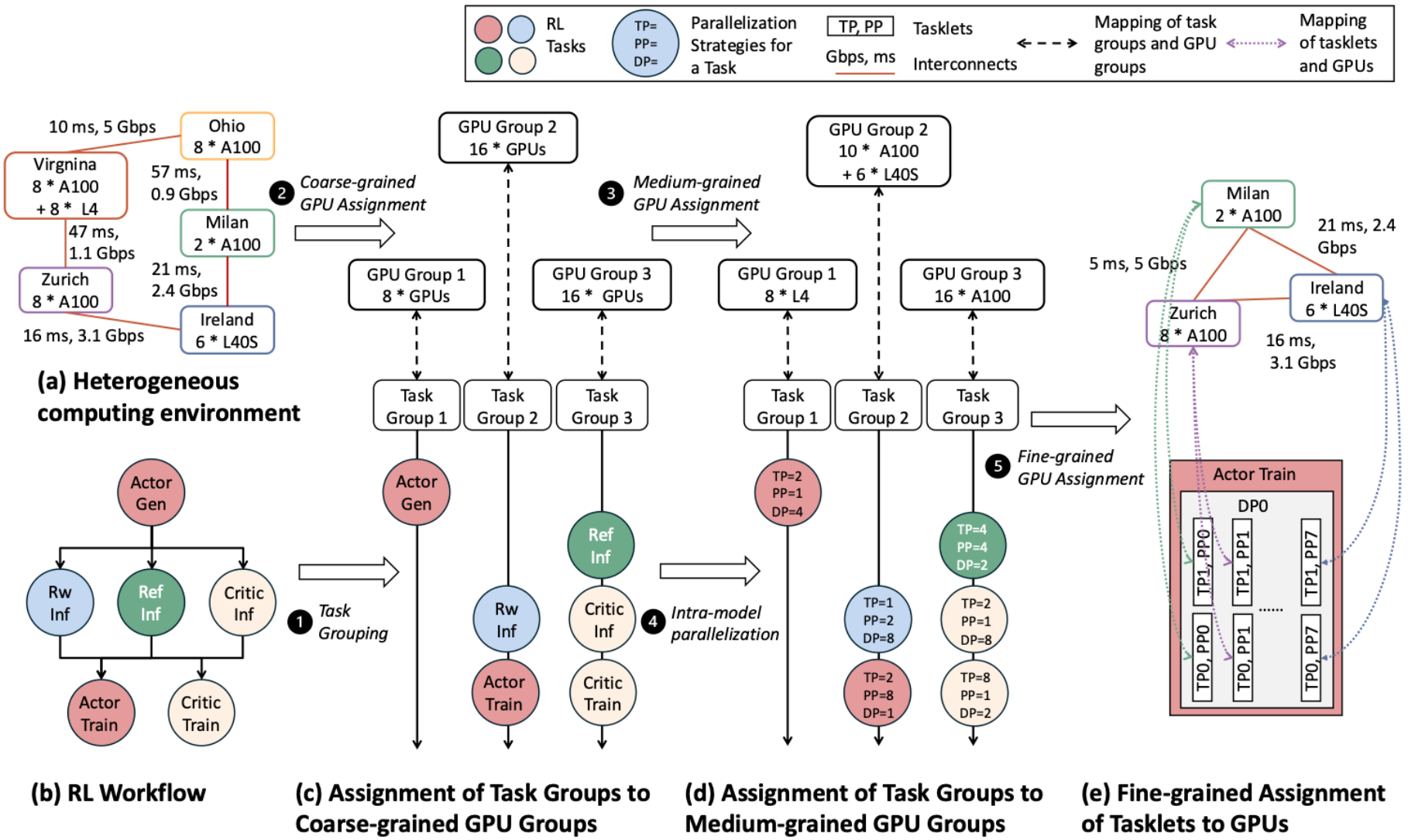
# HetRL Methodology

- Problem formulation: constrained joint optimization problem
  - Setting: given a computational graph $G$ for an RL workflow and a device topology graph $G_D$ for a heterogeneous environment
  - Goal: determine an optimal scheduling strategy, which consists of a partitioning strategy $\rho$ and an assignment strategy $\sigma$, such that the execution time of the RL workflow is minimized and resource constraints are satisfied.
  - Hardness: NP-hard

# HetRL Methodology

- Problem formulation: constrained joint optimization problem
- Multi-Level search framework: coarse-to-fine constructive approach that operationalizes the joint optimization over $(\rho, \sigma)$
  - Level 1: task grouping
  - Level 2: coarse-grained GPU assignment
  - Level 3: medium-grained GPU assignment
  - Level 4: intra-model parallelization
  - Level 5: fine-grained GPU assignment

# HetRL Overview



# HetRL Methodology

- Problem formulation: constrained joint optimization problem
- Multi-Level search framework: coarse-to-fine constructive approach that operationalizes the joint optimization over $(\rho, \sigma)$
- Hybrid scheduling algorithm efficiently identifies near-optimal solutions.
  - Successive halving algorithm for search budget allocation at higher levels (Level 1 and 2)
  - Evolutionary algorithm for low-level plan generation

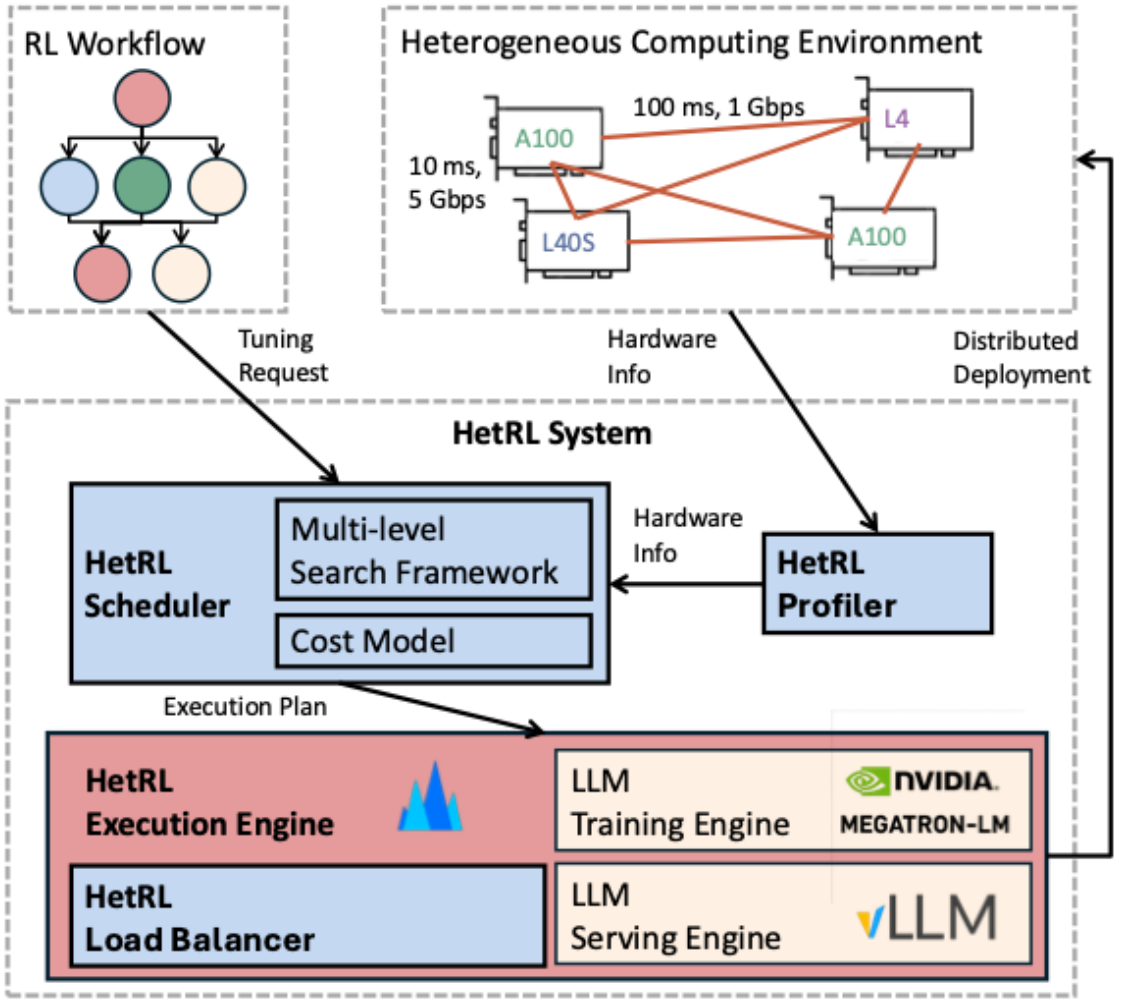
# HetRL Methodology

- Problem formulation: constrained joint optimization problem
- Multi-Level search framework: coarse-to-fine constructive approach that operationalizes the joint optimization over $(\rho, \sigma)$
- Hybrid scheduling algorithm efficiently identifies near-optimal solutions.
- ILP-based scheduling algorithm provides optimal solutions when budgets allows.

# HetRL System

## Principal components

- Scheduler
  - Search framework
  - Cost model
- Profiler
- Execution engine
  - Training engine
  - Serving engine
  - Load Balancer

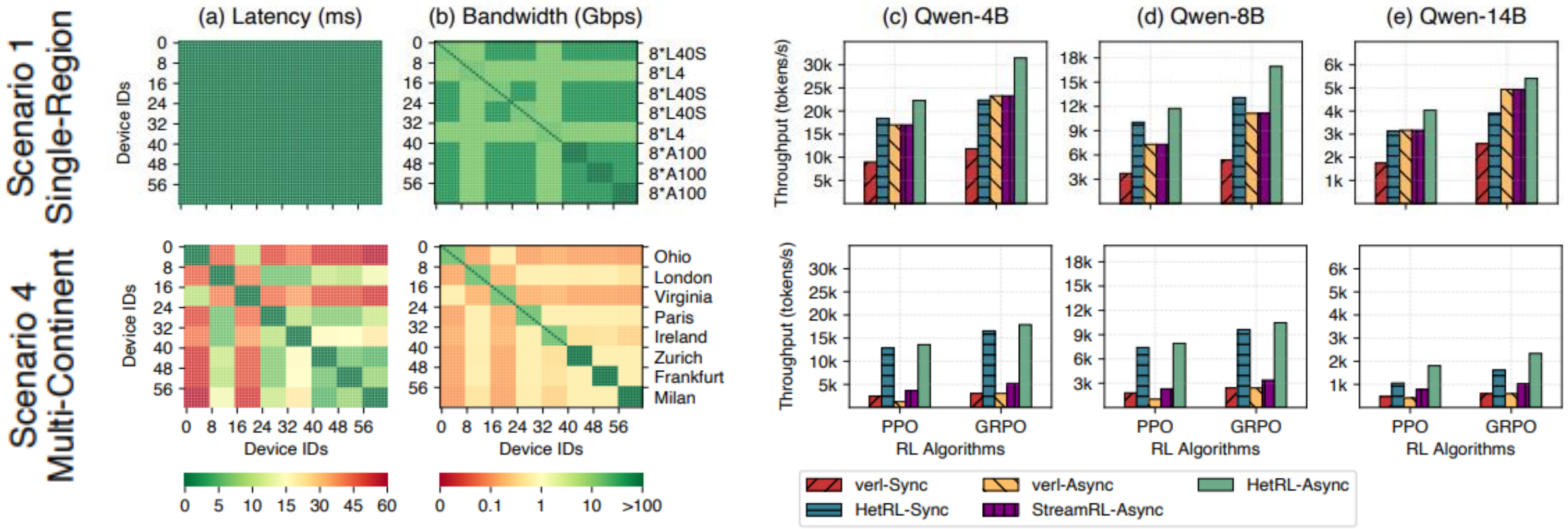


# Evaluation Setup

- Models, datasets, and RL algorithms
  - Qwen 2B, 4B, 8B, and 14B
  - GSM8K and Math-500
  - PPO and GRPO (sync and async)
- Baselines: verl and StreamRL.
- Scenarios: single-region, multi-region, multi-country, and multi-continent.
- GPU specs:

| Model | Arch | Size (GB) | FP16 Perf (TFLOPS) | HBM (GB/s) | Intra (GB/s) |
|---|---|---|---|---|---|
| A100 (24) | Ampere | 40 | 312 | 2039 | 600 |
| L40S (24) | Ada | 48 | 366 | 864 | 64 |
| L4 (16) | Ada | 24 | 121 | 300 | 64 |

# End-to-End Comparison



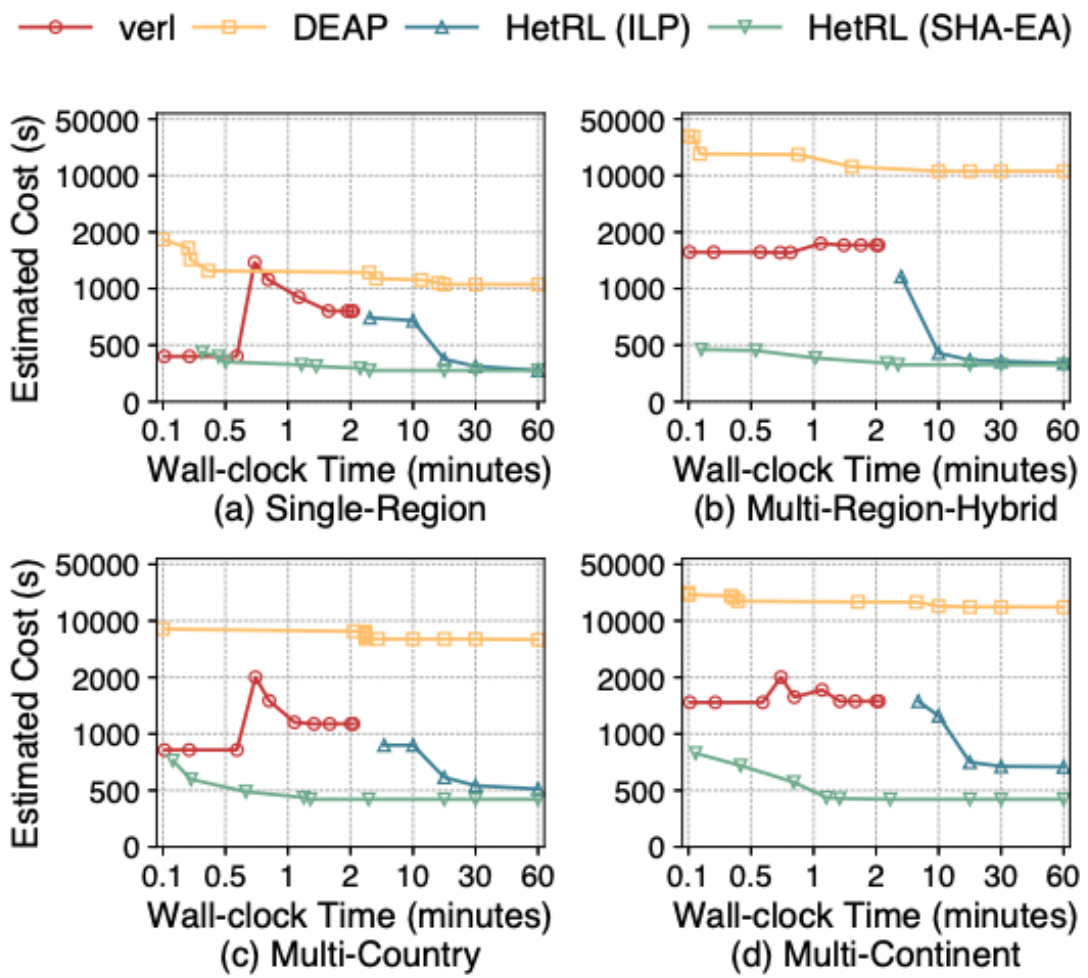
# Search Efficiency Comparison

Hybrid Scheduling Algorithm

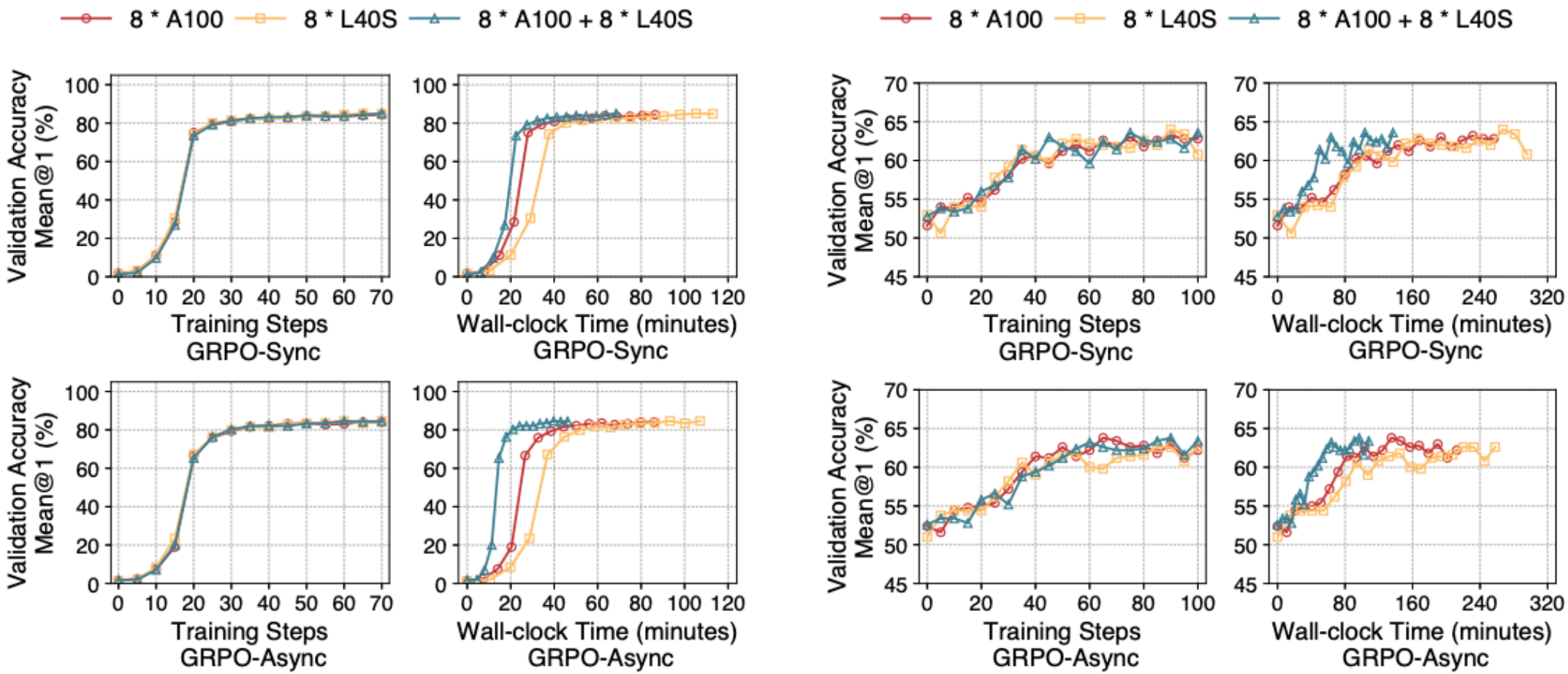
- Efficiently identifies near-optimal solutions

ILP-based scheduling algorithm

- Provides optimal solutions when budgets allows



# Impact on Training Quality




Training dynamics of Qwen3-1.7B-Base on
GSM8K (left) and MATH-500 (right) with GRPO

# Summary



HetRL: Efficient Reinforcement Learning for LLMs in heterogeneous environments

- Support for heterogeneity-aware deployment
  - Alleviating the shortage of homogeneous high-end GPUs within a single AZ
  - Leveraging underutilized mid-range or previous-generation GPUs
- Fast and heterogeneity-aware scheduling
  - Hybrid scheduling algorithm efficiently identifies near-optimal solutions
  - ILP-based scheduling algorithm provides optimal solutions when budgets allows
- Outperforms SoTA systems by 3.17× on average