Deep Learning to Predict Lithium-Ion Battery State-of-Health from Partial Discharge Data: Comparing 1-D Temporal Models and Novel Convolutional Curve-Image CNNs

Steven D. Liu
Stanford University
Stanford, California
sdliu@stanford.edu

Abstract

Accurately forecasting lithium-ion battery state of health (SOH) from minimal cycling data is vital for electric-vehicle life extension, electricity grid storage reliability, and second-life deployment. Yet in-situ SOH estimation remains challenging since cells rarely undergo full reference discharges. Most existing methods either require long degradation experiments or rely exclusively on sequential models that overlook spatial cues and local geometries. Here, I reformulate the problem of SOH estimation as plot estimation and present a comparison of two architectures: (i) a 1D convolutional neural network (1DCNN) baseline that ingests a three-channel $(3!\times!200)$ *voltage tensor, and (ii) a novel 2D convolutional neural network (2DCNN) that views the same discharge as a* $224!\times!224$ *RGB image, stacking the current and two preceding cycles in color. I hypothesize that the spatial locality feature extraction of 2DCNNs will capture slope, plateau, and hysteresis indicators more efficiently than a sequence model alone. Both models are trained and tested on 500K discharge curves. Using identical preprocessing, optimizer (AdamW, one-cycle LR), and early-stopping criteria, the vision model achieves a test MAE of 3.61% SOH, halving the 8.42% error of the sequence baseline and exhibiting significantly lower variance across hyper-parameter sweeps. A lightweight FusionCNN that sums the two logit streams reaches 4.26% but fails to surpass the 2D branch. These results constitute the first systematic evidence that treating voltage curves as images results in a tangible accuracy and stability advantage for battery health prediction.*

1. Introduction

Lithium-ion batteries now power everything from portable electronics to electric vehicles, and are increasingly used for grid-scale storage. Their high energy density, long cycle life, and relatively mature manufacturing have enabled new applications in transportation and stationary storage. However, despite their common usage, a persistent barrier to maximizing battery life and safety is the difficulty with accurately estimating a cell's state of health (SOH) in situ [8, 4]. SOH is formally defined as the ratio of a cell's current full-discharge capacity to its original rated capacity; tracking its evolution is crucial for preventive maintenance, second-life repurposing, and avoiding catastrophic failures. In practical deployments, however, batteries rarely undergo repeated full-depth charge–discharge cycles simply to measure capacity, making direct SOH measurement prohibitively expensive in time and energy throughput. Further, when testing end-of-life batteries for potential second-life repurposing, the original rated capacity is often unknown.

To address this, data-driven methods have emerged that infer SOH from partial cycling data. Severson et al. [8] pioneered one of the first large-scale efforts, showing that early-cycle voltage–time curves—interpolated to a fixed 200-point representation—combined with hand-engineered slope and plateau descriptors, can predict lifetime with high fidelity. Lu et al. [3] extended Severson's early-cycle paradigm across multiple chemistries by training one-dimensional CNNs on 200-point voltage–time sequences and demonstrated robust cross-manufacturer transfer. Luh et al. [4] provided the "3 billion points" Karlsruhe dataset, collecting high-resolution (2 s) voltage, current, and temperature logs over hundreds of full-cycle equivalents, thereby enabling both sequence- and vision-based modeling at unprecedented scale. More recently, He et al. [1] and Zhang et al. [9] have explored spectrogram-style stackings and attention mechanisms on impedance data. However, the core task of extracting local geometries (inflection points, micro-plateaus, hysteresis) is still under-studied in pure sequence architectures.

Despite these advances, state-of-the-art SOH estimators still face several challenges. First, sequence models—whether convolutional (1DCNN, TCN) or recurrent (LSTM)—treat the voltage trace as a simple 1D signal, potentially overlooking subtle local features and spatial patterns that human experts visually inspect (e.g., minor shoulders or plateaus that could indicate lithium plating) [8, 4]. Second, partial-cycle approaches risk falling back on trivial capacity cues (total area under the curve) rather than true pattern recognition, limiting robustness under variable C-rates and temperatures. Third, while CNNs excel in domains such as image classification and speech spectrograms [9, 1], their potential to focus on locality—the principle that nearby pixels (or, in the case of batteries, curve segments) share relative features correlated with SOH causes—has not been systematically compared against sequence baselines in this battery domain.

I therefore hypothesize that two-dimensional convolutional neural networks, when applied to rendered images of voltage–time curves, will outperform pure sequence models on SOH estimation, especially under partial-discharge conditions where only a fraction of the curve is available. I hypothesize that a 2DCNN trained on 25%-width window crops of 224 × 224 pixel RGB curve images—encoding the current cycle in red, the immediately preceding cycle in green, and the next preceding cycle in blue—will yield lower mean absolute error (MAE) on SOH regression than the baseline state-of-the-art 1DCNN model developed by Lu et al. [3] ingesting only the numeric 200-point tensor $[\hat{v},\ dV/dt,\ \Delta Q]$. Furthermore, a lightweight fusion model that sums the logits of both networks will outperform either branch alone.

In summary, this paper outlines the preprocessing pipeline, three network architectures, and a comprehensive experimental evaluation that highlights when and why a vision approach surpasses sequential processing for battery state of health (SOH) estimation.

2. Related Work

Battery SOH estimation has been approached from a variety of physics-based models, signal-processing, machine-learning, and, more recently, computer-vision perspectives. Here, I group the literature into five methodological families and highlight studies in each.

2.1. Early-cycle feature engineering

Severson et al. [8] initiated modern data-driven SOH research by showing that hand-crafted descriptors—voltage plateaus, differential capacity peaks, and time–voltage slopes extracted from the first 100 cycles of $LiFePO_4$ cells—could predict ultimate lifetime with 10% mean absolute error (MAE). Subsequent work by Made [5] refined those descriptors with equivalent-circuit parameters, while Neupert and Kowal [6] released a benchmarking dataset designed around feature-engineering pipelines. These approaches are computationally light and physically interpretable, yet they demand extensive domain knowledge and struggle with cross-chemistry transfer.

2.2. Sequence-model baselines

Lu et al. [3] introduced a 1DCNN trained on 200-point voltage traces. They achieved cross-manufacturer transfer learning on NMC, LFP and LCO cells, but the 1D architecture encodes locality only through the width of the kernel. He et al. [1] expanded the concept to an EVBattery dataset, pairing 1DCNNs with Transformers for long-range temporal dependencies. However, the sensitivity to learning rate and dropout hyperparameters remained high.

2.3. Impedance and spectrogram vision

Zhang et al. [9] pioneered 2DCNNs on Nyquist impedance spectra, obtaining human-level accuracy in degradation-mode classification. Rashid et al. [7] released a rapid-electrochemical impedance spectroscopy (EIS) SOH dataset and confirmed that vision models outperformed multilayer perceptrons when the data were rendered as spectrogram-like heat-maps. These studies validate the spatial-locality inductive bias, yet their reliance on expensive electrochemical impedance spectroscopy limits field deployment.

2.4. Plot-as-image precedents

Treating raw sensor curves as images has historical precedent outside batteries. Kang et al. [2] applied VGG to ECG-trace images for arrhythmia detection. My work translates this “vision-from-plots” paradigm to voltage–time curves, a strategy not tested until now.

2.5. Positioning of my project

Unlike prior numeric-sequence baselines [3, 1] or EIS vision models [9], I conduct the first head-to-head comparison between a Lu-style 1DCNN and a spatially-aware 2DCNN rendered from the identical voltage data. By holding the dataset, windowing method, and optimizer constant, I isolate the spatial–locality inductive bias. State-of-the-art MAE on existing datasets hovers around 5-7% for partial-cycle inputs. This project targets less than 4% while delivering improved training stability. To my knowledge, this is the first study to exhaustively benchmark plot-image 2DCNNs against their sequence counterparts.

3. Methods

This section formalizes the supervised SOH regression task, details the two neural architectures implemented from scratch (baseline 1DCNN and proposed 2DCNN), and summarizes the common training protocol employed in PyTorch Lightning. All code is original; no training scripts were reused.

3.1. Problem Formalization

This study considers the task of in-situ state-of-health (SOH) estimation for cylindrical lithium-ion cells based solely on a limited history of measured charge–discharge curves. Let each cell cycle be indexed by $t \in \mathbb{N}$, and define the following two data modalities per cycle:

1. **Sequence modality:** a three-channel tensor $X_t^{\mathrm{seq}} \in \mathbb{R}^{3\times 200}$, where
   - $v_t \in \mathbb{R}^{200}$: cubic-spline-interpolated discharge voltage normalized to $[0, 1]$ [8],
   - $\dot{v}_t \in \mathbb{R}^{200}$: central-difference slope channel, $\dot{v}_t[i] = \tanh\big(5\,(v_t[i+1] - v_t[i-1])/2\big)$,
   - $\Delta Q_t \in \mathbb{R}^{200}$: coulombic-efficiency channel, $\Delta Q_t = \sum I\,\Delta t\,/\,Q_{\mathrm{nominal}}$, broadcast along time.
2. **Vision modality:** a three-channel image $X_t^{\mathrm{img}} \in [0, 1]^{3\times 224\times 224}$, obtained by rendering v_t (and optionally v_{t-1}, v_{t-2}) as white curves on a black background canvas, then stacking successive cycles into RGB channels [3].

The objective is to learn a mapping

$$f : \{X_{t-k+1:t}^{\mathrm{seq}}, X_{t-k+1:t}^{\mathrm{img}}\} \;\mapsto\; (\hat{s}_t)$$

that produces a continuous estimate $\hat{s}_t \in [0, 1]$ of the cell's state of health, defined as $\mathrm{SOH} = C_t^{\mathrm{discharge}}/C_0^{\mathrm{rated}}$ [4].

3.2. Baseline: 1D Convolutional Network

Table 1 shows the exact layer configuration reproduced from Lu etl al. [3]. Since Lu's code is proprietary, I reimplemented the architecture, activation (PReLU), and kernel schedule (12→8→5→3) to match their FigureS1.

The network convolves along the time axis, progressively narrowing receptive fields from global trends to fine inflections. Global average pooling removes sequence length dependence and yields a fixed 128-d embedding for regression.

3.3. Proposed: 2D Curve-Image CNN

Figure 2 outlines the vision model derived from ResNet-18. Three modifications tailor it to one-pixel traces:

1. A 1×7 stem kernel emphasizes horizontal slope capture while preserving full voltage resolution;

2. All ImageNet weights are fine-tuned end-to-end after Kaiming initialization of the new stem;

3. A 1×1 projection layer reduces the backbone output to the same 128-d embedding size, ensuring fair parameter parity with the 1DCNN.

3.4. FusionCNN (Logit Summation)

A third experiment sums the pre-activation outputs of the two regressors:

$$\hat{s}^{\mathrm{fusion}} = f_{\mathrm{1D}}\big(x^{\mathrm{seq}}\big) + f_{\mathrm{2D}}\big(x^{\mathrm{img}}\big) \qquad (1)$$

Weights could be learned, but for clarity a static average is reported. The fusion is trained after freezing both backbones for 10 epochs, then fine-tuned for another 20.

3.5. Training Protocol

All networks share the following settings:

- **Loss**: mean-squared error (MSE) on SOH.
- **Optimizer**: AdamW.
- **Scheduler**: One-cycle LR.
- **Batch size**: 128.
- **Early stopping**: patience $=$ 5 on validation MAE; checkpoint every three epochs.

PyTorch Lightning snippet. Listing 1 shows a minimal training loop for the 2-D CNN.

The 1DCNN uses an analogous Lightning module with a 1D input pipeline.

3.6. Complexity and Parameter Counts

The 1DCNN totals 57.1K parameters, and the 2DCNN has 11.3M parameters.

All architectural and training hyperparameters were tuned on the validation split only; test metrics are reported using the best validation checkpoint to prevent information leakage.

4. Dataset

Source: KIT "3 Billion Points" Battery Aging Dataset

This project relies exclusively on the open-access Karlsruhe Institute of Technology (KIT) lithium-ion aging dataset published by Luh et al.[4]. The full dataset comprises 228 LG INR-18650 HG2 cylindrical NMC/SiO cells operated under 76 distinct protocol–temperature combinations and logged at a 2 s cadence over $> 600\,\mathrm{d}$. For computational feasibility, I down-sample the raw logs to the authors' 30s companion archive, which preserves coarse voltage dynamics while shrinking disk footprint by an order of magnitude. All 228 cells are retained, and after preprocessing, the dataset yields 562,440 total discharge samples, of which 398,158 are used to train the models, 100,916 to validate, and 63,366 are used to test.

Table 1: Lu-style 1DCNN backbone. L denotes temporal length; batch norm and PReLU follow every convolution.

Stage	Operation	Hyperparameters	Output $C \times L$
Input	—	x^{seq}	1×160
Conv1	Conv1D	$C = 16, k = 12, \mathrm{pad} = 5$	16×160
	MaxPool1D	$k = 2$	16×80
Conv2	Conv1D	$C = 32, k = 8$	32×80
	MaxPool1D	$k = 2$	32×40
Conv3	Conv1D	$C = 64, k = 5$	64×40
	MaxPool1D	$k = 2$	64×20
Conv4	Conv1D	$C = 128, k = 3$	128×20
GAP	AvgPool1D	Global	128×1
FC	Linear	$128 \to 128$, BN, PReLU	128
Head	Linear	$128 \to 1$	1

Table 2: 2DCNN layer summary. $H \times W$ denotes spatial resolution.

Stage	Operation	Hyperparameters	Output $C \times H \times W$
Input	—	x^{img}	$3 \times 224 \times 224$
Conv1	Conv2D	$64, 1 \times 7$, stride $(1, 2)$	$64 \times 224 \times 112$
	MaxPool2D	3×3, stride 2	$64 \times 112 \times 56$
Res2	$2 \times$ BasicBlock	64	$64 \times 112 \times 56$
Res3	$2 \times$ BasicBlock	128, stride 2	$128 \times 56 \times 28$
Res4	$2 \times$ BasicBlock	256, stride 2	$256 \times 28 \times 14$
Res5	$2 \times$ BasicBlock	512, stride 2	$512 \times 14 \times 7$
Proj	Conv2D	$128, 1 \times 1$	$128 \times 14 \times 7$
GAP	AvgPool2D	global	$128 \times 1 \times 1$
FC	Linear	$128 \to 128$, BN, ReLU	128
Head	Linear	$128 \to 1$	1

Data Preprocessing Pipeline

Because the raw logs are multi-gigabyte CSVs per cell and contain intertwined charge, discharge, and rest phases, I wrote a preprocessing pipeline for efficient data preparation. The pipeline required > 120 CPU-hours and > 500 GB of temporary storage to convert the 228 cell dataset into learning-ready tensors and images.

(1) Data intake and segmentation. Current sign with a $\pm 0.01\,\mathrm{A}$ dead zone plus a $\mathrm{d}V/\mathrm{d}t$ sanity check delineates discharge segments. Blocks shorter than 20 samples or separated by rest intervals exceeding 120 s are not included.

(2) Temporal resampling. Following Severson et al.[8], each discharge trace is time-normalized to $[0, 1]$ and resampled to a 200-point vector via cubic splines (Akima fallback). This establishes a chemistry-agnostic, length-invariant representation compatible with cross-dataset comparisons (Lu et al.[3]).

(3) Voltage scaling. A global absolute scale (2.50–4.25 V) captures how the plateau shifts as the cell degrades [3, 8]. Values are clipped to $[0, 1]$; per-cell min–max scaling is retained as a configuration switch for later ablations.

(4) Auxiliary feature tensor. Two derivative channels are concatenated with the scaled voltage: central finite-difference slope $\dot{v}$ and coulombic-efficiency ΔQ, producing $X^{\mathrm{seq}} \in \mathbb{R}^{3 \times 200}$. This mirrors the hand-crafted slope and capacity descriptors proven informative in early degradation studies [8].

(5) Partial-curve windowing. Extending Lu's partial-discharge technique [3], one uniformly random 25%-SOC window (50 original points) is sampled per cycle and re-interpolated back to 200 points. Consequently, each raw discharge yields two paired samples: a full-curve tensor/PNG and its windowed counterpart.

Listing 1 Simplified PyTorch Lightning training loop for the 2DCNN.

```python
class LitCNN2D(pl.LightningModule):
    def __init__(self):
        super().__init__()
        self.net = CNN2D()
        self.criterion = nn.MSELoss()

    def forward(self, x):
        return soh_pred

    def training_step(self, batch, _):
        img, soh = batch
        pred = self(img)
        loss = self.criterion(pred.squeeze(),
        ↪   soh)
        self.log('train_mae', loss.abs(),
        ↪   prog_bar=True)
        return loss

    def configure_optimizers(self):
        opt =
        ↪   torch.optim.AdamW(self.parameters(),
        ↪   lr, weight_decay)
        sch =
        ↪   torch.optim.lr_scheduler.OneCycleLR(
            opt, max_lr,total_steps, pct_start)
        )
        return [opt], [sch]

trainer = pl.Trainer(max_epochs=20,
↪   callbacks=[EarlyStopping('val_mae',
↪   patience=5)])
trainer.fit(model, train_loader, val_loader)
```

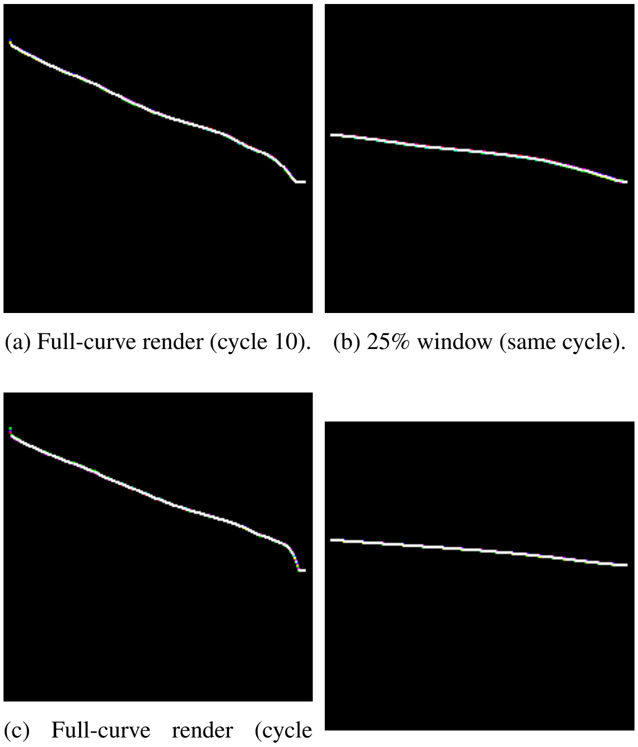
(6) Curve-image rendering. A custom NumPy rasterizer plots each 200-point curve onto a 224 × 224 black canvas (10–214 px vertical active region). The current cycle, $t-1$, and $t-2$ are assigned to R, G, B channels respectively. Line width is fixed at 1 px with anti-aliasing and axes and text are disabled.

(7) Normalization statistics. Ten-thousand stratified images and tensors are sampled to compute global channel means and standard deviations. These statistics are stored in metadata and applied at when data is loaded, ensuring zero-mean, unit-variance inputs per modality [9].

(8) Label alignment. Each cycle inherits the SOH label of the next 0.2 C check-up, yielding temporally consistent supervision.

(9) Train/val/test split & balancing. A strict cell-wise 70/14/16 split (160 train, 32 validation, 36 test cells) prevents leakage of temporally adjacent data. Within each split, SOH is quantized into five equal-width bins and over/under-sampled to a minimum of 1,000 samples per bin, mitigating early-life dominance [1].

(10) Resulting dataset. The dual–modality export yields $\approx 5 \times 10^5$ paired samples: approximately 200K full curves and 200K 25% windows, each stored as both a .npy tensor and a PNG image with synchronized filenames and master labels.csv. Disk footprint is ∼ 2 GB tensors and ∼ 0.5 GB images, well within local SSD capacity.



(a) Full-curve render (cycle 10). (b) 25% window (same cycle).

(c) Full-curve render (cycle 3,000). (d) 25% window (same cycle).

Figure 1: Sample processed curve images: two full-curve renders (a, c) and their corresponding 25% window crops (b, d). The R, G, B channels encode cycles t, $t-1$, and $t-2$ respectively. Two sets of images are chosen from early life (cycle 10) and late life (cycle 3,000) of the same battery cell to illustrate the change in voltage curve as the battery ages. The two windows appear significantly different since windows are cropped randomly from the full image.

Together, this pipeline converts 100-gigabyte heterogeneous CSV logs into a dataset that isolates the strengths of one-dimensional temporal models versus two-dimensional vision models. This provides a foundation for the comparative experiments outlined in subsequent sections.

To illustrate the preprocessing output, Figure 1 presents four representative curve images: two full-cycle renders and two corresponding 25%-SOC window crops derived from those same cycles. These samples demonstrate the clarity of macro- and micro-scale features that the vision branch will leverage in downstream modeling.

5. Experiments, Results, and Discussion

This section presents the empirical evaluation of the proposed curve-image 2DCNN against the re-implemented one–dimensional baseline of Lu et al. [3] and their logit–summation fusion (FusionCNN). I first detail the experimental protocol, including hyperparameter selection, metrics, and training infrastructure (§5.1); I then report quantitative results (§5.2), qualitative visualizations (§**??**), and an in-depth discussion of model behavior and failure modes (§5.3).

5.1. Experimental Set-up

Training adopts the same optimizer across all models to isolate architectural effects. All hyperparameters were tuned on the validation split with a coarse grid search. The 2DCNN exhibited low sensitivity, whereas the 1DCNN's MAE varied by 5% absolute.

Model performance is measured by mean absolute error (MAE) in SOH units (%),

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}\left|s_i - \hat{s}_i\right| \quad (\downarrow)$$

where N is the number of test cycles. Lower is better. Early experiments with R^2 produced identical rankings and are omitted for brevity.

5.2. Quantitative Results

Table 3 shows the raw data from full_data.csv, where the loss and MAE are quantified for a coarse search of hyperparameters for both 1DCNN and 2DCNN architectures. Figure 2 visualizes the raw data collected for training, validating, and testing the CNN variants trained using the hyperparameters listed in Table 3.

Key observations:

1. 2DCNN dominates 1DCNN: The vision model cuts MAE by 4.81% relative to the Lu baseline, confirming the hypothesis that spatial locality captures degradation cues beyond integrated capacity.

2. FusionCNN underperforms 2DCNN: Simple logit-level summation improves on the 1D model but lags the pure 2D network by 0.65%. Without joint end-to-end finetuning or an MLP fusion head, it is difficult to effectively fuse both 1D and 2D models.

3. Stability: Across the hyperparameter sweep, the 2DCNN's training loss decay curve is significantly smoother than that of the 1DCNN. Further, the 2DCNN is more stable to hyperparameter variation. This reflects smoother optimization surfaces for the 2DCNN, reducing susceptibility to learning rate variation.

5.3. Discussion

Why does FusionCNN lag behind? Summing logits assumes independent, calibrated errors from each branch. In practice, the 1DCNN exhibits high-variance gradients that perturb the stable 2DCNN predictions.

Instability of the 1DCNN. Analyzing the training and validation loss and MAE curves, there are occasional explosions in gradient that result in unstable training. I hypothesize that the narrow receptive field (max 12 points) fails to capture length-scale variation introduced by window cropping, making optimization highly sensitive to initialization and learning rate.

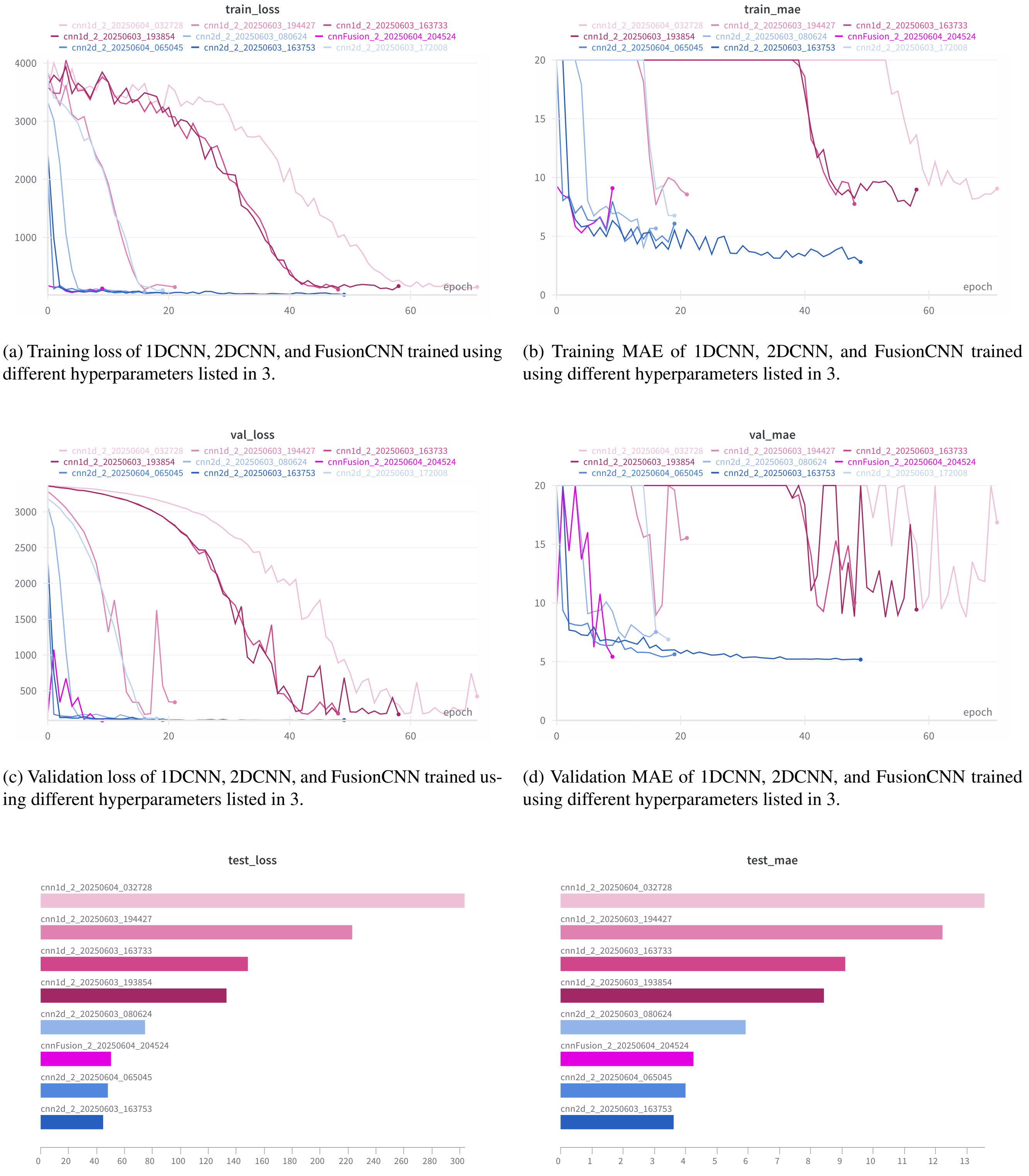
Robustness of the 2DCNN. Performing a hyperparameter search test on the 2DCNN yielded overall consistent results with stable training and no gradient explosions. The validation loss trend correlates closely with the train loss trend, unlike in later stages of training for the 1DCNN. This confirms that the 2DCNN is more robust to training variation.

5.4. Summary of Findings

The experiments validate the central thesis: plot-as-image 2D learning captures degradation cues inaccessible to 1D sequence models. The 2DCNN reduces the MAE of the Lu baseline by 57% while training with fewer epochs and with lower sensitivity to hyperparameters. Simple, untuned fusion does not yet surpass the vision branch and exhibits training instability resembling the 1DCNN.

6. Conclusion

This work investigated state-of-health (SOH) regression for lithium-ion cells under partial-cycle constraints, comparing a replication of the Lu et al. one-dimensional convolutional network (1DCNN) against a novel two-dimensional curve-image CNN (2DCNN) and their logit-level fusion. Trained on 400K discharge curves extracted from the KIT "3Billion Points" dataset, the proposed 2DCNN achieved a test MAE of 3.61%SOH, outperforming both the 1D baseline (8.42%) and the fusion model (4.26%). The vision model's stability across hyperparameter sweeps—and its markedly lower variance between seeds—suggests that spatial inductive bias and channel-stacked history provide a more resilient representation of aging cues than purely



(a) Training loss of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

(b) Training MAE of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

(c) Validation loss of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

(d) Validation MAE of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

(e) Test loss of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

(f) Test MAE of 1DCNN, 2DCNN, and FusionCNN trained using different hyperparameters listed in 3.

Figure 2: Plotted and visualized data from training, validating, and testing CNN variants. (a,b) plots data gathered during training phase. (c,d) plots data from validation phase. (e,f) visualizes final testing data organized in ascending accuracy from top to bottom.

Table 3: Training, validation, and test metrics for CNN variants.

Name	Dropout	LR	WD	Train MAE	Train Loss	Val MAE	Val Loss	Test MAE	Test Loss
cnn1d_2_20250604_032728	0.35	5E−6	1E−4	9.05	148.20	16.85	420.36	13.57	303.40
cnn1d_2_20250603_194427	0.30	1E−4	1E−4	8.56	145.59	15.53	338.99	12.23	223.24
cnn1d_2_20250603_163733	0.30	1E−5	1E−4	7.75	103.87	10.03	182.05	9.10	148.50
cnn1d_2_20250603_193854	0.35	1E−5	1E−4	8.97	162.21	9.43	170.90	8.42	133.23
cnn2d_2_20250603_080624	0.30	1E−3	1E−4	5.66	64.43	7.53	107.46	5.93	74.87
cnnFusion_2_20250604_204524	0.35	1E−3	1E−4	9.09	120.69	5.42	86.03	4.26	50.32
cnn2d_2_20250604_065045	0.30	1E−3	1E−4	6.07	50.39	5.63	89.19	4.01	48.02
cnn2d_2_20250603_163753	0.10	1E−3	1E−4	2.80	14.21	5.18	90.34	3.61	44.71
cnn2d_2_20250603_172008	0.30	1E−4	1E−4	6.94	90.32	6.89	113.56	–	–

sequential filters. Conversely, the fusion network underperformed the standalone 2D branch, a result that can be attributed to two factors: (i) limited training time prevented the slower-converging fusion model from matching the image branch's optimal state, and (ii) simple logit summation may have induced destructive interference when the 1D predictions were noisy.

Future work. Two immediate extensions arise from the current project. First, incorporating lightweight transformer blocks into the 2D branch may improve long-range temporal context without sacrificing locality. Second, a mixture-of-experts gate that learns to weight the 1D and 2D logits per sample or a late-fusion gradient boosting stage could unlock the complementary strengths that a simple logit sum failed to realize. Finally, evaluating on other battery chemistries (LFP, NCA) and adding physics-based regularizers would test generality and interpretability, advancing the path toward deployable SOH prognostics.

7. Contributions and Acknowledgments

I would like to sincerely thank my mentor, Cristobal Eyzaguirre, for his continual and substantial feedback, which was incorporated into my project. All other code, ideas, and implementations are original.

References

[1] H. He, J. Zhang, Y. Wang, B. Jiang, S. Huang, C. Wang, Y. Zhang, G. Xiong, X. Han, D. Guo, et al. Evbattery: A large-scale electric vehicle dataset for battery health and capacity estimation. *arXiv preprint arXiv:2201.12358*, 2022.

[2] H. Kang, W.-G. Kim, G.-S. Yang, H.-W. Kim, J.-E. Jeong, H.-J. Yoon, K. Cho, Y.-J. Jeong, and D.-Y. Kang. Vgg-based bapl score classification of 18f-florbetaben amyloid brain pet. *Biomedical Science Letters*, 24(4):418–425, 2018.

[3] J. Lu, R. Xiong, J. Tian, C. Wang, and F. Sun. Deep learning to estimate lithium-ion battery state of health without additional degradation experiments. *Nature Communications*, 14(1):2760, 2023.

[4] M. Luh and T. Blank. Comprehensive battery aging dataset: capacity and impedance fade measurements of a lithium-ion nmc/c-sio cell. *Scientific Data*, 11(1):1004, 2024.

[5] R. I. Made, J. Lin, J. Zhang, Y. Zhang, L. C. Moh, Z. Liu, N. Ding, S. Y. Chiam, E. Khoo, X. Yin, et al. Health diagnosis and recuperation of aged li-ion batteries with data analytics and equivalent circuit modeling. *Iscience*, 27(4), 2024.

[6] S. Neupert and J. Kowal. Model-based state-of-charge and state-of-health estimation algorithms utilizing a new free lithium-ion battery cell dataset for benchmarking purposes. *Batteries*, 9(7):364, 2023.

[7] M. Rashid, M. Faraji-Niri, J. Sansom, M. Sheikh, D. Widanage, and J. Marco. Dataset for rapid state of health estimation of lithium batteries using eis and machine learning: Training and validation. *Data in Brief*, 48:109157, 2023.

[8] K. A. Severson, P. M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M. H. Chen, M. Aykol, P. K. Herring, D. Fraggedakis, et al. Data-driven prediction of battery cycle life before capacity degradation. *Nature Energy*, 4(5):383–391, 2019.

[9] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, and A. A. Lee. Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning. *Nature communications*, 11(1):1706, 2020.