Learned surrogates for fast-charging optimisation: how accurate must the constraints and their gradients be?

Keywords. Neural operators, derivative-enhanced (Sobolev) training, surrogate-based optimisation, constrained optimisation, lithium-ion fast charging

Supervisor: Shankar A. Deka, shankar.deka@aalto.fi.

Advisor: Hassan Rouhi, hassan.rouhi@aalto.fi.

Research Problem Statement. How fast a battery may be charged is decided by solving an optimisation problem: deliver as much charge as possible without crossing the voltage and temperature limits that protect the cell. Evaluating those limits needs a physics-based model of the cell, which is accurate but slow, so it is increasingly replaced by a learned surrogate [1,2,3].

A surrogate used this way is not only predicting — it is being optimised against, and that changes what it has to get right. Ordinary training fits the predicted trajectory; it does not target the derivatives the optimiser follows, nor accuracy at the moments where a limit is nearly reached.

Two kinds of error matter here, and neither shows up in an average accuracy score. A model can predict the response well and still be wrong about the direction in which the optimiser should move. And a small, almost constant offset in the predicted limit is enough to turn a protocol that looks feasible into one that is not.

This project will measure how those errors propagate into charging decisions, and whether training a surrogate on derivatives as well as values [5] improves the decisions enough to justify its cost. Surrogates of a physics-based model of the cell, including its thermal behaviour, are trained on simulated data [4]; the protocols they produce are then replayed on the simulator and counted for violations, against a baseline in which the simulator is optimised directly.

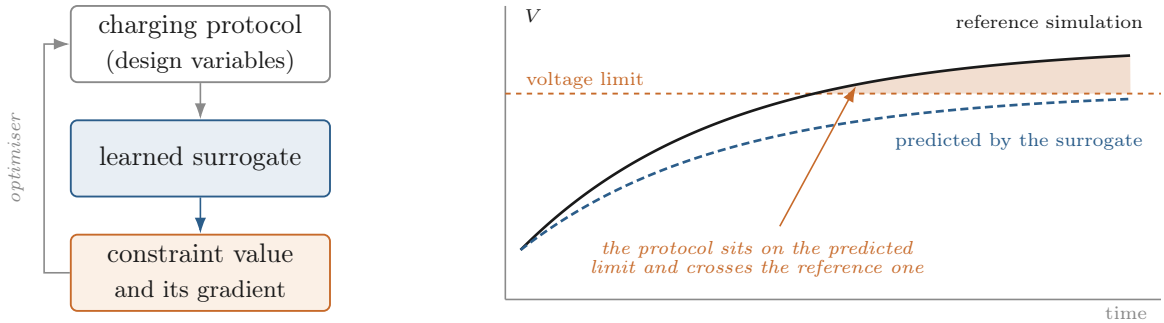

Figure 1: The setting. A protocol optimised against a surrogate comes to rest exactly on the limit the surrogate predicts; a small error in that predicted constraint puts the reference simulation over it. The thesis measures how errors in the constraints, and in their gradients with respect to the design variables, propagate into infeasible charging decisions.

The outcome will be an account of the accuracy a surrogate actually needs to be worth optimising against, the safety margin that goes with it and what that margin costs, and whether the surrogate repays its training cost on this workload. A finding that it does not is equally useful and will be reported as one.

Expected workload: The work will include:

1. Review learned surrogates and operator-learning methods for physical systems and their use for battery models, derivative-enhanced training, and the standard ways of enforcing limits along a trajectory in numerical optimisation.

2. Formulate the charging problem explicitly — maximise the charge delivered within a fixed charging time, over a small number of current amplitudes at fixed switching times, with the voltage and temperature limits enforced at specified points along the trajectory — and establish a direct optimisation baseline on the simulator, warm-started across a family of related problems, so that the comparison is made against a realistic alternative rather than a naive

one.

3. Generate training data together with the corresponding sensitivities, verifying numerical convergence and checking a sample of the sensitivity labels against finite differences, and train surrogates with and without derivative supervision; evaluate the errors the optimiser actually consumes, separating the regions where a limit is nearly reached from the rest of the trajectory.

4. Close the loop: optimise against each surrogate, replay the resulting protocols on the simulator on a finer time grid, count violations, choose a constraint margin on validation problems and measure the violations that remain and the charge that is lost on unseen ones, and account for the total computation, including data generation and training.

Feasibility and completion. Approximately six months of full-time work on Aalto's Triton computing cluster. The model — a single-particle model with electrolyte dynamics, coupled to a lumped thermal description — together with the cell parameter set and the cooling conditions, is fixed at the start and taken from published sources, and all data are generated from an open-source simulator, so there is no data-acquisition risk. The simulator provides reference solutions rather than exact truth, and checking their numerical convergence is part of the work. The principal risk lies in obtaining trustworthy sensitivities and a reliable optimisation benchmark rather than in training the network, so month one is spent showing that the simulator, its derivatives and the direct optimiser work together on a small example. Months two and three cover data generation and training, months four and five the optimisation experiments, and month six analysis and writing.

Optional extension: After the core results, extend from the reduced model to a richer one, or from a fixed family of charging problems to a receding-horizon setting in which the problem is re-solved as the cell charges.

Prerequisite: Good knowledge of machine learning, linear algebra and numerical methods; Python and PyTorch experience. Familiarity with numerical optimisation is useful but can be acquired during the project; prior knowledge of electrochemistry is not required.

Type of work: Approximately 35% methods and analysis, 65% implementation. Writing proceeds throughout the project. Development of new electrochemical models, laboratory measurements and hardware-level controller design are outside scope.

References

[1] P. Brendel, I. Mele, A. Rosskopf, T. Katrašnik and V. Lorentz, "Parametrized physics-informed deep operator networks for Design of Experiments applied to Lithium-Ion-Battery cells," Journal of Energy Storage 128, 117055, 2025. DOI: 10.1016/j.est.2025.117055.

[2] A. A. Panahi, D. Luder, B. Wu, G. Offer, D. U. Sauer and W. Li, "Fast and generalisable parameter-embedded neural operators for lithium-ion battery simulation," Energy and AI 22, 100647, 2025. DOI: 10.1016/j.egyai.2025.100647.

[3] M. Borah, Q. Wang, S. Moura, D. U. Sauer and W. Li, "Synergizing physics and machine learning for advanced battery management," Communications Engineering 3, 134, 2024. DOI: 10.1038/s44172-024-00273-6.

[4] V. Sulzer, S. G. Marquis, R. Timms, M. Robinson and S. J. Chapman, "Python Battery Mathematical Modelling (PyBaMM)," Journal of Open Research Software 9(1), 14, 2021. DOI: 10.5334/jors.309.

[5] Y. Qiu, N. Bridges and P. Chen, "Derivative-enhanced deep operator network," arXiv:2402.19242, 2024.