Battery Health Predictor: WIDS 2025 Project Report

Maitreyee Markale 24b3926

February 1, 2026

Abstract

This report summarizes the development of a battery state-of-health (SoH) prediction system over four weeks. The project progresses from foundational physics-informed neural networks to electrochemical modeling and machine learning approaches, culminating in an integrated transformer-based predictor with physics constraints.

1 Introduction

This repository contains the WIDS 2025 project on battery modeling, physics-informed neural networks, and state-of-health (SoH) estimation. The work spans four weeks, building progressively from basic PINN concepts to advanced transformer models with electrochemical integration.

2 Week 1: Exponential Decay PINN

2.1 Purpose

Train a physics-informed neural network (PINN) to learn the exponential decay ODE $\frac{dy}{dt} + \lambda y = 0$ with initial condition $y(0) = y_0$. The network learns from physics without labeled data.

2.2 Theoretical Background

Physics-Informed Neural Networks (PINNs) combine neural networks with physical laws. For ODEs, the network approximates the solution $y(t) \approx \hat{y}(t; \theta)$, where θ are trainable parameters. Automatic differentiation computes derivatives, allowing enforcement of physics through residual losses:

$$\mathcal{L}_{physics} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{d\hat{y}}{dt}(t_i) + \lambda \hat{y}(t_i) \right)^2$$

Boundary/initial conditions are enforced via:

$$\mathcal{L}_{ic} = (\hat{y}(0) - y_0)^2$$

Total loss: $\mathcal{L} = \mathcal{L}_{physics} + \alpha \mathcal{L}_{ic}$. This enables zero-shot learning from governing equations alone.

2.3 Key Components

- `exponential_decay_pinn.py`: Main training script
- `requirements.txt`: Dependencies (PyTorch, NumPy, Matplotlib)

2.4 Methodology

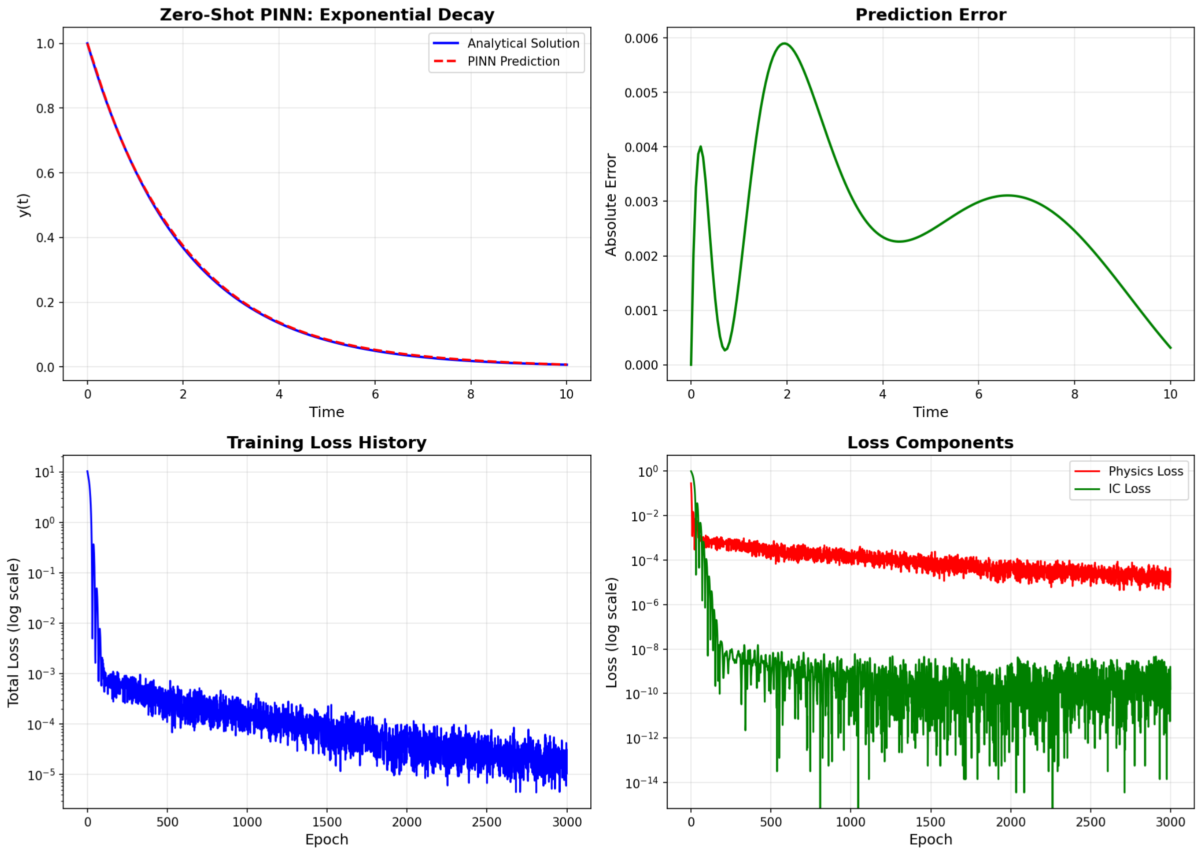
The neural network maps time t to $y(t)$. Automatic differentiation computes $\frac{dy}{dt}$, forming physics loss from ODE residual. Initial condition loss ensures $y(0) = y_0$.

2.5 How to Run

```
pip install -r requirements.txt
python exponential_decay_pinn.py
```

2.6 Results

The trained PINN matches the analytic solution $y(t) = y_0 e^{-\lambda t}$. Training plots compare learned vs. analytic curves.



3 Week 2: SPM 1C Discharge and OCV Fit

3.1 Purpose

Run a single-particle model (SPM) discharge at 1C using PyBaMM, extract concentration profiles, and fit a polynomial SOC $\rightarrow$ Voltage mapping.

3.2 Theoretical Background

The Single-Particle Model (SPM) approximates battery electrodes as spherical particles undergoing diffusion. The governing equation for concentration $c(r,t)$ in a particle is:

$$\frac{\partial c}{\partial t} = D\left(\frac{\partial^2 c}{\partial r^2} + \frac{2}{r}\frac{\partial c}{\partial r}\right)$$

With boundary conditions for flux at surface $r = R$: $-D\frac{\partial c}{\partial r} = \frac{I}{FA}$, where I is current, F Faraday constant, A surface area.

Terminal voltage includes OCV, ohmic drop, and overpotentials. OCV is fitted as a polynomial $V_{oc}(SOC) = \sum_{k=0}^{5} a_k SOC^k$.

3.3 Key Components

- `run_spm.py`: SPM simulation and data extraction
- `ocv_fit.py`: Polynomial fitting utility
- `requirements.txt`: Dependencies including PyBaMM

3.4 Methodology

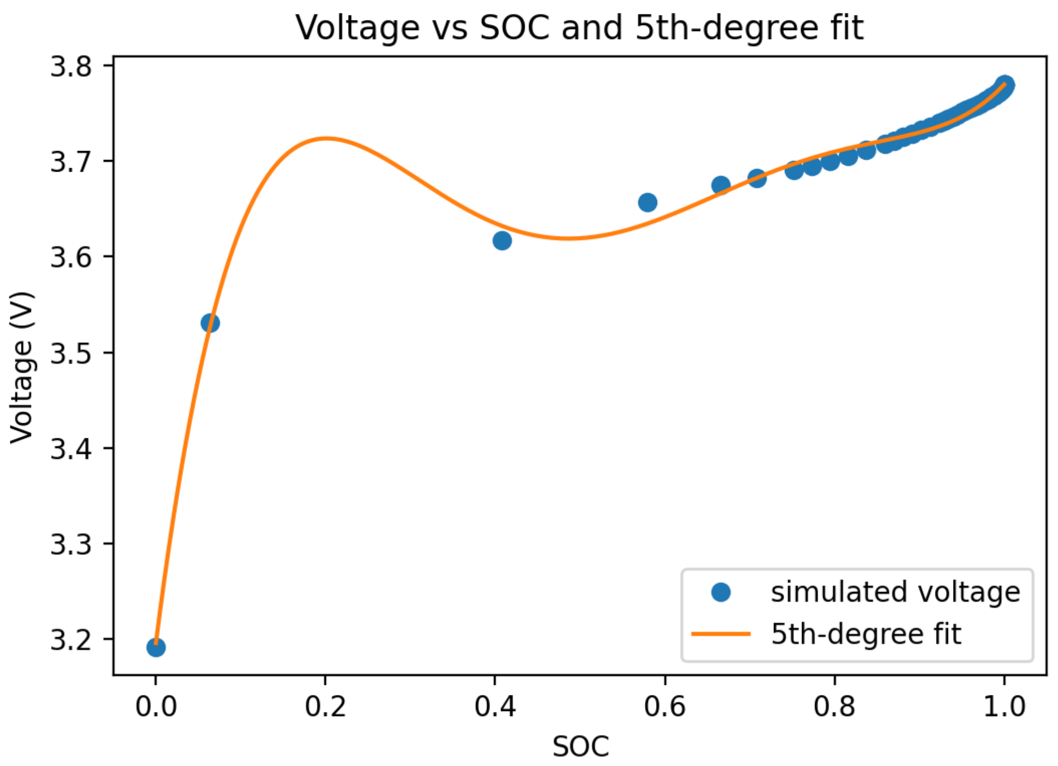
Simulates 1C discharge for 1 hour, extracts negative electrode concentration profiles at multiple timestamps. Fits 5th-degree polynomial to voltage vs. SOC data.

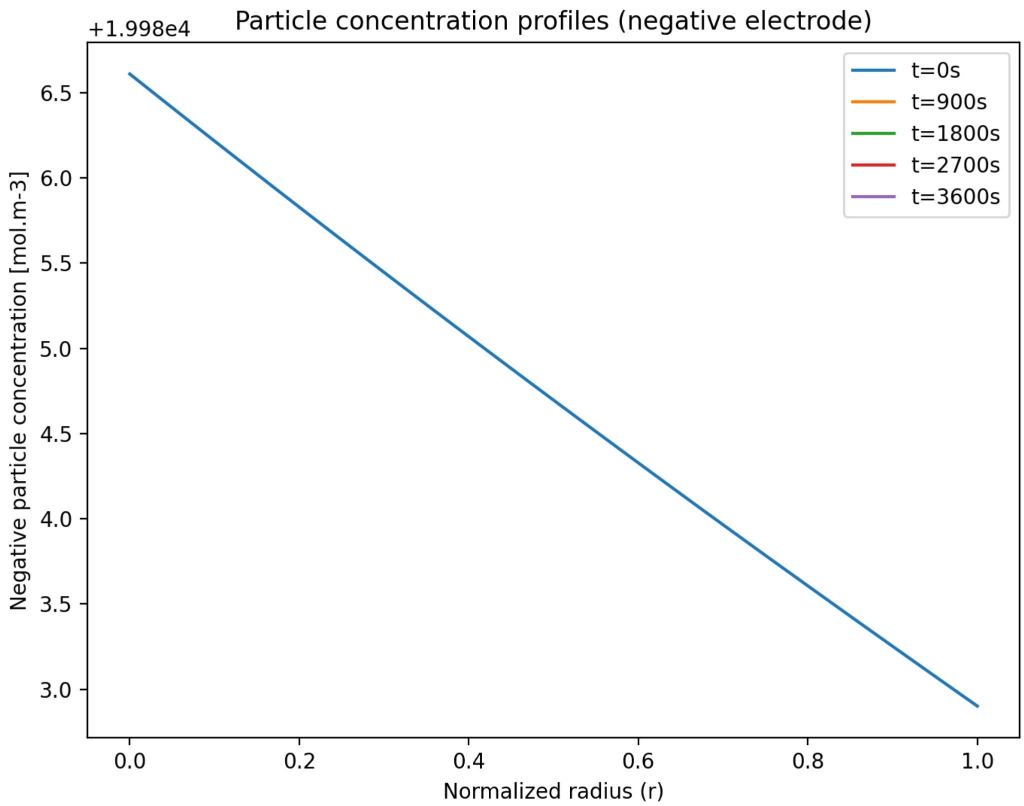
3.5 How to Run

```
python -m venv .venv
.venv\Scripts\Activate.ps1
pip install -r requirements.txt
python run_spm.py
python ocv_fit.py
```

3.6 Outputs

- Concentration profile plots
- Voltage vs. SOC fit plot
- Saved polynomial coefficients (`ocv_coeffs.npy`)





4 Week 3: Transformer-Based SoH Estimation

4.1 Purpose

Build a transformer model to predict SoH, charge capacity (Q), and energy (E) from NASA battery dataset discharge cycles, using physics-constrained losses.

4.2 Theoretical Background

Transformers use self-attention to process sequences. For battery cycles, input sequence is binned discharge data $[V, I, T, SOC, dt]_{1:T}$. Multi-head attention computes:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Physics constraints enforce conservation laws: - Coulomb counting: $Q = \int I dt$ - Energy conservation: $E = \int V I dt$

Loss includes data fidelity + physics penalties: $\mathcal{L} = \mathcal{L}_{data} + \lambda_q \mathcal{L}_{coulomb} + \lambda_e \mathcal{L}_{energy}$.

4.3 Key Components

- `data_processing.py`: Dataset download and preprocessing
- `transformer_model.py`: PyTorch transformer architecture
- `train_transformer.py`: Training with physics constraints
- `evaluate_transformer.py`: Model evaluation

- `visualize.py`: Result visualization

4.4 Methodology

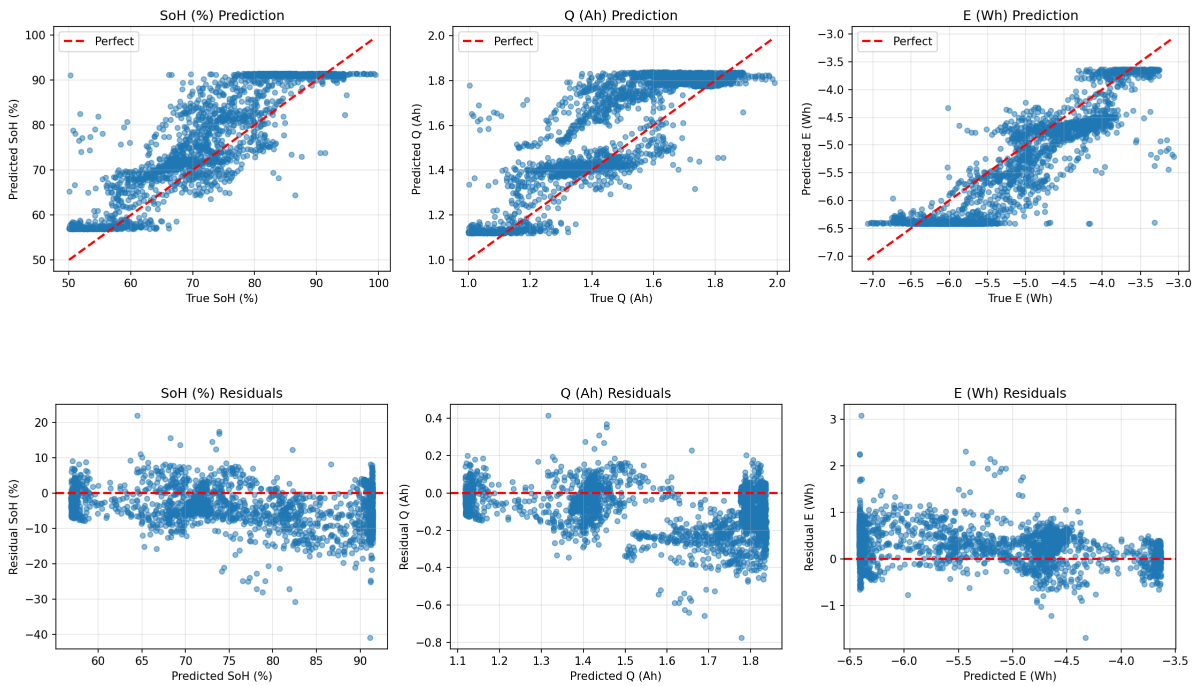
Processes 2694 discharge cycles into binned (20 bins) and physics (200 points) datasets. Transformer takes 5 features (V, I, T, SoC, dt) and predicts [SoH%, Q_Ah, E_Wh] with coulomb counting and energy conservation losses.

4.5 How to Run

```
cd Week3
pip install -r requirements.txt
python data_processing.py --download --out_dir data_out
python train_transformer.py --main_csv data_out/battery_main_binned20.csv --
    phys_csv data_out/battery_phys_resampled200.csv --epochs 100
python evaluate_transformer.py --main_csv data_out/battery_main_binned20.csv --
    phys_csv data_out/battery_phys_resampled200.csv
python visualize.py --main_csv data_out/battery_main_binned20.csv --phys_csv
    data_out/battery_phys_resampled200.csv --output_dir plots
```

4.6 Results

- SoH: MAE=5.2%, R^2=0.70
- Q: MAE=0.12 Ah, R^2=0.58
- E: MAE=0.38 Wh, R^2=0.75



5 Week 4: SPM Integration and Parameter Fitting

5.1 Purpose

Enhance SPM with OCV characterization and parameter fitting from NASA dataset for ensemble predictions with transformer.

5.2 Theoretical Background

Enhanced SPM includes: - OCV from polynomial fit - Ohmic resistance R_0: $V = V_{oc} - IR_0$ - RC circuit for transients: $\tau = R_{ct}C_{dl}$, voltage response $V(t) = V_{oc} + IR_0 + IR_{ct}e^{-t/\tau}$

Parameters fitted via least squares from discharge data. Ensemble with transformer improves robustness by combining data-driven and physics-based predictions.

5.3 Key Components

- `spm_model.py`: Enhanced SPM class with OCV polynomial and resistances
- `train_spm.py`: Parameter fitting from physics dataset

5.4 Methodology

Fits ohmic resistance (R0) and extracts OCV profile from high-resolution discharge data. Simulates voltage curves with transient dynamics.

5.5 How to Run

```
cd Week4
pip install -r requirements.txt
python train_spm.py --phys_csv ../Week3/data_out/battery_phys_resampled200.csv
```

5.6 Outputs

SPM instance for discharge simulation, useful for physics-informed baselines and ensemble methods.

6 Conclusion

The project demonstrates a comprehensive approach to battery SoH prediction, combining physics-informed machine learning (PINNs, transformers) with electrochemical modeling (SPM). Week 1 establishes PINN fundamentals, Week 2 introduces SPM basics, Week 3 implements advanced ML with physics constraints, and Week 4 integrates models for robust predictions.