GREEN: Carbon-efficient Resource Scheduling for Machine Learning Clusters

— a system researcher's perspective

Presenter

Kaiqiang Xu
徐凯强

Authors

Kaiqiang Xu 徐凯强

Decang Sun 孙德仓

Han Tian 田晗

Junxue Zhang 张骏雪

Kai Chen 陈凯

iSingLab HKUST

iSing AI

香港科技大學 THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

A rising challenge for AI Clusters at scale

growing environmental footprint of GPU Clusters

Power Demand	8% of global data center demand → projected 15–20% by 2028
Resource Scarcity	Power and resource limits hurt cluster-wide performance
Design Goal	Smarter, carbon-aware management without slowing down cluster

Characteristics of ML Cluster Schedulers

① Scale-Adaptive

Adjust GPU allocations dynamically

② Model-Agnostic vs. Model-Aware

Tweaking job settings or hyper-parameters

③ Energy-Aware

Optimizing energy use with certain trade-offs

Prior Work	Scale-Adaptive	Energy-Aware	Model-Agnostics
Gandiva	❌	❌	❌
Tiresias	❌	❌	✅
Pollux	✅	❌	❌
Zeus	❌	✅	❌
GREEN	✅	✅	✅

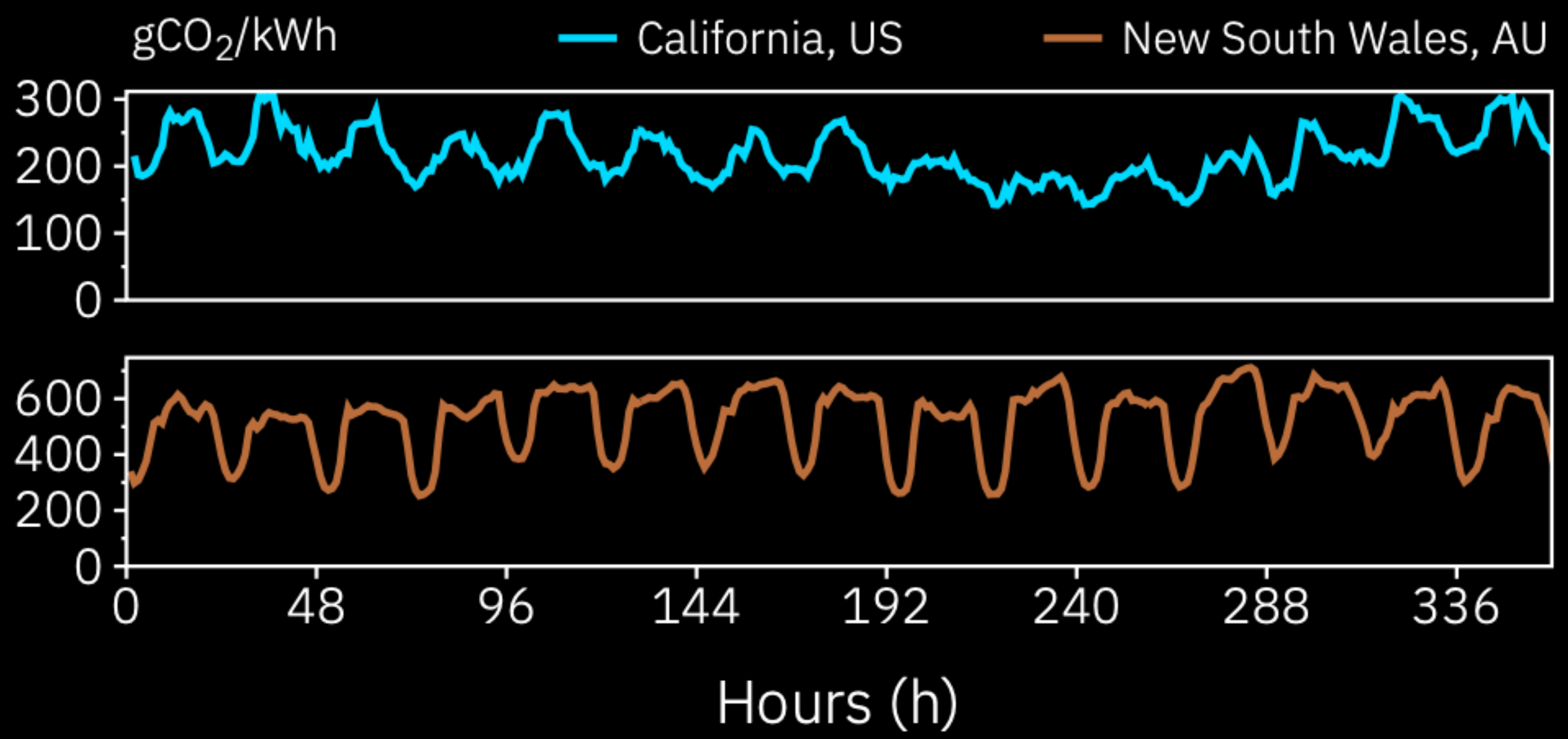
Considerations in Energy Management for ML

Dynamic Voltage and Frequency Scaling (DVFS) or other throttling strategy essentially scaling back the work performed (or capacity)

→ GPU hours are expensive and scaling back resource use is suboptimal

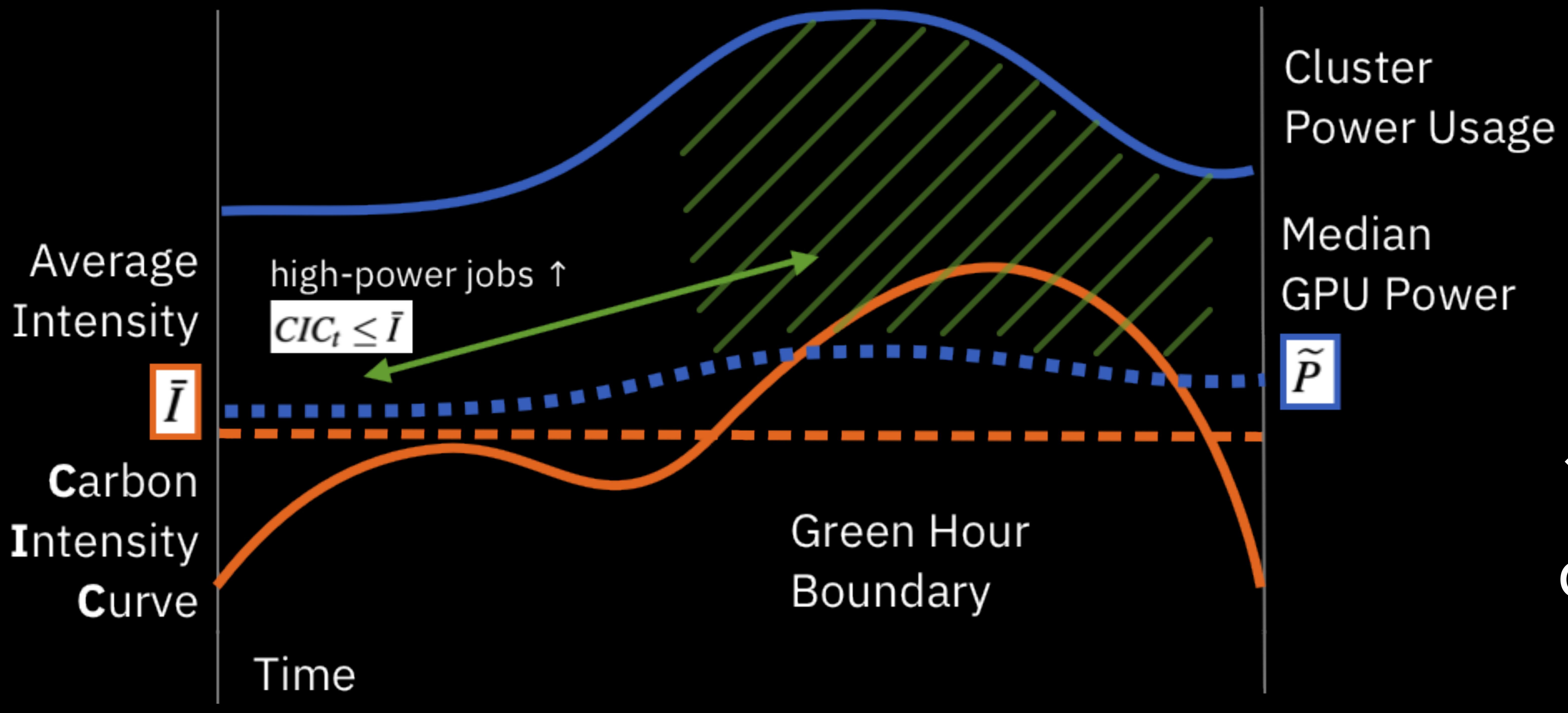
What's more:

- Energy use ≠ Carbon footprint
- Carbon Intensity varies over time



Motivation and Idea of GREEN Scheduling

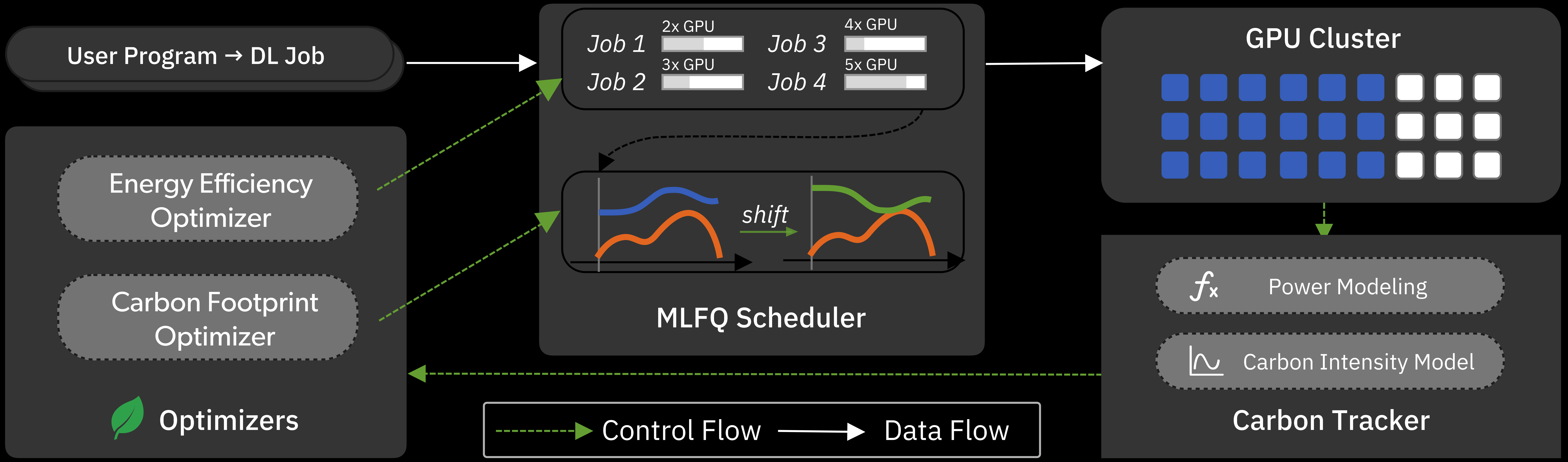
- Different power usage among jobs → Even when using same # of GPU
- Preexisting Job Preemption → Exploit the (natural) temporal flexibility



← Align energy use with low carbon-intensity periods

GREEN Workflow — High-Level View

1. Monitor per-job energy and carbon data
2. Optimize via energy scaling and carbon shifting
3. Schedule using Multilevel Feedback Queue (MLFQ)



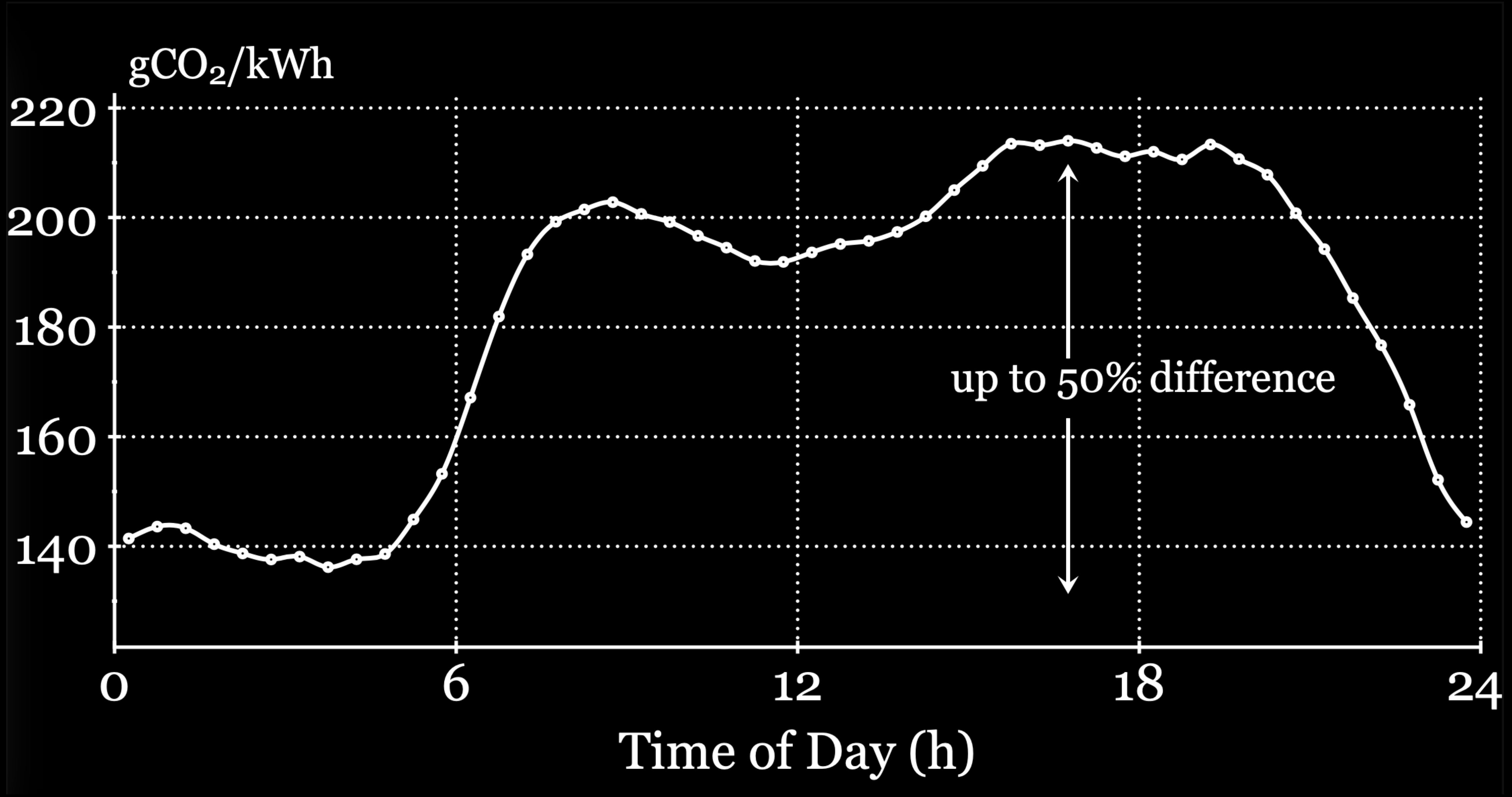
Carbon Tracking and Factor Model

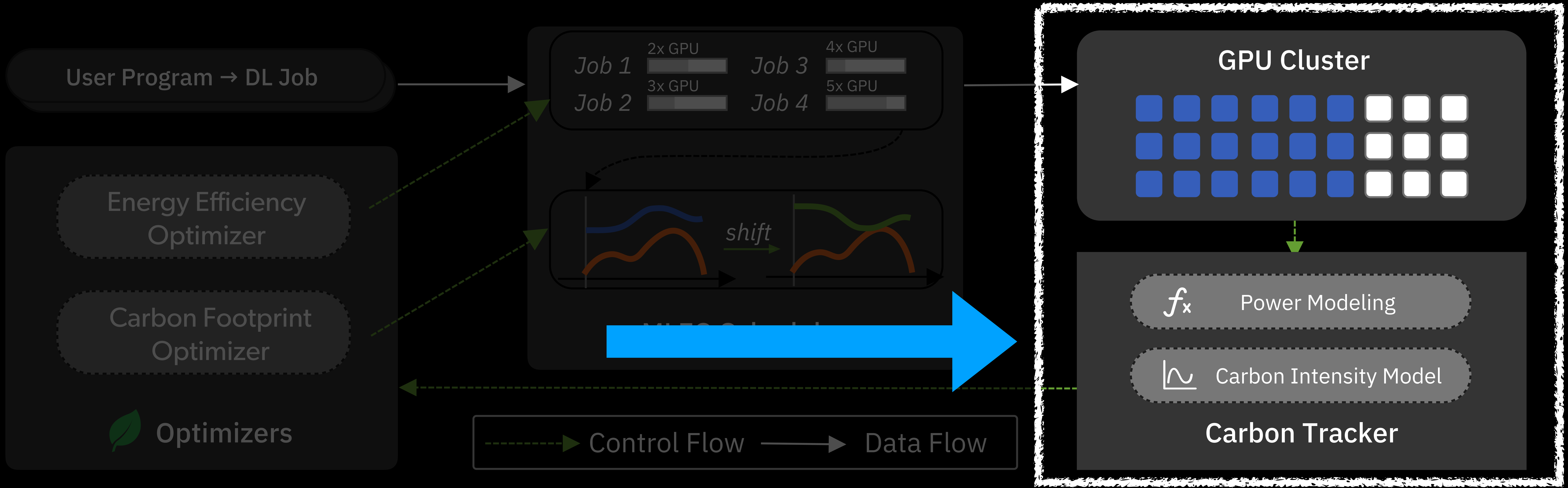
① Monitoring job's energy use

$$P_{\text{job}}(t) = P_{gpu}(t) + P_{cpu\_model}(t) + P_{static}$$

② Calculating job's carbon footprint

$$\texttt{FOOTPRINT}_{\text{job}}(T) = \int_0^T P_{\text{job}}(t) \cdot CIC(t)dt$$

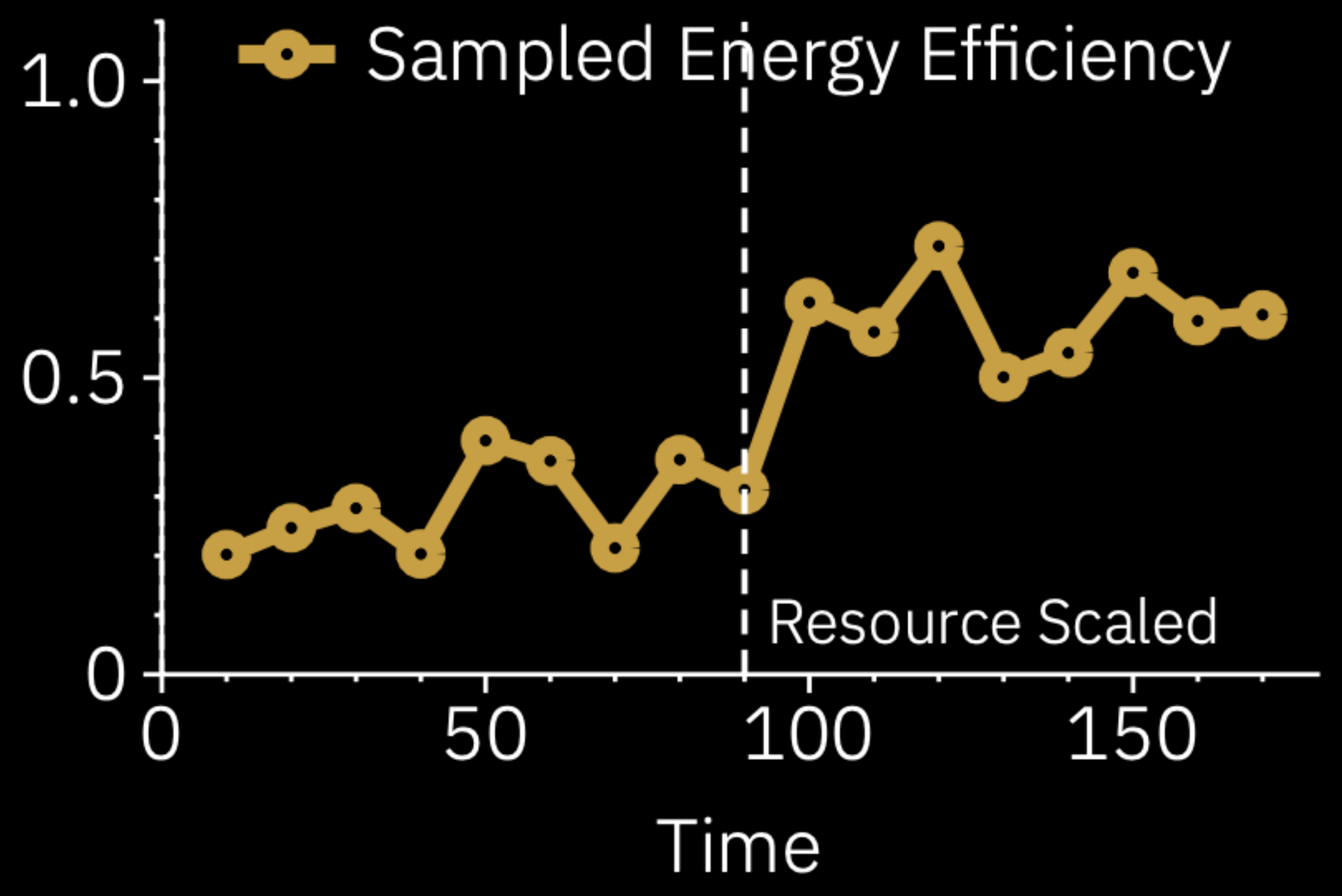


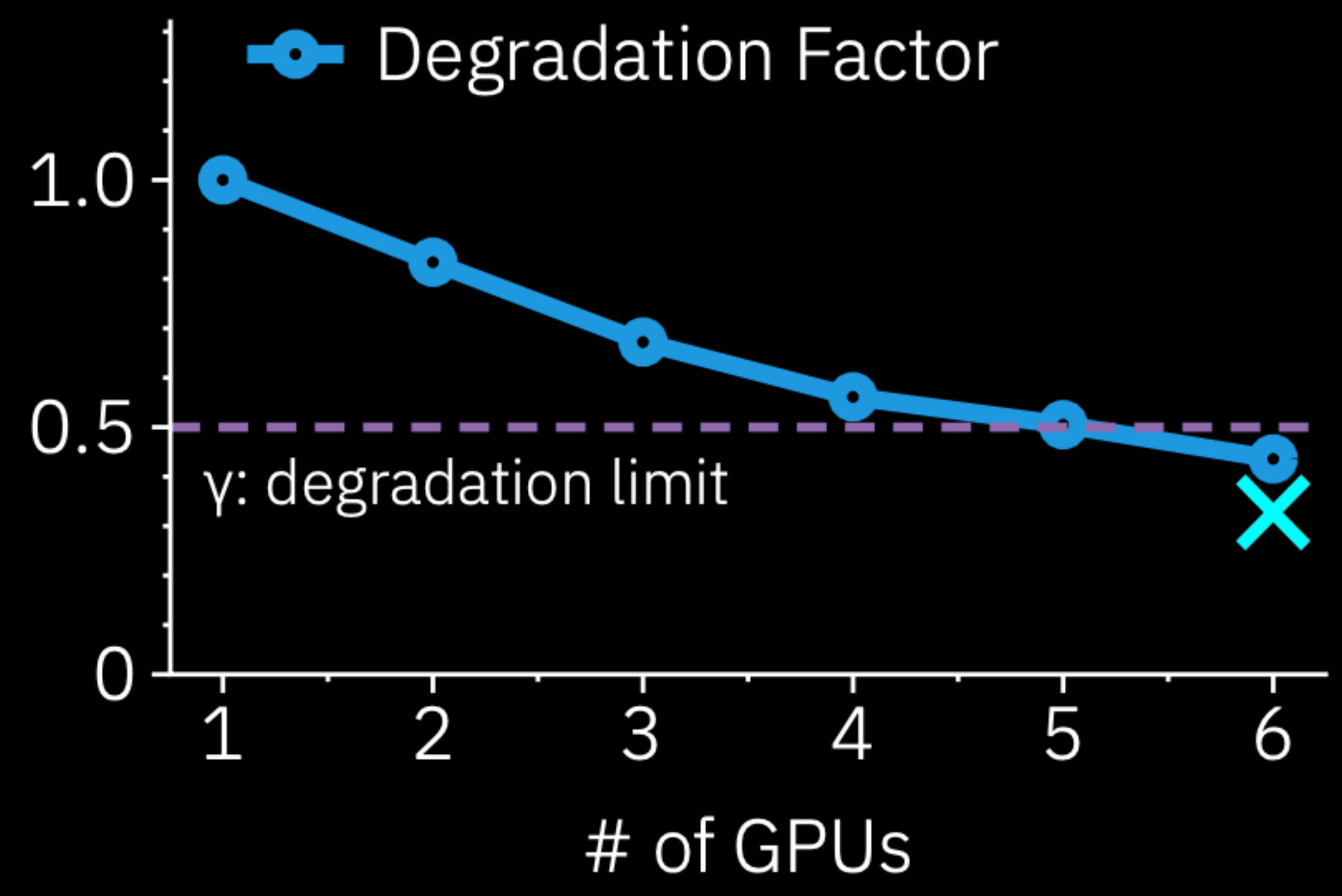


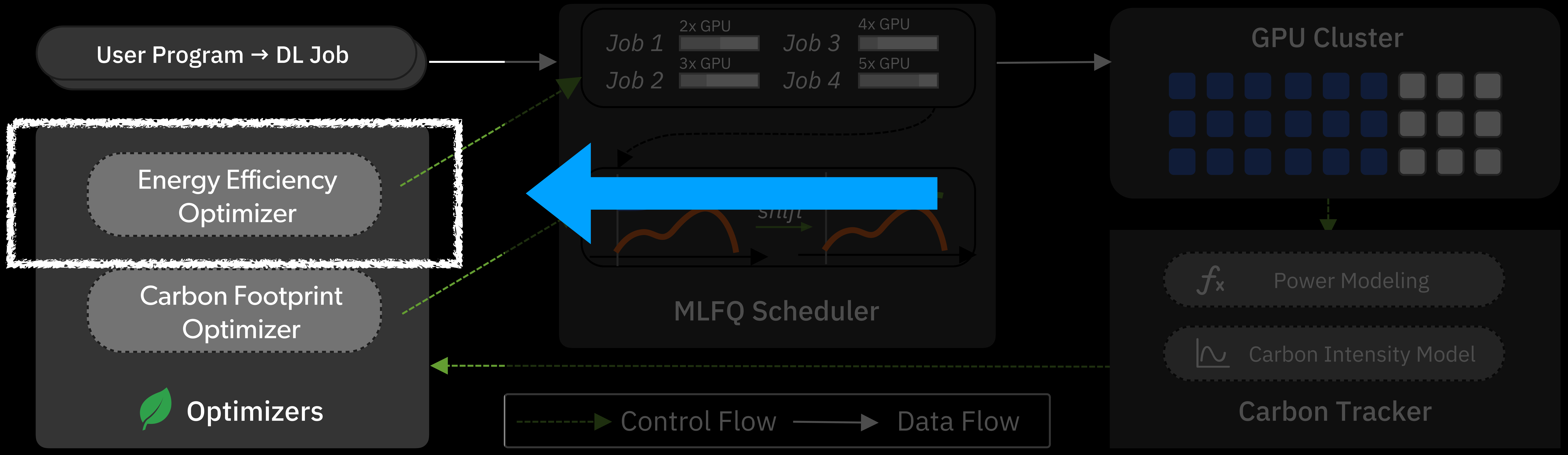
Energy Efficiency Optimizer

Scale jobs' resource allocation to maximize progress per unit energy:

$$\mathtt{EFFICIENCY}_{\mathtt{job}}(t) = \frac{\mathrm{d}Progress_{\mathtt{job}}(t)}{\mathrm{dW}_{\mathtt{job}}(t)}$$



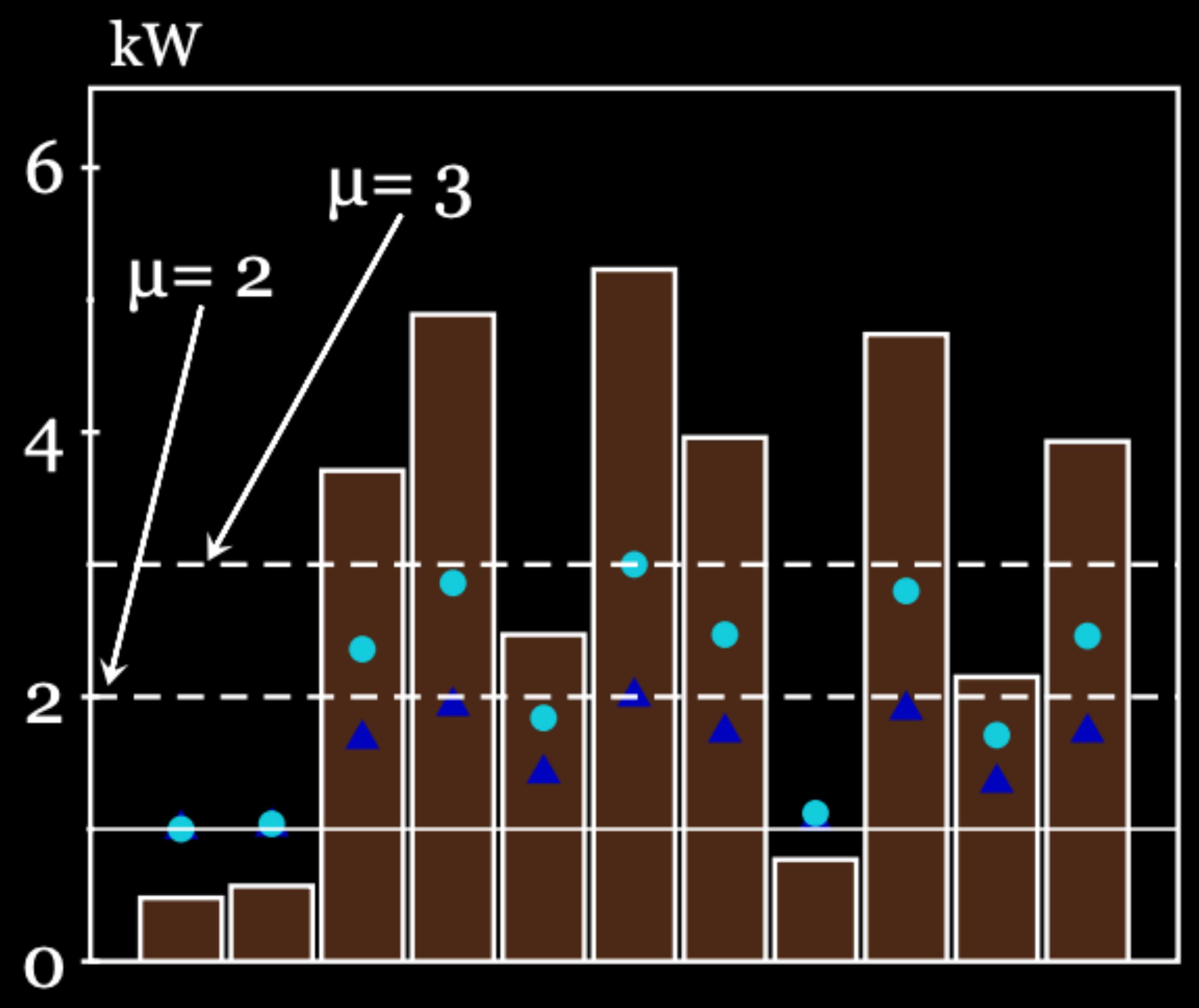


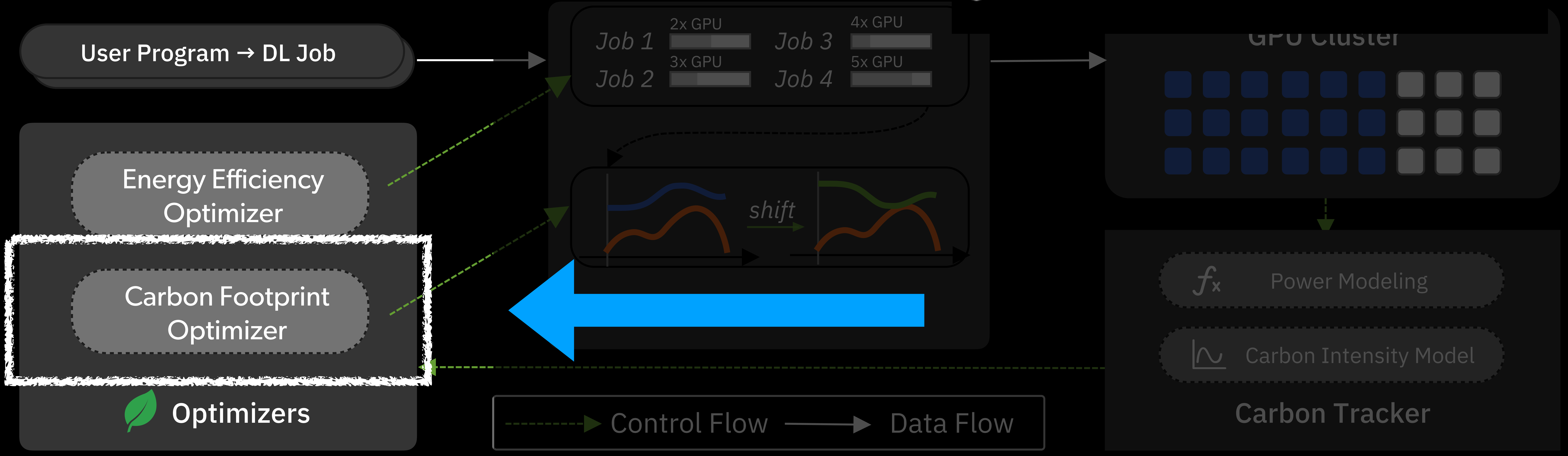


Carbon Footprint Optimizer

Prioritization coefficient by rescaling the job's power consumption to [1,μ]

$$P^*_{\mathtt{job}} = \frac{(P_{\mathtt{job}} - P_{min})}{(P_{max} - P_{min})} \times (\mu - 1) + 1$$

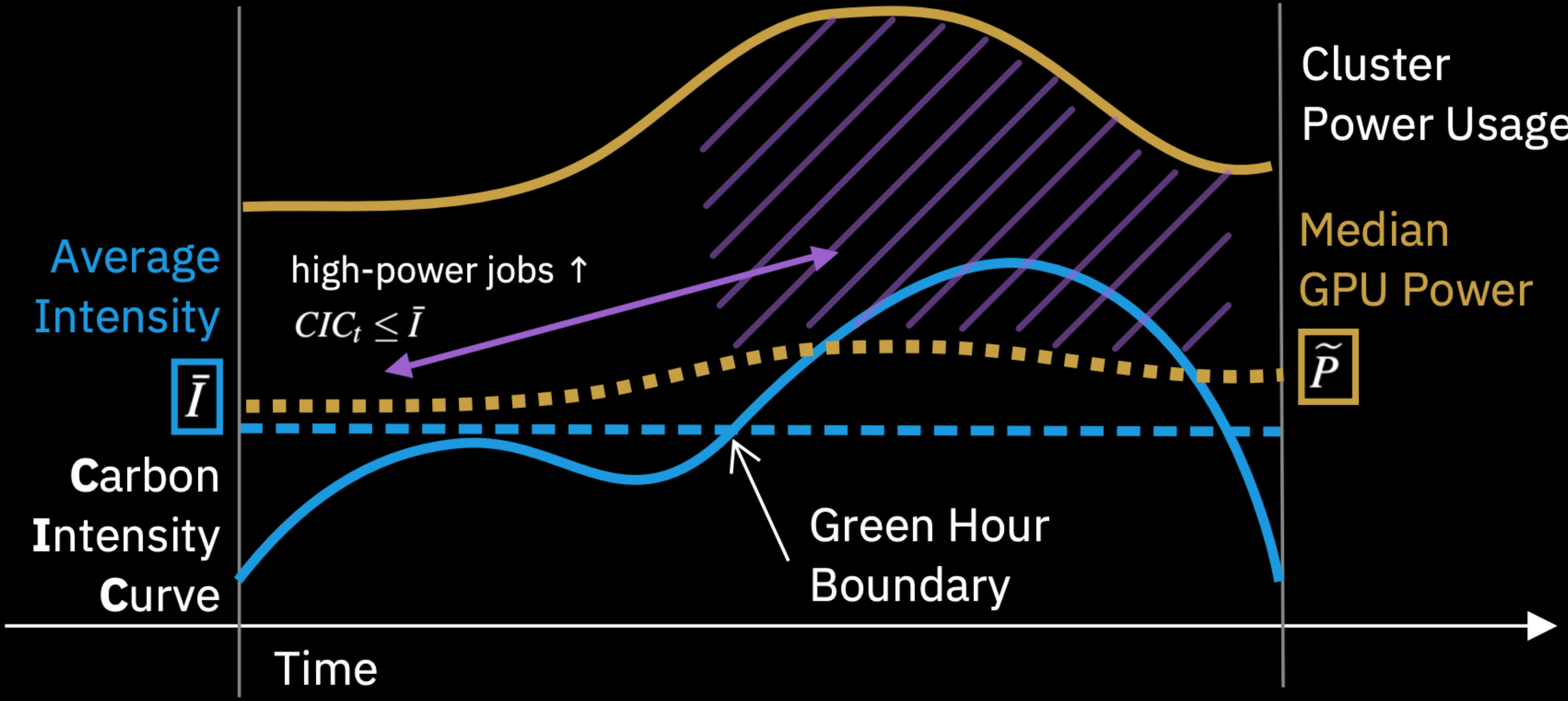




Carbon Footprint Optimizer

$$P^*_{\texttt{job}} = \frac{(P_{\texttt{job}} - P_{min})}{(P_{max} - P_{min})} \times (\mu - 1) + 1$$

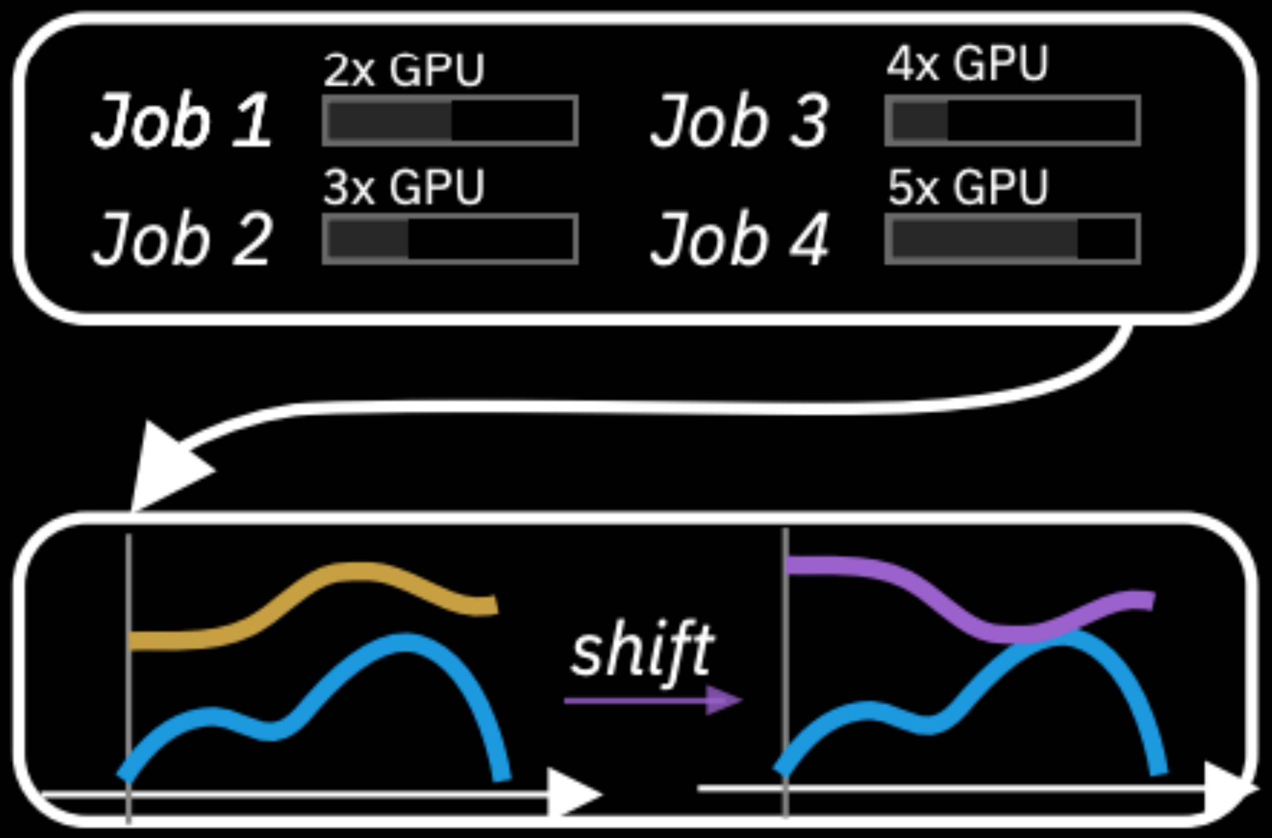
$\texttt{SHIFTING}_{\texttt{job}}$	if $P_{job} \le \widetilde{P}$	if $P_{job} > \widetilde{P}$
if $CIC_t \le \bar{I}$	$P^*_{\texttt{job}}$	$1/P^*_{\texttt{job}}$
if $CIC_t > \bar{I}$	$1/P^*_{\texttt{job}}$	$P^*_{\texttt{job}}$



$$\texttt{PRIORITY}_{\texttt{job}} = \left(\frac{\texttt{FOOTPRINT}_{\texttt{job}}}{\texttt{DEGRADATION}_{\texttt{job}}}\right) \cdot \texttt{SHIFTING}_{\texttt{job}}$$

Workload temporal shifting — based on carbon intensity and jobs' power consumption

Multilevel Feedback Queue (MLFQ) Scheduling



Upper Queue

Profiling and scale resource allocation

Goal: Optimize energy effiency

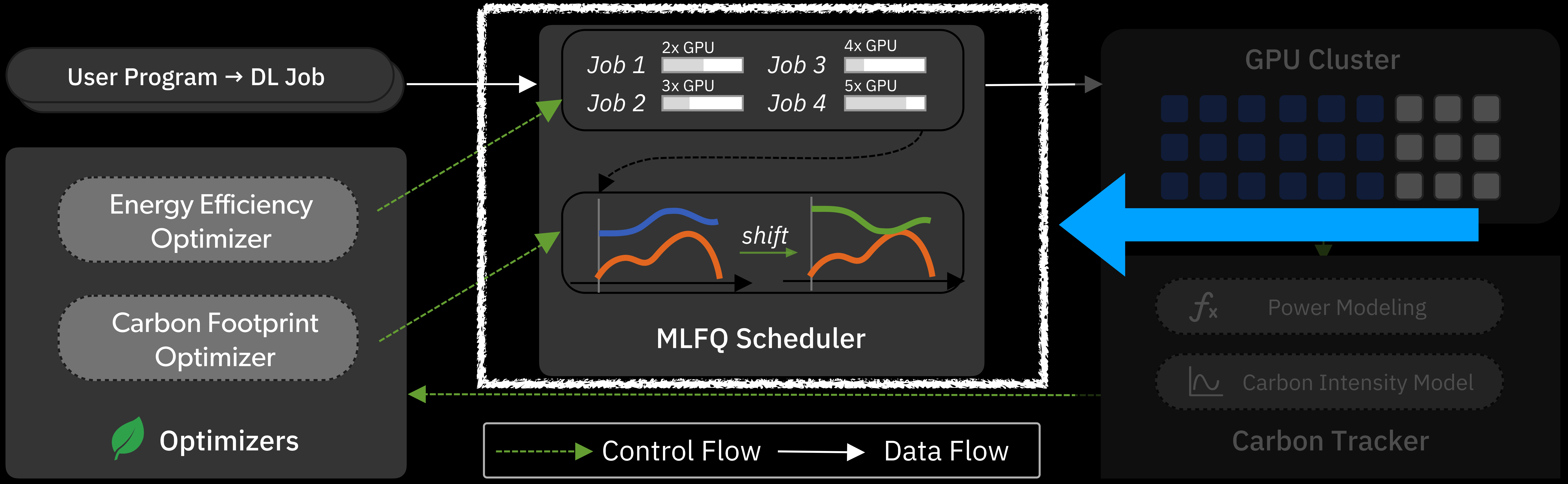
$$\textbf{if } \mathtt{DEGRADATION}_J = \Delta\mathtt{EFFICIENCY}_J \geq \gamma \textbf{ then}$$
$$\quad \text{Scale out } J \text{ with one more GPU}$$

Lower Queue

Shift high-power jobs to greener time

Goal: Optimize cluster carbon footprint

$$\mathtt{PRIORITY}_{\mathtt{job}} = \left(\frac{\mathtt{FOOTPRINT}_{\mathtt{job}}}{\mathtt{DEGRADATION}_{\mathtt{job}}}\right) \cdot \mathtt{SHIFTING}_{\mathtt{job}}$$



Evaluation Setup

Workload: We collected 791 jobs from real users over a 24-hour period on a university-managed production cluster (see SING, ASPLOS '25)

Design and Operation of Shared Machine Learning Clusters on Campus

Authors: Kaiqiang Xu, Decang Sun, Hao Wang, Zhenghang Ren, Xinchen Wan, Xudong Liao, Zilong Wang, Junxue Zhang, Kai Chen | Authors Info & Claims

ASPLOS '25: Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 1
Pages 295 - 310 • https://doi.org/10.1145/3669940.3707266

Baselines: ML cluster schedulers and carbon-aware workload schedulers

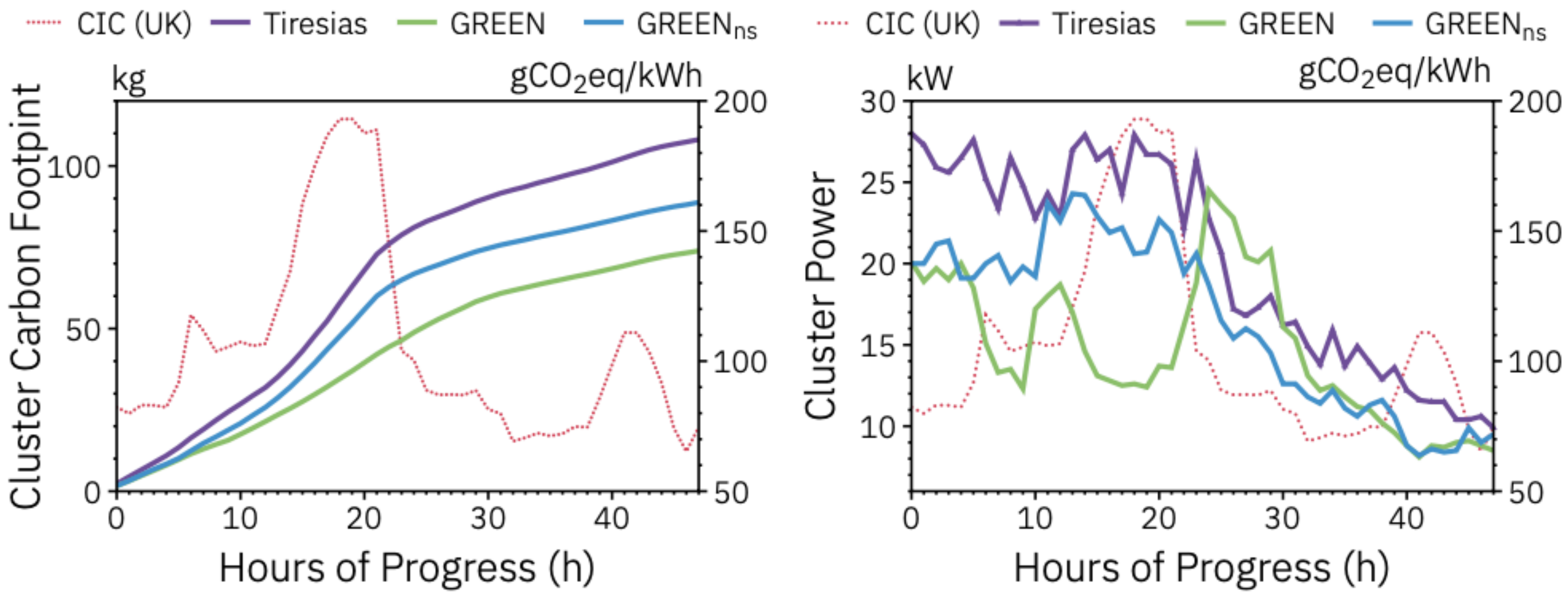
Metrics: JCT, Makespan, Carbon footprint, Cluste-wide Power Draw

Carbon Reductions with Small Speed Tradeoffs

(a) United Kingdom **(b)** California, USA **(c)** Sweden **(d)** Poland

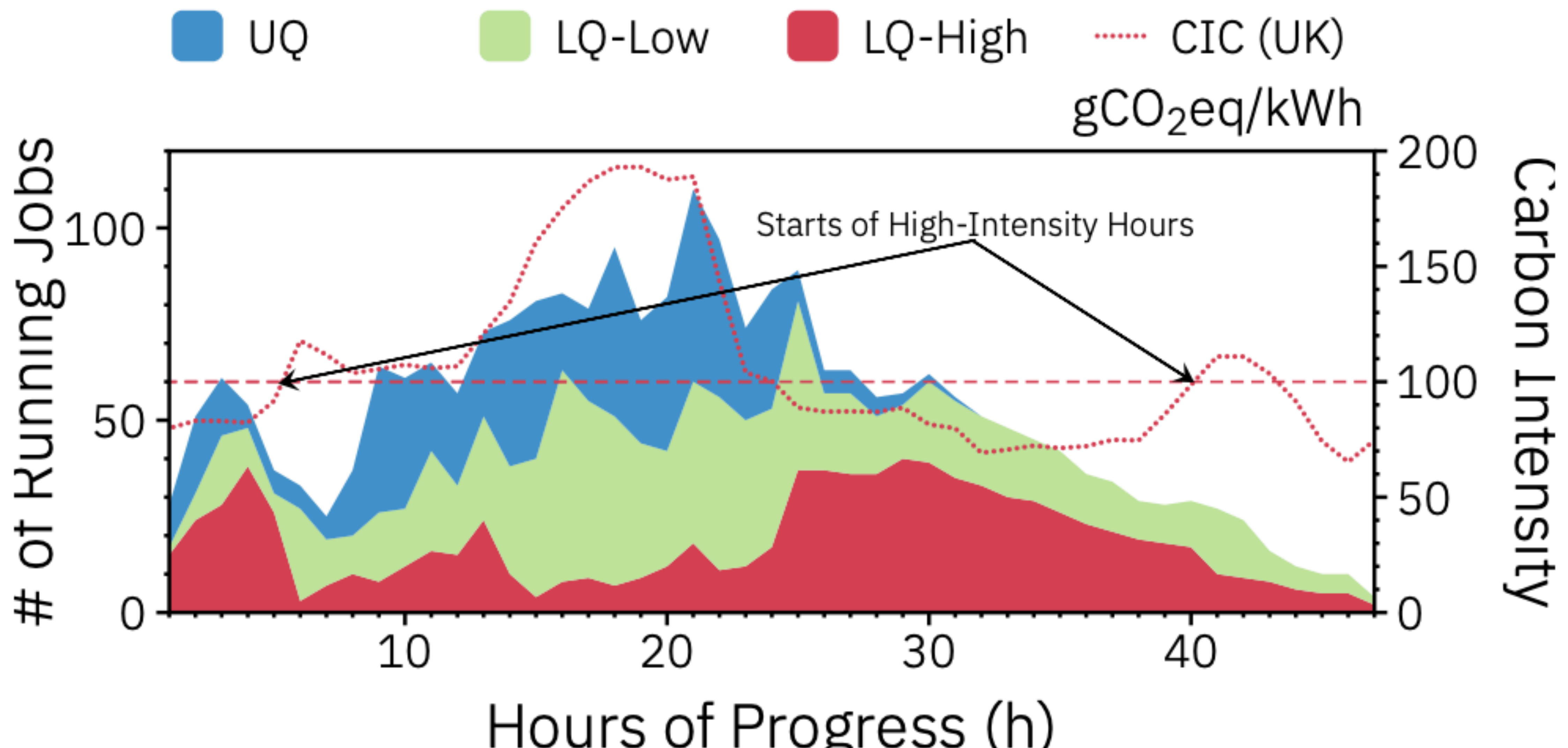
Job Size (% of Total Jobs)	JCT Increase	
	Average	Tail
Extra Small Jobs (0-9 minutes, 22%)	–0.5%	–0.4%
Small Jobs (10-59 minutes, 30%)	1.7%	1.2%
Medium Jobs (1 - 10 hours, 30%)	4.4%	4.8%
Large Jobs ($\geq$ 10 hours, 18%)	6.9%	5.8%

Carbon Reductions with Small Speed Tradeoffs



(a) Cluster-wide carbon footprint (b) Cluster-wide power draw

Cluster-wide carbon emission accumulation and power draw.



Number of running high- and lower-power jobs (left axis) responding to carbon intensity changes.

About the Presenter

Kaiqiang Xu | https://kqxu.com

Final-year PhD @ HKUST | Visiting Researcher @ Google

Research Focus. Developing new abstractions, parallelization strategies, and scheduling algorithms for machine learning computing. My work has been published in NSDI, OSDI, ASPLOS, and SIGMOD.

Next Step: Building AI Clusters for Usability, Efficiency, and Cost Savings.

Co-op welcome!