CONDITION-SPECIFIC WEIGHTED DATA SPARSIFICATION FOR COMPUTE-EFFICIENT PHYSICAL-SYSTEM REGRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Industrial physical-system models are often retrained from dense datasets collected across various operating conditions. Full-data retraining can incur substantial cloud-side computation to process redundant observations, whereas aggressive pruning may discard rare or difficult states. This work introduces a condition-specific data sparsification framework for cost-constrained industrial machine learning on physical-system regression tasks. The framework formulates sparsification as a condition-specific weighted empirical approximation. Geometric representatives define the retained support, assignment counts preserve the empirical mass of omitted observations, and residual feedback corrects task-relevant omissions under a condition-specific budget. The procedure operates at the training-support level, leaving the predictor architectures and forward computations unchanged. The framework is evaluated on battery state-of-charge (SOC) estimation and gas-turbine NO_x prediction using paired model-initialization seeds. The reported analytical cloud-side FLOP counts cover the modeled data-selection, retraining, and validation computations. The final retention ratios are 39.2%–41.4% for SOC and 38.7%–41.7% for NO_x. In the retention-matched battery MLP experiments, the framework obtains lower mean MAE than Random in two of four driving cycles and than GradMatch in three of four cycles, while using 10.1%–16.2% fewer mean analytical cloud-side FLOPs than GradMatch. Across all settings, seed-averaged FLOP reductions are 8.3%–89.2% for SOC and 20.0%–85.1% for NO_x, while setting-level, seed-averaged relative MAE changes range from -9.8% to +18.0% and from -0.9% to +13.4%, respectively. These results characterize when condition-specific data sparsification is beneficial and support the selection of retention budgets for cost-constrained industrial retraining.

1 INTRODUCTION

Cloud platforms support physical-system modeling by providing centralized infrastructure for storing industrial measurements and developing, retraining, and deploying predictive models for manufacturing applications (Chang et al., 2024; Rajashekarappa et al., 2026). Battery state-of-charge (SOC) estimators must maintain accuracy across ambient temperatures and dynamic operating conditions (Zhang et al., 2026; Liu et al., 2026). Gas turbine NO_x prediction models must remain reliable across variations in key combustion (Winarto et al., 2026; Roy et al., 2026). As operational measurements accumulate, repeatedly retraining a model on the complete dataset can increase cloud-side computation, particularly when the dataset contains many redundant samples from similar operating states (Poenaru-Olaru et al., 2025; Cho et al., 2025).

The practical challenge is therefore to reduce training cost without losing coverage of physically relevant operating states or discarding samples that are difficult for the regression model to predict. Random selection may underrepresent rare operating conditions. Gradient-based methods, including GradMatch, select subsets by minimizing the discrepancy between selected-subset and full-data gradients (Killamsetty et al., 2021a), but their optimization-dependent procedures can introduce additional computational overhead and may require a pretraining phase (Ro et al., 2023). Geometry-based methods preserve input-space coverage but may discard samples that remain difficult for the predictor, while model-dependent methods emphasize task relevance without necessarily preserving

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the empirical mass represented by retained samples. These limitations motivate a condition-specific data selection method that accounts for local geometric coverage, unequal sample representation, and regression-relevant omissions, while leaving the predictor architecture unchanged.

This study develops a feedback-corrected geometric sparsification framework for cloud-based regression. The main contributions are twofold. First, a condition-specific weighted empirical approximation is formulated for cost-constrained regression. The formulation couples geometric support construction, representative-frequency weighting for empirical-mass preservation, and residual-guided support correction within a single training-support objective, while leaving the predictor architectures and forward computations unchanged. Second, a paired-seed empirical study across two industrial physical-system regression tasks and three neural network architectures provides analytical cloud-side FLOP estimates under a stated accounting scope and evaluates the resulting accuracy-computation trade-offs under matched retention budgets. Comparisons with a retention-matched random-selection control and GradMatch, together with retention-budget and configuration-level ablation analyses, identify when the approximation is beneficial and how its trade-offs vary across operating conditions and architectures.

2 Related Work

2.1 Coverage- and Task-Aware Data Subset Selection

Data subset selection reduces training costs by replacing the full training set with a compact subset that retains relevant information. Random sampling provides a simple, low-overhead baseline, but it does not explicitly optimize distributional coverage or task-dependent sample utility (Zheng et al., 2023). Existing selection methods therefore differ primarily in whether they prioritize distributional coverage or task-relevant sample characteristics.

Coverage-aware methods select samples that span the input or representation space. Clustering provides a model-independent mechanism for summarizing redundant observations, while density-aware methods adapt sample retention to local geometric structure (Jiao et al., 2026). CCS combines distributional coverage with importance scores to reduce the concentration induced by importance-only pruning at high pruning rates (Zheng et al., 2023). However, input-space coverage does not necessarily reflect the contribution of a sample to model optimization.

Task-aware methods use sample importance, prediction difficulty, uncertainty, information content, redundancy, or estimated influence on generalization. Dataset pruning method estimates the influence of removing individual training samples (Yang et al., 2023). InfoMax uses information-based criteria, whereas BWS selects a contiguous window from difficulty-ranked samples and studies its behavior across retention ratios (Tan et al., 2025; Choi et al., 2024). These approaches improve task relevance, but their selection objectives do not explicitly preserve the empirical mass represented by retained samples within various operating conditions.

2.2 Optimization-Aware and Regression-Oriented Weighted Selection

Optimization-aware methods derive selection signals directly from the training objective or validation performance. CRAIG constructs a weighted coreset by optimizing a submodular facility-location objective in gradient space, thereby approximating the full-data gradient (Mirzasoleiman et al., 2020). GLISTER formulates subset selection as a bilevel optimization problem whose outer objective measures validation performance (Killamsetty et al., 2021b). GradMatch uses orthogonal matching pursuit to identify weighted subsets whose gradients approximate those of the training or validation data (Killamsetty et al., 2021a). These methods incorporate task-dependent optimization signals, but they require gradient or validation-loss evaluations and may introduce additional optimization or pretraining overhead.

Among regression-oriented formulations, SELCON selects samples for L_2-regularized regression by constraining validation-set generalization error (Durga et al., 2021). Its objective emphasizes predictive generalization rather than constructing a condition-specific approximation of the full empirical regression objective.

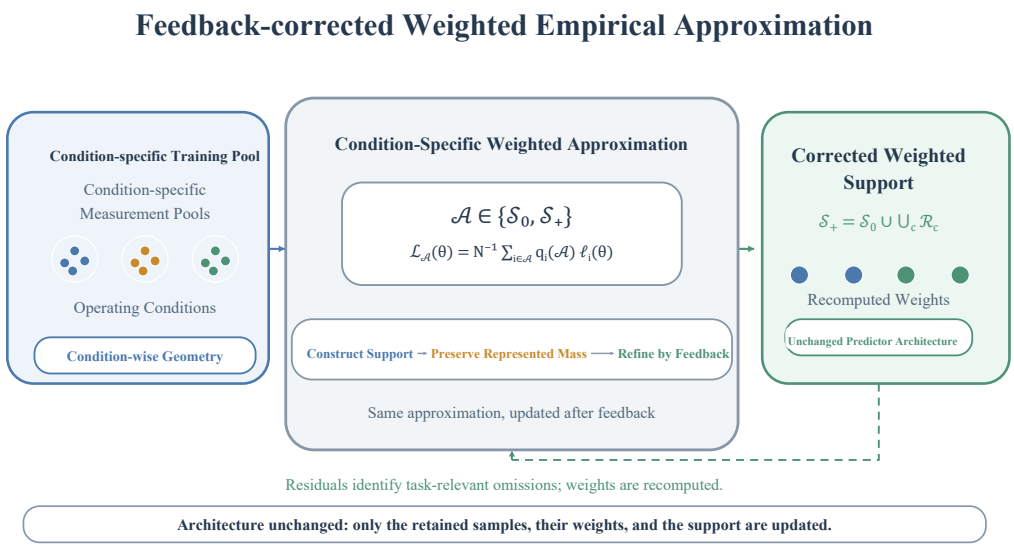

Figure 1: Overview of the proposed framework. Condition-specific geometry constructs an initial support, representative assignments preserve empirical sample mass, and feedback scores guide the re-entry of difficult omitted samples. The predictor architecture is shared by all stages.

The proposed framework casts support construction, representative-frequency weighting, and residual-guided refinement as successive elements of a single condition-specific weighted empirical approximation. Within this formulation, the initial support captures within-condition coverage, assignment counts preserve the empirical mass associated with omitted observations, and predictive feedback identifies task-relevant omissions that guide support refinement. A qualitative comparison with related data-selection methods is provided in Appendix A.

3 Method

To reduce cloud-side retraining cost on dense physical-system datasets collected under multiple operating conditions while preserving coverage of physically relevant states and task-relevant difficult samples, a condition-specific, feedback-corrected weighted-support framework is developed. The framework separates three functions that are often entangled in data reduction. Geometry preserves coverage of the observed input space, representative-frequency weights preserve the empirical mass of omitted samples, and prediction-error feedback restores omitted samples that remain difficult for the predictor. The predictor architecture and target definition are unchanged. This separation allows computational savings to be measured without changing model capacity, as illustrated in Figure 1.

3.1 Problem Formulation

Let the training set be partitioned by observed operating-condition metadata:

$$\mathcal{D} = \bigsqcup_{c=1}^{C} \mathcal{D}_c, \qquad \mathcal{D}_c = \{(\mathbf{x}_i^c, y_i^c)\}_{i=1}^{N_c}, \qquad N = \sum_{c=1}^{C} N_c. \tag{1}$$

Here $\mathbf{x}_i^c \in \mathbb{R}^{T \times F}$ contains T time steps and F input features, and $y_i^c \in \mathbb{R}$ is the regression target. For non-sequential tabular data, $T = 1$. The condition identifier is obtained from dataset metadata during data management and is not inferred from the target or from the proposed selection scores.

For a prediction $\widehat{y}$ and target y, the training loss is the Huber loss

$$\ell_\delta(\widehat{y}, y) = \begin{cases} \frac{1}{2}(\widehat{y} - y)^2, & |\widehat{y} - y| \le \delta, \\ \delta\left(|\widehat{y} - y| - \frac{\delta}{2}\right), & |\widehat{y} - y| > \delta. \end{cases} \tag{2}$$

The full-data empirical objective is

$$\mathcal{L}_{\text{full}}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{c=1}^{C} \sum_{i=1}^{N_c} \ell_\delta(f_{\boldsymbol{\theta}}(\mathbf{x}_i^c), y_i^c)\,. \tag{3}$$

All normalization statistics are fitted on the training portion only and then held fixed for validation and test data. For sequential data, raw train, validation, and test boundaries are defined before window construction.

3.2 Feedback-Corrected Weighted Support

The procedure contains up to three sequential training stages. Let $\boldsymbol{\theta}_0$ denote the paired model initialization. If $E_{\text{w}} > 0$, Stage I trains the predictor on the full training pool for exactly E_{w} epochs. If $E_{\text{w}} = 0$, Stage I is skipped and the paired model initialization is passed directly to Stage II. Validation MAE is evaluated after each epoch, and the checkpoint with the lowest validation MAE is retained as $\boldsymbol{\theta}_{\text{w}}$.

Equivalently, the initialization of Stage II is

$$\boldsymbol{\theta}_{\text{start}} = \begin{cases} \boldsymbol{\theta}_{\text{w}}, & E_{\text{w}} > 0, \\ \boldsymbol{\theta}_0, & E_{\text{w}} = 0. \end{cases} \tag{4}$$

Stage II starts from $\boldsymbol{\theta}_{\text{start}}$ and trains on the weighted initial support $\mathcal{S}_0$ for at most E_{s} epochs. Validation-MAE early stopping with patience p_{s} determines the retained Stage-II checkpoint $\boldsymbol{\theta}_{\text{s}}$. If feedback re-entry is enabled and produces a nonempty correction set, Stage III starts from $\boldsymbol{\theta}_{\text{s}}$ and trains on $\mathcal{S}_+$ for at most E_{r} epochs with patience p_{r}. The checkpoint with the lowest validation MAE in the final executed stage is returned. If Stage III is disabled, has zero budget, or produces no new samples, the Stage-II checkpoint remains the final model.

3.2.1 Condition-Specific Geometric Support

Selection is performed separately for every operating condition in normalized input space. Each input is vectorized as

$$\mathbf{z}_i^c = \text{vec}(\mathbf{x}_i^c).$$

For condition c, MiniBatch K-means partitions the samples into

$$K_c = \min(K_{\max}, N_c), \qquad K_{\max} = 10. \tag{5}$$

The implementation uses a minibatch size of 256 and clustering seed 42. Let $\mathcal{D}_{ck}$ denote the samples assigned to cluster k in condition c. If sample i is assigned to cluster κ_i^c with centroid $\boldsymbol{\mu}_{c\kappa_i^c}$, its centroid distance is

$$d_i^c = \left\| \mathbf{z}_i^c - \boldsymbol{\mu}_{c\kappa_i^c} \right\|_2 . \tag{6}$$

Within each cluster, centroid distances are sorted and smoothed with an odd moving window adapted to the cluster cardinality. For a cluster of size n_{ck}, the nominal window width is $2\lfloor n_{ck}/20 \rfloor + 1$. The width is bounded below by three when $n_{ck} \geq 3$ and clipped to the largest valid odd width for smaller clusters. Rank and smoothed distance are then normalized to $[0, 1]$. The elbow rank is the point with the largest absolute deviation between the resulting curve and the unit diagonal. If τ_{ck}^- denotes the distance immediately preceding that rank, a robust upper threshold is defined by

$$\tau_{ck}^+ = \text{median}(d_{ck}) + s(1.4826)\,\text{MAD}(d_{ck}), \qquad s = 3, \tag{7}$$

where

$$\text{MAD}(V) = \text{median}_{v \in V} \left| v - \text{median}(V) \right|. \tag{8}$$

When the MAD is zero, τ_{ck}^+ is treated as unbounded. The distance-based component is

$$\mathcal{S}_{\text{dist},c} = \left\{ i \in \mathcal{D}_c : \tau_{c\kappa_i^c}^- \leq d_i^c \leq \tau_{c\kappa_i^c}^+ \right\}. \tag{9}$$

This criterion removes the low-distance core and the robustly identified high-distance tail while retaining the intermediate geometric shell selected by the elbow and upper threshold.

Cluster coverage is protected independently of the distance interval. For each nonempty cluster, the $\min(M, |\mathcal{D}_{ck}|)$ samples nearest to its centroid are retained, with $M = 100$:

$$\mathcal{S}_{\text{center},c} = \bigcup_{k=1}^{K_c} \text{Nearest}_M\left(\boldsymbol{\mu}_{ck}, \mathcal{D}_{ck}\right). \tag{10}$$

Thus, every nonempty cluster contributes at least one representative to any subsequent nearest-representative assignment.

For sequential data, a transition score is computed only between adjacent windows from the same contiguous training segment:

$$a_i^c = \left\|\mathbf{x}_i^c[T,:] - \mathbf{x}_{i-1}^c[T,:]\right\|_2. \tag{11}$$

The first window of each segment and every non-sequential sample receive a zero transition score. For the nonzero scores in cluster k, a transition representative satisfies

$$a_i^c > \text{median}(a_{ck}) + s(1.4826)\,\text{MAD}(a_{ck}), \qquad d_i^c \le \tau_{ck}^{+}. \tag{12}$$

The same multiplier $s = 3$ is used for transition filtering. If the transition MAD is zero, the threshold reduces to the median score and only strictly larger scores are retained. The condition-specific geometric support is the union of all three components:

$$\mathcal{S}_{\text{geo},c} = \mathcal{S}_{\text{dist},c} \cup \mathcal{S}_{\text{center},c} \cup \mathcal{S}_{\text{trans},c}, \qquad \mathcal{S}_{\text{geo}} = \bigcup_{c=1}^{C} \mathcal{S}_{\text{geo},c}. \tag{13}$$

An auxiliary reference set $\mathcal{T}_c$ containing up to $B = 8$ landmarks is computed for each condition. In the main configuration, topology protection is disabled, so landmarks are not directly inserted into the training support. They determine only the number of ordinary replacement samples:

$$b_c = |\mathcal{T}_c \setminus \mathcal{S}_{\text{geo},c}|. \tag{14}$$

The b_c replacements are selected outside $\mathcal{S}_{\text{geo},c}$ with a fixed replacement seed, prioritizing clusters that contain unretained landmarks. Reference landmarks themselves are excluded from this ordinary replacement pool. Let the selected set be $\mathcal{O}_c$. The initial support is

$$\mathcal{S}_0 = \mathcal{S}_{\text{geo}} \cup \bigcup_{c=1}^{C} \mathcal{O}_c. \tag{15}$$

3.2.2 Representative-Frequency Weighting

Let $\mathcal{A} \subseteq \mathcal{D}$ denote the current support, either $\mathcal{S}_0$ or its feedback-corrected extension $\mathcal{S}_+$. For each condition and cluster, define $\mathcal{A}_{ck} = \mathcal{A} \cap \mathcal{D}_{ck}$. Every omitted sample is assigned to the nearest retained representative in the same condition and cluster:

$$\pi_{\mathcal{A}}(j) = \arg\min_{i \in \mathcal{A}_{ck}} \left\|\mathbf{z}_j^c - \mathbf{z}_i^c\right\|_2. \tag{16}$$

The raw representative-frequency weight for $i \in \mathcal{A}_{ck}$ is

$$q_i(\mathcal{A}) = 1 + \sum_{j \in \mathcal{D}_{ck} \setminus \mathcal{A}_{ck}} \mathbb{I}[\pi_{\mathcal{A}}(j) = i]. \tag{17}$$

The initial unit contribution accounts for the retained sample itself. Because every omitted sample is assigned exactly once, the weights preserve empirical sample mass:

$$\sum_{i \in \mathcal{A}} q_i(\mathcal{A}) = N. \tag{18}$$

The normalized weight used by mini-batch training is

$$w_i(\mathcal{A}) = \frac{|\mathcal{A}|}{N} q_i(\mathcal{A}). \tag{19}$$

This normalization makes the mean support weight equal to one while retaining the relative frequency represented by each sample. With $\ell_i(\boldsymbol{\theta}) = \ell_\delta(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i)$, the weighted objective is

$$\begin{aligned}\mathcal{L}_{\mathcal{A}}(\boldsymbol{\theta}) &= \frac{1}{|\mathcal{A}|}\sum_{i\in\mathcal{A}} w_i(\mathcal{A})\ell_i(\boldsymbol{\theta}) \\ &= \frac{1}{N}\sum_{i\in\mathcal{A}} q_i(\mathcal{A})\ell_i(\boldsymbol{\theta}).\end{aligned} \tag{20}$$

Therefore, a retained sample contributes both its own loss and the loss mass of omitted samples assigned to it. Stage II minimizes $\mathcal{L}_{\mathcal{S}_0}$ and produces $\boldsymbol{\theta}_{\mathrm{s}}$.

3.2.3 Prediction-Error Feedback Correction

After Stage II, one inference pass is performed over the full training pool $\mathcal{D}$ using $\boldsymbol{\theta}_{\mathrm{s}}$. This pass evaluates all training samples for computational accounting, but only samples outside the initial support are eligible for re-entry. Validation and test samples are never ranked or inserted. For each operating condition c, define

$$\mathcal{S}_{0,c} = \mathcal{S}_0 \cap \mathcal{D}_c, \qquad \mathcal{U}_c = \mathcal{D}_c \setminus \mathcal{S}_{0,c}, \tag{21}$$

where $\mathcal{U}_c$ denotes the discarded training samples eligible for feedback-guided re-entry. If $\mathcal{U}_c =$, condition c is skipped and its restoration quota is set to zero.

For each eligible sample $j \in \mathcal{U}_c$, define the absolute prediction error and Huber loss in normalized target space:

$$e_j = \left|f_{\boldsymbol{\theta}_{\mathrm{s}}}(\mathbf{x}_j^c) - y_j^c\right|, \qquad h_j = \ell_\delta\left(f_{\boldsymbol{\theta}_{\mathrm{s}}}(\mathbf{x}_j^c), y_j^c\right). \tag{22}$$

Because the two quantities have different numerical scales, they are normalized separately within each operating condition:

$$s_{e,c} = \mathrm{median}_{u\in\mathcal{U}_c}(e_u) + \varepsilon, \qquad s_{h,c} = \mathrm{median}_{u\in\mathcal{U}_c}(h_u) + \varepsilon, \tag{23}$$

where $\varepsilon > 0$ is a fixed numerical constant. The normalized feedback terms are

$$\widetilde{e}_j = \frac{e_j}{s_{e,c}}, \qquad \widetilde{h}_j = \frac{h_j}{s_{h,c}}. \tag{24}$$

The feedback score is

$$g_j = \alpha\widetilde{e}_j + (1-\alpha)\widetilde{h}_j, \qquad \alpha = 0.5. \tag{25}$$

The coefficient α is fixed across all experimental settings and is not tuned using validation or test data. The normalized absolute error and Huber loss jointly characterize prediction difficulty.

Given the restoration fraction ρ, the condition-specific restoration quota is

$$m_c = \min(|\mathcal{U}_c|, \lceil \rho N_c \rceil), \qquad \rho = 0.05. \tag{26}$$

The m_c samples in $\mathcal{U}_c$ with the largest feedback scores form the re-entry set $\mathcal{R}_c$. The feedback-corrected support is

$$\mathcal{S}_+ = \mathcal{S}_0 \cup \bigcup_{c=1}^{C} \mathcal{R}_c, \qquad |\mathcal{S}_{+,c}| = |\mathcal{S}_{0,c}| + m_c. \tag{27}$$

Thus, ρ specifies a condition-specific restoration quota rather than the final retention ratio. Representative assignments and frequency weights are recomputed on $\mathcal{S}_+$ before Stage III. Starting from $\boldsymbol{\theta}_{\mathrm{s}}$, Stage III minimizes $\mathcal{L}_{\mathcal{S}_+}$ for at most E_{r} epochs with validation-MAE early stopping. If no sample is re-entered, the Stage-II checkpoint remains the final model.

For a single-pool dataset, $C = 1$ and the same procedure reduces to feedback-guided restoration within one training pool. In retention-matched comparisons, Random and GradMatch use the realized condition-specific final cardinalities $|\mathcal{S}_{+,c}|$. Random replaces geometric ranking and feedback-guided re-entry with uniform random selection and uniform random re-entry while preserving support sizes, representative-frequency weighting, training budgets, paired seeds, and checkpoint rules.

4 EXPERIMENT SETUP

The evaluation uses two public industrial regression datasets. For battery SOC estimation, the Panasonic 18650PF dataset contains HWFET, LA92, UDDS, and US06 cycles recorded at $0\,^{\circ}\mathrm{C}$. A leave-one-cycle-out protocol holds out one cycle for testing, while the other three form the development pool. The validation segment for the first condition is the initial one-fifth of its record, the validation segment for the second condition is the middle one-fifth, and the validation segment for the third condition is the final one-fifth. All remaining records are used for training. Windows of length five are then generated independently within each contiguous training or validation segment, so no window crosses a train-validation boundary. For gas-turbine NO_x prediction, the UCI Gas Turbine CO and NO_x Emission Data Set is evaluated independently on the 2013, 2014, and 2015 subsets using target-stratified 70%/20%/10% train-validation-test splits. Each year is treated as one non-temporal training pool.

MLP, FCN, and CNN predictors are evaluated. Within each setting, all methods use the same architecture, data split, optimization configuration, applicable training budgets, and checkpoint rule. Model-initialization seeds are paired across methods and set to $\{42, 2026, 3026\}$, while data splitting, selection, and training-order seeds are fixed to 42. The main retention-matched comparison is conducted in the battery MLP setting. Random and GradMatch use the realized condition-specific final support sizes of the proposed method. Independently trained full-data models provide the accuracy and computational references. Component ablations and retention-ratio analyses are also conducted in this setting. All models use Adam with learning rate 10^{-4}, batch size 32, and Huber loss. Per-setting epoch budgets and realized epochs are reported in Appendix C. Results are reported as mean $\pm$ standard deviation over the three paired seeds.

Reported analytical cloud-side FLOPs include the modeled MiniBatch K-means and reachability computations, together with the counted training and validation computations. They exclude full-pool feedback inference, representative reassignment, frequency-weight recomputation, optimizer updates, data movement, and checkpoint I/O. One multiply-accumulate operation counts as two FLOPs. For method m and paired seed r,

$$\Delta^{(m,r)}_{\mathrm{FLOPs}} = 100\frac{F^{(r)}_{m} - F^{(r)}_{\mathrm{full}}}{F^{(r)}_{\mathrm{full}}}. \tag{28}$$

Negative values indicate computational savings; setting-level values are reported as the mean and standard deviation over the three paired seeds.

5 RESULTS AND ANALYSIS

5.1 ACCURACY-COMPUTATION TRADE-OFF

Table 1 summarizes setting-level, seed-averaged changes in test MAE and accounted analytical FLOPs relative to independently trained, seed-matched full-data models. The final retention ratio is 39.2%–41.4% for battery SOC and 38.7%–41.7% for gas-turbine NO_x, with identical retention ratios across architectures within each condition or year.

Across battery settings, absolute MAE changes range from -0.0079 to $+0.0045$, relative MAE changes range from -9.8% to $+18.0\%$, and FLOP reductions range from 8.3% to 89.2%. Across gas-turbine settings, the corresponding ranges are -0.029 to $+0.279\ \mathrm{mg/m^3}$, -0.9% to $+13.4\%$, and 20.0% to 85.1%, respectively. Thus, the proposed pipeline reduces accounted analytical computation in every evaluated setting, while the accuracy-cost trade-off remains condition- and architecture-dependent. The per-seed MAE, FLOP, and accuracy-change values underlying these setting-level ranges are reported in Appendix D

5.2 COMPARISON WITH DATA-SELECTION BASELINES

Table 2 compares the proposed framework with Random and GradMatch for battery MLP SOC estimation. Random and GradMatch use exactly the same final condition-specific support cardinalities as the proposed method for every paired seed. Random replaces only the initial and re-entry sample-identity selection rules with budget-matched random selection. All other pipeline components are

Table 1: Ranges of setting-level, seed-averaged changes in test MAE and accounted analytical cloud-side FLOPs relative to the corresponding seed-matched full-data models. For each setting, the MAE and FLOP changes are first computed separately for each paired seed and are then averaged across the three model-initialization seeds. The reported ranges are taken over these setting-level averages. Battery SOC MAE changes are reported in normalized SOC units, whereas gas-turbine NO_x MAE changes are reported in $\mathrm{mg/m^3}$ on the original target scale. The signed FLOP change Δ_{FLOPs} is defined in Eq. 28; negative values indicate computational savings, whereas positive values indicate additional computational cost.

Task	Model	$\Delta\mathrm{MAE}_{\mathrm{abs}}$	$\Delta\mathrm{MAE}_{\mathrm{rel}}$ (%)	Δ_{FLOPs} (%)
Battery SOC	MLP	-0.0071 to $+0.0014$	-8.5 to $+3.2$	-49.4 to -8.3
Battery SOC	FCN	$+0.0002$ to $+0.0036$	$+0.4$ to $+13.5$	-89.2 to -70.2
Battery SOC	CNN	-0.0079 to $+0.0045$	-9.8 to $+18.0$	-64.6 to -44.3
Gas NO_x	MLP	-0.029 to $+0.269$	-0.9 to $+6.7$	-85.1 to -76.4
Gas NO_x	FCN	$+0.135$ to $+0.212$	$+4.2$ to $+10.5$	-57.8 to -20.0
Gas NO_x	CNN	$+0.235$ to $+0.279$	$+7.6$ to $+13.4$	-39.3 to -29.9

Table 2: Retention-matched comparison for battery MLP SOC estimation. Values are mean $\pm$ standard deviation over three paired seeds. Random and GradMatch use the same final support cardinalities as the proposed method. Negative FLOP changes indicate savings relative to the corresponding seed-matched full-data model. Random is a selection-only control; the reported FLOP changes correspond only to GradMatch and Proposed.

Condition	Random MAE	GradMatch MAE	Proposed MAE	$\Delta^{\mathrm{GM}}_{\mathrm{FLOPs}}$ (%)	$\Delta^{\mathrm{prop}}_{\mathrm{FLOPs}}$ (%)
HWFET	0.04453 ± 0.00057	0.04574 ± 0.00218	0.04501 ± 0.00227	-39.7 ± 3.6	-49.4 ± 3.0
LA92	0.03211 ± 0.00140	0.03019 ± 0.00031	0.03004 ± 0.00002	$+4.4 \pm 22.1$	-12.2 ± 18.6
UDDS	0.02350 ± 0.00137	0.02286 ± 0.00046	0.02402 ± 0.00191	-28.4 ± 21.6	-35.7 ± 19.4
US06	0.08661 ± 0.03262	0.08424 ± 0.02689	0.07635 ± 0.02162	$+9.4 \pm 19.3$	-8.3 ± 16.2

held fixed, so Random is used to isolate the accuracy effect of sample selection rather than to provide an independent FLOP comparison.

The proposed method obtains lower mean MAE than Random on LA92 and US06, whereas Random is better on HWFET and UDDS. Relative to GradMatch, the proposed method is better on HWFET, LA92, and US06, while GradMatch is better on UDDS. The proposed method reduces mean accounted analytical FLOPs by 49.4%, 12.2%, 35.7%, and 8.3% on HWFET, LA92, UDDS, and US06, respectively. GradMatch reduces FLOPs on HWFET and UDDS but incurs additional cost on LA92 and US06. Across all four conditions, the proposed pipeline uses 10.1%–16.2% fewer mean analytical cloud-side FLOPs than GradMatch. These results indicate condition-dependent advantages rather than uniform accuracy superiority. The corresponding per-seed accuracy and computational results are reported in Appendix E.

5.3 Component Ablation

The component ablation is conducted in the battery MLP setting. Unweighted and Weighted use identical geometric supports and retention ratios, isolating the effect of representative-frequency weighting. The remaining re-entry configurations use the same final support budgets but differ in optimizer-state handling and checkpoint selection. Therefore, these comparisons measure configuration-level effects rather than the isolated effect of feedback re-entry alone. All configurations use the same data split, shared pipeline seed, training-order seed, and three paired model-initialization seeds.

Table 3 reports the setting-level test MAE. Frequency weighting improves the mean on HWFET and US06 but is not uniformly beneficial on LA92 and UDDS. The proposed configuration gives the lowest mean MAE on LA92, UDDS, and US06 among the evaluated ablation variants, while the Weighted configuration is best on HWFET. The seed-level results and complete configuration definitions are provided in Appendix F.

Table 3: Core component ablation for battery MLP SOC estimation. Values are test MAE mean ± standard deviation over three paired seeds; lower is better.

Configuration	HWFET	LA92	UDDS	US06
Unweighted	0.04703 ± 0.00102	0.03101 ± 0.00148	0.02439 ± 0.00030	0.09403 ± 0.04656
Weighted	$\mathbf{0.04418 \pm 0.00066}$	0.03171 ± 0.00071	0.02720 ± 0.00782	0.07975 ± 0.02555
Shared Adam / global checkpoint	0.04649 ± 0.00230	0.03163 ± 0.00181	0.02739 ± 0.00544	0.08084 ± 0.02903
Reset Adam / global checkpoint	0.04641 ± 0.00236	0.03126 ± 0.00170	0.02546 ± 0.00133	0.08084 ± 0.02903
Proposed	0.04501 ± 0.00227	$\mathbf{0.03004 \pm 0.00002}$	$\mathbf{0.02402 \pm 0.00191}$	$\mathbf{0.07635 \pm 0.02162}$

Table 4: Pareto-efficient operating points in the battery MLP retention-budget analysis. The candidate set contains the 20%, 40%, 60%, and 80% sparse configurations and the independently trained, seed-matched full-data reference. Values are mean ± standard deviation over three paired seeds.

Condition	Operating point	Test MAE	Δ_{FLOPs} (%)
HWFET	20%	0.04792 ± 0.00052	-57.5
HWFET	40%	0.04648 ± 0.00411	-49.6
HWFET	80%	0.04620 ± 0.00113	-33.7
HWFET	Full data	0.04361 ± 0.00138	0.0
LA92	20%	0.03167 ± 0.00078	-25.5
LA92	40%	0.03147 ± 0.00099	-11.7
LA92	60%	0.03142 ± 0.00035	$+2.2$
LA92	80%	0.03138 ± 0.00111	$+16.1$
UDDS	20%	0.03328 ± 0.00284	-45.7
UDDS	40%	0.02722 ± 0.00139	-35.5
UDDS	80%	0.02490 ± 0.00186	-15.3
UDDS	Full data	0.02412 ± 0.00064	0.0
US06	20%	0.07944 ± 0.03061	-23.6
US06	80%	0.07910 ± 0.03101	$+19.2$

5.4 Retention-Budget Sensitivity

The retention-budget analysis evaluates target retention ratios of 20%, 40%, 60%, and 80% in the battery MLP setting. Independently trained, seed-matched full-data models are included as reference operating points. Pareto efficiency jointly minimizes setting-level mean test MAE and accounted analytical cloud-side FLOPs. Table 4 reports the resulting Pareto-efficient operating points.

The 20% configuration is Pareto-efficient for all four conditions and reduces mean accounted analytical FLOPs by 23.6%–57.5%. The 40% configuration remains Pareto-efficient for HWFET, LA92, and UDDS, where it improves mean MAE over the corresponding 20% configuration while reducing mean FLOPs by 11.7%–49.6%. These results support approximately 40% retention as a intermediate operating point for most SOC conditions, rather than a universally optimal budget. At higher retention, the modeled selection operations and additional training and validation passes can offset the savings from sparse training. The corresponding trade-offs are analyzed in Appendix G

6 Conclusion

This study introduces a condition-specific weighted-support framework for cost-constrained industrial regression. Across battery SOC and gas-turbine NO_x tasks using MLP, FCN, and CNN predictors, estimated accounted analytical cloud-side FLOPs decreased by 8.3%–89.2% and 20.0%–85.1%, respectively. The corresponding relative MAE changes ranged from -9.8% to $+18.0\%$ and from -0.9% to $+13.4\%$. Under matched final retention in the battery MLP task, the framework obtained lower mean MAE than Random in two of four conditions and than GradMatch in three of four conditions, while requiring 10.1%–16.2% fewer mean analytical cloud-side FLOPs than GradMatch across the four conditions.

The reported analytical FLOPs include explicitly modeled support construction and all executed training and validation passes. The evaluation is limited to two industrial regression tasks, three predictor architectures, and the examined operating-conditions. The results characterize an application-dependent accuracy-computation trade-off rather than universal superiority, and support retention-budget selection under the analytical accounting scope.

AI USE STATEMENT

Generative AI tools were used during the preparation of this submission for literature retrieval and discovery, research ideation and execution support, drafting portions of the manuscript, and language editing. In particular, these tools were used to help identify potentially relevant prior work, organize and refine research ideas, improve the clarity and concision of technical explanations, and polish the English presentation of selected sections.

Generative AI tools were not used to generate, fabricate, or modify experimental data, numerical results, figures, tables, citations, or bibliographic records. They were also not used as a substitute for the authors' scientific judgment in defining the research question, designing the core method, selecting experimental settings, interpreting results, or drawing the final conclusions. All experiments, implementations, analyses, and final scientific decisions were conducted and verified by the authors.

All AI-assisted text, code suggestions, literature suggestions, and other outputs were reviewed by the authors. Suggested references were checked against the original sources, and AI-assisted code was inspected, executed, and tested for consistency with the reported experimental protocol. The authors also checked the manuscript for unsupported claims, citation errors, inconsistencies between the method and implementation, and potential plagiarism. The authors take full responsibility for the final content of this work, including all text, claims, code, figures, tables, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work uses publicly available industrial datasets and does not involve human or animal subjects, personally identifiable information, or sensitive personal data. The datasets are used under their respective research-use conditions and are properly cited. The proposed method is evaluated only for reducing cloud-side training costs in industrial regression and is not presented as a substitute for domain-specific validation in safety-critical deployment. The authors declare no conflicts of interest.

REPRODUCIBILITY STATEMENT

The paper specifies the proposed data-sparsification procedure, support construction, representative-frequency weighting, feedback-based restoration, and sequential training stages in Section 3. The experimental datasets, preprocessing procedures, data partitions, baseline protocols, optimization settings, random seeds, evaluation metrics, and analytical FLOP accounting scope are described in Section 4. Appendix B reports the hardware and software environment, while Appendix C provides the predictor architectures, Huber-loss threshold, epoch budgets, early-stopping rules, and stage-local checkpoint configuration. Appendix D reports the complete per-seed accuracy and computational results, and Appendices E–G provide the baseline, ablation, and retention-budget analyses.

The experiments use publicly available datasets with the corresponding sources cited in the reference list. Together, the stated procedures, fixed configurations, random seeds, accounting definition, and per-condition results are intended to facilitate independent reproduction of the reported experiments. The computational results should be interpreted as analytical cloud-side FLOP estimates under the accounting scope defined in Section 4, rather than as direct measurements of wall-clock time or energy consumption.

REFERENCES

Ray-I Chang, Jia-Ying Lin, and Yu-Hsin Hung. Cloud-based machine learning methods for parameter prediction in textile manufacturing. *Sensors*, 24(4):1304, February 2024. doi: 10.3390/s24041304. URL https://doi.org/10.3390/s24041304.

Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10602–10643. PMLR, July 2025. URL https://proceedings.mlr.press/v267/cho25e.html.

Hoyong Choi, Nohyun Ki, and Hye Won Chung. BWS: Best window selection based on sample scores for data pruning across broad ranges. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 8672–8701. PMLR, July 2024. URL https://proceedings.mlr.press/v235/choi24c.html.

S. Durga, Rishabh Iyer, Ganesh Ramakrishnan, and Abir De. Training data subset selection for regression with controlled generalization error. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9202–9212. PMLR, July 2021. URL https://proceedings.mlr.press/v139/s21a.html.

Zhengxing Jiao, Yaxin Hou, Jun Ma, Yuhang Li, Ding Ding, Yuheng Jia, Hui Liu, and Junhui Hou. Samples are not equal: A sample selection approach for deep clustering. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/63ef323523f3be8b58ed9277cc747485-Paper-Conference.pdf.

Krishnateja Killamsetty, Durga S, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. GRAD-MATCH: Gradient matching based data subset selection for efficient deep model training. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 5464–5474. PMLR, July 2021a. URL https://proceedings.mlr.press/v139/killamsetty21a.html.

Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. GLISTER: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8110–8118. AAAI Press, 2021b. doi: 10.1609/aaai.v35i9.16988. URL https://ojs.aaai.org/index.php/AAAI/article/view/16988.

Qiao Liu, Zhuo Zhang, Xinyue Shu, Mingyang Zhang, and Xuelin Xu. A gaussian filtering enhanced TCN-transformer hybrid network for robust state of charge estimation of lithium-ion batteries under complex operating conditions. *Ionics*, 32(9):9753–9767, 2026. doi: 10.1007/s11581-026-07361-x. URL https://doi.org/10.1007/s11581-026-07361-x.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6950–6960. PMLR, July 2020. URL https://proceedings.mlr.press/v119/mirzasoleiman20a.html.

Lorena Poenaru-Olaru, June Sallou, Luis Cruz, Jan S. Rellermeyer, and Arie van Deursen. Sustainable machine learning retraining: Optimizing energy efficiency without compromising accuracy. In *2025 11th International Conference on ICT for Sustainability (ICT4S)*, pp. 100–111, Dublin, Ireland, June 2025. IEEE. doi: 10.1109/ICT4S68164.2025.00019. URL https://doi.org/10.1109/ICT4S68164.2025.00019.

Mohan Rajashekarappa, Ebru Turanoglu Bekar, Alexander Karlsson, Jon Bokrantz, Mukund Subramaniyan, and Anders Skoogh. Industrial MLOps: A systematic review of architectures and implementation challenges. *Production & Manufacturing Research*, 14(1):2658878, 2026. doi: 10.1080/21693277.2026.2658878. URL https://doi.org/10.1080/21693277.2026.2658878.

Yeonju Ro, Zhangyang Wang, Vijay Chidambaram, and Aditya Akella. Lowering the pre-training tax for gradient-based subset training: A lightweight distributed pre-training toolkit. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 29130–29142. PMLR, July 2023. URL https://proceedings.mlr.press/v202/ro23a.html.

Atanu Roy, Sabyasachi Pramanik, Kalyan Mitra, and Manashi Chakraborty. Emission reduction in gas turbine: Improving CO and NOx emission prediction using modified CNN-Bi-LSTM extrinsic attention regressor. *Environment, Development and Sustainability*, 2026. doi: 10.1007/s10668-025-07209-w. URL https://doi.org/10.1007/s10668-025-07209-w. Online first.

Haoru Tan, Sitong Wu, Wei Huang, Shizhen Zhao, and Xiaojuan Qi. Data pruning by information maximization. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/7d848891e365ca623dc8352db9782585-Abstract-Conference.html.

Rudy Winarto, Wiwik Anggraeni, and Mauridhi Hery Purnomo. Industrial gas turbine emissions prediction using data-driven hybrid stacked ensemble machine under dynamic operating conditions. *Egyptian Informatics Journal*, 35:101041, 2026. doi: 10.1016/j.eij.2026.101041. URL https://doi.org/10.1016/j.eij.2026.101041.

Shuo Yang, Zeke Xie, Hanyu Peng, Min Xu, Mingming Sun, and Ping Li. Dataset pruning: Reducing training data by examining generalization influence. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=4wZiAXD29TQ.

Qin Zhang, Sufang Wen, Shunli Wang, Yuanru Zou, Shaoqing Chen, Carlos Fernandez, and Frede Blaabjerg. Improved hawk difference fusion: Hybrid neural network modeling for state of charge estimation of lithium-ion batteries adaptive to complex temperature conditions. *Electrical Engineering*, 108(7):311, 2026. doi: 10.1007/s00202-026-03662-6. URL https://doi.org/10.1007/s00202-026-03662-6.

Haizhong Zheng, Rui Liu, Fan Lai, and Atul Prakash. Coverage-centric coreset selection for high pruning rates. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.15809.

Table 5: Qualitative positioning of the proposed framework relative to closely related data-selection methods.

Method	Geometric support and coverage	Task-dependent signal	Input-space mass accounting	Residual correction	Condition-specific support and re-entry control
GradMatch	–	✓	–	–	–
CCS	✓	✓	–	–	–
BWS	–	✓	–	–	–
CRAIG	–	✓	–	–	–
SELCON	–	✓	–	–	–
Proposed	✓	✓	✓	✓	✓

A Qualitative Comparison with Related Methods

Table 5 positions the proposed framework relative to closely related data-selection methods. A checkmark indicates that the corresponding property is an explicit design objective or mechanism of the method. A dash indicates that the property is not explicitly formulated in the cited method; it does not imply that the method cannot be adapted to provide that property. Input-space mass accounting refers specifically to weights obtained from assignments to input-space representatives and excludes gradient-space multiplicities used for gradient approximation.

B Hardware and Software Environment

The reported experiments were executed on a Windows 11 workstation equipped with an Intel(R) Core(TM) i9-14900HX CPU, 31.8 GiB of RAM, and an NVIDIA GeForce RTX 4060 Laptop GPU with 8.0 GiB of GPU memory. The software environment used Python 3.8.10, PyTorch 2.4.1+cu124, CUDA 12.4, and NVIDIA driver 560.94. All runs used the same hardware and software environment unless otherwise stated.

C Training Protocol and Epoch Configuration

The executed training protocol is summarized here to clarify the entries in Tables 6 and 7. The procedure uses up to three sequential stages: full-data warm-up, initial weighted-support training, and feedback-guided re-entry training.

When $E_w^{\max} > 0$, Stage I runs for its complete epoch budget without early stopping. Validation MAE is evaluated after every epoch, and the checkpoint with the lowest validation MAE initializes Stage II. When $E_w^{\max} = 0$, Stage I is skipped and the paired model initialization is passed directly to Stage II. Stages II and III use validation-MAE-based early stopping with effective patience

$$P_q^{\text{eff}} = \min\left(P_q, E_q^{\max}\right), \qquad q \in \{s, r\}. \tag{29}$$

The best checkpoint from each executed stage initializes the next stage or is used for final evaluation. If Stage III has zero budget, produces no new samples, or is disabled for a particular setting, the Stage-II checkpoint remains the final model. The tables report the setting-specific maximum budgets, effective patience values, and stage-local best epochs over the paired model-initialization seeds $\{42, 2026, 3026\}$.

The configurations reflect the different convergence behavior of the architectures and tasks. Full-data warm-up is omitted for the battery CNN and FCN settings but retained for the battery MLP and gas-turbine settings. The selected stage-local epochs vary across conditions and seeds, showing that the final checkpoint is not determined solely by the maximum training budget. Feedback-corrected Stage III is omitted only for the gas-turbine FCN settings from 2013 and 2014.

All full-data, proposed, and baseline models use the same predictor architecture within each task and architecture family. The only difference between methods is the training support, sample weighting, and sample selection procedure. Battery SOC models use a window of five time steps and three input variables per time step. Gas-turbine observations are non-temporal and contain nine input variables; they are represented with $T = 1$ for compatibility with the convolutional implementations.

Table 6: Per-setting maximum epoch budgets and effective early-stopping patience for the proposed method. Stage I runs for its complete warm-up budget without early stopping. For Stages II and III, the reported effective patience is the smaller of the configured patience and the corresponding maximum epoch budget.

Task	Model	Condition	Stage I budget $E_w^{\max}$	Stage II budget $E_s^{\max}$	Stage II patience	Stage III budget $E_r^{\max}$	Stage III patience
battery SOC	CNN	HWFET	0	20	20	10	10
battery SOC	CNN	LA92	0	20	20	10	10
battery SOC	CNN	UDDS	0	20	20	10	10
battery SOC	CNN	US06	0	20	20	10	10
battery SOC	FCN	HWFET	0	3	3	3	3
battery SOC	FCN	LA92	0	3	3	3	3
battery SOC	FCN	UDDS	0	3	3	3	3
battery SOC	FCN	US06	0	10	10	20	20
battery SOC	MLP	HWFET	30	20	20	20	20
battery SOC	MLP	LA92	30	20	20	20	20
battery SOC	MLP	UDDS	30	20	20	20	20
battery SOC	MLP	US06	30	20	20	20	20
gas turbine NO_x	CNN	2013	200	200	50	200	20
gas turbine NO_x	CNN	2014	200	200	50	200	20
gas turbine NO_x	CNN	2015	150	200	30	200	20
gas turbine NO_x	FCN	2013	300	10	10	0	—
gas turbine NO_x	FCN	2014	100	10	10	0	—
gas turbine NO_x	FCN	2015	300	200	20	200	20
gas turbine NO_x	MLP	2013	20	100	10	100	10
gas turbine NO_x	MLP	2014	20	100	10	100	10
gas turbine NO_x	MLP	2015	20	100	10	100	10

Notes. A dash in Stage III indicates that feedback-corrected re-entry was not executed.

Table 7: Per-setting maximum epoch budgets and stage-local best epochs, continued.

Task	Model	Condition	Stage I best epoch	Stage II best epoch	Stage III best epoch
battery SOC	CNN	HWFET	—	12–16	1–4
battery SOC	CNN	LA92	—	12–18	4–6
battery SOC	CNN	UDDS	—	3–15	4–8
battery SOC	CNN	US06	—	15–18	1–3
battery SOC	FCN	HWFET	—	3	1–3
battery SOC	FCN	LA92	—	3	2
battery SOC	FCN	UDDS	—	1–3	2–3
battery SOC	FCN	US06	—	6–7	6–14
battery SOC	MLP	HWFET	25–29	9–10	15–20
battery SOC	MLP	LA92	26–29	1–3	2–13
battery SOC	MLP	UDDS	25–26	5	14–18
battery SOC	MLP	US06	27–28	4–11	4–13
gas turbine NO_x	CNN	2013	182–199	3–24	4–28
gas turbine NO_x	CNN	2014	155–183	7–24	1–5
gas turbine NO_x	CNN	2015	131–141	2–19	2–12
gas turbine NO_x	FCN	2013	272–296	4–10	—
gas turbine NO_x	FCN	2014	92–99	2–9	—
gas turbine NO_x	FCN	2015	295	2–12	3–12
gas turbine NO_x	MLP	2013	20	22	1–5
gas turbine NO_x	MLP	2014	20	1	1–3
gas turbine NO_x	MLP	2015	20	26–34	4–21

Notes. All best epochs are indexed locally within their corresponding stages rather than cumulatively across stages. A dash indicates that the corresponding stage was not executed.

For battery SOC estimation, the MLP flattens the 5×3 input window and uses fully connected layers $15 \to 64 \to 32 \to 1$. SELU activations are applied after the first two linear layers. The battery FCN uses one-dimensional convolutions with channel widths $3 \to 128 \to 256 \to 128$, kernel size 3, and padding 1. Each convolution is followed by batch normalization and ReLU, and a final 1×1 convolution maps the 128 channels to one output channel. Global average pooling produces

the scalar regression output. The battery CNN uses two convolutional blocks with channel widths $3 \to 128 \to 256$, kernel size 3, padding 1, batch normalization, ReLU activations, and max-pooling with kernel size 2 after each block. The resulting representation is flattened and passed through fully connected layers $256 \to 128 \to 1$, with a ReLU activation between the two linear layers.

For gas-turbine NO_x prediction, the MLP uses the nine-dimensional input vector followed by fully connected layers $9 \to 128 \to 64 \to 1$, with SELU activations after the first two linear layers. The gas-turbine FCN treats the nine features as a one-dimensional input with one input channel and uses convolutional channel widths $1 \to 128 \to 256 \to 128$, kernel size 3, and padding 1. Batch normalization and ReLU follow each convolution, and a final 1×1 convolution followed by global average pooling produces the scalar output. The gas-turbine CNN also uses one input channel, two convolutional blocks with channel widths $1 \to 128 \to 256$, kernel size 3, padding 1, batch normalization, ReLU activations, and max-pooling with kernel size 2. The pooled representation is flattened and passed through fully connected layers $512 \to 128 \to 1$, because the nine-feature input has lengths $9 \to 4 \to 2$ after the two pooling operations.

All models are optimized with Adam using a learning rate of 10^{-4} and a mini-batch size of 32. The Huber loss is used for training and for the per-sample feedback score. Its threshold is fixed to $\delta = 1$ in normalized target space:

$$\ell_\delta(\widehat{y}, y) = \begin{cases} \frac{1}{2}(\widehat{y} - y)^2, & |\widehat{y} - y| \le \delta, \\ \delta\left(|\widehat{y} - y| - \frac{1}{2}\delta\right), & |\widehat{y} - y| > \delta, \end{cases} \qquad \delta = 1. \tag{30}$$

The same fixed value of δ is used in full-data warm-up, Stage-II weighted-support training, Stage-III feedback-corrected training, and feedback-score computation. Input and target normalization statistics are fitted using the training portion only and then held fixed for validation and test evaluation. Gas-turbine predictions are inverse-transformed to the original NO_x scale before reporting MAE, MSE, and RMSE.

The data-split, selection, and training-order seeds are fixed at 42. The paired model-initialization seeds are $\{42, 2026, 3026\}$. All methods within a given setting use the same initialization seeds, data partitions, optimizer configuration, epoch budgets, checkpoint-selection rule, and applicable early-stopping settings.

D Per-seed Accuracy and Computational Results

This appendix expands the setting-level summaries reported in the main text into per-seed results. Six tables are provided. Tables 8, 9, and 10 report the MLP, FCN, and CNN results for battery SOC estimation, respectively. Tables 11, 12, and 13 report the corresponding gas-turbine NO_x results. The battery tables cover the HWFET, LA92, UDDS, and US06 driving cycles, whereas the gas-turbine tables cover the 2013, 2014, and 2015 records.

Each row corresponds to one model-initialization seed $\{42, 2026, 3026\}$. The reported columns include the final training-data retention ratio γ, the seed-matched full-data and proposed-model MAEs, the absolute and relative MAE changes, the raw analytical cloud-side FLOPs, and the signed FLOP change Δ_{FLOPs}. Accuracy changes are computed as proposed minus full-data, so a positive value indicates higher prediction error and a negative value indicates lower prediction error. The *Mean* row in each table reports the arithmetic mean of the three displayed seed-level values for every numerical column. Percentage changes are averaged directly and are not recomputed from the mean MAEs or mean FLOPs.

The full-data reference is trained independently using the same model-initialization seed as the corresponding proposed run. The reported FLOPs follow the analytical accounting scope defined in Section 4. They include the explicitly modeled MiniBatch K-means and reachability-based support-construction operations, together with all counted training and validation model passes, including Stage III when executed. They exclude full-pool feedback inference, diagnostic validation passes, representative reassignment through cKDTree, representative-frequency weight recomputation, optimizer updates, data movement, checkpoint I/O, and other non-modeled operations. The FLOP unit for each table is stated in its caption. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost.

Table 8: Per-seed accuracy and accounted analytical FLOPs for MLP-based battery SOC estimation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{11} and follow the accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each cycle, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Cycle	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
HWFET	42	40.4	0.04221	0.04594	+0.00373	+8.83	3.9270	2.0566	−47.63
HWFET	2026	40.4	0.04365	0.04242	−0.00123	−2.81	3.9270	2.0566	−47.63
HWFET	3026	40.4	0.04497	0.04667	+0.00170	+3.79	4.3634	2.0566	−52.87
HWFET	*Mean*	40.4	0.04361	0.04501	+0.00140	+3.27	4.0725	2.0566	−49.38
LA92	42	39.2	0.03122	0.03002	−0.00120	−3.84	1.7568	1.7824	+1.45
LA92	2026	39.2	0.03257	0.03005	−0.00252	−7.74	1.8714	1.7824	−4.76
LA92	3026	39.2	0.03096	0.03006	−0.00090	−2.91	2.6734	1.7824	−33.33
LA92	*Mean*	39.2	0.03158	0.03004	−0.00154	−4.83	2.1005	1.7824	−12.21
UDDS	42	39.7	0.02477	0.02395	−0.00081	−3.29	1.5483	1.3192	−14.80
UDDS	2026	39.7	0.02412	0.02215	−0.00198	−8.19	2.8152	1.3192	−53.14
UDDS	3026	39.7	0.02348	0.02596	+0.00248	+10.57	2.1677	1.3192	−39.14
UDDS	*Mean*	39.7	0.02412	0.02402	−0.00010	−0.30	2.1771	1.3192	−35.69
US06	42	41.4	0.11393	0.09940	−0.01453	−12.75	2.1780	2.2001	+1.01
US06	2026	41.4	0.07672	0.07316	−0.00357	−4.65	3.0122	2.2001	−26.96
US06	3026	41.4	0.05968	0.05650	−0.00318	−5.32	2.1780	2.2001	+1.01
US06	*Mean*	41.4	0.08344	0.07635	−0.00709	−7.57	2.4561	2.2001	−8.31

Table 9: Per-seed accuracy and accounted analytical FLOPs for FCN-based battery SOC estimation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{13} and follow the accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each cycle, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Cycle	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
HWFET	42	40.4	0.05161	0.04843	−0.00318	−6.16	3.0504	0.3399	−88.86
HWFET	2026	40.4	0.04837	0.05007	+0.00170	+3.52	2.7852	0.3399	−87.79
HWFET	3026	40.4	0.04550	0.04752	+0.00202	+4.45	2.7852	0.3399	−87.79
HWFET	*Mean*	40.4	0.04849	0.04867	+0.00018	+0.60	2.8736	0.3399	−88.15
LA92	42	39.2	0.03108	0.03327	+0.00219	+7.05	2.6700	0.2895	−89.16
LA92	2026	39.2	0.03210	0.03337	+0.00127	+3.96	2.6700	0.2895	−89.16
LA92	3026	39.2	0.03163	0.03074	−0.00088	−2.79	2.6700	0.2895	−89.16
LA92	*Mean*	39.2	0.03160	0.03246	+0.00086	+2.74	2.6700	0.2895	−89.16
UDDS	42	39.7	0.02795	0.03035	+0.00239	+8.55	1.9681	0.2159	−89.03
UDDS	2026	39.7	0.02614	0.03118	+0.00504	+19.29	1.9681	0.2159	−89.03
UDDS	3026	39.7	0.02709	0.03059	+0.00350	+12.92	1.9681	0.2159	−89.03
UDDS	*Mean*	39.7	0.02706	0.03071	+0.00364	+13.59	1.9681	0.2159	−89.03
US06	42	41.4	0.08010	0.07766	−0.00244	−3.05	5.9159	1.8744	−68.32
US06	2026	41.4	0.08499	0.08282	−0.00217	−2.55	5.9159	1.8744	−68.32
US06	3026	41.4	0.07862	0.08471	+0.00608	+7.74	7.1836	1.8744	−73.91
US06	*Mean*	41.4	0.08124	0.08173	+0.00049	+0.71	6.3385	1.8744	−70.18

D.1 Battery SOC results

For the battery MLP, the proposed method increases the mean MAE on HWFET by $+0.00140$, but reduces the mean MAE on LA92, UDDS, and US06 by -0.00154, -0.00010, and -0.00709, respectively. The corresponding accounted analytical FLOPs are lower than those of the independently trained full-data references in all four cycles.

For the battery FCN, the proposed method produces small mean MAE increases on all four cycles. The absolute changes are $+0.00018$, $+0.00086$, $+0.00364$, and $+0.00049$ for HWFET, LA92, UDDS, and US06, respectively. Despite these accuracy increases, the FCN configuration provides the largest computational reductions among the battery architectures.

Table 10: Per-seed accuracy and accounted analytical FLOPs for CNN-based battery SOC estimation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{13} and follow the accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each cycle, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Cycle	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
HWFET	42	40.4	0.06170	0.05439	−0.00732	−11.86	0.7315	0.4002	−45.28
HWFET	2026	40.4	0.05078	0.04689	−0.00389	−7.65	0.6679	0.4002	−40.07
HWFET	3026	40.4	0.05064	0.05300	+0.00236	+4.66	0.7633	0.4002	−47.56
HWFET	*Mean*	40.4	0.05437	0.05143	−0.00295	−4.95	0.7209	0.4002	−44.30
LA92	42	39.2	0.03081	0.03429	+0.00349	+11.31	0.5846	0.3407	−41.72
LA92	2026	39.2	0.03559	0.03343	−0.00216	−6.08	0.6402	0.3407	−46.79
LA92	3026	39.2	0.03545	0.03074	−0.00472	−13.31	0.6402	0.3407	−46.79
LA92	*Mean*	39.2	0.03395	0.03282	−0.00113	−2.69	0.6217	0.3407	−45.10
UDDS	42	39.7	0.02359	0.02716	+0.00357	+15.14	0.7182	0.2541	−64.62
UDDS	2026	39.7	0.02365	0.02748	+0.00382	+16.17	0.7182	0.2541	−64.62
UDDS	3026	39.7	0.02684	0.03279	+0.00596	+22.19	0.7182	0.2541	−64.62
UDDS	*Mean*	39.7	0.02469	0.02914	+0.00445	+17.83	0.7182	0.2541	−64.62
US06	42	41.4	0.07482	0.06718	−0.00764	−10.21	1.7226	0.4339	−74.81
US06	2026	41.4	0.08898	0.07727	−0.01171	−13.16	0.7769	0.4339	−44.15
US06	3026	41.4	0.07938	0.07490	−0.00448	−5.65	0.7769	0.4339	−44.15
US06	*Mean*	41.4	0.08106	0.07312	−0.00794	−9.67	1.0921	0.4339	−54.37

Table 11: Per-seed accuracy and accounted analytical FLOPs for MLP-based gas-turbine NO_x prediction. MAE and absolute changes are reported in $\mathrm{mg/m^3}$ on the original NO_x scale after inverse transformation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{10} and follow the analytical accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each year, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Year	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
2013	42	41.7	4.53199	4.58641	+0.05442	+1.20	6.5058	1.2574	−80.67
2013	2026	41.7	4.54331	4.58873	+0.04542	+1.00	6.5058	1.3183	−79.74
2013	3026	41.7	4.47741	4.53223	+0.05483	+1.22	6.5058	1.2574	−80.67
2013	*Mean*	41.7	4.51757	4.56912	+0.05156	+1.14	6.5058	1.2777	−80.36
2014	42	38.7	3.97301	4.23043	+0.25742	+6.48	6.5114	0.9509	−85.40
2014	2026	38.7	4.04650	4.34447	+0.29797	+7.36	6.5114	0.9796	−84.96
2014	3026	38.7	4.06796	4.32012	+0.25216	+6.20	6.5114	0.9796	−84.96
2014	*Mean*	38.7	4.02916	4.29834	+0.26918	+6.68	6.5114	0.9700	−85.11
2015	42	39.4	3.14929	3.15012	+0.00083	+0.03	6.7167	1.4741	−78.05
2015	2026	39.4	3.18693	3.12254	−0.06439	−2.02	6.7167	1.7289	−74.26
2015	3026	39.4	3.19928	3.17710	−0.02218	−0.69	6.7167	1.5613	−76.75
2015	*Mean*	39.4	3.17850	3.14992	−0.02858	−0.89	6.7167	1.5881	−76.35

For the battery CNN, the proposed method reduces mean MAE on HWFET, LA92, and US06 by -0.00295, -0.00113, and -0.00794, respectively, while the mean MAE on UDDS increases by $+0.00445$. These results show that the accuracy effect of sparsification depends on the driving cycle and is not uniform across architectures.

D.2 Gas-turbine NO_x results

For the gas-turbine MLP, the proposed method increases the mean MAE on the 2013 and 2014 records by $+0.05156$ and $+0.26918\ \mathrm{mg/m^3}$, respectively. It slightly reduces the mean MAE on the 2015 record by $-0.02858\ \mathrm{mg/m^3}$. The MLP nevertheless reduces the accounted analytical FLOPs for all three years.

Table 12: Per-seed accuracy and accounted analytical FLOPs for FCN-based gas-turbine NO_x prediction. MAE and absolute changes are reported in $\mathrm{mg/m^3}$ on the original NO_x scale after inverse transformation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{13} and follow the analytical accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each year, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Year	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
2013	42	41.7	3.11262	3.31829	+0.20567	+6.61	2.9349	1.7893	−39.03
2013	2026	41.7	3.22586	3.24738	+0.02152	+0.67	2.1056	1.7893	−15.02
2013	3026	41.7	3.16669	3.34300	+0.17632	+5.57	2.4291	1.7893	−26.34
2013	*Mean*	41.7	3.16839	3.30289	+0.13450	+4.28	2.4899	1.7893	−26.80
2014	42	38.7	2.19740	2.40854	+0.21115	+9.61	1.6659	0.6119	−63.27
2014	2026	38.7	2.14776	2.40951	+0.26175	+12.19	1.6541	0.6119	−63.01
2014	3026	38.7	2.24437	2.38521	+0.14084	+6.28	1.1596	0.6119	−47.23
2014	*Mean*	38.7	2.19651	2.40109	+0.20458	+9.36	1.4932	0.6119	−57.84
2015	42	39.4	2.02019	2.20313	+0.18294	+9.06	2.2528	1.9618	−12.92
2015	2026	39.4	2.03183	2.25600	+0.22417	+11.03	2.6414	1.9619	−25.73
2015	3026	39.4	2.02866	2.25821	+0.22955	+11.32	2.5017	1.9672	−21.36
2015	*Mean*	39.4	2.02689	2.23911	+0.21222	+10.47	2.4653	1.9636	−20.00

Table 13: Per-seed accuracy and accounted analytical FLOPs for CNN-based gas-turbine NO_x prediction. MAE and absolute changes are reported in $\mathrm{mg/m^3}$ on the original NO_x scale after inverse transformation. Full-data and proposed runs are paired by model-initialization seed, and γ denotes the final retention ratio. Accuracy changes are computed as proposed minus full-data, so positive values indicate higher prediction error. FLOPs are reported in units of 10^{12} and follow the analytical accounting scope defined in Section 4. Negative Δ_{FLOPs} values indicate computational savings, whereas positive values indicate additional accounted cost. For each year, the *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column; percentage changes are averaged directly.

Year	Seed	γ (%)	Full MAE	Proposed MAE	Δ_{abs}	Δ_{rel} (%)	Full FLOPs	Proposed FLOPs	Δ_{FLOPs} (%)
2013	42	41.7	3.14699	3.32367	+0.17668	+5.61	5.2821	3.6532	−30.84
2013	2026	41.7	3.08359	3.36223	+0.27864	+9.04	5.5292	3.6726	−33.58
2013	3026	41.7	3.06556	3.31589	+0.25033	+8.17	5.2512	3.9177	−25.39
2013	*Mean*	41.7	3.09871	3.33393	+0.23522	+7.61	5.3542	3.7478	−29.94
2014	42	38.7	2.07525	2.28513	+0.20988	+10.11	5.3794	3.5895	−33.27
2014	2026	38.7	2.10855	2.32992	+0.22137	+10.50	6.0750	3.6865	−39.32
2014	3026	38.7	2.11988	2.49668	+0.37680	+17.77	6.7706	3.6954	−45.42
2014	*Mean*	38.7	2.10123	2.37058	+0.26935	+12.79	6.0750	3.6571	−39.34
2015	42	39.4	2.12002	2.30893	+0.18891	+8.91	3.7950	2.8026	−26.15
2015	2026	39.4	2.07576	2.30209	+0.22633	+10.90	4.2574	2.7528	−35.34
2015	3026	39.4	2.04649	2.46929	+0.42280	+20.66	6.0752	2.9325	−51.73
2015	*Mean*	39.4	2.08076	2.36010	+0.27935	+13.49	4.7092	2.8293	−37.74

For the gas-turbine FCN, the mean MAE increases by $+0.13450$, $+0.20458$, and $+0.21222\ \mathrm{mg/m^3}$ for 2013, 2014, and 2015, respectively. The accounted analytical FLOPs remain lower than the full-data reference in every year, although the magnitude of the reduction depends on the year-specific training budget.

For the gas-turbine CNN, the mean MAE increases by $+0.23522$, $+0.26935$, and $+0.27935\ \mathrm{mg/m^3}$ for 2013, 2014, and 2015, respectively. The CNN therefore exhibits a clearer accuracy–compute trade-off than the gas-turbine MLP, while still reducing the accounted analytical FLOPs in all evaluated years.

Overall, the per-seed results confirm that the proposed framework consistently reduces the modeled analytical cloud-side computation, whereas its effect on prediction accuracy depends on the physical task, operating condition, architecture, and model-initialization seed. The setting-level ranges

Table 14: Per-seed exact-budget comparison for MLP-based battery SOC estimation. The symbol γ denotes the final training-data retention ratio. The reported analytical cloud-side FLOPs follow the accounting scope defined in Section 4 and are expressed in units of 10^{11}. Random is a selection-only control and is compared using MAE. Analytical cloud-side FLOPs are reported separately for Proposed and GradMatch. Each *Mean* row reports the arithmetic mean of the three displayed seed-level values in each numerical column. Lower MAE and fewer accounted FLOPs are better.

Cycle	Seed	γ (%)	Random MAE	GradMatch MAE	Proposed MAE	Proposed FLOPs	GradMatch FLOPs
HWFET	42	40.4	0.04490	0.04569	0.04594	2.0566	2.4513
HWFET	2026	40.4	0.04483	0.04358	0.04242	2.0566	2.4513
HWFET	3026	40.4	0.04388	0.04794	0.04667	2.0566	2.4513
HWFET	*Mean*	40.4	0.04454	0.04574	0.04501	2.0566	2.4513
LA92	42	39.2	0.03176	0.03044	0.03002	1.7824	2.1186
LA92	2026	39.2	0.03093	0.02984	0.03005	1.7824	2.1186
LA92	3026	39.2	0.03366	0.03028	0.03006	1.7824	2.1186
LA92	*Mean*	39.2	0.03212	0.03019	0.03004	1.7824	2.1186
UDDS	42	39.7	0.02461	0.02332	0.02395	1.3192	1.4679
UDDS	2026	39.7	0.02391	0.02285	0.02215	1.3192	1.4679
UDDS	3026	39.7	0.02197	0.02239	0.02596	1.3192	1.4679
UDDS	*Mean*	39.7	0.02350	0.02285	0.02402	1.3192	1.4679
US06	42	41.4	0.12312	0.11426	0.09940	2.2001	2.6257
US06	2026	41.4	0.07639	0.07611	0.07316	2.2001	2.6257
US06	3026	41.4	0.06033	0.06236	0.05650	2.2001	2.6257
US06	*Mean*	41.4	0.08661	0.08424	0.07635	2.2001	2.6257

reported in the main text are obtained by averaging the three paired seed-level results shown in these tables.

E Baseline Study

This study compares the proposed framework with retention-matched Random selection and GradMatch in the battery MLP setting. All methods use the same data splits, predictor architecture, optimization configuration, model-initialization seeds, and condition-specific initial and final retention budgets. Random is a selection-only control that replaces the proposed sample-identity selection rules with budget-matched random selection while retaining the remaining pipeline components. GradMatch independently selects samples using batch-gradient representations and non-negative orthogonal matching pursuit weights, followed by a second selection step when the final budget exceeds the initial budget. In contrast, the proposed framework constructs a geometric support and uses prediction-error feedback to select samples for re-entry.

Table 14 reports the results for all four driving cycles, three paired model-initialization seeds $\{42, 2026, 3026\}$, and the corresponding arithmetic mean for each cycle. The table reports MAE for all three methods and the accounted analytical cloud-side FLOPs for Proposed and GradMatch. Random is used to isolate the effect of sample-identity selection under the same final support budgets and is therefore not assigned a separate FLOP total.

The proposed framework obtains lower mean MAE than Random on LA92 and US06, whereas Random obtains lower mean MAE on HWFET and UDDS. Relative to GradMatch, the proposed framework obtains lower mean MAE on HWFET, LA92, and US06, while GradMatch performs better on UDDS. These comparisons indicate condition-dependent accuracy benefits rather than uniform superiority.

The seed-level outcomes further reveal condition-dependent sensitivity to model initialization. Random achieves the lowest MAE for two of the three HWFET seeds, whereas the proposed framework performs best for seed 2026. On LA92, the proposed framework performs best for seeds 42 and 3026, while GradMatch performs best for seed 2026. On UDDS, Random, GradMatch, and the proposed framework each achieve the lowest MAE for one seed. In contrast, the proposed framework achieves the lowest MAE for all three US06 seeds.

The corresponding MAE variability is also condition-dependent. Across the three paired seeds, the proposed method has standard deviations of 0.00227, 0.00002, 0.00191, and 0.02162 on HWFET,

Table 15: Qualitative configuration analysis for the MLP battery ablation study. The table records the component switches, optimizer-state rule, checkpoint rule, and final retention range used by each configuration. It is a configuration table rather than a performance table. "Reset" indicates that a new Adam optimizer is initialized between executed training stages. "Global" selects the checkpoint with the lowest validation MAE across all executed stages, whereas "stage-local" selects the best checkpoint from the final executed stage. The first two configurations retain protected topology landmarks, whereas the re-entry configurations disable topology protection and use cardinality-matched ordinary replacements.

Configuration	Frequency weighting	Feedback re-entry	Topology protection	Matched replacement	Adam reset	Checkpoint	Final retention (%)
Unweighted	No	No	Yes	No	Yes	Stage-local	34.2–36.4
Weighted	Yes	No	Yes	No	Yes	Stage-local	34.2–36.4
Shared/local	Yes	Yes	No	Yes	No	Stage-local	39.2–41.4
Shared/global	Yes	Yes	No	Yes	No	Global	39.2–41.4
Reset/global	Yes	Yes	No	Yes	Yes	Global	39.2–41.4
Proposed	Yes	Yes	No	Yes	Yes	Stage-local	39.2–41.4

LA92, UDDS, and US06, respectively. These values are lower than both baselines on LA92 and US06, but not on HWFET or UDDS. Thus, the results suggest improved initialization robustness in selected operating conditions rather than a uniform stability advantage across all cycles. Because each setting uses three paired seeds, these observations should be interpreted as descriptive evidence of seed sensitivity rather than as a formal statistical test.

Under the implemented matched-budget protocol, Random is used only to isolate the accuracy effect of sample-identity selection. The computational comparison therefore focuses on the proposed framework and GradMatch. Relative to GradMatch, the proposed framework uses 16.1%, 15.9%, 10.1%, and 16.2% fewer accounted analytical FLOPs on HWFET, LA92, UDDS, and US06, respectively. These reductions are computed under the analytical accounting scope defined in Section 4 and do not represent direct wall-clock or energy measurements.

Overall, the proposed framework provides the lowest mean MAE among the three methods on two of the four cycles, achieves lower accounted analytical FLOPs than GradMatch on all four cycles, and exhibits lower seed-level MAE variability on LA92 and US06. The results support a condition-dependent accuracy-stability-computation trade-off rather than a universal advantage over every baseline and operating condition.

F Ablation Study

The ablation analysis contains two complementary parts. First, Table 15 provides a qualitative configuration analysis of the design choices used in the battery MLP experiments. This table does not report prediction performance; instead, it specifies which components are enabled or disabled in each configuration and clarifies the differences between the evaluated variants. Second, Table 16 reports the corresponding quantitative per-seed test MAEs.

The Unweighted and Weighted configurations use the same geometric support, topology protection, final retention ratio, optimizer-reset rule, and stage-local checkpoint rule. Their only difference is whether representative-frequency weights are applied. Their comparison therefore isolates the effect of weighting. The Shared/local configuration enables feedback-guided re-entry and matched ordinary replacement, while inheriting the Adam state between stages and selecting the best checkpoint within the final stage. The Shared/global configuration keeps the same re-entry and optimizer-state rules but selects the best checkpoint across all executed stages. The Reset/global configuration further resets the Adam optimizer between stages while retaining global checkpoint selection. The proposed configuration uses the same re-entry and optimizer-reset rules as Reset/global, but applies stage-local checkpoint selection.

The transition from the non-re-entry configurations to the re-entry configurations changes more than feedback scoring alone. Specifically, it also disables topology protection and introduces cardinality-matched ordinary replacements. Therefore, comparisons between Weighted and the re-entry variants should be interpreted as configuration-level comparisons, rather than as isolated estimates of the

Table 16: Per-seed quantitative test MAE for the MLP battery ablation study. Only the model-initialization seed varies; the pipeline, data-selection, and training-order seeds are fixed and shared across configurations. The Unweighted and Weighted configurations use identical retained supports, so their comparison isolates representative-frequency weighting. The re-entry configurations use matched final retention budgets and differ in optimizer-state handling and checkpoint selection. For each cycle, the *Mean* row reports the arithmetic mean of the three displayed seed-level MAEs. Lower MAE is better.

Cycle	Seed	Unweighted	Weighted	Shared Adam global CKPT	Reset Adam global CKPT	Proposed
HWFET	42	0.04628	0.04359	0.04532	0.04533	0.04594
HWFET	2026	0.04819	0.04489	0.04500	0.04478	0.04242
HWFET	3026	0.04662	0.04405	0.04914	0.04912	0.04667
HWFET	*Mean*	0.04703	0.04418	0.04649	0.04641	0.04501
LA92	42	0.02954	0.03179	0.02955	0.02959	0.03002
LA92	2026	0.03250	0.03096	0.03286	0.03298	0.03005
LA92	3026	0.03098	0.03238	0.03248	0.03122	0.03006
LA92	*Mean*	0.03101	0.03171	0.03163	0.03126	0.03004
UDDS	42	0.02416	0.03622	0.03359	0.02395	0.02395
UDDS	2026	0.02473	0.02276	0.02516	0.02648	0.02215
UDDS	3026	0.02428	0.02260	0.02343	0.02596	0.02596
UDDS	*Mean*	0.02439	0.02719	0.02739	0.02546	0.02402
US06	42	0.14698	0.10796	0.11393	0.11393	0.09940
US06	2026	0.07560	0.07308	0.06890	0.06890	0.07316
US06	3026	0.05949	0.05819	0.05968	0.05968	0.05650
US06	*Mean*	0.09402	0.07974	0.08084	0.08084	0.07635

feedback component. The controlled one-factor comparisons are the Unweighted versus Weighted pair, the Shared/local versus Shared/global pair, the Shared/global versus Reset/global pair, and the eset/global versus Proposed pair.

The quantitative results show that representative-frequency weighting has condition-dependent effects. Relative to the unweighted configuration, weighting decreases the mean MAE on HWFET from 0.04703 to 0.04418 and on US06 from 0.09402 to 0.07974. However, it increases the mean MAE on LA92 from 0.03101 to 0.03171 and on UDDS from 0.02439 to 0.02719. Thus, frequency weighting improves the representation of omitted samples in some conditions but does not provide a uniform improvement across all operating conditions.

Among the re-entry configurations reported in the per-seed table, the proposed reset-Adam and stage-local checkpoint design obtains the lowest mean MAE on all four conditions. Its mean MAE is 0.04501, 0.03004, 0.02402, and 0.07635 on HWFET, LA92, UDDS, and US06, respectively. These values are lower than those of both global-checkpoint variants on every condition. Considering all configurations, the proposed method obtains the lowest mean MAE on LA92, UDDS, and US06, whereas the Weighted configuration remains slightly better on HWFET with a mean MAE of 0.04418 compared with 0.04501 for the proposed method.

The seed-level results also reveal several high-error observations. For the Weighted configuration, the UDDS result for seed 42 is 0.03622, compared with 0.02276 and 0.02260 for seeds 2026 and 3026. The same seed also produces a relatively high US06 MAE of 0.10796. The Unweighted configuration has an even larger US06 error for seed 42 (0.14698). In addition, the Shared/global and Reset/global configurations produce elevated HWFET errors for seed 3026, with MAEs of 0.04914 and 0.04912, respectively.

These observations should be interpreted as seed-level sensitivity rather than as confirmed training failures, because only three paired seeds are evaluated and the available summaries do not diagnose the cause of each high-error run. Nevertheless, the proposed configuration reduces the largest observed US06 error to 0.09940 and achieves the lowest MAE for all three US06 seeds. It also provides the best mean performance among the evaluated re-entry variants on every condition. The results therefore support a conditional stability and accuracy advantage for the complete configuration, rather than uniform superiority for every seed and every operating condition.

Table 17: Per-seed retention-budget results for HWFET. FLOPs are reported in units of 10^{11}. A dagger marks a retention ratio on the Pareto frontier determined from setting-level averages over the three seeds.

γ (%)	Seed	Test MAE	Test MSE	Test RMSE	Full FLOPs	Sparse FLOPs	Δ_{FLOPs} (%)
$20^{\dagger}$	42	0.048482	0.003213	0.056684	3.9270	1.727225	−56.02
$20^{\dagger}$	2026	0.047460	0.003183	0.056418	3.9270	1.727225	−56.02
$20^{\dagger}$	3026	0.047831	0.003295	0.057401	4.3634	1.727225	−60.42
$40^{\dagger}$	42	0.044006	0.002831	0.053205	3.9270	2.049452	−47.81
$40^{\dagger}$	2026	0.044199	0.002903	0.053880	3.9270	2.049452	−47.81
$40^{\dagger}$	3026	0.051222	0.003670	0.060582	4.3634	2.049452	−53.03
60	42	0.047789	0.003160	0.056211	3.9270	2.371670	−39.61
60	2026	0.048101	0.003291	0.057371	3.9270	2.371670	−39.61
60	3026	0.046526	0.003027	0.055023	4.3634	2.371670	−45.65
$80^{\dagger}$	42	0.044894	0.002919	0.054030	3.9270	2.693897	−31.40
$80^{\dagger}$	2026	0.046827	0.003160	0.056212	3.9270	2.693897	−31.40
$80^{\dagger}$	3026	0.046877	0.003203	0.056596	4.3634	2.693897	−38.26

Table 18: Per-seed retention-budget results for LA92. FLOPs are reported in units of 10^{11}. A dagger marks a retention ratio on the Pareto frontier determined from setting-level averages over the three seeds.

γ (%)	Seed	Test MAE	Test MSE	Test RMSE	Full FLOPs	Sparse FLOPs	Δ_{FLOPs} (%)
$20^{\dagger}$	42	0.031220	0.001682	0.041012	1.7568	1.511679	−13.95
$20^{\dagger}$	2026	0.032568	0.001738	0.041689	1.8714	1.511679	−19.22
$20^{\dagger}$	3026	0.031220	0.001614	0.040177	2.6734	1.511679	−43.45
$40^{\dagger}$	42	0.030635	0.001636	0.040443	1.7568	1.793720	+2.10
$40^{\dagger}$	2026	0.032568	0.001738	0.041689	1.8714	1.793720	−4.15
$40^{\dagger}$	3026	0.031220	0.001614	0.040177	2.6734	1.793720	−32.90
$60^{\dagger}$	42	0.031220	0.001682	0.041012	1.7568	2.075754	+18.16
$60^{\dagger}$	2026	0.031822	0.001645	0.040554	1.8714	2.075754	+10.92
$60^{\dagger}$	3026	0.031220	0.001614	0.040177	2.6734	2.075754	−22.36
$80^{\dagger}$	42	0.031220	0.001682	0.041012	1.7568	2.357796	+34.21
$80^{\dagger}$	2026	0.032568	0.001738	0.041689	1.8714	2.357796	+25.99
$80^{\dagger}$	3026	0.030360	0.001561	0.039507	2.6734	2.357796	−11.81

G PARETO ANALYSIS

The retention-budget analysis evaluates the MLP battery setting at four condition-specific final retention ratios, $\gamma \in \{20\%, 40\%, 60\%, 80\}$. The per-seed results are reported in four tables: Table 17 for HWFET, Table 18 for LA92, Table 19 for UDDS, and Table 20 for US06. Each table reports test MAE, test MSE, test RMSE, full-data FLOPs, sparse-model FLOPs, and the signed FLOP change for seeds $\{42, 2026, 3026\}$.

For each $\gamma < 100\%$, five percentage points of the final budget are reserved for prediction-error-guided re-entry. The resulting support therefore matches the prescribed condition-specific final retention ratio. Each sparse configuration is compared with an independently trained full-data model using the same operating condition and model-initialization seed. The full-data model is used as the reference operating point and is not treated as a within-pipeline 100% retention configuration. No separate sparse 100%-retention run is included.

The signed FLOP change is computed according to Eq. 28. Negative values indicate fewer accounted analytical FLOPs than the paired full-data model, whereas positive values indicate additional accounted cost. All tables use the same analytical accounting scope as Section 4. The accounting includes explicitly modeled support-construction operations and counted training and validation model passes. Full-pool feedback inference, sample reassignment, representative- frequency weight recomputation, optimizer updates, data movement, and checkpoint I/O are excluded.

Pareto efficiency is determined from the setting-level averages over the three seeds using test MAE and accounted analytical FLOPs. The full-data reference is included as a candidate operating point.

Table 19: Per-seed retention-budget results for UDDS. FLOPs are reported in units of 10^{11}. A dagger marks a retention ratio on the Pareto frontier determined from setting-level averages over the three seeds.

γ (%)	Seed	Test MAE	Test MSE	Test RMSE	Full FLOPs	Sparse FLOPs	Δ_{FLOPs} (%)
$20^{\dagger}$	42	0.031558	0.001566	0.039574	1.5483	1.114278	−28.03
$20^{\dagger}$	2026	0.036561	0.001989	0.044596	2.8152	1.114278	−60.42
$20^{\dagger}$	3026	0.031729	0.001621	0.040258	2.1677	1.114278	−48.60
$40^{\dagger}$	42	0.027502	0.001183	0.034390	1.5483	1.322178	−14.60
$40^{\dagger}$	2026	0.025714	0.001058	0.032520	2.8152	1.322178	−53.03
$40^{\dagger}$	3026	0.028450	0.001308	0.036169	2.1677	1.322178	−39.01
60	42	0.031025	0.001445	0.038013	1.5483	1.530070	−1.18
60	2026	0.028596	0.001255	0.035425	2.8152	1.530070	−45.65
60	3026	0.027519	0.001216	0.034877	2.1677	1.530070	−29.42
$80^{\dagger}$	42	0.025574	0.001062	0.032591	1.5483	1.737970	+12.25
$80^{\dagger}$	2026	0.026330	0.001070	0.032718	2.8152	1.737970	−38.26
$80^{\dagger}$	3026	0.022793	0.000879	0.029648	2.1677	1.737970	−19.82

Table 20: Per-seed retention-budget results for US06. FLOPs are reported in units of 10^{11}. A dagger marks a retention ratio on the Pareto frontier determined from setting-level averages over the three seeds.

γ (%)	Seed	Test MAE	Test MSE	Test RMSE	Full FLOPs	Sparse FLOPs	Δ_{FLOPs} (%)
$20^{\dagger}$	42	0.113926	0.017366	0.131780	2.1780	1.834136	−15.79
$20^{\dagger}$	2026	0.068903	0.007567	0.086989	3.0122	1.834136	−39.11
$20^{\dagger}$	3026	0.055478	0.005098	0.071401	2.1780	1.834136	−15.79
40	42	0.113926	0.017366	0.131780	2.1780	2.176345	−0.08
40	2026	0.068903	0.007567	0.086989	3.0122	2.176345	−27.75
40	3026	0.059679	0.005336	0.073048	2.1780	2.176345	−0.08
60	42	0.113926	0.017366	0.131780	2.1780	2.518569	+15.64
60	2026	0.068903	0.007567	0.086989	3.0122	2.518569	−16.39
60	3026	0.057574	0.005102	0.071427	2.1780	2.518569	+15.64
$80^{\dagger}$	42	0.113926	0.017366	0.131780	2.1780	2.860778	+31.35
$80^{\dagger}$	2026	0.068903	0.007567	0.086989	3.0122	2.860778	−5.03
$80^{\dagger}$	3026	0.054465	0.004858	0.069699	2.1780	2.860778	+31.35

A dagger (†) marks a sparse retention ratio on the resulting Pareto frontier; the dagger does not indicate that every individual seed is Pareto-efficient.

The resulting frontiers are condition-dependent. HWFET retains the 20%, 40%, and 80% configurations on the setting-level frontier, whereas the 60% configuration is dominated. All four sparse ratios remain non-dominated for LA92. UDDS retains the 20%, 40%, and 80% configurations, while the 60% configuration is dominated. For US06, only the 20% and 80% configurations are Pareto-efficient, while the 40% and 60% configurations are dominated.

Although the 40% configuration is not Pareto-efficient for every operating condition, it provides a practical accuracy-computation compromise for the proposed framework. The realized final retention ratios of the main experiments are approximately 39.2%–41.4%, which is close to the 40% budget. Relative to 20% retention, the 40% configuration reduces the setting-average MAE for HWFET, LA92, and UDDS while requiring fewer FLOPs than the 60% and 80% configurations. The improvement is particularly clear for UDDS, where the mean MAE decreases from approximately 0.0333 at 20% retention to 0.0272 at 40%. For US06, increasing retention from 20% to 40% does not improve the mean MAE, indicating that the near-40% operating point should be interpreted as a practical default rather than a universally optimal budget.

Thus, the Pareto analysis does not support a single globally optimal retention ratio. Instead, it shows that a final retention level near 40% often captures most of the accuracy benefit available at low retention while avoiding the additional computational cost of substantially larger supports.