Intelligent Surrogates Pay Attention to Data, Improving Multi-Objective HPC Optimization

Ashna Nawar Ahmed, Banooqa Banday,
Tanzima Z. Islam (Advisor)
Texas State University
San Marcos, Texas, USA
qvu10,bhb37,tanzima@txstate.edu

Terry Jones (advisor)
Oak Ridge National Laboratory
Knoxville, Tennessee, USA
trjones@ornl.gov

Abstract

High-performance computing (HPC) schedulers must balance runtime and power. We present a surrogate-assisted multi-objective Bayesian optimization (MOBO) framework using TabNet regressors and models trained on attention-based embeddings, coupled with active-learning sample selection. The surrogates predict runtime and power, enabling MOBO to efficiently discover Pareto-optimal node allocations. We quantify trade-offs with Pareto fronts, Hypervolume (HV), and Spread across PM100 and Adastra production traces. MOBO improves HV over single-objective baselines by 24% (PM100) and 37% (Adastra) and attains lower Spread in 75% of surrogate families. Active learning reduces evaluations by 53–70%. To our knowledge, this is the first demonstration of embedding-informed surrogates for MOBO applied to HPC scheduling, optimizing runtime–power trade-offs on production datasets.

ACM Reference Format:
Ashna Nawar Ahmed, Banooqa Banday, Tanzima Z. Islam (Advisor) and Terry Jones (advisor). 2025. Intelligent Surrogates Pay Attention to Data, Improving Multi-Objective HPC Optimization. In *Proceedings of Supercomputing (SC '25).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Motivation & Approach

Exascale HPC must balance two competing forces: runtime for throughput and power for sustainability. Traditional heuristics such as FCFS with backfilling or bin-packing require manual tuning and cannot capture runtime–power interactions, leaving inefficiencies at scale [6]. Automated, data-driven methods promise better solutions, but face three persistent challenges.

First, HPC telemetry is massive yet irregular, with missing values and inconsistent counters [8]. Direct surrogate training often collapses. We introduce intelligent sample acquisition via active learning to query the most informative data, improving efficiency and robustness.

Second, information is scattered across modalities: categorical fields (queue, partition), numerical metrics (runtime, memory, power), and irregular signals (GPU bursts, I/O spikes). Noise and redundancy make feature selection difficult, and surrogates destabilize on log-transformed targets [3]. Attention-based embeddings address this by highlighting informative signals and suppressing noise, enabling lightweight regressors (RF, XGBoost) to outperform heavier transformer architectures while remaining deployable.

Finally, runtime and power conflict inherently [5]. Prior HPC surrogate modeling has used single-objective BO [4], obscuring trade-offs. We instead integrate surrogates with multi-objective BO (MOBO) to expose Pareto fronts of runtime–power balances. MOBO has succeeded in accelerator tuning [9] and laser–plasma design [7], but has not been applied to HPC scheduling. This motivates our two hypotheses: (**H1**) attention-based embeddings improve surrogate quality; and (**H2**) MOBO captures runtime–power trade-offs more effectively than SOBO or random baselines.

2 Experimental Setup

We evaluate on two real HPC job-log datasets: PM100 [2] and Adastra [1]. Preprocessing handles missing values, aggregates node-level power, and aligns categorical, numerical, and time-series features. Surrogates are trained either as direct TabNet regressors or as embedding-informed models (TabNet embeddings + RF/XGBoost/LightGBM). These surrogates drive multi-objective Bayesian optimization (MOBO) to explore runtime–power trade-offs. Baselines include SOBO (runtime-only, power-only) and Random. Experiments were run on TACC's Stampede3 [10] supercomputer.

3 Preliminary Results

We evaluate results by contribution, using PM100 and Adastra datasets.

Impact of Intelligent Sample Acquisition
Active-learning–guided surrogates converge faster and are more stable than training on the full dataset (Fig. 4, 5). This occurs because the targets exhibit enormous variability, and naïvely using all samples exposes models to noisy and uninformative regions of the space, causing them to fail to generalize. We achieved nearly identical surrogate accuracy (MAPE $\approx$ 0.99 for both runtime and power) using only 47% of the original PM100 samples (231,238 $\longrightarrow$ 109,202) and 30% of the Adastra samples (15,285 $\longrightarrow$ 4,547) through intelligent sample acquisition, substantially reducing training cost without sacrificing model quality.

Impact of Attention-Based Embeddings (Hypothesis 1)
Embedding-informed surrogates (RF/XGB/LGBM) generally outperform TabNet regressors. On PM100, embeddings reduce Spread by ~99%, and under SOBO achieve orders-of-magnitude higher HV than TabNet, though they collapse under MOBO. On Adastra,

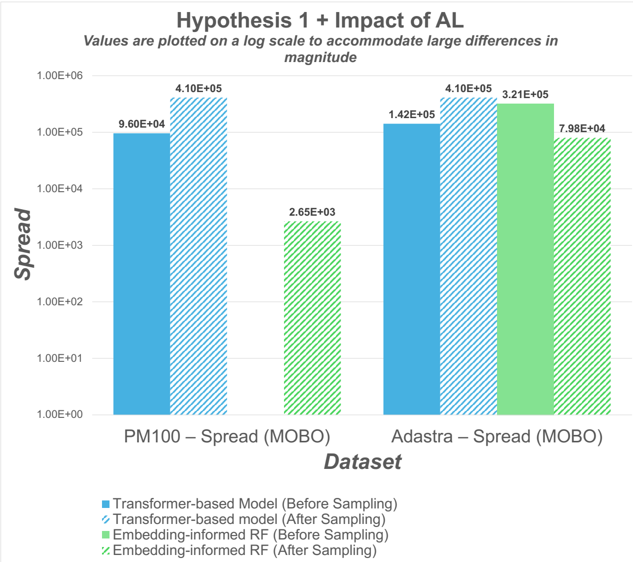

Figure 1: On PM100 and Adastra, embedding-informed surrogates achieve higher HV than TabNet regressors, confirming improved surrogate quality.

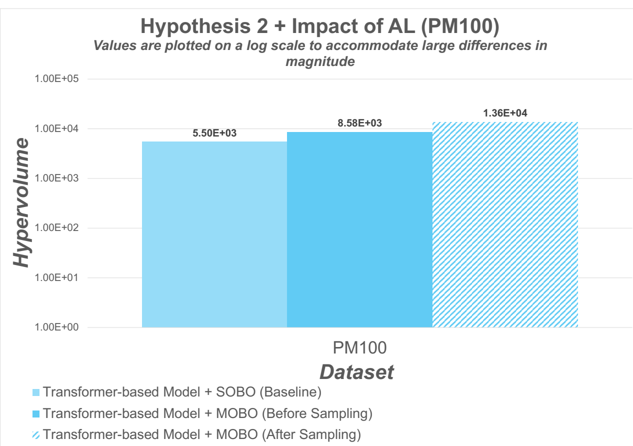

Figure 2: On PM100, MOBO outperforms SOBO with higher HV, capturing broader runtime–power trade-offs.

embeddings consistently improve HV by 37% and reduce Spread by ~90% (Table 1, Fig. 1). Attention-based embeddings highlight relevant signals while suppressing redundancy and noise, enabling lightweight models to train faster and generalize better. **Overall, these results support Hypothesis 1**.

Impact of MOBO on Runtime–Power Trade-offs (Hypothesis 2)
MOBO outperforms SOBO in 3/4 surrogate families, improving HV on PM100 and reducing Spread on Adastra (Figs. 2, 3). Gains are uneven, and Random baselines sometimes yield inflated metrics due to outlier dominance. These issues may be mitigated through robust metrics or bounded reference points. MOBO's ability to jointly optimize runtime and power yields more balanced Pareto fronts. **Overall, these findings support Hypothesis 2**.

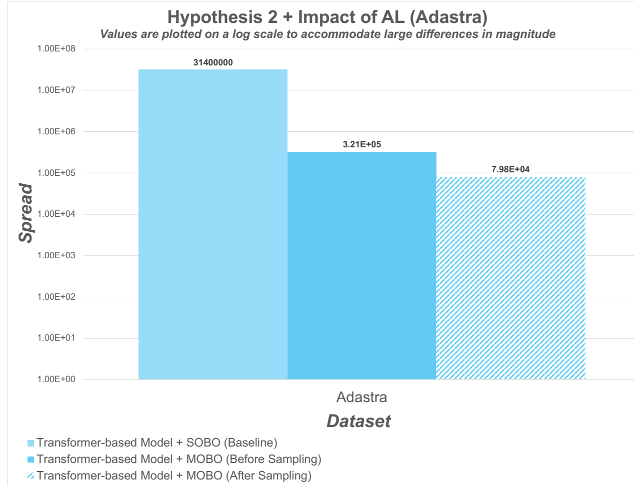

Figure 3: On Adastra, MOBO achieves lower Spread than SOBO baselines, producing more balanced Pareto solutions.

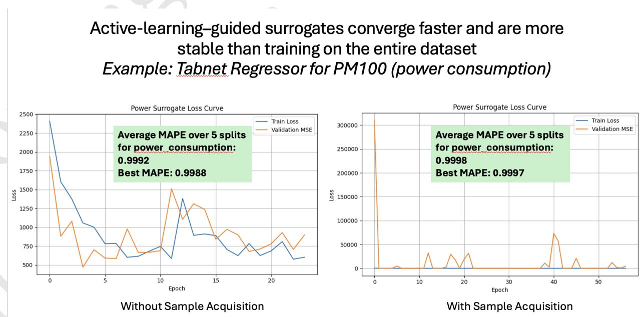

Figure 4: Power Consumption Surrogate Comparison

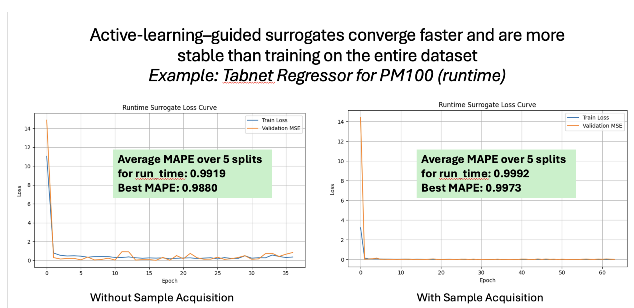

Figure 5: Runtime Surrogate Comparison

4 Conclusions

This work proposes an AI/ML framework for making surrogate modeling data-efficient and intelligent. We use sample acquisition for irregular telemetry, embeddings for noisy features, and MOBO for runtime–power trade-offs. Together, these advances position

Table 1: Summary of results supporting Hypotheses 1 and 2. Full results are available via the QR code in the poster.

Dataset	Metric	Hypothesis 1 (Embeddings vs. TabNet)	Hypothesis 2 (MOBO vs. Baselines)
PM100	HV	✓ Embeddings have *orders of magnitude* higher HV than Regressor	✓ Improved HV 24% vs. SOBO–Runtime (TabNet Regressor)
	Spread	✓ Embeddings have ∼99% lower Spread than Regressor	✓ MOBO has best in 3/4 families (75%)
Adastra	HV	✓ Embeddings have 37% more HV than Regressor	✓ MOBO improved HV 37% vs. SOBO–Runtime (TabNet Regressor)
	Spread	✓ Embeddings have ∼90% lower Spread than Regressor	✓ MOBO best in 3/4 families (75%)

surrogate modeling to enhance HPC schedulers with AI/ML-based decision-making.

Acknowledgments

This material is based upon work supported by the U.S. Department of Energy, Office of Science under Award Number DE-SC0022843. Additional support was provided by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Next-Generation Scientific Software Technologies program under Contract Number DE-AC05-00OR22725.

References

[1] 2024. Adastra: Log data from the HPE-Cray EX supercomputer at CINES. CINES documentation; TOP500 system description; GENCI reports.

[2] Francesco Antici, Mohsen Seyedkazemi Ardebili, Andrea Bartolini, and Zeynep Kiziltan. 2023. PM100: A Job Power Consumption Dataset of a Large-scale Production HPC System. In *Proceedings of the SC '23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis* (Denver, CO, USA) *(SC-W '23)*. Association for Computing Machinery, New York, NY, USA, 1812–1819. doi:10.1145/3624062.3624263

[3] Sercan Ö. Arik and Tomas Pfister. 2021. TabNet: Attentive Interpretable Tabular Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 35, 8 (May 2021), 6679–6687. doi:10.1609/aaai.v35i8.16826

[4] Alexandru Calotoiu, Torsten Hoefler, Marius Poke, and Felix Wolf. 2013. Using automated performance modeling to find scalability bugs in complex codes. In *SC '13: Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. 1–12. doi:10.1145/2503210.2503277

[5] Kalyan Deb. 2001. *Multiobjective Optimization Using Evolutionary Algorithms. Wiley, New York.*

[6] Yuping Fan, Zhiling Lan, Taylor Childers, Paul Rich, William Allcock, and Michael E. Papka. 2021. Deep Reinforcement Agent for Scheduling in HPC. arXiv:2102.06243 [cs.DC] https://arxiv.org/abs/2102.06243

[7] F. Irshad, S. Karsch, and A. Döpp. 2023. Multi-objective and multi-fidelity Bayesian optimization of laser-plasma acceleration. *Phys. Rev. Res.* 5 (Jan 2023), 013063. Issue 1. doi:10.1103/PhysRevResearch.5.013063

[8] Byung H. Park, Saurabh Hukerikar, Ryan Adamson, and Christian Engelmann. 2017. Big Data Meets HPC Log Analytics: Scalable Approach to Understanding Systems at Extreme Scale. In *2017 IEEE International Conference on Cluster Computing (CLUSTER)*. 758–765. doi:10.1109/CLUSTER.2017.113

[9] Ryan Roussel, Auralee L. Edelen, Tobias Boltz, Dylan Kennedy, Zhe Zhang, Fuhao Ji, Xiaobiao Huang, Daniel Ratner, Andrea Santamaria Garcia, Chenran Xu, Jan Kaiser, Angel Ferran Pousa, Annika Eichler, Jannis O. Lübsen, Natalie M. Isenberg, Yuan Gao, Nikita Kuklev, Jose Martinez, Brahim Mustapha, Verena Kain, Christopher Mayes, Weijian Lin, Simone Maria Liuzzo, Jason St. John, Matthew J. V. Streeter, Remi Lehe, and Willie Neiswanger. 2024. Bayesian optimization algorithms for accelerator physics. *Phys. Rev. Accel. Beams* 27 (Aug 2024), 084801. Issue 8. doi:10.1103/PhysRevAccelBeams.27.084801

[10] Texas Advanced Computing Center. 2025. Stampede3 User Guide. https://docs.tacc.utexas.edu/hpc/stampede3/. Accessed: 2025-08-18.