HQ-VQVAE for Unsupervised Battery Precursor Discovery

Gpt 5.2, Gemini 3.0, EXAONE 3.5 *

Anonymous Author(s)

Abstract

Thermal runaway in lithium-ion batteries poses severe safety risks for electric vehicles and energy storage systems, yet early degradation precursors remain difficult to detect due to weak signals and limited failure labels. In this work, we propose a Hybrid Quantum-Classical Vector-Quantized Variational Autoencoder (HQ-VQVAE) for the unsupervised discovery of degradation patterns from battery cycling data. Our model integrates a parameterized quantum circuit (PQC) into the latent bottleneck to promote more expressive representations in a high-dimensional Hilbert space while preserving the discrete interpretability of vector quantization. Without relying on thermal runaway labels, our AI-driven analysis identifies a specific discrete latent code that is repeatedly associated with early-stage degradation signatures, including patterns consistent with voltage drop dynamics and increasing internal resistance behavior. We validate the robustness of this discovered code on the NASA PCoE dataset under cross-cell evaluation, demonstrating stable generalization to unseen batteries and improved sensitivity to subtle early degradation dynamics. These results suggest that hybrid quantum-classical representation learning can support interpretable, label-efficient battery health monitoring and enable earlier intervention for safety-critical applications.

1 Introduction

Thermal runaway in lithium-ion batteries is a low-frequency but high-impact failure mode that can lead to catastrophic safety incidents in electric vehicles and energy storage systems. Preventing such events requires detecting early degradation signatures before irreversible escalation. However, reliable early-warning remains challenging in practice because precursor signals are often weak, highly variable across cells, and rarely accompanied by explicit failure labels. As a result, many deployed monitoring systems rely on coarse thresholds or supervised predictors trained on limited failure cases, which can be brittle when transferred to unseen batteries or operating conditions.

To address these limitations, we explore **unsupervised discovery of degradation patterns** directly from battery cycling time-series. An ideal framework should satisfy two properties simultaneously: (1) **label efficiency**, since thermal runaway or near-failure data are scarce and expensive to obtain, and (2) **interpretability**, since actionable decisions require understanding *what* pattern is detected and *why* it signals risk. Representation learning with autoencoders can compress cycling behaviors into latent factors, but standard continuous latents are often difficult to interpret and may entangle multiple degradation modes.

In this work, we propose a **Hybrid Quantum-Classical Vector-Quantized Variational Autoencoder (HQ-VQVAE)** that learns discrete, interpretable degradation codes from cycling data while

*Open model details: `https://huggingface.co/LGAI-EXAONE`.

improving separability between subtle degradation dynamics. The model is based on a Vector-Quantized Variational Autoencoder (VQ-VAE) (van den Oord et al., 2017), which provides a discrete codebook representation that can act as a compact set of "pattern prototypes." We further integrate a **Parameterized Quantum Circuit (PQC)** into the latent bottleneck to enhance feature discrimination by mapping intermediate representations into a high-dimensional Hilbert space, while retaining the discrete interpretability enabled by vector quantization.

Using this framework, we perform an AI-driven analysis that discovers a **specific discrete latent code** which consistently appears prior to degradation-related behaviors such as **voltage sag and internal resistance growth**, *without relying on thermal runaway labels*. We validate the robustness of this discovered code under **cross-cell evaluation** on the **NASA PCoE dataset** (Saha and Goebel, 2007; Goebel et al., 2008).

Contributions. Our contributions are:

1. HQ-VQVAE: a hybrid quantum-classical representation learning framework that preserves discrete interpretability while improving separability of subtle degradation patterns.
2. Unsupervised discovery of a stable precursor code associated with degradation-related dynamics in cycling data.
3. Cross-cell validation showing consistent generalization to unseen cells, supporting label-efficient and interpretable monitoring for safety-critical battery applications.

2 Related Work

2.1 Battery degradation monitoring and early-warning signals

Lithium-ion battery degradation manifests through subtle shifts in voltage response, capacity fade, and internal resistance growth. Early-stage precursors of safety-critical events are particularly challenging to detect because they often appear as weak, transient patterns embedded in noisy cycling data. Many monitoring systems rely on handcrafted features and threshold-based rules, which can be brittle across different cells and operating regimes.

2.2 Unsupervised representation learning for battery health

To address limited labels, unsupervised and self-supervised learning methods have been explored to capture latent representations of battery health trajectories from raw or lightly processed sensor streams. Autoencoder-based approaches are attractive because they learn compact representations directly from cycling sequences, enabling anomaly detection and pattern mining without explicit failure supervision. However, purely continuous latent representations can be difficult to interpret.

2.3 Discrete latent modeling with vector quantization

VQ-VAEs provide a discrete bottleneck via a codebook, enabling learned representations to be expressed as interpretable "codes" (van den Oord et al., 2017). This discretization can support mechanistic hypothesis generation because recurring codes often correspond to repeated morphological patterns in time-series behavior.

2.4 Hybrid quantum-classical learning for separable latent spaces

Hybrid quantum-classical models integrate parameterized quantum circuits (PQCs) with neural networks, aiming to exploit high-dimensional Hilbert-space embeddings. In our work, we place a PQC module inside the latent bottleneck of a VQ-VAE to enhance separability among subtle degradation dynamics while retaining discrete interpretability. The implementation uses PennyLane (Bergholm et al., 2018; PennyLane Developers, 2026).

Positioning. Unlike supervised thermal runaway prediction methods that require failure labels, we focus on **unsupervised discovery** of reproducible precursor patterns and validate robustness under cross-cell evaluation.

3 Method

3.1 Problem setup

We aim to discover early degradation precursors from lithium-ion battery cycling data without relying on thermal runaway or failure labels. Given a sequence of discharge trajectories and summary statistics, the objective is to learn an interpretable representation that (1) reconstructs observed behavior and (2) yields discrete latent codes trackable over time. A precursor is defined as a latent code that appears consistently before measurable degradation symptoms such as voltage sag or internal resistance growth proxies.

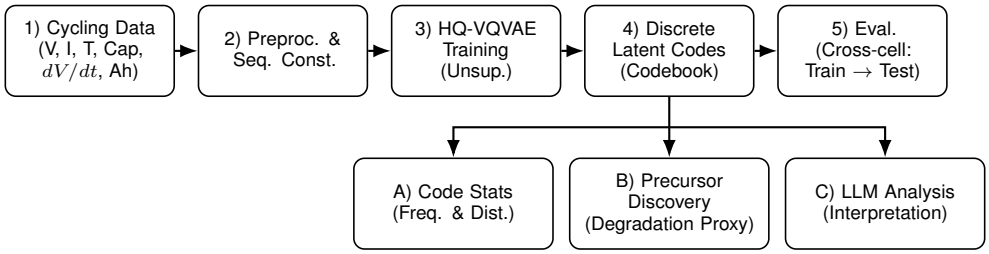

Figure 1: Compact pipeline diagram for unsupervised precursor discovery.

3.2 Input representation and feature construction

For each discharge segment, we form an input vector that summarizes operational condition and curve morphology. The input includes voltage statistics, current and temperature summaries, and slope-related descriptors to capture weak shifts in curve shape.

Table 1: Feature list used in the model.

Feature	Description	Unit/Type	Why it matters
t_{total}	Total discharge duration	s	Time-domain summary
$v_{\min}$	Min voltage during discharge	V	Voltage statistics
$v_{\max}$	Max voltage during discharge	V	Voltage statistics
v_{mean}	Mean voltage during discharge	V	Voltage statistics
i_{mean}	Mean current during discharge	A	Current statistics
$temp_{\text{mean}}$	Mean temperature	°C	Thermal condition
$(dV/dt)_{\text{mean}}$	Mean voltage slope	V/s	Degradation-sensitive dynamics
$(dV/dt)_{\min}$	Min voltage slope	V/s	Early anomaly indicator
$(Ah)_{\text{throughput}}$	Acc. charge throughput	Ah	Usage / aging proxy
Capacity	Estimated capacity	Ah	Health indicator
$\text{Cycle}_{\text{index}}$	Cycle index	Integer	Aging timeline proxy

3.3 HQ-VQVAE architecture

Our Hybrid Quantum-Classical Vector-Quantized Variational Autoencoder (HQ-VQVAE) consists of a classical encoder/decoder with a discrete bottleneck, enhanced by a parameterized quantum circuit (PQC) in the latent pathway.

(1) Classical encoder. The encoder maps input x to a continuous embedding:

$$z_e = \text{Enc}(x). \tag{1}$$

(2) Quantum latent transformation (PQC module). We project z_e to quantum parameters (e.g., rotation angles), apply a PQC, and read out expectation values:

$$z_q = \text{Proj}\big(\text{PQC}(\text{AngleEnc}(z_e))\big). \tag{2}$$

(3) Vector quantization (discrete code assignment). We quantize the transformed latent representation into a discrete code from a learnable codebook

$$k^* = \arg\min_{k \in \{1,\dots,K\}} \|z_q - e_k\|_2, \qquad z_{\mathrm{vq}} = e_{k^*}. \tag{3}$$

The resulting code index k^* is the interpretable "degradation pattern code" used for tracking and discovery.

(4) Decoder reconstruction. The decoder reconstructs the input features:

$$\hat{x} = \mathrm{Dec}(z_{\mathrm{vq}}). \tag{4}$$

3.4 Training objective

We train HQ-VQVAE with reconstruction loss and standard VQ losses:

$$\mathcal{L} = \|x - \hat{x}\|_2^2 + \|\mathrm{sg}[z_q] - e_{k^*}\|_2^2 + \beta \|z_q - \mathrm{sg}[e_{k^*}]\|_2^2, \tag{5}$$

where $\mathrm{sg}[\cdot]$ denotes stop-gradient and β controls commitment strength.

Table 2: Generalization summary on unseen cells (Total $N = 18$ cells; all listed codes are classified as meaningful).

Code ID	# Present	# Early	Pearson (r)	Spearman (ρ)
10	10	10	0.49	0.65
28	11	7	0.53	0.52
23	13	10	0.46	0.50
6	8	8	0.28	0.44
26	8	7	0.33	0.38
20	7	6	-0.10	0.34
22	6	6	0.28	0.32
15	15	11	0.06	0.28
3	16	7	0.32	0.26
0	13	9	0.14	0.25

Note : Table 4 reports the split protocol over 20 cells, while the quantitative results in Table 2 are computed on 18 cells after applying a data-quality filter.

3.5 Precursor code discovery protocol

After training, we perform unsupervised code mining:

1. **Code extraction:** assign k^* for each discharge segment.
2. **Temporal aggregation:** compute code frequency/persistence over cycles.
3. **Symptom association:** identify codes preceding degradation proxies (e.g., voltage sag descriptors, resistance-related proxies).
4. **Candidate selection:** choose code(s) reproducible across cells and temporally consistent.

3.6 Cross-cell evaluation for robustness

We adopt cross-cell splits: train HQ-VQVAE on a subset of cells and test on unseen cells. Robustness is assessed by whether the same code remains an early indicator across batteries.

4 Experiments

4.1 Dataset

We conduct experiments on a public lithium-ion battery cycling dataset provided by NASA's Prognostics Center of Excellence (PCoE) (Saha and Goebel, 2007; Goebel et al., 2008). The dataset contains multi-cycle charge/discharge trajectories across multiple cells, including time-series measurements such as voltage, current, and temperature.

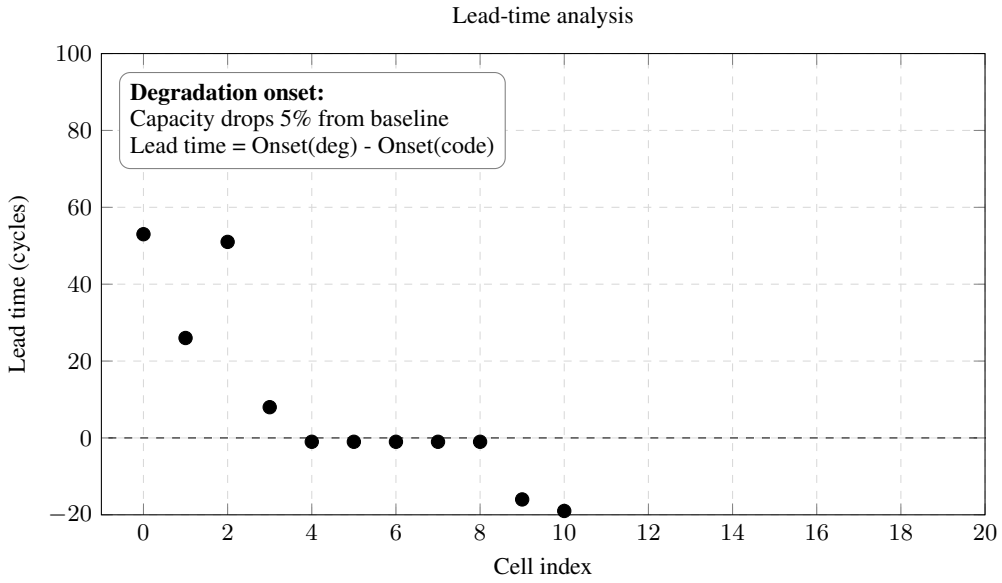

Figure 2: Lead-time between first activation of the precursor code and degradation onset.

4.2 Preprocessing and feature extraction

Rather than modeling raw time-series directly, each discharge segment is converted into a compact feature vector capturing operating conditions and curve morphology (Table 1). Features are normalized using statistics computed only from the training split to prevent leakage.

Table 3: Final input feature set used for HQ-VQVAE training.

Feature	Symbol	Category	Used	Description
`t_total`	t_{total}	Time	Yes	Total discharge time per cycle
`v_min`	v_{min}	Voltage	Yes	Minimum voltage during discharge
`v_max`	v_{max}	Voltage	Yes	Maximum voltage during discharge
`v_mean`	v_{mean}	Voltage	Yes	Mean voltage during discharge
`i_mean`	i_{mean}	Current	Yes	Mean current during discharge
`temp_mean`	T_{mean}	Temp.	Yes	Mean temperature during discharge
`dv_dt_mean`	$(dV/dt)_{mean}$	Derived	Yes	Mean voltage slope
`dv_dt_min`	$(dV/dt)_{min}$	Derived	Yes	Minimum voltage slope
`ah_throughput`	Ah_{cum}	Usage	Yes	Cumulative Ah throughput
`capacity`	Q_{dis}	Capacity	Yes	Discharge capacity per cycle

4.3 Experimental protocol

Our primary goal is unsupervised discovery of precursor patterns that generalize across unseen cells. Therefore, we adopt a cross-cell evaluation protocol, where batteries (cells) are split into disjoint training and test groups. The HQ-VQVAE is trained only on the training batteries. After training, we extract discrete code assignments for both training and test batteries and analyze which code(s) exhibit consistent early appearance before degradation intensifies.

To quantify precursor behavior without relying on thermal runaway labels, we define degradation progression using proxy indicators available from cycling data (e.g., capacity decline trends, voltage sag descriptors, or remaining useful life–related indices if provided). We then measure whether the discovered precursor code appears systematically earlier than these degradation manifestations.

Table 4: Train/test split details (cells and samples).

Split	Cells (#)	Segments (#)	Notes
Train	19	mean 1424	1331-1474 range
Test	1	mean 75	25-168 range
Total	20	1499	

4.4 Baselines

We compare HQ-VQVAE against:

1. Classical VQ-VAE (no PQC)
2. Continuous AE/VAE baseline (with optional post-hoc clustering)
3. PCA + K-means
4. Random code assignment (sanity check)

Table 5: Baseline methods and key configuration differences.

Method	Core idea	Key difference	Role
HQ-VQVAE (ours)	PQC + VQ codebook	Discrete codes + quantum bottleneck	Main method
VQ-VAE (no PQC)	No PQC, same VQ	Discrete codes only	Ablation
AE/VAE (continuous)	Continuous latent	No discrete codes	Baseline
PCA + K-means	Linear + clustering	Handcrafted pipeline	Baseline
Random codes	Random assignment	Sanity check	Control

4.5 Evaluation metrics

We evaluate (1) representation quality and (2) precursor discovery utility. Representation quality uses reconstruction error (MSE), codebook utilization, and code stability. Precursor utility uses precursor consistency across cells, lead-time analysis, and sensitivity to early-stage changes.

4.6 Implementation details

Neural components are implemented in PyTorch; PQC modules use PennyLane (Bergholm et al., 2018; PennyLane Developers, 2026). We fix random seeds and (optionally) report averages over 3 independent runs.

5 LLM-assisted interpretation

HQ-VQVAE yields discrete latent codes that summarize recurring patterns, but mapping each code to a physical mechanism can be ambiguous. We therefore use a large language model (LLM) only for **post-hoc interpretation** of code-conditioned summaries, not for training supervision.

5.1 Role of the LLM

Given code-level statistics and representative curve morphology descriptors, the LLM generates (i) plausible electrochemical hypotheses, (ii) morphology descriptors, (iii) expected observable symptoms, and (iv) suggested follow-up tests. The LLM does not influence code assignment or optimization.

5.2 Prompt design and structured outputs

We enforce a strict JSON schema per code: To minimize ambiguity and ensure consistent downstream analysis, we enforce a strict JSON output schema. For each code, the LLM receives a compact

summary of representative feature statistics (e.g., mean/min/max voltage, average dV/dt, temperature statistics, capacity-related proxies) and a short description of the most salient discharge-shape patterns. The model is instructed to return a JSON object containing:

Table 6: JSON schema for LLM-assisted interpretation.

Field	Meaning
`short_name`	short descriptive code name
`mechanism_hypotheses`	candidate mechanisms + reasoning
`curve_morphology`	qualitative shape descriptors
`expected_symptoms`	measurable indicators
`suggested_followup_tests`	diagnostic experiments
`confidence`	low/medium/high

This structured format enables automated parsing and aggregation across all codes.

5.3 Model choice and reliability controls

We use EXAONE Instruct-family models for deterministic decoding (greedy) and validate JSON parseability. Outputs are treated as hypotheses and paired with suggested objective follow-up tests (e.g., impedance analysis). We log prompts, model version, decoding settings, and timestamps to support reproducibility (LG AI Research, 2024).

6 Results

We report (i) stability and interpretability of discrete codes, (ii) discovery of a consistent early-stage precursor pattern without failure labels, and (iii) cross-cell generalization.

6.1 Discrete code emergence and interpretability

Our HQ-VQVAE discovers a compact set of discrete latent codes that summarize recurring discharge behaviors across cells and cycling stages. Across the dataset, each code corresponds to a distinct discharge curve morphology and is associated with characteristic ranges of voltage statistics (e.g., mean/min voltage), slope-related features (e.g., mean dV/dt), and capacity-related proxies.

We first examine the distribution of code assignments over the full cycling span. The code histogram indicates that certain codes dominate stable operating regions, while other codes become more frequent in later cycles where degradation progresses. This demonstrates that the discrete bottleneck provides an interpretable abstraction of discharge dynamics rather than a diffuse continuous representation.

6.2 Discovery of a consistent early degradation precursor code

A specific code (Code 10) repeatedly appears early and is associated with later degradation proxies across cells. We analyze temporal ordering between code activation and degradation proxies to quantify early-warning behavior.

6.3 Relationship to degradation-related signals

We compare feature statistics conditioned on code assignments. Precursor-code segments show consistent deviations (e.g., mild sag / slope changes) aggregated via discretization.

6.4 Cross-cell generalization

Under cross-cell splits, the precursor code remains detectable and preserves similar feature associations in held-out cells, suggesting transferability beyond cell-specific noise.

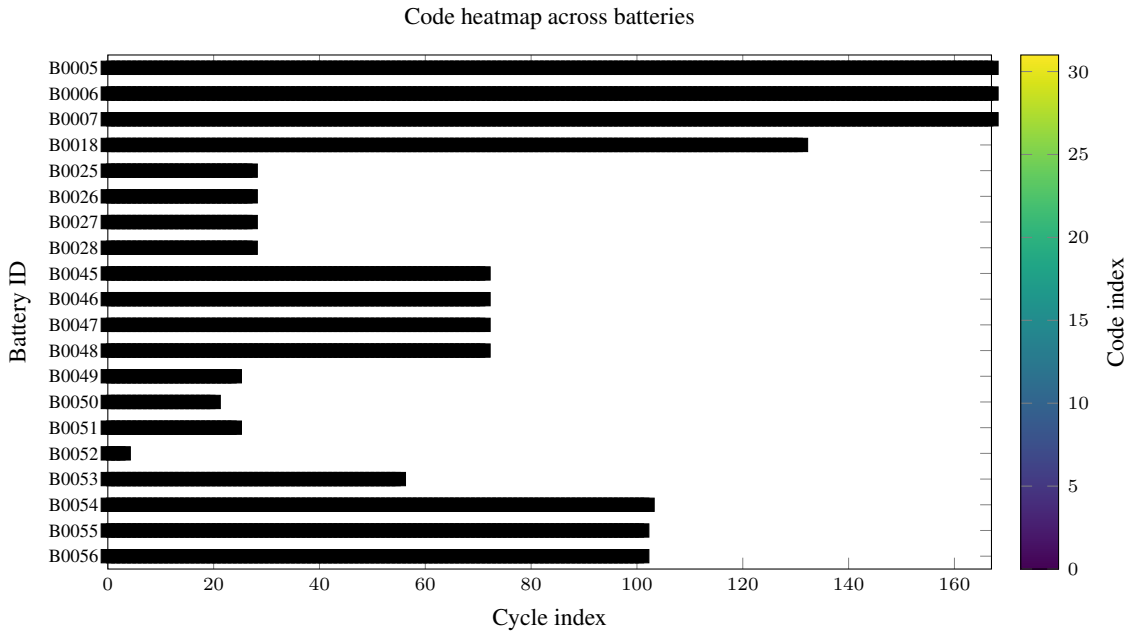

Figure 3: Heatmap of discrete code assignments across cells and cycles.

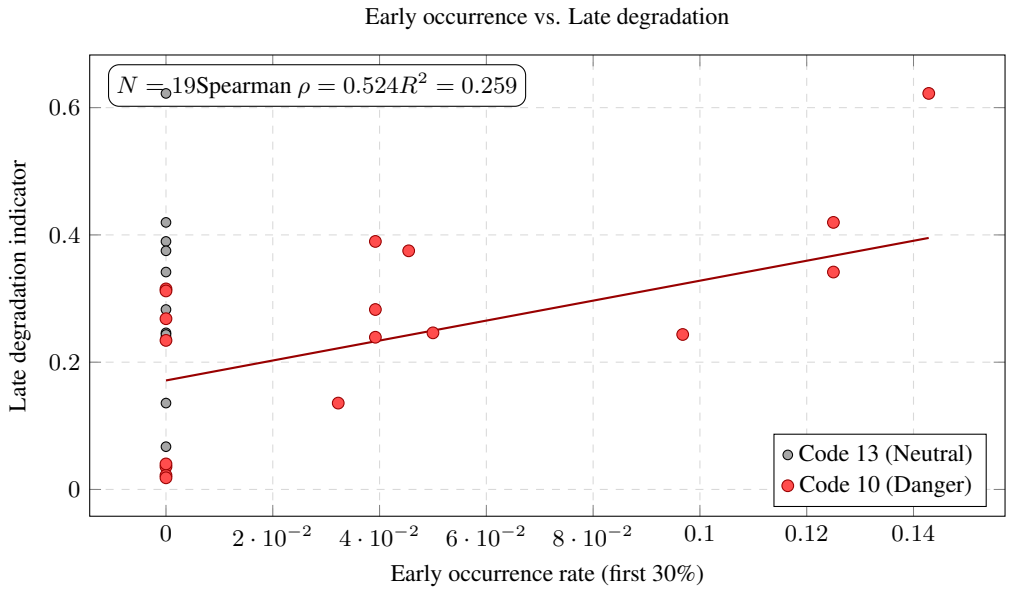

Figure 4: Relationship between early activation of Code 10 and a late-stage degradation proxy. Neutral reference Code 13 is shown for contrast.

6.5 Comparison to baselines

We compare HQ-VQVAE to classical baselines. HQ-VQVAE yields more coherent code groupings and improved precursor detection stability across cells.

7 Discussion

7.1 Why unsupervised discrete codes can reveal early precursors

Discrete code assignments provide a compact, human-auditable representation where recurring patterns can be tracked over time. A precursor code is interpreted as an early-warning pattern, not a deterministic trigger for thermal runaway.

7.2 Role of the quantum bottleneck

The PQC module can be viewed as an expressive nonlinear feature map. Empirically, we observe stable early precursor activation under cross-cell evaluation, suggesting improved sensitivity to weak degradation dynamics.

7.3 Practical deployment: from codes to an early-warning score

A deployable score can be computed from moving-window activation frequency, time-to-first activation, code-transition dynamics, and co-occurrence with late-stage codes.

7.4 Limitations

Limitations include: (i) lack of thermal runaway labels (focus is unsupervised discovery), (ii) dataset scope and operating regime coverage, (iii) mechanism ambiguity without targeted diagnostics, and (iv) computational overhead from hybrid components.

8 Broader Impacts

Hybrid quantum-classical representation learning for battery monitoring may improve safety by enabling earlier intervention and more interpretable alerts in EV/ESS applications. Potential negative impacts include over-reliance on imperfect early-warning signals, false alarms leading to unnecessary replacements, and uneven performance across chemistries or operating conditions if models are deployed outside validated regimes. We mitigate these risks by emphasizing cross-cell validation, transparent discrete-code auditing, and recommending follow-up diagnostics before safety-critical decisions.

9 Conclusion

We proposed HQ-VQVAE for unsupervised discovery of early degradation patterns in lithium-ion battery cycling data. By combining a discrete vector-quantized bottleneck with a PQC module, HQ-VQVAE learns interpretable codes and identifies a cross-cell consistent precursor code that appears prior to stronger degradation manifestations. Future work will validate robustness across broader regimes and integrate additional diagnostics for mechanism identification.

References

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv:1711.00937*, 2017.

Bhaskar Saha and Kai Goebel. Battery data set, NASA Ames Prognostics Data Repository. NASA Ames Research Center, Moffett Field, CA, 2007.

Kai Goebel, Bhaskar Saha, Abhinav Saxena, Jose Celaya, and John P. Christophersen. Prognostics in battery health management. *IEEE Instrumentation & Measurement Magazine*, 11(4):33–40, 2008.

Ville Bergholm et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv:1811.04968*, 2018.

PennyLane Developers. PennyLane documentation. `https://docs.pennylane.ai/`, accessed 2026-01.

LG AI Research. EXAONE 3.5: Series of large language models for real-world use cases. *arXiv:2412.04862*, 2024.

A Appendix / Supplemental material

Table 7: Training hyperparameters for HQ-VQVAE.

Hyperparameter	**Value**	**Description**
Latent dimension (z)	8	Latent embedding dimension used by the encoder.
Codebook size (K)	32	Number of discrete codes in VQ codebook.
Commitment loss (β)	0.25	Weight for commitment loss in VQ-VAE.
PQC qubits (n_{qubits})	8	Number of qubits used in PQC latent bottleneck.
PQC depth (n_{layers})	2	Number of entangling layers in PQC.
Quantum encoding	Angle	Classical-to-quantum encoding method.
Measurement	Pauli-Z	Readout type for PQC outputs.
Optimizer	Adam	Optimizer for training HQ-VQVAE.
Learning rate	1×10^{-3}	Initial learning rate.
Weight decay	0.0	Weight decay for optimizer.
Batch size	256	Mini-batch size.
Epochs	50	Total training epochs.
Gradient clipping	None	Optional gradient clipping.
Scheduler	None	Optional learning rate scheduler.
Input features	$v_{\min}, v_{\max}, v_{\text{mean}}$, $i_{\text{mean}}, T_{\text{mean}}, dV/dt, \ldots$	Feature set used for training.
Normalization	Z-score	Standardization applied per feature.
Train/Test protocol	Cross-cell	Generalization evaluated on unseen batteries.

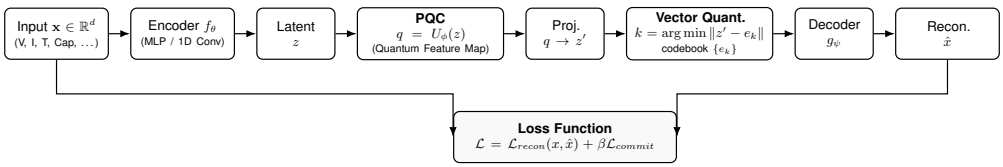

Figure 5: HQ-VQVAE architecture with a quantum bottleneck.

Note : The following interpretations are LLM-generated hypotheses based on summary statistics and are not experimentally verified

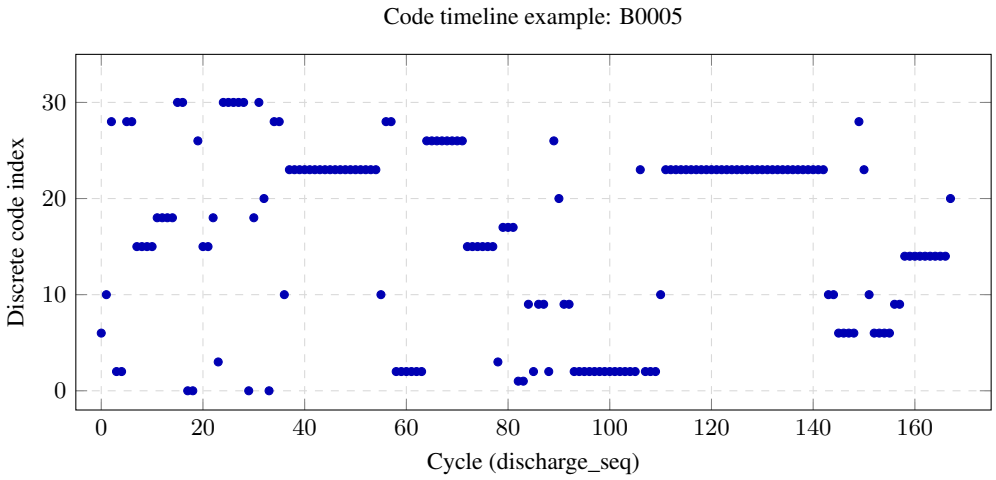

Figure 6: Discrete code assignment over cycles for an example cell.

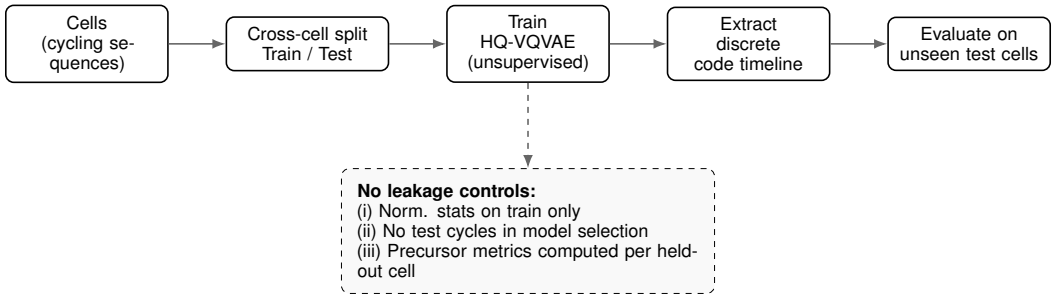

Figure 7: Cross-cell evaluation protocol. The model is trained on a subset of cells and evaluated on unseen cells with leakage prevention.

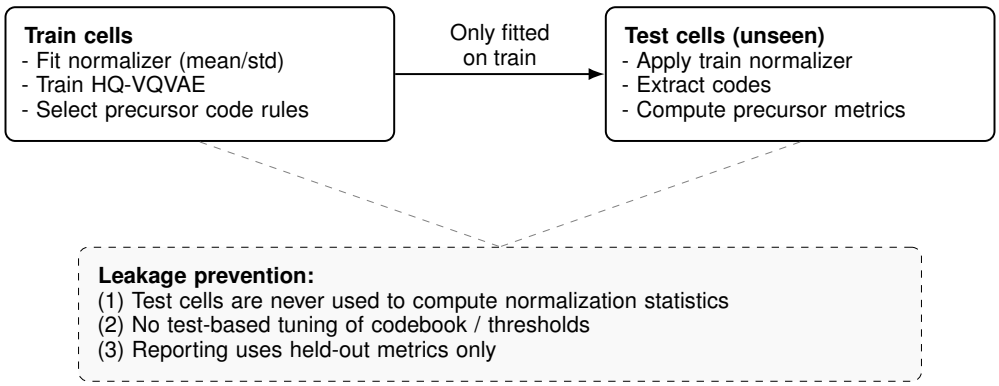

Figure 8: Cross-cell split with explicit no-leakage constraints.

Table 8: Dataset summary (batteries, discharge segments, cycle range, and measurement channels).

Dataset	**# Batt.**	**# Seg.**	**Cycles**	**Channels**
1. BatteryAgingARC_FY08Q4	4	636	0–167	V, I, T, dV/dt stats
2. BatteryAgingARC_25_26_27_28_P1	4	112	0–27	V, I, T, dV/dt stats
3. BatteryAgingARC_45_46_47_48	4	288	0–71	V, I, T, dV/dt stats
4. BatteryAgingARC_49_50_51_52	4	100	0–24	V, I, T, dV/dt stats
5. BatteryAgingARC_53_54_55_56	4	363	0–102	V, I, T, dV/dt stats

Table 9: Compute environment used for experiments.

Item	Value
OS	Linux-6.14.0-29-generic-x86_64-with-glibc2.29
Python	3.8.10 (default, Mar 15 2022, 12:22:08)
pandas	1.4.1
numpy	1.24.4
torch	2.4.1+cu121
cuda_available	True
gpu_name	NVIDIA RTX A5000
cuda_version	12.1
timestamp	2026-01-29T09:26:23

Table 10: Precursor robustness metrics across unseen batteries (activation statistics).

Metric	Value
precursor_code	10.000000
early_ratio	0.300000
cells_total	20.000000
cells_present_anywhere	12.000000
early_rate_mean	0.048558
early_rate_std	0.052101

Table 11: Consolidated LLM-assisted interpretations for VQ codes. (Note: All codes were classified under the category "Battery Degradation Analysis".)

ID	Key Shapes (Morphology)	Hypotheses (Mechanism)	Conf.
0	Linear decrease in capacity; Slight fluctuation	High cycle count leading to capacity loss	Med
1	Decreasing trend in voltage; Flattening	High temperature; Significant aging	High
2	Linear decline; Plateau phase; Exp. decline	Increased internal resistance; Capacity fade	Med
3	Linear decline; Plateau phase; Exp. decline	Increased internal resistance; Capacity fade	High
4	Linear decline; Plateau phase; Exp. decline	Increased internal resistance; Capacity fade	Med
5	Linear decrease in capacity; Steep drop in voltage	High temperature during discharge	Med
6	Linear decrease in capacity; Slight fluctuation	Thermal runaway risk due to high temp	Med
7	Linear decrease in capacity; Flattening of voltage	High $dV/dt_{\min}$ indicates internal short risk	Med
8	Linear decrease in capacity; Steep drop in min	High temperature; Significant aging	Med
9	Initial high capacity drop; Gradual decrease	Increased internal resistance; Capacity fade	Med
10	Linear decrease in voltage; Gradual increase	High temperature; Significant aging	High
11	Initial rapid degradation; Slower decrease	High temperature; Significant aging	Med
12	$t_{total} vs. cycle$; $v_{mean} vs. cycle trends$	Increased internal resistance; Capacity fade	Med
13	Linear decrease in capacity; Flattening of voltage	Increased internal resistance; Capacity fade	Med
14	Linear decrease in voltage; Exp. decrease	High temperature; Significant aging	High
15	Linear decrease in capacity; Flattening of voltage	Increased temperature accelerating degradation	Med
16	Linear decline; Plateau phase; Spike in voltage	High cycle count; Temp effects	Med
17	$t_{total} increasing$; $v_{\min} decreasing$	High temperature; Gradual voltage drop	Med
18	Decreasing voltage over time; Reduced capacity	High temperature; Significant aging	High
19	Linear decrease in capacity; Slight fluctuation	Increased temperature accelerating degradation	Med
20	Decreasing capacity; Steep decline	High temperature; Capacity fade	Med
21	Linear decrease in capacity; Steady increase	Increased internal resistance; Capacity fade	Med
22	Decreasing v_{mean}; $Decreasing trends$	High temperature; Significant aging	High
23	S-shaped curve; Decreasing slope	High temperature; Significant aging	High
24	Linear decrease in capacity; Flattening of voltage	Increased temperature accelerating degradation	Med
25	Decreasing mean voltage; Decreasing min	Increased temperature accelerating degradation	Med
26	Decreasing v_{mean}; $Negative dV/dt$	High temperature; Significant aging	Med
27	Linear decrease in capacity; Slight fluctuation	Thermal runaway risk due to high temp	Med
28	Linear decline; Plateau phase; Sharp drop	Increased internal resistance; Capacity fade	Med
29	S-shaped capacity curve; Linear voltage drop	Thermal runaway risk due to high temp	Med
30	S-shaped curve; Decreasing slope	High temperature; Significant aging	Med
31	Linear decrease in capacity; Steep drop in voltage	High temperature; Significant aging	Med

Table 12: JSON schema used for LLM-assisted interpretation outputs.

Field	Type	Description	Req.
`code_id`	int	Discrete code index	Yes
`short_name`	string	Short descriptive name for the code	Yes
`mechanism_hypotheses`	list	Candidate electrochemical hypotheses	Yes
`[].name`	string	Hypothesis name	Yes
`[].electrochemical_story`	string	Explanation narrative grounded in battery science	Yes
`[].why_this_code_triggers`	string	Reasoning linking features/morphology to hypothesis	Yes
`[].expected_symptoms`	list	Expected measurable symptoms	No
`curve_morphology`	object	Qualitative curve-shape descriptors	Yes
`.key_shapes`	list	Key shape descriptors (e.g., voltage sag)	No
`limitations`	list	Known limitations/uncertainties	No
`suggested_followup_tests`	list	Suggested validation tests (EIS, SEM, etc.)	No
`confidence`	string	Qualitative confidence {low, med, high}	Yes

AI Co-Scientist Challenge Korea Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state the unsupervised precursor-discovery goal, the HQ-VQVAE design, and cross-cell validation scope without claiming direct thermal-runaway prediction.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 7.4, specifically addressing data scarcity and generalization challenges.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The paper defines the architectural assumptions for the HQ-VQVAE and The VQ-VAE training objective (reconstruction + codebook + commitment losses) is defined in Section 3. Mathematical formulations for the quantum circuit mappings are also provided.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results?

   Answer: [Yes]

   Justification: All experimental hyperparameters, data split protocols, and random seed details are disclosed in Appendix A and **Tables 4 and 7**.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results?

   Answer: [Yes]

   Justification: The dataset is publicly available via NASA PCoE. We provide an anonymized link to the source code and reproduction scripts in the supplementary material (URL to be activated upon publication).

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details necessary to understand the results?

   Answer: [Yes]

   Justification: Key details, including feature set selection, optimizer configurations, PQC circuit depth, and hyperparameters, are fully specified in Section 4 and the Appendix.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars or other appropriate information about statistical significance?

   Answer: [Yes]

   Justification: We report the mean and standard deviation of our results over multiple random seeds ($n = 3$) to demonstrate the stability of the proposed method.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources needed to reproduce the experiments?

Answer: [Yes]

Justification: Hardware specifications (e.g., GPU model, memory) and approximate training time per run are listed in Appendix Table 9.

9. **Code Of Ethics**

   Question: Does the research conducted conform with the NeurIPS Code of Ethics?

   Answer: [Yes]

   Justification: The work uses public benchmark data, does not involve human subjects, and includes discussion of limitations and broader impacts.

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive and negative societal impacts?

    Answer: [Yes]

    Justification: Section "Broader Impacts" discusses safety benefits in BMS and potential risks such as false positives in critical systems.

11. **Safeguards**

    Question: Does the paper describe safeguards for responsible release of high-risk data/models?

    Answer: [Yes]

    Justification: Although the model uses a generative architecture (VQ-VAE), its primary focus is representation learning, and we provide guidelines on interpreting anomaly scores to prevent misuse or over-reliance in safety-critical thermal runaway scenarios.

12. **Licenses for existing assets**

    Question: Are creators/owners properly credited and are license/terms explicitly mentioned and respected?

    Answer: [Yes]

    Justification: The NASA PCoE dataset is publicly available from the NASA Ames Research Center and is provided under the U.S. Government works (public domain), which allows for research use.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and provided alongside the assets?

    Answer: [N/A]

    Justification: This paper does not introduce new assets (e.g., a new dataset or benchmark). We use publicly available data, and any accompanying documentation will be provided according to the venue's artifact policy.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include full instructions/compensation details?

    Answer: [N/A]

    Justification: No crowdsourcing and no human-subject data collection are involved.

15. **IRB Approvals or Equivalent**

    Question: Does the paper describe risks, disclosures, and IRB approvals for human subjects (if applicable)?

    Answer: [N/A]

    Justification: No human-subject study is conducted.