Advanced Analytics for EV-batteries

Dr. C. Watkinson, Dr. T. McAloone, Dr. W. Handley, Prof. M. Hobson
*(PolyChord Ltd, UK).*

Juergen Schneider, Alexander Ulz
*(AVL, Austria).*

**Abstract**

In this talk, we present a novel data-science pipeline for accelerating inherently-slow, complex simulations by many orders of magnitude. This means a simulation user gets reactive answers that would otherwise have taken hours or days to obtain, drastically speeding up design processes, enabling more full investigations of even high-dimensional simulations, and allowing complex simulations to be applied to problems that require sampling (e.g. making predictions from real-world data). This pipeline also delivers fresh insights into simulations by more fully investigating the influence of all parameters on the simulation output, and provides a ranking of input parameters by the strength of their influence on the simulation output. The pipeline emerged from a joint endeavour by PolyChord Ltd (a spin-out from the University of Cambridge Astrophysics group) in collaboration with AVL Advanced Simulations, through AVL’s accelerator program ‘Creators Expedition’. It has been developed for use with AVL’s physics-based simulations of Electric-Vehicle (EV) batteries. In this talk, we will discuss how we apply a cutting-edge nested sampling algorithm to provide both a fuller investigation of the original simulation than is otherwise possible, and the ideal dataset for training an AI twin of the original simulation. We will outline the AI approaches we use to deliver an emulator with uncertainty quantification built in, providing stable and unbiased results when applying our AI twin in the field. In tackling this problem, we have identified a near-term application in improving the efficiency and quality of the EV battery design process, and longer-term applications in battery-management systems and warranty/insurance risk analysis.

# Overview of Methods

Thanks to a collaboration between PolyChord Ltd and AVL Advanced Simulations, we have developed a novel data-science pipeline for accelerating inherently slow, complex simulations by many orders of magnitude. Using AVL’s simulation software, we designed a 39-parameter model which simulated an EV battery undergoing repeated charge/discharge cycles for 50 hours. We could then assess a number of key metrics which indicate changes in the battery’s performance at the end of this process, e.g. State of Health, Power Loss, Film Layer Thickness. In the development of our pipeline we focussed on investigating the Cell Voltage after the 50 simulated hours. Our goal was to produce a pipeline which could take this model and create an AI twin that could learn how the simulation ultimately calculates the Cell Voltage given the complex interplay between the 39 parameters under consideration.

The pipeline begins by using our proprietary PolyChord sampling algorithm [1]: a next-generation adaptation of Nested Sampling [2]. The manner in which the PolyChord sampler navigates parameter space provides several distinct advantages for this type of problem. The algorithm maintains a number of ‘live points’, which are samples taken of the output of the target function (in this case the simulation output) at specific points in parameter space. The live points with the lowest value outputs are systematically replaced by generating new higher-value points, whilst keeping track of every sample we take. As the algorithm proceeds, its samples exponentially hone-in on the peaks of the function. This approach efficiently maps out a function’s output hypersurface in input-parameter space, whilst being resistant to getting stuck in local maxima – a common problem among other optimisation/sampling algorithms such as Markov Chain Monte Carlo. The PolyChord algorithm’s effective exploration of a simulation’s parameter space, can uncover previously unseen insights into the simulation’s behaviour. In addition, we obtain a set of samples which are spread out across the entire hypersurface of the function and exponentially cluster around the peaks – usually the regions of most interest.

Modern simulations, such as AVL’s physics-based simulations of Electric-Vehicle batteries, are typically complex, containing many parameters which describe a high-dimensional, multimodal surface. If one were to sample this parameter space using a more naïve approach, such as Uniform Random sampling, or Latin Hypercube sampling, the resulting dataset will not include samples from all the peaks of the target function. Therefore, when compared with more standard approaches, the samples generated using the PolyChord algorithm form an ideal dataset for training an AI twin of a simulation. Our pipeline uses PolyChord to sample the simulation output and use these samples to train a Neural Network to learn the mapping of simulation inputs to outputs.

# Neural Network Model

During development of this Proof-of-Concept pipeline, the architecture of the Neural Network (NN) was kept very simple, to ensure that the results of our testing could be connected directly to the use of PolyChord for generating training data, unmired by complicated Machine-Learning machinery. We used a fully connected feed-forward NN, consisting of a 39-node input layer (for the 39 parameters in the AVL simulation), followed by two 26-node hidden layers and a single output node representing the learned output. We use more than one hidden layer to more accurately approximate non-linear functions, and the number of nodes in the hidden layers was chosen using a common rule of thumb that the hidden layers should be 2/3 the size of the input layer. The resultant NN is simple, but still flexible enough to learn what we need it to. It is worth noting that by recording the values of other metrics of interest (e.g. State of Health) during sampling, we could extend this network design to have two or more nodes in the output layer, effectively allowing it to learn how to map multiple outputs at once.

# Results

In our initial exploration, we produced two NNs using the architecture described above: one trained on the PolyChord samples; and one trained on a set of samples drawn randomly. Figure 1 shows a comparison of the values predicted by each of these networks against the ‘true’ from AVL’s simulation. We can clearly see that, apart from a handful of outliers amongst the over 100,000 samples in this plot, the predictions from both networks tend to cluster fairly 

1. A comparison of the True vs. Predicted values for the Neural Networks trained on random samples (red) and PolyChord (blue). The points in this plot were selected by reserving a random 20% of the PolyChord samples as a test set which were not used in training for either of the networks.

closely around the black line, representing what we’d expect from a perfect predictor. That is, until we reach higher values of cell voltage, where the predictions of the NN trained on random samples begin to diverge. This is a result of the random sampling missing the peaks of the surface in parameter space, leading to underpredictions in the final model. The PolyChord trained network, however, is able to give relatively accurate predictions even at high voltages.

Of course, no predictor is going to be perfectly accurate, especially given that we have used a very simple NN architecture for both of our models, in order to highlight the difference between the two sampling methods. However, to more closely investigate the spread in both networks’ predictions, we can examine Figure 2, which shows the distribution of the percentage error of predictions for both networks. Again, we can see a marked difference between the two models, with the NN trained on random samples once more showing signs of underpredicting. In addition, by calculating these errors, we can identify which reasons of the parameter space generally give better/worse predictions. This gives an in-built uncertainty that is tied to our predictions and can inform potential users of the AI twin that they may want to verify some predictions using the original simulation if that uncertainty is high. This also will indicate the regions to focus on when generating additional samples to retrain the model through active learning. We are now developing uncertainty-aware neural networks by transferring methods developed by the company directors for use in astrophysics.

1. A comparison of the percentage error on the predictions of both networks. Here we can see that the distribution of errors from the PolyChord-trained NN is much tighter and is centred far closes to zero as one might hope. The NN trained on random samples, however, gives a distribution that is much more spread out, with its bulk far from zero, again indicating the issue of chronic underprediction.

# References

[1] Handley, W. J. and Hobson, M. P. and Lasenby, A. N. (2015). Monthly Notices of the Royal Astronomical Society Volume 453: Oxford University Press (OUP). p4385-4399.

[2] Skilling, John (2004). Bayesian Inference and Maximum Entropy Methods in Science and Engineering: 24th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering. p395-405.