EnergyLLM: Dynamic LLM Edge Inference

Team 7

12/03/2025

Haebin Do

haebin_do@g.harvard.edu

Kevin He

kevinhe@g.harvard.edu

Alexander Ingare

zingare@g.harvard.edu

Angelica Kim

angelicakim@g.harvard.edu

Problem Statement and Motivations

Challenges

- Transformer LLMs (1B+ params) have high energy requirements.
- Edge devices operate under tight power/thermal constraints (2W-50W)
- With passive cooling on edge devices, Dynamic Voltage/Frequency Scaling is not optimized for unique memory and compute requirements of LLMs
- Optimal configuration changes drastically between thermal regimes

Evaluation Metrics

- **RL Search Overhead against Grid Search**
- **Dynamic Change of Knobs over Time**
- **Energy**, **End-to-End latency, Device Temperature**

Research Question: *How can we adapt LLM configurations to a device's energy budget to deliver the best user experience under strict resource constraints?*

Proposed Approach

Prior Work:

- Xu et al., "Camel: Energy-Aware LLM Inference on Resource-Constrained Devices," 2024

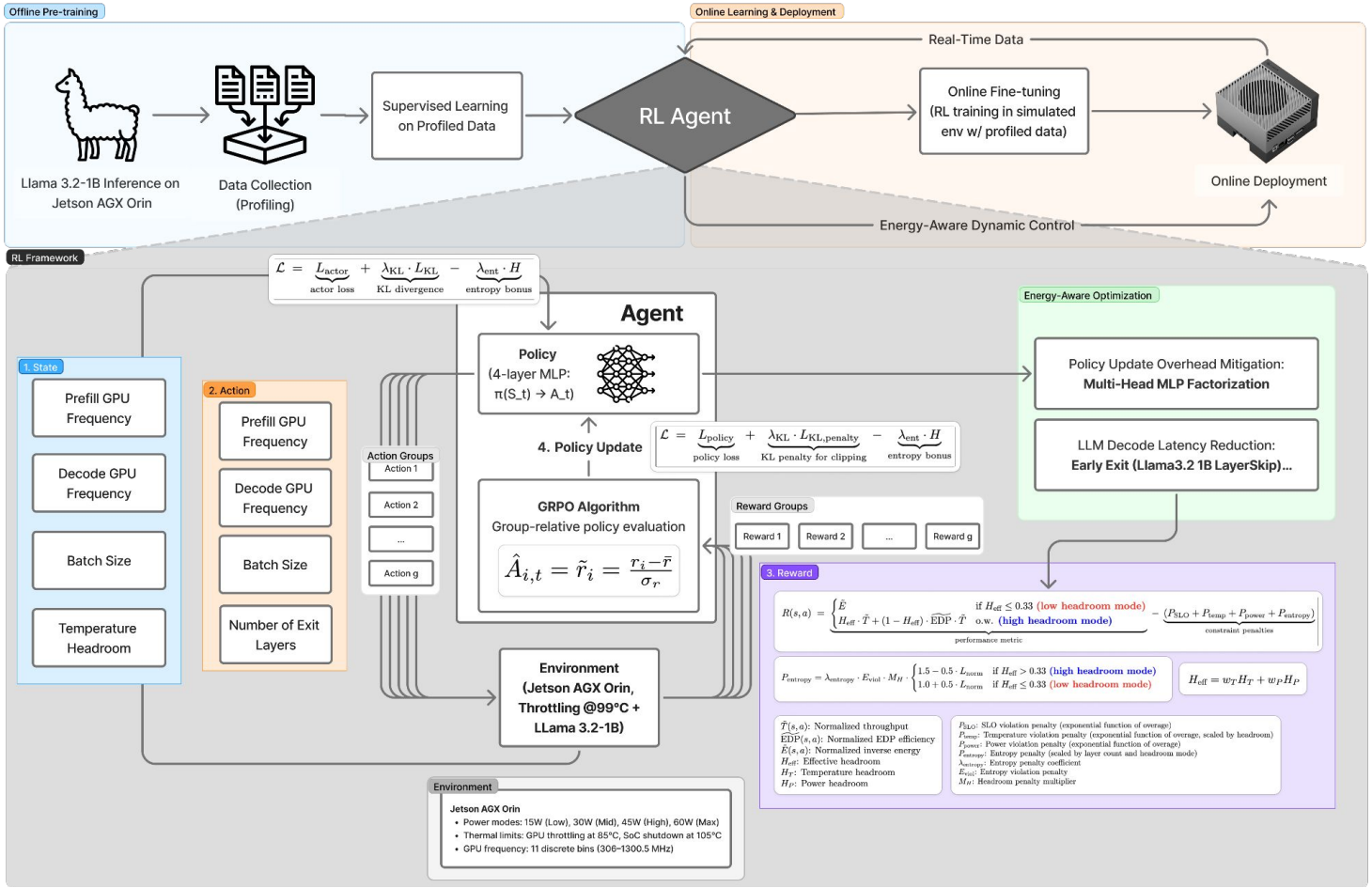
Secret Weapon (RAPS)

- **Reusable**: Software-managed LLM hardware optimization applicable any edge device
- **Adaptable**: Fast-adapting ***GRPO*** RL Agent dynamically learns optimal parameters online
- **Performance**: Lower per-request latency than DVFS by reducing batch size and maintaining high frequency
- **Scalable:** Plug new hardware and runtime knobs into the RL policy with minimal effort
- **Dynamic Energy-based Early Layer Exit** for maintaining high token latency at high temperatures

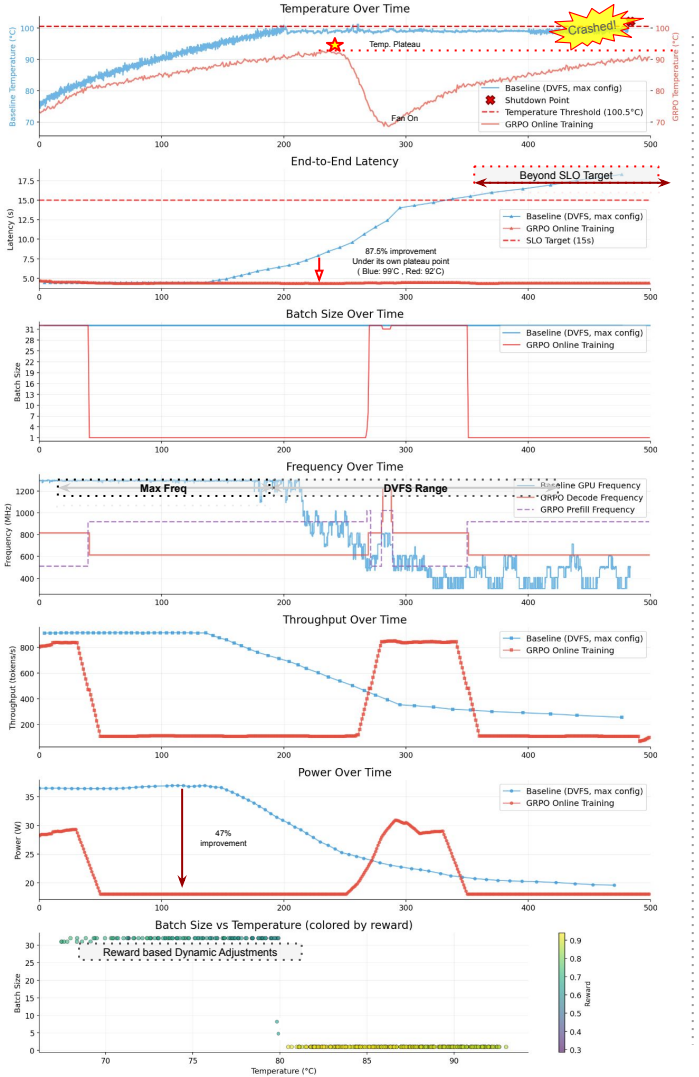
Novelty

- **No system-level edge-device solution predicts power/temperature and adaptively prevents harsh hardware throttling/shutdown for LLM workloads**
  - Aggressive LLM usage on edge devices leads to crashes from harsh throttling.
- **Implementation of different Prefill (GEMM) and decode (GEMV) optimization strategies.**
- **Multi-Head MLP Optimization** for energy efficiency of **RL**
- **Use of Batch-size to reduce power & temperature overheads**
- No previous dynamic tradeoff of accuracy for latency **with early exit**

Proposed Approach



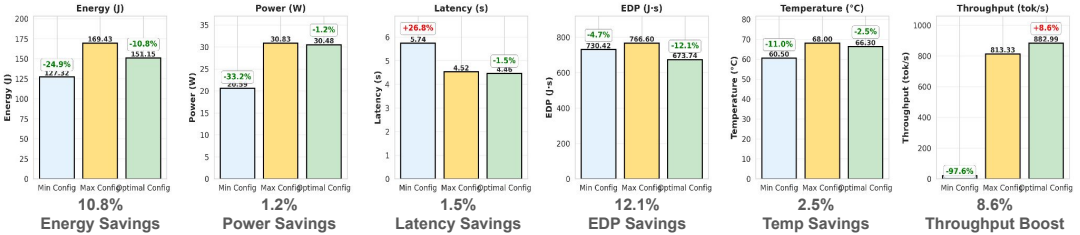
Performance Comparison - DVFS vs GRPO Deployed



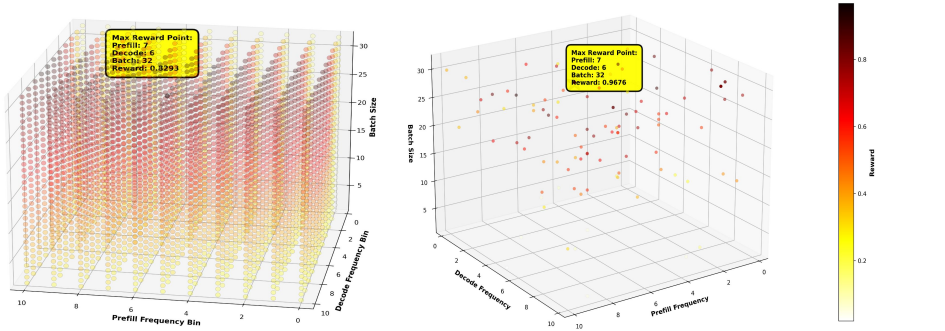
Optimal Configuration (Fan On vs Fan Off)

	Prefill Freq (MHz)	Decode Freq	Batch Size
Low Temp	1020	918	32
High Temp(80'C)	918	612	1

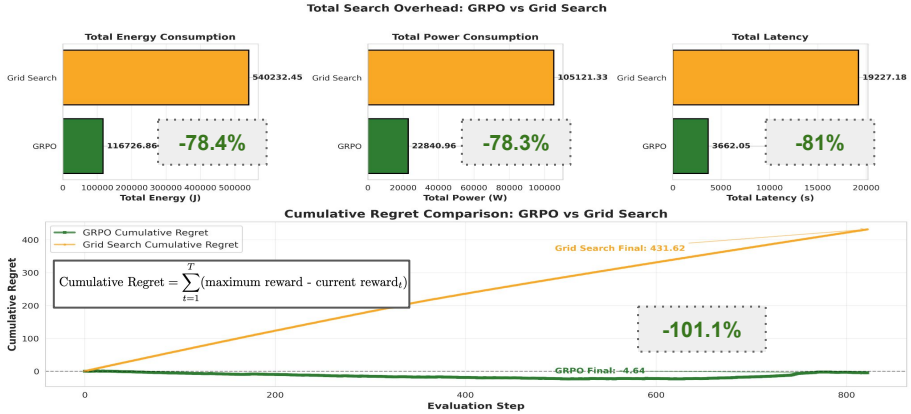
Performance Comparison Across Configurations



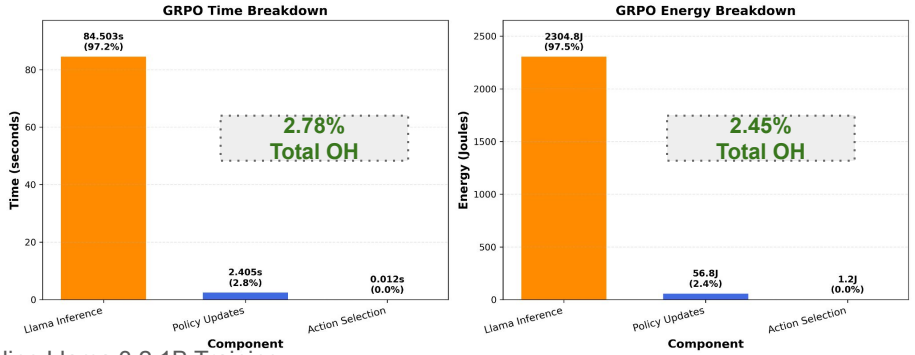
Search Space Exploration (Grid Search vs GRPO)



Search Overhead - GRPO vs Grid Search



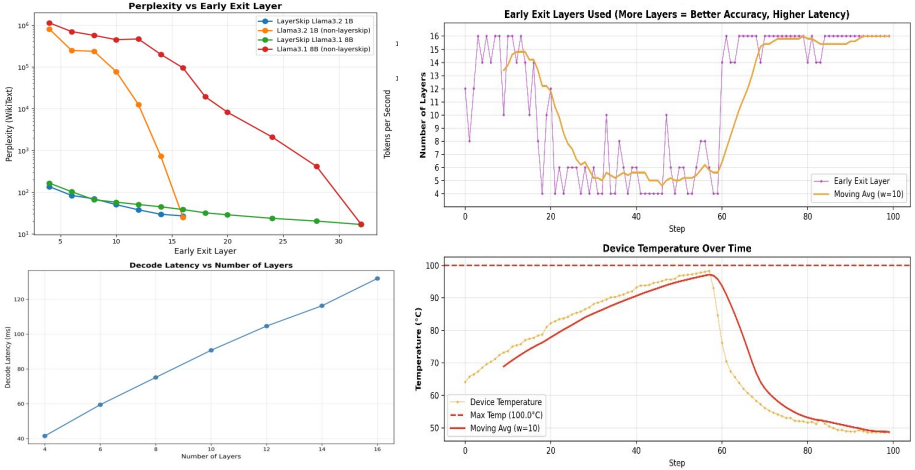
GRPO Overhead Breakdown



Online Llama 3.2 1B Training
(5 steps)

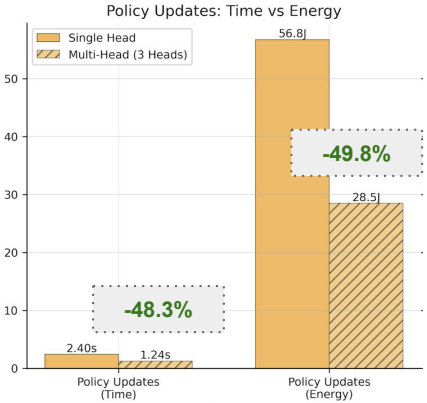
Summary:
Time: Total RL Overhead = 2.42 sec (2.78%)
Energy: Total RL Overhead = 57.94 J (2.45%)

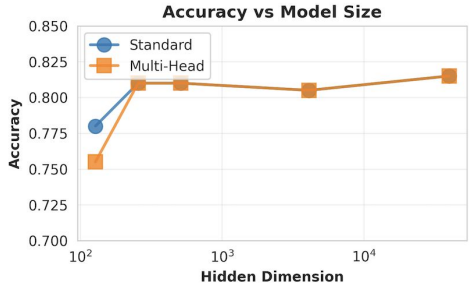
LLM Decode Latency Reduction via Early Exit



Llama3.2 1B LayerSkip

Policy Update Overhead Mitigation via Multi-Head Policy





Memory Footprint reduction by 96.1%

Conclusion

Lessons Learned

- Reward designs dictate behavior
- KL tuning governs stability (KL coefficients (for policy update), KL clipping (for PPO/GRPO stability), and temperature scaling)
- Add any knobs that can be dynamically traded would make your LLM more optimized

Additional Work

- Early exit latency-accuracy tradeoff analysis
- Extra LLM models

Anticipated Conclusion

- Similar Improvements over Grid-Search / DVFS would be observed across other models

Group Contribution

Haebin Do: Designed RL pipeline from pre-training to online training(fine-tuning) & State, Reward Function(Throughput/EDP/Energy/Headroom/Penalty) for high/low temperature state/power with penalties & Tuned policy Hyper-parameters (KL loss) for Jetson Orin, GRPO energy & latency overhead, MLP layer optimization design.

Kevin He: Setup of Jetson Orin, Setup of running LLM with PPO policy, Baseline DVFS & static comparisons, investigated dynamic pruning and early layer exit perplexity, latency tradeoffs.

Alexander Ingare: LLM inference early exit design, added entropy-based penalty to latency-energy reward model, RL entropy pre training with synthetic data generation to change layers according to the states.

Angelica Kim: Designed RL pipeline from online training(fine-tuning) to deployment & State, Reward(Throughput/EDP/Energy/Headroom/Penalty) Function for high/low temperature state/power with penalties for Jetson Orin, Tuned policy Hyper-parameters (KL loss), Designed Random-search / Grid-search Process across different temperatures on Jetson.