ELTO: Energy-Latency Trade-off Optimization for Machine Learning Inference with Dynamic Batching

Ivan Dokuchaev, Ali Kadhum Idrees and Rolf Schuster

Fachhochschule Dortmund, Germany

Fachhochschule Dortmund
University of Applied Sciences and Arts

Motivation

EMULATE

Rising GPU Workloads: The rapid growth of AI inference at scale—especially in edge and cloud environments—drives increasing GPU utilization and energy demand.

Environmental Impact: This expansion comes with a **significant carbon footprint** and **high operational costs**, often neglected in system design.

Current Focus: Traditional QoS optimization emphasizes **low latency and high throughput**, with **little attention to energy efficiency**.

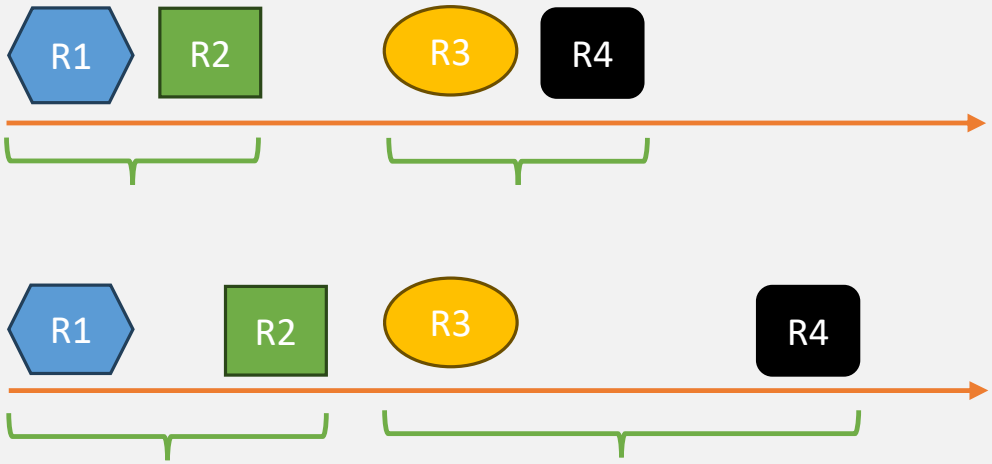
Research Gap: How can we design **systems** that dynamically **configure and adapt ML inference workloads** to balance energy efficiency and latency in real-world deployments?

Fachhochschule Dortmund
University of Applied Sciences and Arts

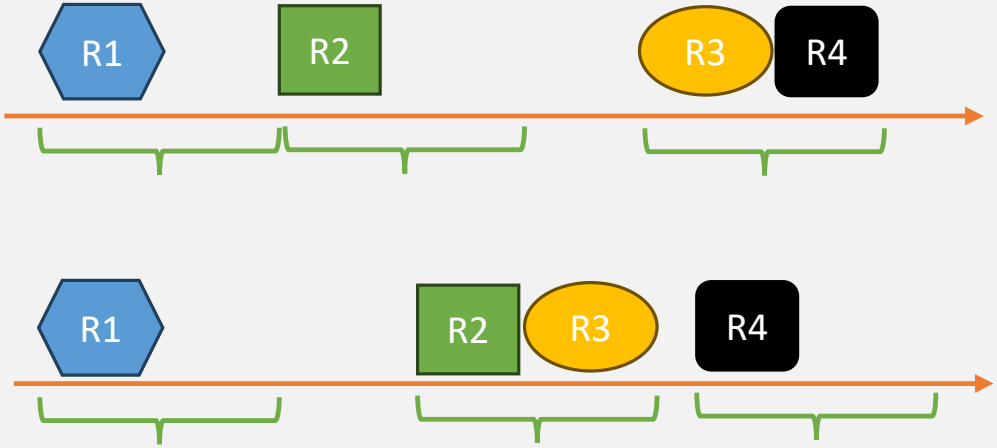
Static and dynamic batching

EMUJATE

- Static Batching (Batch size = 2)



- Dynamic Batching (BS = 2 or Timeout)



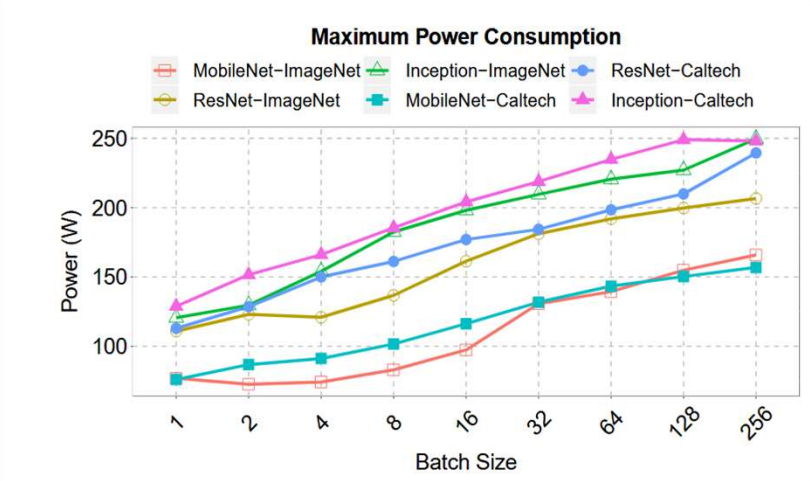
Batching - grouping multiple input requests for simultaneous processing to leverage parallel processing capabilities

Fachhochschule Dortmund
University of Applied Sciences and Arts

Power consumption on GPU

EMUJATE

- Larger batches = lower energy per inference

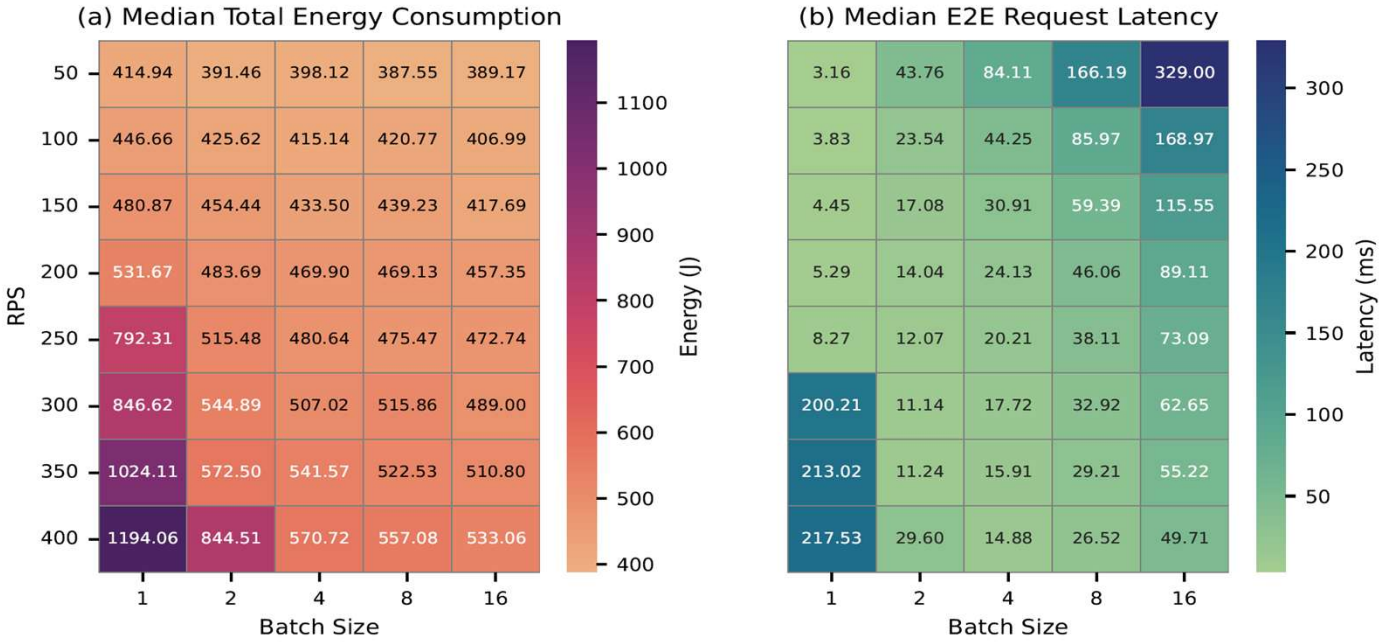


Impact of the batch size on om power consumption of DNN inference

Source: S. M. Nabavinejad, S. Reda, and M. Ebrahimi, "Coordinated Batching and DVFS for DNN Inference on GPU Accelerators," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 10, pp. 2496–2508, Oct. 2022, doi: 10.1109/TPDS.2022.3144614.

Fachhochschule Dortmund
University of Applied Sciences and Arts

Batch size VS Request rate



Median energy consumption (a) and E2E latency (b) profiles for ResNet18 showing relationships with client batch size B and request rate λ

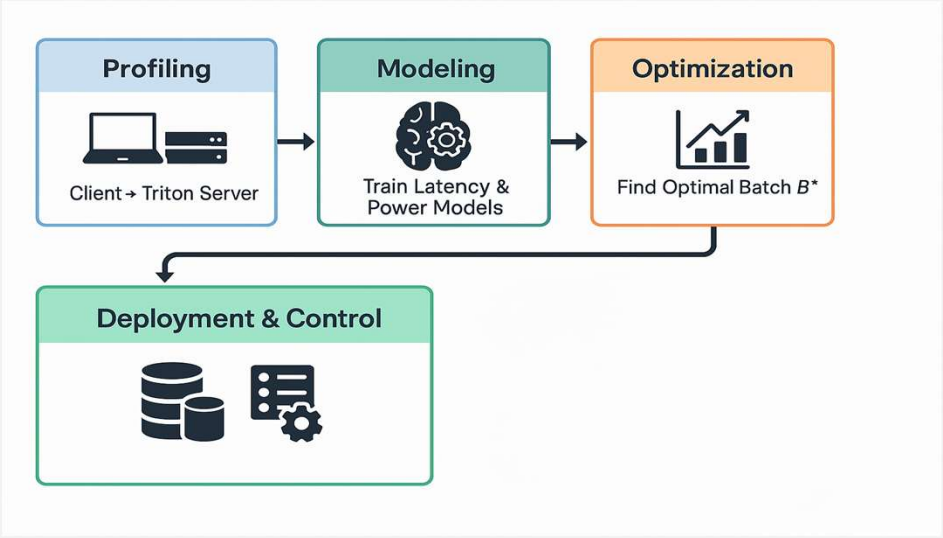
Fachhochschule Dortmund
University of Applied Sciences and Arts

ELTO - Optimizing the Energy-Latency Trade-off

EMULATE

- **Problem:** For any deployed ML model, there's a fundamental trade-off. Using larger batches of requests can improve energy efficiency and throughput, but it often increases latency.
- **Goal:** How do we find the optimal batch size when the workload is constantly changing?

- Proposed ELTO Framework

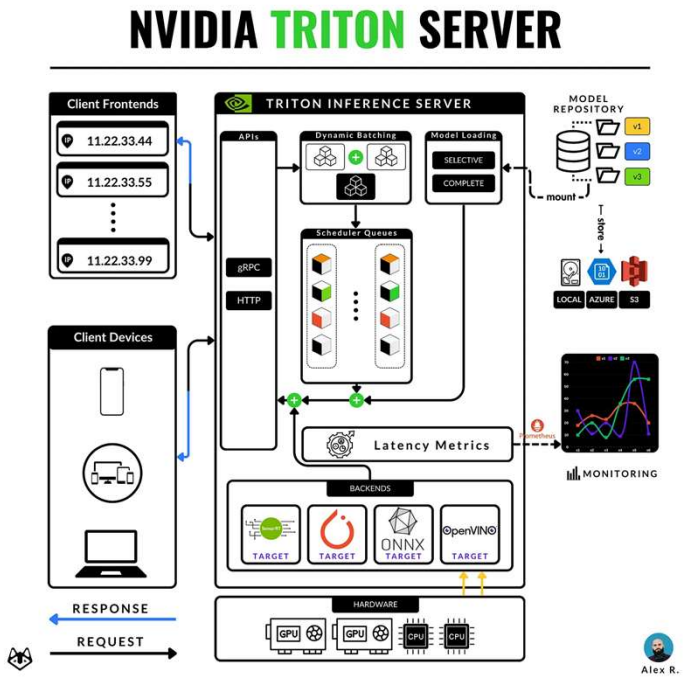


Fachhochschule Dortmund
University of Applied Sciences and Arts

Profiling Phase

EMUJATE

- Systematically vary batch sizes (B) and request rates (λ)
- Record latency and energy consumption for each configuration
- Build comprehensive dataset for predictive model training



- Source: https://medium.com/neuralbits/deploying-deep-learning-models-at-scale-triton-inference-server-0-to-100-ae0f5e7d88b5

Fachhochschule Dortmund
University of Applied Sciences and Arts

Modeling Phase

EMUJATE

- Build ML models to predict energy and latency for any batch size and request rate
- Models evaluated: Linear Regression, Decision Tree, Random Forest, SVR
- Tree-based models capture complex non-linear relationships in the data

Backbone	Model	MAE (J)	MAE (ms)
ResNet18	DecisionTree	19.83	10.20
	RandomForest	19.43	11.54
	LinearRegression	62.59	39.57
	SVR	49.77	44.15
ResNet50	DecisionTree	49.45	40.12
	RandomForest	50.89	40.40
	SVR	324.40	80.87
	LinearRegression	112.27	103.56

MAE Comparison of Latency and Energy Models with ResNet Backbones

Fachhochschule Dortmund
University of Applied Sciences and Arts

Cost-Based Optimization

EMUJATE

- Combine predicted latency and energy into single cost metric
- Flexible weighting allows tuning based on priorities (latency vs. energy)
- Search across all batch sizes to minimize total cost
- Generate runtime policy: optimal batch size for each request rate
- **Result**: Automated, adaptive batch size selection

Given a user-defined latency weight $w_L \in [0, 1]$, the composite cost $C(x)$ based on predictions is:

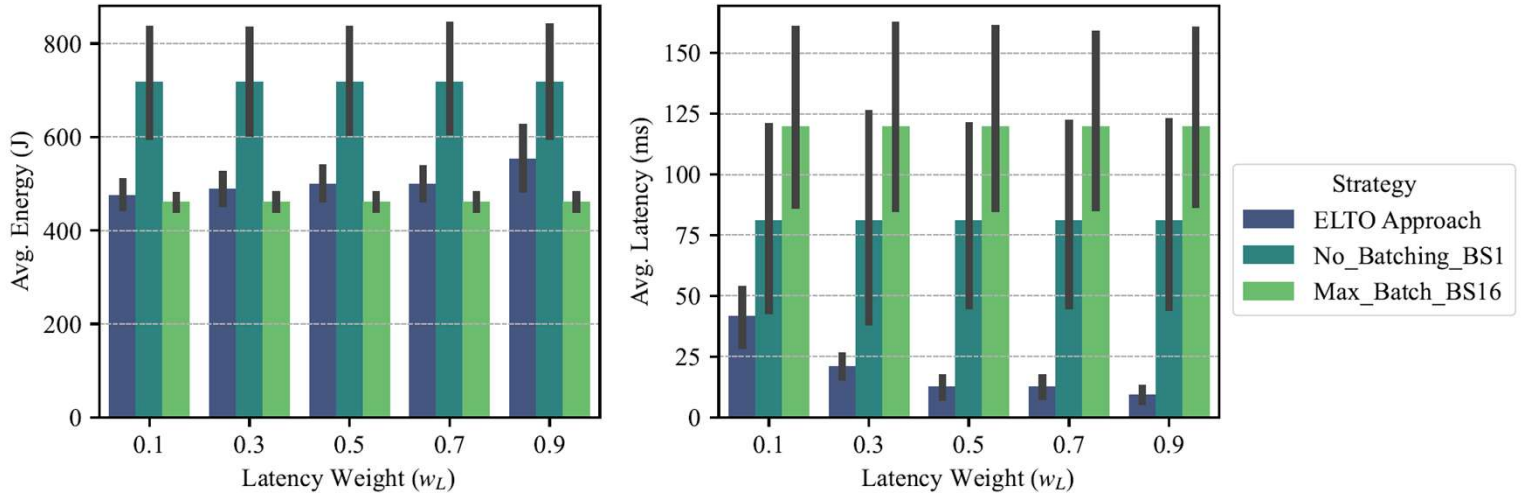
$$C(x) = w_L \cdot L_{\text{sc}}(x) + (1 - w_L) \cdot E_{\text{sc}}(x). \tag{7}$$

The optimal batch size $B^*(\lambda)$ for a given λ is found by minimizing this predicted cost over the set of available batch sizes $\mathcal{B}_{\text{avail}}$:

$$B^*(\lambda) = \arg\min_{B \in \mathcal{B}_{\text{avail}}} C(\lambda, B). \tag{8}$$

Fachhochschule Dortmund
University of Applied Sciences and Arts

Performance of the ELTO approach versus static batching strategies



Performance of the ELTO approach versus static batching strategies (“No Batching BS1” and “Max Batch BS16”) for
the ResNet18 model.

Fachhochschule Dortmund
University of Applied Sciences and Arts

Regret metric to measure final perfomance

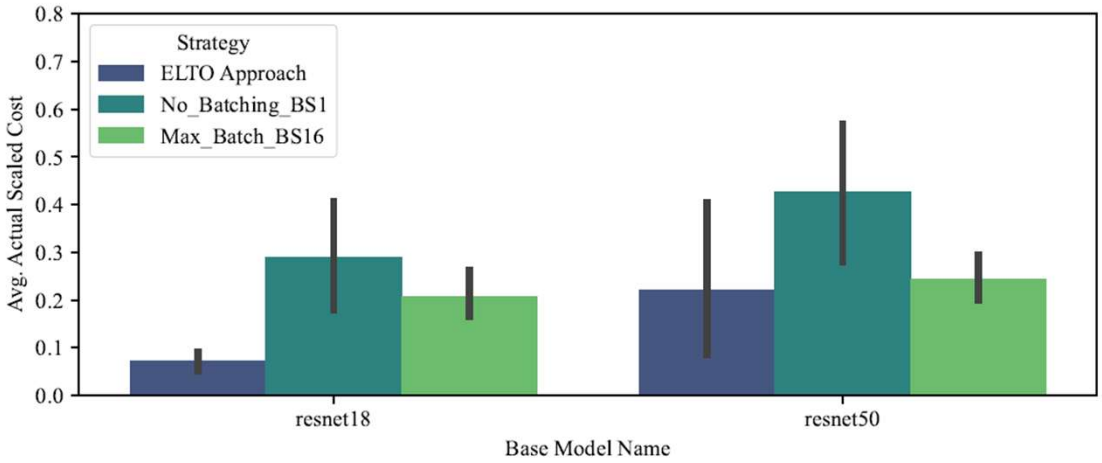
To evaluate the framework's performance, we use a regret metric. First, define the actual cost $C_{\text{act}}(x)$ using actual scaled latency and energy values from $\mathcal{D}_{\text{val}}$:

$$C_{\text{act}}(x) = w_L \cdot L_{\text{sc}}(x) + (1 - w_L) \cdot E_{\text{sc}}(x), \tag{9}$$

where $L_{\text{sc}}(x)$ and $E_{\text{sc}}(x)$ are derived by applying the scaling defined in (6) to the actual $L(x)$ and $E(x)$ values. The regret $R(\lambda)$ is the difference between the actual cost incurred by choosing $B^*(\lambda)$ and the minimum possible actual cost:

$$R(\lambda) = C_{\text{act}}\big(\lambda, B^*(\lambda)\big) - \min_{B \in \mathcal{B}_{\text{avail}}} C_{\text{act}}(\lambda, B). \tag{10}$$

$$X_{scaled} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{6}$$



Comparison of the average actual scaled cost of the proposed ELTO Approach

Fachhochschule Dortmund
University of Applied Sciences and Arts

Conclusion & Future directions

EMULATE

What We Accomplished:

- Energy-latency optimization framework (ELTO)
- 75.5% and 48.2% cost reduction vs. baselines
- Dynamic batch sizing beats static strategies
- Sustainable AI with tradable performance loss

Next Steps:

- Expand the framework to handle multiple, potentially conflicting constraints simultaneously. For example: minimize energy subject to hard constraints on latency, throughput, *and* memory footprint.
- Incorporate a wider array of ML model architectures
- Extending to multimodal serving environments.