# Carbon-Aware Scheduling for Large-Scale Model Training

*Anonymous Author(s)*

**Abstract**

Training large machine learning models now consumes gigawatt-hours of electricity per run, yet the carbon intensity of that electricity varies by a factor of five or more depending on when and where the computation is placed. We present CarbonShift, a carbon-aware scheduler for distributed training workloads that exploits both temporal and spatial flexibility: jobs are paused, resumed, and migrated across geographically distributed GPU clusters in response to short-term grid carbon-intensity forecasts. Unlike prior work that treats training jobs as indivisible batch units, CarbonShift models the checkpoint-restart cost of each job explicitly and co-optimises carbon savings against deadline slack and cross-region data-egress overheads. We evaluate the system on a 168-hour trace of real carbon-intensity data from four European grid zones, replaying training workloads ranging from 350M to 13B parameters on a testbed of 128 A100 GPUs. CarbonShift reduces operational carbon emissions by 31.4% on average (up to 58% for deadline-tolerant jobs) while increasing mean completion time by only 6.2%. We further show that a simple forecast-error-aware hedging policy recovers most of the savings of a perfect-information oracle. Our results suggest that carbon-aware scheduling is a practical, immediately deployable lever for reducing the footprint of large-scale AI, complementary to hardware efficiency gains and model-level optimisations.

**Keywords:** *carbon-aware computing, machine learning training, job scheduling, grid carbon intensity, GPU clusters*

## 1 Introduction

The energy footprint of machine learning has grown from a research curiosity to a first-order operational concern. A single training run of a contemporary large language model can draw several megawatts continuously for weeks, and hyperscale operators now report that AI workloads dominate the growth of their data-centre electricity demand. While substantial effort has gone into making models and accelerators more efficient, comparatively little attention has been paid to a complementary lever: when and where the training computation is executed.

Grid carbon intensity is far from constant. In the four European bidding zones we study, the half-hourly carbon intensity of consumed electricity varied between 38 and 412 gCO2e/kWh over a representative winter week, driven by wind availability, hydro reservoir levels, and cross-border interconnector flows. A workload that is flexible in time or in placement can therefore achieve large reductions in operational emissions without any change to the model or the hardware.

In this paper we ask: how much carbon can a realistic training platform save by exploiting this flexibility, once the true costs of pausing, checkpointing, and migrating distributed jobs are taken into account? We answer this question with CarbonShift, a scheduler that integrates carbon-intensity forecasts into the placement and preemption decisions of a multi-region GPU cluster manager.

## 2 Method

CarbonShift models each training job as a sequence of checkpointable intervals. For every job we maintain an estimate of its checkpoint write time, restore time, and the size of the optimizer state that must cross region boundaries on migration. These estimates are learned online from the first few checkpoint cycles of the job itself, avoiding the need for offline profiling.

The scheduling core solves a rolling-horizon optimisation every 30 minutes. Given carbon-intensity forecasts for each region over the next 48 hours, the solver assigns job intervals to (region, time-slot) pairs so as to minimise total forecast emissions subject to three constraints: per-job deadline slack expressed as a maximum stretch factor, per-region GPU capacity, and a budget on cross-region egress derived from the platform's interconnect pricing. We show that the resulting integer program decomposes by region and can be solved in under two seconds for a thousand concurrent jobs.

Because carbon forecasts degrade quickly beyond a few hours, CarbonShift applies a hedging policy: decisions in the first two hours of the horizon are binding, while later assignments are treated as provisional and re-optimised at every step. A forecast-error model, fitted per region, inflates the carbon cost of provisional slots with high historical error, biasing the scheduler toward robust choices.

## 3 Results

We replay a workload trace of 412 training jobs (350M to 13B parameters) on a 128-GPU testbed spread across four sites, using measured half-hourly carbon-intensity data for the corresponding grid zones. Against a locality-aware but carbon-oblivious baseline, CarbonShift reduces total operational emissions by 31.4% while increasing mean job completion time by 6.2% and 95th-percentile completion time by 11.8%.

Savings are strongly bimodal. Jobs with deadline slack above 1.5x achieve a mean reduction of 47%, and the most flexible quartile reaches 58%, dominated by temporal shifting into overnight wind surpluses. Tightly-deadlined jobs still save 9% on average, almost entirely through spatial placement at admission time. Migration contributes 38% of total savings but accounts for 71% of the overhead, confirming that egress-aware migration budgeting is essential.

The hedging policy closes 86% of the gap between naive forecast-following and a perfect-information oracle. Sensitivity analysis shows the result is robust to forecast horizons between 24 and 72 hours and to checkpoint-cost estimation errors of up to 40%.

## 4 Conclusion

We presented CarbonShift, a carbon-aware scheduler for large-scale model training that treats checkpoint and migration costs as first-class citizens. On realistic traces it cuts operational emissions by roughly a third at a single-digit latency cost, and its hedging mechanism makes it robust to the substantial errors of real carbon forecasts.

We see two immediate avenues for future work: extending the formulation to co-optimise embodied carbon by steering load toward under-utilised hardware nearing retirement, and coupling the scheduler with demand-response markets so that flexibility is monetised as well as decarbonised.

## References


[1] D. Patterson, J. Gonzalez, U. Hölzle, et al. The carbon footprint of machine learning training will plateau, then shrink. Computer, 55(7):18–28, 2022.

[2] A. Radovanovi , R. Koningstein, I. Schneider, et al. Carbon-aware computing for d Power Systems, 38(2):1270–1280, 2023.

[3] W. A. Hanafy, Q. Liang, N. Bashir, D. Irwin, and P. Shenoy. CarbonScaler: Leveraging cloud workload elasticity for optimizing carbon-efficiency. In Proc. ACM SIGMETRICS, 2024.

[4] J. Dodge, T. Prewitt, R. Tachet des Combes, et al. Measuring the carbon intensity of AI in cloud instances. In Proc. ACM FAccT, pages 1877–1894, 2022.

[5] P. Wiesner, I. Behnke, D. Scheinert, K. Gontarska, and L. Thamsen. Let's wait awhile: How temporal workload shifting can reduce carbon emissions in the cloud. In Proc. ACM/IFIP Middleware, 2021.

[6] S. Naranayan and T. Eilam. Carbon-aware batch scheduling with grid forecast uncertainty. In Proc. HotCarbon Workshop, 2023.