RETHINKING REPRESENTATIVENESS AND DIVERSITY IN DYNAMIC DATA SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Dynamic data selection accelerates training by sampling a changing subset of the dataset while preserving accuracy. We re-examine two core notions underlying sample selection: representativeness and diversity. We define representativeness as weighted coverage of dataset-level high-frequency feature factors, moving beyond local geometric centrality, and treat diversity as a process-level constraint that requires the selection trajectory to gradually include complementary rare factors over training, rather than dispersion within a single subset. Based on this view, we propose a dynamic selection framework with three components. First, we score representativeness in a plug-in feature space to prioritize samples covering high-frequency feature factors. We instantiate this with a sparse autoencoder trained on the target dataset, using sparse unit activations to summarize both individual samples and dataset-wide factor statistics. Second, we realize process-level diversity by combining rare-factor sampling with a usage-frequency penalty that promotes sample rotation, provably discourages monopoly, and reduces gradient bias. Third, we couple the two-dimensional scoring with a smooth scheduler that transitions selection from core-pattern consolidation to rare-factor exploration, without extra gradients, influence estimates, or second-order computations on the training model. Extensive experiments across vision, text, medical, language modeling, and generative multimodal tasks and different models show that our method yields a better accuracy-efficiency frontier than prior selection methods: it retains most of the full-data accuracy at 1.5–$2.5\times$ lower training cost, and under a matched-propagation protocol it even surpasses full-data training on ImageNet-1K. *Code is provided in the supplementary material.*

1 INTRODUCTION

The performance of deep learning models often comes at the cost of prohibitive computation from training on massive datasets (Liu et al., 2024; Touvron et al., 2021; 2023). Data selection aims to extract a small yet high-value subset, reducing cost and wall-clock time while maintaining or improving accuracy (Albalak et al., 2024).

Early work mostly adopts static data selection strategies (Xie et al., 2023; Har-Peled & Mazumdar, 2004; Sener & Savarese, 2018; Mirzasoleiman et al., 2020), i.e., choosing a fixed subset before training that balances representativeness and diversity (Yang et al., 2025a; Agarwal et al., 2020; Ge et al., 2024; Yu et al., 2025; Du et al., 2025). Such methods score an example by its position, by the response of a proxy model, or by set-level statistics. Geometry-based criteria (Sener & Savarese, 2018; Mirzasoleiman et al., 2020) use the distance from a point to selected centers or class means. Importance-based ones (Paul et al., 2021; Coleman et al., 2020; Sorscher et al., 2022; Xia et al., 2023) use loss, confidence, or influence from a lightweight proxy model. Set-level ones (Maharana et al., 2024; Zheng et al., 2023; Tan et al., 2025) aggregate quantities computed over point embeddings, and measure them inside the candidate subset rather than over the dataset. Because the subset is never revised, it must anticipate the model's evolving capacity in advance, and in practice the selection ratio is kept at roughly 70%, below which performance degrades significantly (Tan et al., 2023; Xia et al., 2023; Yang et al., 2025a; Mirzasoleiman et al., 2020).

Dynamic data selection, which re-derives the subset during training instead of fixing it in advance, has therefore attracted attention: it is more robust at low ratios than a fixed subset, and at a 30% ratio

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.

even uniform resampling surpasses several static baselines (Okanovic et al., 2024). Model-driven variants remove the fixed-subset constraint by re-scoring examples every epoch with signals from the current model, such as small loss for pruning (Qin et al., 2024) or uncertainty for retaining (Hassan et al., 2025; Raju et al., 2021; He et al., 2024).

Static criteria and dynamic signals differ in what they measure, but neither is an attribute statistic of the dataset, and neither keeps track of which factors the trajectory has already covered. A frequent factor such as a common color or a material is distributed across the feature space, not concentrated in a cluster, so a subset covering all cluster centers, or consisting of the most difficult examples, may still miss it, while the only example carrying an infrequent factor combination scores low and is discarded. The dynamic signal adds the model state, but not the selection history: since the model responds most strongly to examples on which it is still failing, the same region is selected in many consecutive epochs, and the union of examples it trains on shrinks, so gradients come from a narrower part of the dataset. Uniform resampling avoids such concentration but does not target the uncovered factors.

We address the missing dataset statistic and the missing selection history by scoring examples with statistics over the whole dataset (Figure 1). A sparse autoencoder is trained on features from a plug-in encoder (CLIP by default), and we record which sparse units each example activates; summing over the dataset gives each unit an activation frequency and a class coverage. First, units activated with high frequency mark frequent attribute factors, and we define representativeness as their weighted coverage, where a unit's weight decreases with the number of classes it is activated in, so that a factor present in every class has little influence. Second, the same statistics give a unit's rarity as the inverse of the number of examples that activate it, and we define diversity over the trajectory rather than within a subset: an example is scored by the rarity of its units, minus a usage-frequency penalty $\lambda \log(1 + u_i(t))$, where $u_i(t)$ counts previous selections. Selection thus rotates instead of concentrating on one region. Third, a scheduler $\alpha(t)$ interpolates between the two scores during training, from coverage of frequent factors to exposure to rare ones, consolidating frequent patterns early and covering missing ones later. All three use dataset-level statistics only, with no gradients or influence estimates on the training model.

Our framework is model-agnostic and extends beyond image classification: we evaluate it on CIFAR-10/100, Tiny-ImageNet, and ImageNet-1K with ResNet-18/34/50, VGG, and ViT backbones, and transfer to text, medical, and language modeling tasks and to generative multimodal fine-tuning, even when the extractor is weak as a zero-shot solver. It improves the accuracy-efficiency trade-off over prior methods.

Our contributions can be summarized as follows:

- **Rethinking representativeness: from points to dataset-level factor coverage.** Representativeness is a property of the factors a subset covers rather than of individual points: we instantiate it as weighted coverage of high-frequency factors read off from sparse unit activations in a plug-in feature space. The factors are learned from the dataset, so coverage needs no external target distribution.
- **Rethinking diversity: from subset dispersion to trajectory-level rotation.** Diversity is a property of the trajectory rather than of a single subset: rare factors are prioritized, and a Usage-Frequency Penalty couples each example's score to how often it has been used, turning a fixed ranking into rotation. This provably discourages sample monopoly and reduces the resulting gradient bias.
- **Coupling the two scores, not scoring them separately.** A lightweight scheduler shifts the focus from factor coverage to rare-factor exploration; an ablation shows both are necessary.

2 Related Work

Static data selection. Early methods fix a subset before training via geometry-based coresets (Sener & Savarese, 2018; Mirzasoleiman et al., 2020) or importance-based scoring (Paul et al., 2021; Coleman et al., 2020; Sorscher et al., 2022; Xia et al., 2023). Recent extensions add set-level diversity and coverage ($\mathbb{D}^2$ (Maharana et al., 2024), CCS (Zheng et al., 2023), InfoMax (Tan et al., 2025)) or training-dynamics importance (CLD (Nagaraj et al., 2025), Dyn-Unc (He et al., 2024), TDDS (Zhang

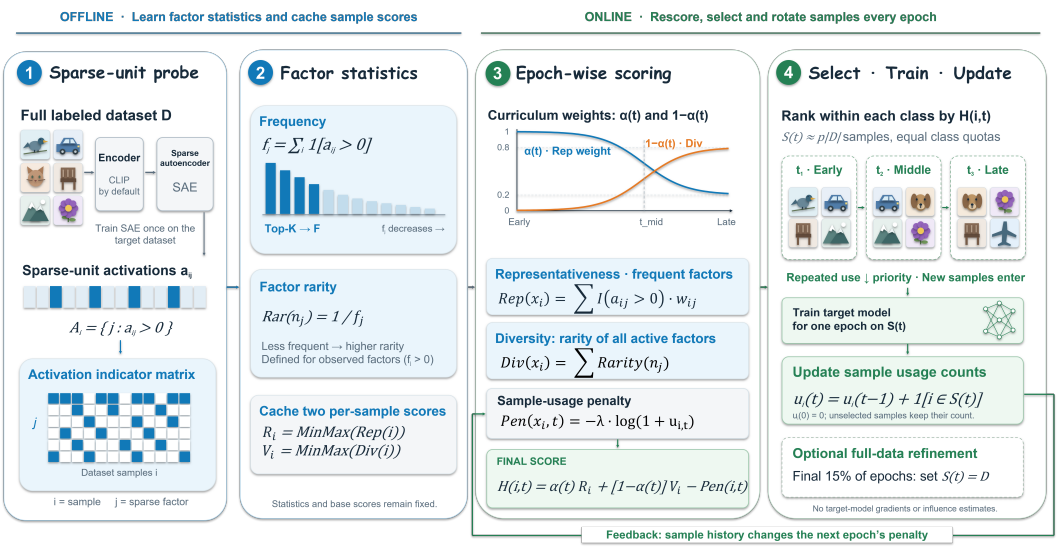


Figure 1: **Our framework for dynamic data selection.** A plug-in feature encoder (CLIP by default) and a sparse autoencoder map inputs to sparse unit activations, cached once offline. Online, at epoch t, each example is rescored by representativeness (weighted coverage of high-frequency factors), diversity (factor rarity), and a usage-frequency penalty that discourages repeated selection and enforces rotation. A scheduler $\alpha(t)$ smoothly balances representativeness and diversity, transitioning from core-pattern consolidation to rare-feature exploration.

et al., 2024), DUAL (Cho et al., 2025)), yet all commit to a fixed subset and a single view of the data distribution. Representation-based scoring in strong pretrained spaces, e.g., the CLIP-powered framework (Yang et al., 2025a), has proven robust and transferable across tasks and inspires our plug-in use of a pretrained encoder.

Dynamic data selection. Dynamic methods resample the subset along training using model-state feedback (Raju et al., 2021; Qin et al., 2024; Hassan et al., 2025); RS2 (Okanovic et al., 2024) shows that even random resampling is already strong. However, these methods rely on noisy instantaneous signals without organizing which semantic factors are covered over time, and focus primarily on reducing the total training cost while preserving accuracy.

Sparse autoencoders. SAEs decompose representations into sparse, interpretable units (Makhzani & Frey, 2013; Gao et al., 2025; Huben et al., 2024) and have been used for data selection (Yang et al., 2025b); we instead use sparse units as a probe for dataset-level factor statistics. A detailed discussion is provided in Appendix E.

3 Method

3.1 Problem Setup and Overview

Let $\mathcal{D} = \{d_i\}_{i=1}^{|\mathcal{D}|}$ be a labeled dataset, where x_i is an input instance and y_i its label. We perform dynamic data selection once per epoch. At epoch t, we construct a subset $\mathcal{S}(t) \subseteq \mathcal{D}$ with selection ratio p by ranking examples using a time-dependent score $\mathcal{H}(d_i, t)$:

$$\mathcal{S}(t) = \arg\max_{\mathcal{S} \subseteq \mathcal{D}} \sum_{d_i \in \mathcal{S}} \mathcal{H}(d_i, t) \quad \text{s.t.} \quad \frac{|\mathcal{S}|}{|\mathcal{D}|} = p. \tag{1}$$

We rethink two core notions for scoring examples: representativeness and diversity. Representativeness (Section 3.3) is defined as weighted coverage of dataset-level high-frequency feature factors, moving beyond local geometric centrality. Diversity (Section 3.4) is treated as a process-level constraint that promotes complementary rare-factor exposure over epochs, rather than dispersion within a single subset. We combine the two signals with a smooth curriculum scheduler (Section 3.5) to obtain $\mathcal{H}(d_i, t)$.

3.2 Feature Factors via Sparse Units

Our scoring relies on a set of *sparse units* that serve as latent feature factors for characterizing each example. Given an input x_i, we denote by a_{ij} the activation of unit n_j and define the active set

$$\mathcal{A}_i \triangleq \{j \mid \mathbb{I}(a_{ij} > 0) = 1\}. \tag{2}$$

These units provide a compact factorization of the chosen feature space and support dataset-wide statistics used by our representativeness and diversity scores.

Instantiation. We treat the feature extractor as a plug-in module. By default, we follow Yang et al. (2025a), use CLIP encoder to extract features and train a sparse autoencoder on the target dataset to obtain sparse unit activations $\{a_{ij}\}$. We refer to this component as a *sparse-unit probe*; the remaining selection pipeline is unchanged when swapping the feature extractor. Importantly, the framework is not restricted to CLIP, as further supported by our plug-in robustness analysis in Appendix C.8.

3.3 Rethinking Representativeness: Coverage over High-Frequency Factors

We define representativeness to reflect *dataset-level coverage* of high-frequency feature factors, instead of local geometric centrality.

Let $\mathcal{F}$ be the set of high-frequency factors, instantiated as the top-K sparse units with the highest activation frequency over the full dataset. A sample is representative if it activates many factors in $\mathcal{F}$. To avoid over-valuing ubiquitous factors that appear broadly across classes, we weight each factor by the inverse of its class coverage. Formally, for unit $n_j \in \mathcal{F}$, let c_j be the number of distinct classes in which n_j is activated at least once and define $w_j = 1/c_j$. We score representativeness as

$$Rep(i) \triangleq \sum_{j \in \mathcal{F}} \mathbb{I}(a_{ij} > 0) \cdot w_j. \tag{3}$$

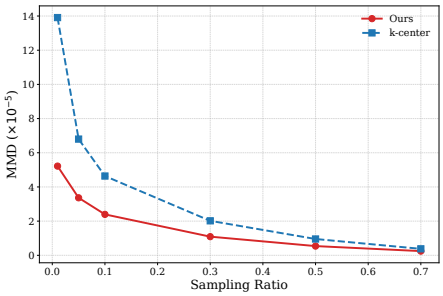


Figure 2: MMD comparison between subsets selected by our representativeness score (Rep-TopK) and the geometric baseline K-Center under different selection ratios. Lower is better.

To assess whether a subset is faithful at the *dataset level*, we measure the Maximum Mean Discrepancy (MMD) between the subset and the full dataset in the same feature space; lower MMD indicates a closer match to the full-data distribution. Figure 2 reports this quantity for our representativeness score and the geometry-based K-Center baseline across selection ratios. As the figure shows, our subsets attain lower MMD at every ratio, showing that representativeness defined as weighted coverage of high-frequency factors tracks the dataset-level distribution more faithfully than local geometric coverage.

3.4 Rethinking Diversity: Process-Level Diversity

In the previous methods, diversity is often defined within a single subset, e.g., by encouraging geometric dispersion or low pairwise similarity, which is insufficient for dynamic selection. Even if a selected subset is diverse at a single epoch, training may still repeatedly concentrate on a small set of consistently high-scoring samples, resulting in poor factor coverage *over time*. We therefore rethink diversity in dynamic selection as a *process-level* property over the entire training trajectory. This requires both exposing complementary rare factors and preventing repeated reselection of the same samples across epochs.

Rare-Factor Diversity. To capture the semantic complement of representativeness, we favor samples that activate rare factors. We measure the rarity of a sparse unit by the inverse of its dataset-level activation count:

$$Rar(n_j) \triangleq \frac{1}{\sum_{i \in \mathcal{D}} \mathbb{I}(a_{ij} > 0)}. \tag{4}$$

A sample receives a higher diversity score if it activates rarer units:

$$Div(i) \triangleq \sum_{j \in \mathcal{A}_i} Rar(n_j). \tag{5}$$

This rarity-based score complements representativeness: while $Rep(i)$ favors coverage of high-frequency, dataset-level factors, $Div(i)$ encourages exposure to rare but informative semantic factors that would otherwise be under-covered.

Usage-Frequency Penalty for Sample Rotation. Rare-factor preference alone is still insufficient. In dynamic selection, a common failure mode is *sample monopoly*: a small set of high-scoring samples dominates the selected subsets across many epochs. To address this issue, we introduce a usage-frequency penalty that explicitly encourages sample rotation over time.

Let $u_i(t)$ denote the cumulative number of times sample i has been selected up to epoch t. We define a sublinear log penalty

$$Pen(i,t) = \lambda \log(1 + u_i(t)), \tag{6}$$

which is subtracted from the sampling score so that overused samples gradually lose priority. The sublinear form is deliberate: the penalty grows fastest over the first few reselections, where the risk of monopoly is highest, and then saturates, discounting heavily used samples rather than excluding them permanently.

A gradient-bias view of repeated reselection, together with the anti-monopoly property of the proposed penalty, is provided in Appendix B.1.

3.5 Curriculum Scheduling of Representativeness and Process-Level Diversity

We use a lightweight curriculum to balance representativeness and process-level diversity over training. At epoch t, the final sampling score is

$$\begin{aligned} \mathcal{H}(i,t) &\triangleq s_i(t) - Pen(i,t), \\ s_i(t) &\triangleq \alpha(t)\, Rep(i) + \big(1 - \alpha(t)\big)\, Div(i), \end{aligned} \tag{7}$$

where $s_i(t)$ interpolates the two scores through the scheduler weight $\alpha(t)$, and $Pen(i,t)$ further enforces sample rotation over time.

A larger $\alpha(t)$ early on emphasizes coverage of high-frequency factors, while a smaller $\alpha(t)$ later increases exposure to rare ones, yielding a transition from core-pattern consolidation to broader process-level exploration; the schedule relies on no gradients or higher-order information. Optionally, we run a short full-data refinement stage in the last T_{full} epochs by setting $\mathcal{S}(t) = \mathcal{D}$, further reducing any residual bias. To mitigate class imbalance, we apply a simple class-balanced constraint by selecting an equal number of samples from each class at every epoch.

We adopt a smooth sigmoid schedule with a floor $\alpha_{\min}$:

$$\alpha(t) = \alpha_{\min} + \big(1 - \alpha_{\min}\big)\Big(1 - \sigma\big(k(t - t_{\text{mid}})\big)\Big), \sigma(x) = \frac{1}{1 + e^{-x}}. \tag{8}$$

Here t_{mid} sets the transition point and k controls the sharpness of the change. Unless otherwise specified we set $\alpha_{\min} = 0.2$, $\lambda = 0.2$, $k = 0.05$, and place t_{mid} at 60% of training. These values were selected by a small hyperparameter search on a validation split and then held fixed across all datasets, backbones, and tasks, without per-setting tuning. The overall procedure is summarized in Algorithm 1.

4 Experiments

4.1 Experiment Setup

Datasets, tasks, and backbones. We evaluate on standard image-classification benchmarks (CIFAR-10/100 (Krizhevsky, 2009), Tiny-ImageNet (Chrabaszcz et al., 2017), ImageNet-1K) and broader transfer tasks: text classification (RSD_15K), medical classification (BRISC), language modeling

Table 1: Accuracy (%) comparison on CIFAR-10/100. Random* denotes dynamic random selection. The reported ratios denote the **selection ratios**. **Ours** is highlighted in grey. ✓ = Dynamic, ✗ = Static. †Results reported in the original papers under their own one-shot static-pruning protocols, converted to the corresponding selection ratios (ResNet-18).

Backbone	Method	Type	**CIFAR-10** (Selection Ratio) 70%	30%	**CIFAR-100** (Selection Ratio) 70%	30%
ResNet-18	Random	✗	$94.6_{\downarrow 1.5}$	$90.2_{\downarrow 5.9}$	$73.8_{\downarrow 4.9}$	$69.7_{\downarrow 9.0}$
	MoDS (Xia et al., 2023)	✗	$93.9_{\downarrow 2.2}$	$90.6_{\downarrow 5.5}$	$74.6_{\downarrow 3.6}$	$65.3_{\downarrow 12.9}$
	MoSo (Tan et al., 2023)	✗	$95.3_{\downarrow 0.8}$	$91.1_{\downarrow 6.0}$	$77.5_{\downarrow 0.7}$	$70.9_{\downarrow 7.3}$
	DP (Yang et al., 2023)	✗	$94.9_{\downarrow 1.2}$	$90.8_{\downarrow 5.3}$	$77.2_{\downarrow 1.0}$	–
	Random*	✓	$94.8_{\downarrow 1.3}$	$93.0_{\downarrow 2.6}$	$77.3_{\downarrow 0.9}$	–
	ϵ-greedy (Raju et al., 2021)	✓	$95.2_{\downarrow 0.9}$	$94.1_{\downarrow 2.0}$	$76.4_{\downarrow 1.8}$	–
	UCB (Raju et al., 2021)	✓	$95.3_{\downarrow 0.8}$	$93.9_{\downarrow 2.2}$	$77.3_{\downarrow 0.9}$	–
	InfoBatch (Qin et al., 2024)	✓	$95.6_{\downarrow 0.5}$	$94.7_{\downarrow 1.4}$	$78.2_{\uparrow 0.0}$	$76.5_{\downarrow 1.7}$
	RS2 w/r (Okanovic et al., 2024)	✓	$95.3_{\downarrow 0.8}$	$94.2_{\downarrow 1.9}$	$77.1_{\downarrow 1.1}$	$74.9_{\downarrow 3.3}$
	RS2 w/o (Okanovic et al., 2024)	✓	$94.9_{\downarrow 1.2}$	$94.1_{\downarrow 2.0}$	$76.9_{\downarrow 1.3}$	$75.8_{\downarrow 2.4}$
	RCAP (Hassan et al., 2025)	✓	$95.9_{\downarrow 0.2}$	$94.4_{\downarrow 1.7}$	$77.6_{\downarrow 0.7}$	$75.6_{\downarrow 2.6}$
	DUAL† (Cho et al., 2025)	✗	95.5†	93.0†	77.9†	69.3†
	CCS† (Zheng et al., 2023)	✗	95.3†	93.0†	76.8†	68.9†
	TDDS† (Zhang et al., 2024)	✗	95.5†	93.0†	77.6†	67.8†
	Ours	✓	$\mathbf{96.1}_{\uparrow 0.0}$	$\mathbf{95.3}_{\downarrow 0.8}$	$\mathbf{78.4}_{\uparrow 0.2}$	$\mathbf{76.7}_{\downarrow 1.5}$
	Whole Dataset	–	$96.1_{\pm 0.1}$		$78.2_{\pm 0.1}$	
ResNet-50	Random	✗	$95.0_{\downarrow 0.8}$	$90.3_{\downarrow 5.5}$	–	$53.6_{\downarrow 25.5}$
	MoDS (Xia et al., 2023)	✗	$95.3_{\downarrow 0.5}$	$91.1_{\downarrow 4.7}$	–	$57.8_{\downarrow 21.3}$
	Self-sup. (Sorscher et al., 2022)	✗	$95.0_{\downarrow 0.8}$	$90.1_{\downarrow 5.7}$	–	$54.6_{\downarrow 24.5}$
	CLIP (Yang et al., 2025a)	✗	$95.4_{\downarrow 0.4}$	$91.1_{\downarrow 4.7}$	–	$62.3_{\downarrow 12.8}$
	RS2 w/r (Okanovic et al., 2024)	✓	$95.4_{\downarrow 0.4}$	$94.0_{\downarrow 1.8}$	$78.4_{\downarrow 0.7}$	$75.1_{\downarrow 3.3}$
	RS2 w/o (Okanovic et al., 2024)	✓	$95.3_{\downarrow 0.5}$	$93.1_{\downarrow 2.7}$	$75.3_{\downarrow 3.8}$	$72.7_{\downarrow 6.4}$
	RCAP (Hassan et al., 2025)	✓	$95.2_{\downarrow 0.6}$	$94.4_{\downarrow 1.4}$	$78.0_{\downarrow 1.1}$	$74.2_{\downarrow 4.9}$
	Ours	✓	$\mathbf{95.6}_{\downarrow 0.2}$	$\mathbf{94.5}_{\downarrow 1.3}$	$\mathbf{79.1}_{\uparrow 0.0}$	$\mathbf{75.7}_{\downarrow 3.4}$
	Whole Dataset	–	$95.8_{\pm 0.1}$		$79.1_{\pm 0.1}$	

(WikiText-2), and generative multimodal fine-tuning (AQUA with LLaVA-v1.5 LoRA). Backbones span ResNet-18/34/50, VGG, and ViT for vision, RoBERTa for text, GPT-2 for language modeling, and LLaVA-v1.5 LoRA for multimodal fine-tuning, allowing us to test transferability across model families and task types. Unless otherwise stated, CIFAR-10/100 results are summarized in Table 1, ImageNet-1K in the left panel of Table 4, and cross-task results in Table 2. Tiny-ImageNet details and results are given in Appendix C.5.

Dynamic selection protocol. Unless stated otherwise, the reported ratio is the **per-epoch selection ratio** p: each epoch we re-sample a subset and train only on the selected examples (equivalently, the complementary pruning ratio is $1 - p$). For all comparisons, training schedules and optimization hyperparameters are identical across methods; only the data stream induced by each selection strategy differs. Unless noted otherwise, we train on the full dataset for the last 15% epochs as a short refinement stage. This refinement stage is optional: results remain strong without it (e.g., removing it changes the CIFAR-10 final accuracy only from **96.1%** to **95.9%**; Appendix C.4). Additional implementation details are in Appendix C.3.

4.2 Main Results on Standard Benchmarks

CIFAR-10/100: comparison with state-of-the-art baselines. We first compare against representative *static* and *dynamic* data selection baselines on CIFAR-10/100 under two per-epoch **selection ratios** (30% and 70%). Baselines span static proxy- and feature-space criteria (MoDS (Xia et al., 2023), MoSo (Tan et al., 2023), DP (Yang et al., 2023)) and dynamic methods that update the data stream across epochs (ϵ-greedy/UCB (Raju et al., 2021), InfoBatch (Qin et al., 2024), RS2 (Okanovic et al., 2024), RCAP (Hassan et al., 2025)), all trained under a consistent protocol.

As shown in Table 1, our method is the most accurate dynamic method in every dataset–ratio cell, with the largest margins on the more challenging CIFAR-100 benchmark. Two patterns stand out. First, dynamic selection is inherently more robust than static selection at low ratios: even *dynamic random* surpasses several static baselines at 30%, since epoch-wise re-sampling already alleviates the

Table 2: **Cross-task generalization summary.** Under the same general framework, the proposed method remains effective across vision, text, medical, language modeling, and generative multimodal tasks. The **Selection Ratio** column reports the fraction of training data used (per-epoch for dynamic settings); AQUA is additionally trained for only 6 epochs.

Dataset	Task	Model	Selection Ratio	Full	Ours
ImageNet-1K	Vision cls. (Top-1)	ResNet-34	50%	73.1	72.8
RSD-15K	Text cls. (Acc.)	RoBERTa	70%	71.0	72.5
BRISC	Medical cls. (Acc.)	VGG16	50%	96.9	98.3
WikiText-2	LM (PPL↓)	GPT-2	60%	25.6	26.1
AQUA	Gen. MLLM FT (R-L F1)	LLaVA-v1.5 LoRA	50% (6 ep)	34.9	34.4

brittleness of a single fixed subset. Second, our method improves on the strongest dynamic baseline in all eight cells, by **0.1** to **0.7** points. On CIFAR-100 with ResNet-50 at 30%, our **75.7%** is the best of all listed methods, ahead of RS2 w/r (**75.1%**) and RCAP (**74.2%**). This is consistent with our design: early representativeness keeps a faithful proxy of the full data, while later rare-factor exposure and sample rotation broaden training signals over time. Across ResNet-18/50 on CIFAR-10/100, our method most consistently matches full-data training under aggressive data reduction. A strictly budget-matched analysis of the optional refinement stage, holding total data exposure fixed, is provided in Appendix C.4.

ImageNet-1K: large-scale evidence. We next evaluate our framework on ImageNet-1K with ResNet-34 under two complementary protocols. First, to isolate data-organization quality from under-training, we keep the total number of forward passes identical to full-data training at every selection ratio (a *matched-propagation* protocol; left panel of Table 4). Under this protocol, our method reaches **73.4%/73.7%/73.8%** top-1 at **70%/50%/30%** ratios, exceeding the **73.1%** full-data reference at every ratio (by **0.3**–**0.7** points) and providing *flexible operating points* on the accuracy–efficiency frontier. Second, for realistic end-to-end acceleration, we reduce the total amount of data processed (a *reduced-data* protocol; Table 3 and Table 2).

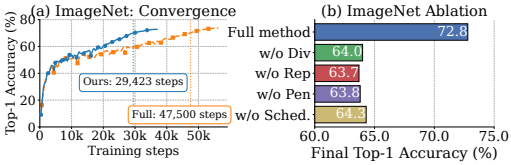


Figure 3: **Additional large-scale evidence on ImageNet.** (a) Convergence speed comparison between full-data training and our 50% selection setting. (b) Large-scale ablations under the same 50% protocol.

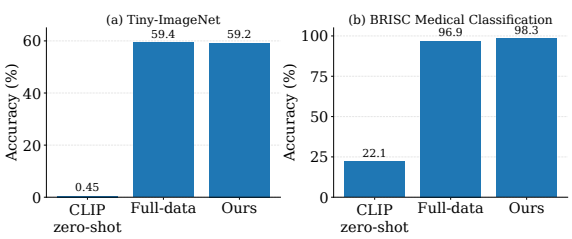
Under the reduced-data protocol, our method keeps most of the full-data accuracy at a fraction of the cost: **72.8%** top-1 at **50%** (only **0.3%** below the **73.1%** reference) and **73.06%** at **70%** with about **25%** training acceleration. Figure 3 shows faster convergence at scale: **70%** top-1 in **29,423** steps versus **47,500** for full data. Ablations confirm every component matters: removing Rep, Div, the penalty, or the scheduler drops reduced-data top-1 to **63.7%/64.0%/63.8%/64.3%**.

4.3 Generalization Across Tasks, Models, and Feature Spaces

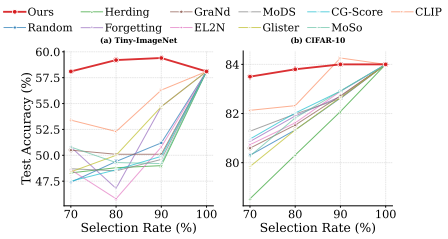
Generalization across tasks, models, and datasets. As summarized in Tables 2 and 9 and Figure 4, the framework generalizes beyond standard image classification: it transfers across CNNs, vision transformers, and datasets, and remains effective on text, medical, language modeling, and generative multimodal tasks. This suggests that the sparse-unit probe captures reusable factor structure for sample ranking rather than a narrow task- or dataset-specific heuristic; detailed settings are provided in Appendices C.6 and C.7.

Robustness to plug-in feature quality. The method does not require strong task-specific zero-shot performance from the plug-in feature extractor. As shown in Figure 4a, effective selection is still obtained even when CLIP zero-shot accuracy is very weak on the downstream task. This supports our view that the plug-in feature space serves as a factor-analysis probe for sample ranking, rather than directly solving the target task in a zero-shot manner. Further results are discussed in Appendix C.8.

Robustness to learning-rate schedules. We also evaluate our method under three common schedulers while keeping other training settings fixed. Across all schedules, the method preserves accuracy and delivers consistent training-time reductions; see Appendix C.9 for full settings and results.



(a) Plug-in feature robustness. Weak CLIP zero-shot task performance does not prevent effective sample selection.



(b) Accuracy under different selection ratios on additional model–dataset pairs.

Figure 4: **Generalization across feature spaces and model–dataset pairs.** (a) The plug-in feature space serves as a factor-analysis probe for sample ranking, rather than requiring strong task-specific zero-shot performance. (b) Our method maintains competitive performance under different selection ratios on VGG-16 with Tiny-ImageNet and ViT-small with CIFAR-10.

Table 3: **Efficiency, accuracy, and overhead summary. Left:** training efficiency and accuracy comparison on CIFAR-10 and Tiny-ImageNet. **Right:** overhead summary on ImageNet under the default protocol. Overhead in the left table includes CLIP feature extraction, SAE training, and sample scoring. The ImageNet summary corresponds to the default **50% selection + final 15% full-data refinement** setting.

Dataset	Method	Acc. (%)	Overhead (GPU h)	Speedup
C-10	Full	$96.1_{\pm 0.1}$	–	1.0×
	Ours	$95.3_{\pm 0.1}$	**0.0028**	**2.51**×
T-Img	Full	$59.4_{\pm 0.1}$	–	1.0×
	Ours	$59.2_{\pm 0.2}$	**0.12**	**1.72**×

(a) Accuracy–efficiency comparison.

Overhead Summary on ImageNet	
SAE time / epoch	37 s
Total SAE training time	30 min 50 s
Full training accuracy	73.1%
Full training time (90 epochs)	405 min
Selection ratio	50%
Final accuracy	72.8%
Estimated speedup	1.54×

(b) Overhead summary under the default setting.

4.4 Efficiency Comparison

Convergence speed. We first examine optimization speed on CIFAR-10 with ResNet-18. Figure 5 reports the convergence curves, and we measure the steps required to reach **90%** validation accuracy (*step@90*). In the early stage, our curve closely matches full-data training, indicating that representativeness-dominant selection provides a faithful proxy of the full dataset. In the mid-to-late stage, our method improves faster and overtakes full-data training, suggesting that rare-factor exposure and sample rotation accelerate learning by introducing complementary training signals across epochs. Overall, our method reaches the target accuracy in the fewest steps.

Additional overhead. Our framework separates offline scoring from lightweight online updates: after training the sparse-unit probe once, scores are precomputed offline, while online training only updates the usage-frequency penalty and the scheduler weight each epoch. As summarized in Table 3, this yields favorable end-to-end trade-offs on both CIFAR-10 and Tiny-ImageNet: **2.51**× acceleration with a small accuracy drop (**96.1** to **95.3**) on CIFAR-10, and **1.72**× speedup with closely matched performance (**59.4** vs. **59.2**) on Tiny-ImageNet. On ImageNet, the one-time SAE-related preprocessing cost (**30** min) remains small relative to full-model training (**405** min).

4.5 Ablation Study

We ablate each design choice on CIFAR-100 with ResNet-50 (200 epochs). Table 4 reports accuracy at 50 epochs (optimization progress) and the final accuracy.

High-frequency-factor coverage supports early progress. Removing Rep (*w/o Rep*) lowers both the 50-epoch and final accuracy, indicating that prioritizing coverage of high-frequency factors improves the early optimization trajectory.

Table 4: **Performance under different selection ratios and ablation results.** Left: Top-1 accuracy (%) on ImageNet-1K and RSD_15K under different selection ratios. The **Ours** row on ImageNet-1K uses a matched total model-propagation budget across selection ratios (data-organization quality), whereas static baselines and rows marked † use a fixed-epoch protocol and are reported as in the original papers (ResNet-34, selection ratios matched) as contextual references. Right: Ablation on CIFAR-100 (ResNet-50, 200 epochs).

(a) Different selection ratios

Dataset	Method	100%	70%	50%	30%
ImageNet-1K (ResNet-34)	Random	73.1	72.2	70.3	66.7
	Entropy	73.1	72.3	70.8	64.1
	$\mathbb{D}^2$	73.1	73.0	71.4	64.8
	Moderate-DS	73.1	72.0	70.3	65.9
	K-center	73.1	72.2	67.2	48.8
	Forgetting	73.1	72.6	70.9	66.5
	Ours	73.1	**73.4**	**73.7**	**73.8**
RSD_15K (RoBERTa)	K-Center	71.0	71.6	69.7	69.1
	Ours	71.0	**72.5**	**71.5**	**69.5**

(b) Ablation study

Variant	R	D	P	α	50 ep.	Final
Full Method	✓	✓	✓	✓	**57.7**	**79.1**
w/o Rep	✗	✓	✓	✓	57.1	78.1
w/o Div	✓	✗	✓	✓	57.5	77.2
w/o Pen	✓	✓	✗	✓	54.7	76.4
w/o α	✓	✓	✓	✗	54.5	78.2
Static	✓	✓	✗	✗	53.8	76.3
only Div	✗	✓	✗	✗	57.1	77.1
only Rep	✓	✗	✗	✗	54.1	76.8
only Pen	✗	✗	✓	✗	54.6	78.1
w/o Full-data	✓	✓	✓	✓	57.6	78.3

Rare-factor exposure matters for the end performance. Removing Div (*w/o Div*) yields a small gap at 50 epochs but a larger one at the end, suggesting process-level exposure to complementary rare factors matters in the later stage.

Rotation via the usage-frequency penalty is critical. Disabling the penalty (*w/o Pen*) causes the largest single-component drop (**79.1** → **76.4**), consistent with greedy re-selection concentrating training on a small subset; conversely, the penalty alone (*only* Pen) is already a strong baseline that spreads selection over the dataset.

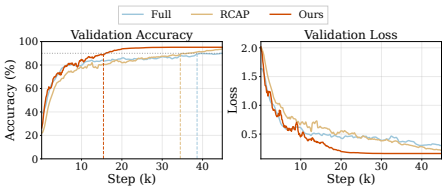


Figure 5: Convergence speed comparison on CIFAR-10 (ResNet-18). **Ours** denotes our method, **RCAP** denotes the previous SOTA method, and **Full** uses the full dataset.

Scheduling improves the trade-off beyond a fixed mixture. Without the curriculum schedule (*w/o* α) performance drops, and a static mixing variant (*Static*) performs substantially worse, showing the benefit of coupling time-varying emphasis with rotation.

Full-data refinement stabilizes the final model. Removing the final full-data stage (*w/o Full-data*) decreases final accuracy, suggesting a short full-data refinement reduces residual selection-induced bias and consolidates the representation.

4.6 Limitations

Because the rare-factor preference may assign high scores to corrupted inputs, visual outliers, or meaningless rare patterns, we do not assume that every rare factor is reliable; filtering for factor reliability is left as future work. Due to computational resource limits, we could not validate the framework on very large-scale training or substantially larger models, and we verify probe transfer on a single cross-dataset case, ImageNet-1K to CIFAR-10.

5 Conclusion

This work rethinks dynamic data selection through two complementary lenses: representativeness as dataset-level coverage of high-frequency factors, and diversity as a process-level requirement enforced over training. Using a sparse-unit probe in a plug-in feature space, we precompute offline scores for representativeness and diversity and update selection online only through a usage-frequency penalty and a lightweight scheduler. Across standard benchmarks and broader tasks, the resulting framework yields a stronger accuracy–efficiency frontier; we hope this perspective and protocol will facilitate further study of scalable dynamic data selection.

AI USE STATEMENT

In this work, we used generative AI tools (e.g., Claude) for tasks with required disclosure: specifically, as a writing and editing assistant to help draft, polish, and rephrase the manuscript text, to refine wording for clarity and conciseness, and to help edit and format mathematical formulas and LaTeX equations. We did not use generative AI tools for conducting the experiments, for generating or altering experimental results, numerical data, tables, or figures, or for writing the experimental code that produced the reported results; these tasks were performed by the authors and are not applicable to this work. Additionally, we used generative AI tools for tasks with recommended disclosure, including proofreading and assisting with the structural organization of the manuscript and its LaTeX typesetting. We have reviewed all AI-assisted work: AI-suggested text and equations were carefully checked for factual and technical accuracy, and the experimental results and figures were produced and verified by the authors by running the corresponding experiments. We take responsibility for the final content of this work, including text, equations, claims, and artifacts produced with the aid of generative AI.

REFERENCES

Sharat Agarwal, Himanshu Arora, Saket Anand, and Chetan Arora. Contextual diversity for active learning. In *European Conference on Computer Vision*, pp. 137–153, 2020.

Alon Albalak, Yanai Elazar, Sang Michael Xie, Shayne Longpre, Nathan Lambert, Xinyi Wang, Niklas Muennighoff, Bairu Hou, Liangming Pan, Haewon Jeong, Colin Raffel, Shiyu Chang, Tatsunori Hashimoto, and William Yang Wang. A survey on data selection for language models. *Transactions on Machine Learning Research*, 2024.

Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, pp. 10602–10643, 2025.

Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017.

Cody Coleman, Christopher Yeh, Stephen Mussmann, Baharan Mirzasoleiman, Peter Bailis, Percy Liang, Jure Leskovec, and Matei Zaharia. Selection via proxy: Efficient data selection for deep learning. In *International Conference on Learning Representations*, 2020.

Yupei Du, Yingjin Song, Hugh Mee Wong, Daniil Ignatev, Albert Gatt, and Dong Nguyen. Disentangling the roles of representation and selection in data pruning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16791–16809, 2025.

Amirreza Fateh, Yasin Rezvani, Sara Moayedi, Sadjad Rezvani, Fatemeh Fateh, Mansoor Fateh, and Vahid Abolghasemi. Brisc: Annotated dataset for brain tumor segmentation and classification. *Scientific Data*, 2026.

Leo Gao, Tom Dupre la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *The Thirteenth International Conference on Learning Representations*, 2025.

Noa Garcia, Chentao Ye, Zihua Liu, Qingtao Hu, Mayu Otani, Chenhui Chu, Yuta Nakashima, and Teruko Mitamura. A dataset and baselines for visual question answering on art. In *Proceedings of the European Conference in Computer Vision Workshops*, 2020.

Yuan Ge, Yilun Liu, Chi Hu, Weibin Meng, Shimin Tao, Xiaofeng Zhao, Mahong Xia, Zhang Li, Boxing Chen, Hao Yang, et al. Clustering and ranking: Diversity-preserved instruction selection through expert-aligned quality estimation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 464–478, 2024.

Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pp. 291–300, 2004.

Atif Hassan, Swanand Khare, and Jiaul H. Paik. RCAP: Robust, class-aware, probabilistic dynamic dataset pruning. In *The 41st Conference on Uncertainty in Artificial Intelligence*, 2025.

Muyang He, Shuo Yang, Tiejun Huang, and Bo Zhao. Large-scale dataset pruning with dynamic uncertainty. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 7713–7722, 2024.

Robert Huben, Hoagy Cunningham, Logan Riggs Smith, Aidan Ewart, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, 2019.

Hantao Lou, Changye Li, Jiaming Ji, and Yaodong Yang. SAE-v: Interpreting multimodal models for enhanced alignment. In *Forty-second International Conference on Machine Learning*, 2025.

Adyasha Maharana, Prateek Yadav, and Mohit Bansal. $\mathbb{D}^2$ pruning: Message passing for balancing diversity & difficulty in data pruning. In *The Twelfth International Conference on Learning Representations*, 2024.

Alireza Makhzani and Brendan Frey. K-sparse autoencoders. *arXiv preprint arXiv:1312.5663*, 2013.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pp. 6950–6960, 2020.

Manish Nagaraj, Deepak Ravikumar, and Kaushik Roy. Coresets from trajectories: Selecting data via correlation of loss differences. *Transactions on Machine Learning Research*, 2025.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos E. Nikolakakis, Amin Karbasi, Dionysios S. Kalogerias, Nezihe Merve Gürel, and Theodoros Rekatsinas. Repeated random sampling for minimizing the time-to-accuracy of learning. In *The Twelfth International Conference on Learning Representations*, 2024.

Xingyuan Pan, Luyang Huang, Liyan Kang, Zhicheng Liu, Yu Lu, and Shanbo Cheng. G-dig: Towards gradient-based diverse and high-quality instruction data selection for machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15395–15406, 2024.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 20596–20607, 2021.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Daquan Zhou, Lei Shang, Baigui Sun, Xuansong Xie, and Yang You. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *The Twelfth International Conference on Learning Representations*, 2024.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Ravi S Raju, Kyle Daruwalla, and Mikko Lipasti. Accelerating deep learning with dynamic data pruning. *arXiv preprint arXiv:2111.12621*, 2021.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 19523–19536, 2022.

Haoru Tan, Sitong Wu, Fei Du, Yukang Chen, Zhibin Wang, Fan Wang, and Xiaojuan Qi. Data pruning via moving-one-sample-out. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 18251–18262, 2023.

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.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pp. 10347–10357, 2021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 1121–1128, 2009.

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2023.

Sang Michael Xie, Shibani Santurkar, Tengyu Ma, and Percy S Liang. Data selection for language models via importance resampling. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 34201–34227, 2023.

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.

Suorong Yang, Peng Ye, Wanli Ouyang, Dongzhan Zhou, and Furao Shen. A CLIP-powered framework for robust and generalizable data selection. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Xianjun Yang, Shaoliang Nie, Lijuan Liu, Suchin Gururangan, Ujjwal Karn, Rui Hou, Madian Khabsa, and Yuning Mao. Diversity-driven data selection for language model tuning through sparse autoencoder. *arXiv preprint arXiv:2502.14050*, 2025b.

Qifan Yu, Zhebei Shen, Zhongqi Yue, Yang Wu, Bosheng Qin, Wenqiao Zhang, Yunfei Li, Juncheng Li, Siliang Tang, and Yueting Zhuang. Mastering collaborative multi-modal data selection: A focus on informativeness, uniqueness, and representativeness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 155–165, 2025.

Xin Zhang, Jiawei Du, Yunsong Li, Weiying Xie, and Joey Tianyi Zhou. Spanning training progress: Temporal dual-depth scoring (tdds) for enhanced dataset pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26223–26232, 2024.

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.

Shouwen Zheng, Yingzhi Tao, and Taiqi Zhou. Rsd-15k: A large-scale user-level annotated dataset for suicide risk detection on social media. In *2025 IEEE 41st International Conference on Data Engineering Workshops*, pp. 190–196, 2025.

A Training and Analysis of Sparse Autoencoder

A.1 Motivation

Modern vision-language models (e.g., CLIP) provide strong but dense embeddings, where multiple factors are mixed across dimensions. For our selection scores, we therefore introduce a sparse factorization that turns dense features into sparse unit activations, enabling simple dataset-level statistics over *high-frequency* versus *rare* factors.

Specifically, we train a Top-k Sparse Autoencoder (SAE) on the chosen embedding space (CLIP by default). The SAE produces an overcomplete and sparse latent code, where each sample activates only a small set of units. These sparse activations serve as our sparse-unit probe, from which we compute weighted coverage of high-frequency factors for representative and rarity-based signals for process-level diversity.

A.2 Model Architecture

Let $x \in \mathbb{R}^D$ be the CLIP-ViT/L-14 embedding of an image. We define a two-layer SAE with tied weights and Top-K sparsity:

$$\mathbf{z}_{\text{pre}} = \mathbf{W}_e\mathbf{x} + \mathbf{b}_e, \quad \mathbf{W}e \in \mathbb{R}^{N_{\text{lat}} \times D}, \tag{9}$$

$$\mathbf{z} = \text{TopK}(\mathbf{z}_{\text{pre}}, k),\ \hat{\mathbf{x}} = \mathbf{W}_d\mathbf{z} + \mathbf{b}_d, \tag{10}$$

where $N_{\text{lat}} \gg D$ denotes latent dimension, $\mathbf{W}_d = \mathbf{W}_e^\top$, k denotes sparsity budget, $TopK(z, k)$ retains only the k largest entries and zeros out the rest:

$$\text{TopK}(\mathbf{z}, k)_i = \begin{cases} z_i, & \text{if } i \in \mathcal{I}_{\text{top-}k}(\mathbf{z}), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

with $\mathcal{I}_{\text{top-}k}(\mathbf{z})$ being the index set of the k largest entries in $\mathbf{z}$ by absolute value.

A.3 Training Details

The SAE is trained by minimizing the reconstruction loss, augmented with a *dead neuron revival* term to ensure full utilization of the latent space. Let $\mathbf{z} = \text{TopK}(\mathbf{W}_e\mathbf{x} + \mathbf{b}_e, k)$ denote the sparse activation. A neuron j is considered *dead* if its activation count remains zero over a monitoring window. Every T_{check} batches, we identify the set of dead neurons $\mathcal{D} = \{j \mid \sum_{i=1}^{B} \mathbb{I}(z_{i,j} > 0) = 0\}$.

For each $j \in \mathcal{D}$, we construct a *revival sample* $\mathbf{x}_j^{\text{rev}}$ as the training instance that maximally excites neuron j under its current pre-activation:

$$\mathbf{x}_j^{\text{rev}} = \arg\max_{\mathbf{x}_i \in \mathcal{B}} \left[(\mathbf{W}_e\mathbf{x}_i + \mathbf{b}_e)_j\right], \tag{12}$$

where $\mathcal{B}$ is the current mini-batch.

We then reconstruct $\mathbf{x}_j^{\text{rev}}$ using *only* neuron j, i.e., define a fake activation vector $\tilde{\mathbf{z}}^{(j)}$ as:

$$\tilde{\mathbf{z}}_l^{(j)} = \begin{cases} (\mathbf{W}_e\mathbf{x}_j^{\text{rev}} + \mathbf{b}_e)_j, & l = j, \\ 0, & l \neq j, \end{cases} \tag{13}$$

and compute its reconstruction $\hat{\mathbf{x}}_j^{\text{from dead}} = \mathbf{W}_d\tilde{\mathbf{z}}^{(j)} + \mathbf{b}_d$.

The revival loss is then summed over all dead neurons in $\mathcal{D}$:

$$\mathcal{L}_{\text{revive}} = \sum_{j \in \mathcal{D}} \left\|\mathbf{x}_j^{\text{rev}} - \hat{\mathbf{x}}_j^{\text{from dead}}\right\|_2^2, \tag{14}$$

and the total training objective is:

$$\mathcal{L} = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2}_{\text{Reconstruction Loss}} + \lambda_{\text{revive}} \cdot \mathcal{L}_{\text{revive}}. \tag{15}$$

This procedure encourages dormant neurons to specialize on under-represented patterns, significantly improving latent coverage and downstream curriculum quality.

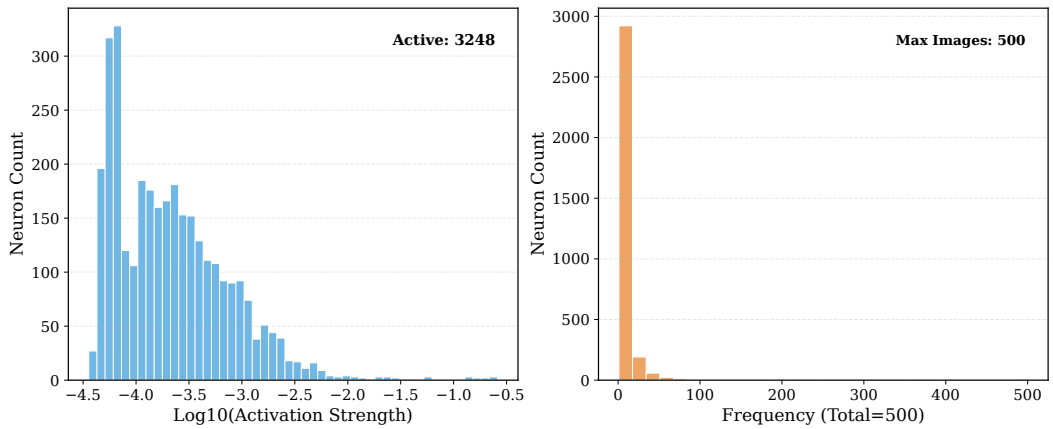


Figure 6: SAE analysis of the class 'cat'. The left panel shows the activation intensity of each neuron; selecting nearly the top 10% of neurons as high-frequency neurons covers all neurons with high activation intensity. The right panel displays the number of samples that activate each neuron, where a maximum value of 500 indicates that all images in this class activate at least one neuron.

A.4 Visualization of our trained SAE

To validate the semantic disentanglement of our SAE, we perform a multi-level visualization of the latent space.

First, we project the entire latent representation using UMAP onto a 2D plane in Figure 9a. The resulting plot reveals that samples from the same class form compact, well-separated clusters, while semantically related classes (e.g., car and bus) lie in proximity, and dissimilar classes (e.g., tulip and bicycle) are distant, as shown in Figure 9b. This geometric structure confirms that the SAE has learned a hierarchically organized semantic space, where Euclidean distance aligns with human-perceived semantic similarity, which is a prerequisite for meaningful sample scoring.

Second, to examine the internal structure of a single class, we analyze the activation patterns for the 'cat' category in Figure 6. We find that 5768 neurons are activated by at least one cat image, and the distribution of average activation strength is long-tailed, indicating that only a small subset of neurons are strongly activated; these likely correspond to core, discriminative features (e.g., "pointy ears", "whiskers"). Moreover, the activation frequency is highly skewed: a few neurons are activated by all 5000 cat images, while most are active in only a few samples, suggesting the presence of both universal and rare, fine-grained visual concepts within the class.

As shown in Figure 7, high frequency neurons can be divided into two categories: class-specific and universal. In our methods, we take the universal ones into consideration, which are considered to represent the implicit feature of the full dataset.

A.5 Sensitivity to SAE Hyperparameters

We further examine the sensitivity of our framework to the sparse-unit probe configuration. Specifically, we vary the SAE latent dimension and the Top-k sparsity budget while keeping the downstream training protocol fixed. For a fair comparison, all configurations are evaluated under the same **60% selection ratio**.

Table 5 shows that downstream performance remains strong across a broad range of SAE configurations. Although reconstruction quality and dead-neuron ratio vary noticeably with the latent dimension and sparsity level, the resulting downstream accuracy is relatively stable, indicating that the proposed scoring framework is not overly sensitive to a narrow choice of SAE hyperparameters.

Several trends are worth noting. First, increasing k generally improves reconstruction error and reduces the dead-neuron ratio, as more latent units are allowed to participate in reconstruction. Second, larger latent dimensions do not monotonically improve downstream accuracy, suggesting that better reconstruction alone does not directly translate into better data selection. Overall, the

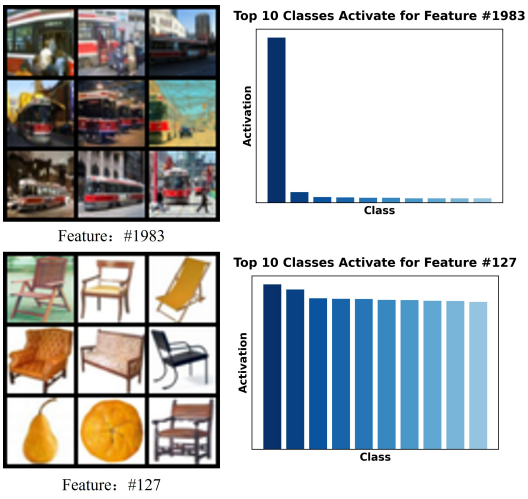


Figure 7: Visualization of two types of highly activated neurons and the nine images (on the left column) that most strongly activate them. The top row shows a class-specific neuron, which is strongly activated only in the class of street car. The bottom row shows a universal neuron, which is strongly activated across multiple classes and images with a plain background.

Table 5: **Sensitivity to SAE hyperparameters.** Under the same **60% selection ratio**, downstream performance remains strong across 12 SAE configurations. The default configuration is **16384 / 64**.

Latent dim	k	Recon. MSE	Dead ratio	Best acc. (%)
4096	16	0.000092	0.3386	95.86
4096	32	0.000068	0.0388	95.94
4096	64	0.000050	0.0100	95.98
4096	128	0.000036	0.0095	**96.14**
8192	16	0.000091	0.6168	95.84
8192	32	0.000066	0.2498	95.88
8192	64	0.000048	0.0453	96.02
8192	128	0.000033	0.0066	95.61
16384	16	0.000091	0.7642	95.93
16384	32	0.000064	0.3868	95.80
16384	64	0.000046	0.1542	96.09
16384	128	0.000031	0.0521	95.72

default setting **16384 / 64** achieves a strong accuracy–robustness trade-off. These results suggest that our method is robust to a reasonably wide range of sparse-unit probe configurations.

B Method Details

B.1 Why Usage-Frequency Penalty Mitigates Repeated Reselection

We provide a simple gradient-bias view to explain why repeated reselection is undesirable in dynamic data selection, and then show that the proposed usage-frequency penalty has an anti-monopoly effect on the ranking dynamics.

Gradient-bias view of repeated reselection. Let $N = |\mathcal{D}|$ and let

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell_i(\theta) \tag{16}$$

denote the empirical risk over the full dataset. The corresponding full-dataset gradient is

$$g_{\text{full}}(\theta) = \nabla\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\nabla\ell_i(\theta). \tag{17}$$

At epoch t, dynamic selection trains on a subset $\mathcal{D}_t \subseteq \mathcal{D}$ of size $K = \lfloor pN \rfloor$, yielding the subset gradient

$$g_t(\theta) = \frac{1}{K}\sum_{i\in\mathcal{D}_t}\nabla\ell_i(\theta). \tag{18}$$

Define the long-term inclusion frequency of sample i as

$$\pi_i \triangleq \lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\mathbb{I}\{i \in \mathcal{D}_t\}. \tag{19}$$

Then the expected subset gradient can be written as

$$\mathbb{E}[g_t(\theta)] = \sum_{i=1}^{N}\frac{\pi_i}{K}\nabla\ell_i(\theta), \tag{20}$$

which induces the following bias relative to full-data training:

$$\begin{aligned}\Delta g(\theta) &\triangleq \mathbb{E}[g_t(\theta)] - g_{\text{full}}(\theta)\\ &= \sum_{i=1}^{N}\left(\frac{\pi_i}{K} - \frac{1}{N}\right)\nabla\ell_i(\theta).\end{aligned} \tag{21}$$

Equation (21) shows that if a small subset dominates selection over many epochs, the inclusion frequencies $\{\pi_i\}$ become highly non-uniform, causing systematic deviation from the full-data gradient. This motivates encouraging broader and more balanced participation over training.

Anti-monopoly property of the usage-frequency penalty. Recall the base score before applying the usage penalty:

$$s_i(t) \triangleq \alpha(t)\,Rep(i) + \big(1-\alpha(t)\big)\,Div(i). \tag{22}$$

Assume $s_i(t) \in [s_{\min}, s_{\max}]$ and define $\Delta = s_{\max} - s_{\min}$. Let the penalized score be

$$\tilde{s}_i(t) = s_i(t) - \lambda\log(1+u_i(t)), \tag{23}$$

where $u_i(t)$ is the cumulative number of times sample i has been selected up to epoch t.

Theorem 1 (Sample Rotation / Anti-Monopoly)**.** *For any two samples i and j, if*

$$u_i(t) \geq (1+u_j(t))\exp(\Delta/\lambda) - 1, \tag{24}$$

then $\tilde{s}_i(t) \leq \tilde{s}_j(t)$; that is, sufficiently over-sampled instances cannot dominate the ranking indefinitely.

Proof. We have

$$\begin{aligned}\tilde{s}_i(t) - \tilde{s}_j(t) &= (s_i(t) - s_j(t)) - \lambda\log\frac{1+u_i(t)}{1+u_j(t)}\\ &\leq \Delta - \lambda\log\frac{1+u_i(t)}{1+u_j(t)}.\end{aligned} \tag{25}$$

By the condition in Eq. (24),

$$\log\frac{1+u_i(t)}{1+u_j(t)} \geq \Delta/\lambda,$$

which implies

$$\tilde{s}_i(t) - \tilde{s}_j(t) \leq 0.$$

Hence $\tilde{s}_i(t) \leq \tilde{s}_j(t)$. □

The theorem formalizes the anti-monopoly effect of the usage-frequency penalty: once a sample has been selected sufficiently often, its ranking priority is necessarily reduced relative to less-used samples. Together with the gradient-bias view above, this explains why explicit sample rotation is beneficial in dynamic data selection. We further evaluate this effect under noisy or hard examples in Appendix B.3, where repeated reselection can otherwise dominate the training signal.

Table 6: Effect of sparse-unit probe instantiation on ResNet-18. We compare probes trained on CLIP features and downstream ResNet-18 representations under different selection ratios.

Probe Feature Space	**70%**	**50%**	**30%**
CLIP embeddings	96.1	95.6	95.3
ResNet-18 embeddings	95.9	95.4	95.1

B.2 Distributional Faithfulness Evaluation via MMD

We provide an auxiliary evaluation to assess whether the subsets induced by our representativeness score better preserve the feature distribution of the full dataset. Specifically, we measure *distributional faithfulness* using Maximum Mean Discrepancy (MMD) in a fixed embedding space.

Embedding space. Unless otherwise stated, we compute features using CLIP ViT-L/14 image embeddings, which are also used as the default feature space for the sparse-unit probe. All embeddings are ℓ_2-normalized before kernel computation.

Subset construction. For a given sampling ratio p, we compare two subset-selection strategies: (i) **Rep-TopK**, which ranks examples by the precomputed representativeness score $Rep(i)$ and selects the top-K examples, where $K = \lfloor p|\mathcal{D}| \rfloor$; and (ii) **K-Center**, which performs greedy k-center selection in the same embedding space and takes the first K selected samples. The latter is purely geometry-based and serves as a representative baseline for local-structure-based coverage.

MMD definition. Let $\phi(\cdot)$ denote the fixed feature extractor, let $\mathcal{V} = \{\phi(x_i)\}_{i=1}^{N}$ be the full set of features, and let $\mathcal{U} = \{\phi(x) \mid x \in \mathcal{S}\}$ denote the features of a selected subset $\mathcal{S}$ of size K. Given a characteristic kernel $k(\cdot, \cdot)$, the squared MMD is defined as

$$\mathrm{MMD}^2(\mathcal{U}, \mathcal{V}) = \Big\| \frac{1}{K} \sum_{u \in \mathcal{U}} \psi(u) - \frac{1}{N} \sum_{v \in \mathcal{V}} \psi(v) \Big\|_{\mathcal{H}}^2, \tag{26}$$

where $\psi(\cdot)$ is the implicit feature map associated with k and $\mathcal{H}$ is the corresponding reproducing kernel Hilbert space.

Equivalently, the squared MMD can be written as

$$\mathrm{MMD}^2(\mathcal{U}, \mathcal{V}) = \mathbb{E}_{u,u' \sim \mathcal{U}}[k(u, u')] + \mathbb{E}_{v,v' \sim \mathcal{V}}[k(v, v')] - 2\,\mathbb{E}_{u \sim \mathcal{U},\, v \sim \mathcal{V}}[k(u, v)]. \tag{27}$$

In practice, we estimate MMD using the standard unbiased estimator based on this kernel form.

Kernel setting. We use an RBF kernel $k(\cdot, \cdot)$. Unless otherwise stated, we follow the default setting in `sklearn.metrics.pairwise.rbf_kernel`, i.e., $\gamma = 1/d$, where d is the feature dimension.

Implementation details. To control computational cost, we randomly subsample up to 5,000 examples from $\mathcal{U}$ and $\mathcal{V}$ when estimating the kernel expectations in Eq. (27). All subset-selection methods are evaluated in the same embedding space for fair comparison.

Protocol. We evaluate sampling ratios $p \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.7\}$. For each ratio, we compute the MMD between the selected subset and the full dataset in the chosen embedding space. Lower MMD indicates that the selected subset better matches the full-data feature distribution. We report the quantitative trend in Fig. 2 and provide t-SNE visualizations as qualitative complements.

Remark. This evaluation is not used for training or subset optimization. It serves only as an analysis tool to examine whether the proposed representativeness score yields subsets that more faithfully preserve the full-data distribution in the chosen feature space.

B.3 Noise Robustness

To evaluate robustness to imperfect supervision, we construct a noisy training set for CIFAR-100 by injecting **20% symmetric label noise** into the training split while keeping the validation/test split

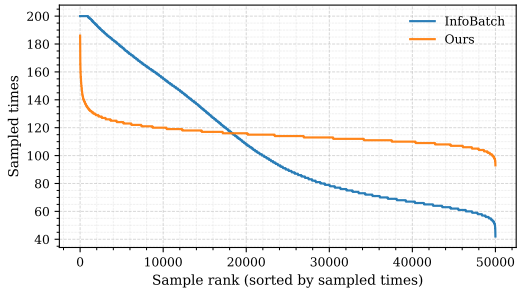


Figure 8: Per-sample usage statistics under 20% symmetric label noise on CIFAR-100 (ResNet-50): samples are sorted by sampled times in descending order. Loss-driven selection exhibits a more concentrated usage distribution, while our method yields a flatter curve, indicating more even sample participation and reduced long-term selection bias.

Table 7: Noise robustness on CIFAR-100 with ResNet-50 under 20% label noise. We report top-1 accuracy (%) for clean and noisy training, and the drop Δ (Clean – Noisy).

Setting	**Selection ratio = 20%**			**Selection ratio = 30%**		
	Clean	**Noisy**	**Δ**	**Clean**	**Noisy**	**Δ**
CLIP	54.10	46.05	8.05	61.90	58.34	3.56
MoSo	-	31.01	-	58.27	43.73	14.54
MoDS	51.89	40.25	11.64	56.00	48.53	7.47
InfoBatch	56.90	54.49	2.41	63.81	55.67	8.14
Ours	**64.29**	**60.33**	3.96	**75.70**	**60.86**	14.84

unchanged (Table 7). Concretely, we randomly sample 20% training instances and replace their labels with a uniformly sampled class index in $\{0, \ldots, 99\}$, and then save the corrupted set in the original CIFAR binary format for a drop-in replacement during training.

We additionally visualize the per-sample usage statistics under label noise to reveal the dynamics of different selection strategies (Figure 8). For loss-driven methods, the usage distribution becomes highly concentrated, indicating that noisy samples with persistently high loss are repeatedly selected. Such behavior amplifies sampling variance and introduces long-term bias in the effective training distribution.

Our method yields a markedly flatter usage curve, indicating more even sample participation over training (Figure 8). This stability arises from relying on semantics-aware scores computed offline and an explicit usage-frequency penalty, rather than noisy, step-wise loss feedback. As a result, our approach mitigates selection-induced bias and maintains robust performance in noisy settings (Table 7 and Figure 8).

B.4 Overall Dynamic Selection Procedure

For completeness, we summarize the overall dynamic data selection procedure in Algorithm 1. Our framework precomputes the representativeness and diversity scores offline, and updates only the scheduler weight and usage-frequency penalty during training. At each epoch, the final sampling score combines representativeness, rarity-based diversity, and the usage-frequency penalty. A short full-data refinement stage can optionally be applied in the last training epochs to further reduce residual selection bias.

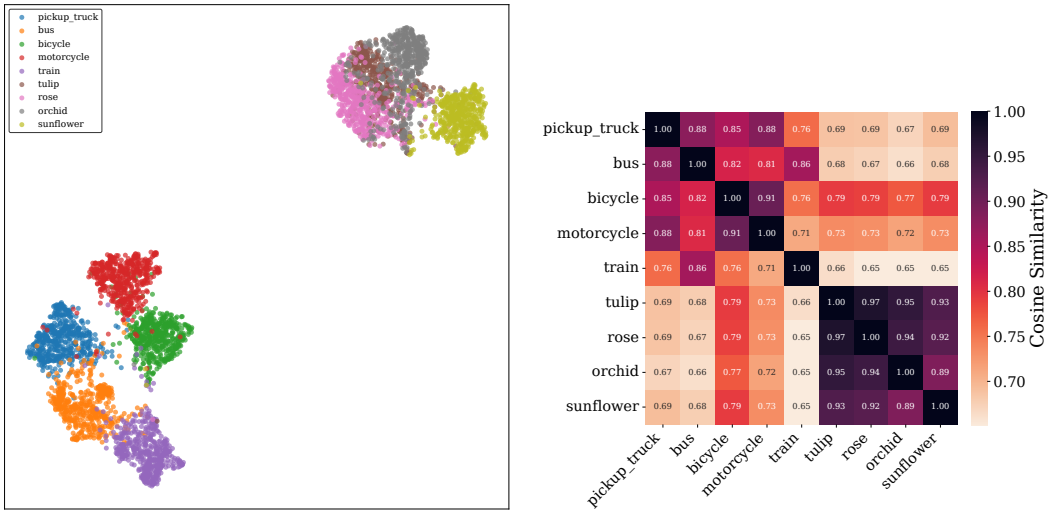
C Experimental Details

C.1 Datasets and Tasks

We evaluated the effectiveness of our approach on multiple image classification datasets, including CIFAR-10/100, Tiny-ImageNet, and ImageNet-1K.

Algorithm 1: Dynamic data selection with full-data refinement

Input: Dataset $\mathcal{D}$, selection ratio p, total epochs T
Input: Refinement ratio ρ, precomputed scores $Rep(i)$ and $Div(i)$
Input: Scheduler $\alpha(t)$ in Eq. 8, penalty weight λ
Output: Epoch-wise selected subsets $\mathcal{S}(t)$
Initialize $u_i(0) \leftarrow 0$ for all $i \in \mathcal{D}$;
for $t \leftarrow 1$ **to** T **do**
  if $t > (1-\rho)T$ **then**
    $\mathcal{S}(t) \leftarrow \mathcal{D}$;
  else
    Compute $\alpha(t)$ by Eq. 8;
    foreach $i \in \mathcal{D}$ **do**
      $Pen(i,t) \leftarrow \lambda \log(1 + u_i(t-1))$;
      $\mathcal{H}(i,t) \leftarrow \alpha(t)\, Rep(i) + (1-\alpha(t))\, Div(i) - Pen(i,t)$;
    Select the top-$\lfloor p|\mathcal{D}| \rfloor$ samples ranked by $\mathcal{H}(i,t)$ to form $\mathcal{S}(t)$;
  Train the model for one epoch on $\mathcal{S}(t)$;
  $u_i(t) \leftarrow u_i(t-1) + \mathbf{1}\{i \in \mathcal{S}(t)\}$ for all $i \in \mathcal{D}$;



(a) UMAP-latent space

(b) Class semantic relationship of SAE activations

Figure 9: Visualization of the SAE latent space. (a) presents the UMAP dimensionality reduction analysis of partial vehicle and plant classes. We observe that classes with similar semantic information are closely distributed in the SAE latent space, while those with high semantic discriminability are far apart. The heatmap in (b) further validates the successful training of our SAE.

CIFAR-10/100. Two classic image datasets, each comprising 50,000 training images and 10,000 test images. Images have a resolution of 32×32 and include 10/100 classes of natural images such as apples, bicycles, fish, etc.

Tiny-ImageNet. Tiny-ImageNet is a downscaled variant of the ImageNet dataset, containing 200 object categories. Each image is resized to 64 × 64 pixels and in RGB format. The dataset consists of 100,000 training images, 10,000 validation images, and 10,000 test images (with labels withheld in the official test set). It is commonly used for prototyping vision models under moderate computational budgets.

ImageNet-1K. ImageNet-1K (also known as ILSVRC-2012) is a widely used benchmark dataset for large-scale image classification. It contains 1,000 object categories and over 1.2 million training images, along with 50,000 validation images and a held-out 100,000-image test set (labels unavailable publicly). Input images are of varying resolutions, typically center-cropped and resized to 224 × 224

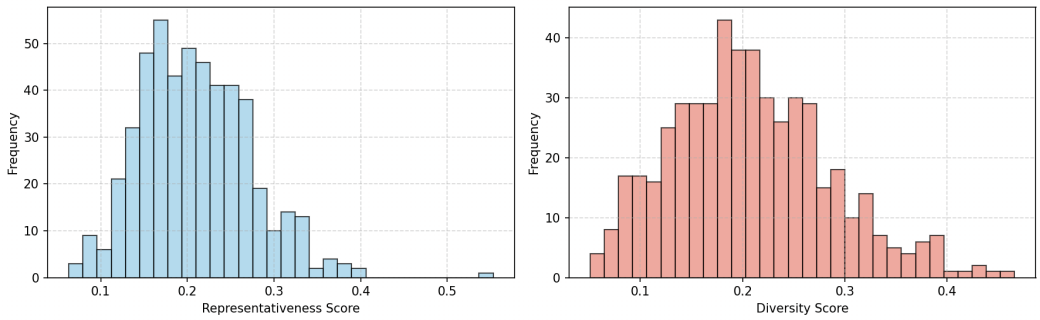


Figure 10: The score distribution of our method.

pixels for standard evaluation. Due to its scale and diversity, ImageNet-1K serves as a cornerstone for evaluating representation learning, model architectures, and training strategies in computer vision.

RSD_15K. RSD-15K is a large-scale, user-annotated dataset for suicide risk detection on social media. It contains roughly 15,000 user-level post histories, making it one of the largest datasets of its kind. Each user is associated with a complete chronological posting timeline to support analysis of risk evolution over time, and annotations follow a four-level risk grading scheme for finer-grained assessment. Labels were produced under expert supervision with multiple rounds of cross-validation to improve reliability and consistency.

C.2 BACKBONES

We evaluate our method on a diverse set of downstream architectures, spanning both convolutional networks and vision transformers, to verify that our selection scores transfer across model families. For CNN backbones, we include ResNet-18/34/50 implemented with the standard BasicBlock/Bottleneck configurations, as well as VGG-16. For transformer backbones, we consider two ViT variants: (i) a ViT-B/16 initialized from ImageNet-1K pretrained weights with the classification head replaced to match the target label space, and (ii) a lightweight ViT-Small tailored for 32×32 images, trained under the same experimental protocol. Unless otherwise noted, all backbones are trained with identical optimization settings, and only the per-epoch data stream is changed according to the selection strategy.

C.3 IMPLEMENT DETAILS

For experiments on CIFAR10/100 and TinyImageNet, we employed a single NVIDIA RTX A6000 GPU. All models were implemented from scratch following the official architectures, with dataset loading performed using the official torchvision library. Except for ViT, all implementations employed the SGD optimizer with momentum set to 0.9. For ResNet architectures, we applied weight decay of 5e-4, a batch size of 128, and an initial learning rate of 0.1. For VGG architectures, we used weight decay of 1e-4, a batch size of 256, and an initial learning rate of 0.1. None of the experiments utilized a warm-up. Unless otherwise specified, the Cos Annealing learning rate scheduler is employed.

For our proposed method, some parameters can be easily set. The minimum weights of the representativeness score $\alpha_{\min} = 0.2$, $t_{mid} = 0.6$, $k = 0.05$, and $\lambda = 0.2$ are default set if not specified. Unless otherwise specified, images have only undergone common augmentation operations such as normalization, random cropping, and horizontal flipping. Following previous research, we trained ResNet-18/50 on CIFAR-10/100, with an SGD optimizer with a momentum of 0.9, weight decay of $5e-4$, a batch size of 128, an initial learning rate of 0.1, and a total training epoch of 200 and VGG-16, with an SGD optimizer with a momentum of 0.9, a weight decay of $1e-4$, a batch size of 128, an initial learning rate of 0.1, and a total training epoch of 200. We trained ResNet-34 on ImageNet-1K using SGD with a momentum of 0.9, a weight decay of 5×10^{-4}, a batch size of 256, an initial learning rate of 0.1, and a total training epoch of 90.

We fine-tune roberta-base (Liu et al., 2019) for RSD_15K. We use the official train/test split, and further split the test set into validation and final test with a 50/50 stratified split for checkpoint

Table 8: **Tiny-ImageNet: comparison against backbone-specific SOTA baselines.** For each backbone (ResNet-50 / VGG-16), we compare with the strongest prior method we reproduce. **Cost** is the normalized training cost (Full-data $= 1.00\times$), and **Speedup**$= 1/\text{Cost}$. Our method uses a 70% (ResNet-50) and 50% (VGG-16) per-epoch selection ratio; baselines are evaluated at their own reported operating points, so speedups are not matched across rows.

Backbone	Method	Acc. (%)	Speedup
ResNet-50	Full-data	63.5	1.00×
	InfoBatch	63.4	**1.71**×
	Ours (70%)	**63.7**	1.26×
VGG-16	Full-data	59.4	1.00×
	CLIP	56.34	1.10×
	Ours (50%)	59.2	**1.72**×

selection and reporting. Our method is instantiated as a static selector on the training split: we compute a combined score $s_i = \text{Rep}(i) + \text{Div}(i)$ and select the top 70% training samples for fine-tuning (3 epochs, AdamW, lr 2×10^{-5}, batch size 16), while evaluation always uses the full validation/test data.

We evaluate on ImageNet-1K using `ImageFolder` with the standard train/val split. Images are preprocessed with RandomResizedCrop(224) and RandomHorizontalFlip for training, and Resize(256) + CenterCrop(224) for validation, followed by ImageNet normalization. The downstream classifier is a ResNet-34 built with the standard `BasicBlock` design, trained for 90 epochs with SGD (momentum 0.9, weight decay 5×10^{-4}, Nesterov) and CosineAnnealingLR ($T_{\max} = 90$, $\eta_{\min} = 10^{-6}$). All runs use distributed training with `torchrun` and `FullyShardedDataParallel`; we adopt `DistributedSampler` for both train and val and aggregate validation metrics via all-reduce across ranks.

C.4 Budget-Matched Analysis of the Full-Data Refinement Stage

Because the main-text numbers in Table 1 use the optional full-data refinement stage (nominal selection ratios do not reflect total exposure), we additionally report strictly budget-matched comparisons on CIFAR-10 (ResNet-18) that hold total exposure fixed. Without refinement, our method obtains **95.9%/94.7%** at **70%/30%** total-exposure budgets; when the last 15% epochs use the full dataset and the preceding per-epoch selection ratios are reduced to **64.71%/17.65%** to keep exposure exactly **70%/30%**, the results are **95.7%/94.3%**. The refinement-free variants are thus **0.2/0.4** points better under equal exposure, confirming that the reported gains do not come from extra full-data training. We therefore treat full-data refinement as an optional stabilization step and report actual exposure and compute explicitly for every setting.

C.5 Tiny-ImageNet Results

For Tiny-ImageNet, we followed the settings of InfoBatch (Qin et al., 2024), using batch size 128, image resize 64, SGD as the optimizer, and a learning rate of 0.1 at the start of cosine annealing. We compare against the SOTA methods on ResNet-50 and VGG-16 respectively; our approach attains the highest accuracy on both backbones, and on VGG-16 it does so with a substantially larger speedup than the strongest baseline.

C.6 Cross-Task Generalization Details

To further examine whether the proposed framework extends beyond standard image classification, we evaluate it on a broader set of task types, including text classification, medical image classification, standard language modeling, and generative multimodal fine-tuning. A summary of the corresponding results is reported in Table 2; here we provide additional context for each task.

Text classification on RSD-15K. We fine-tune RoBERTa on the RSD-15K (Zheng et al., 2025) benchmark under the same general selection framework. Using a 70% selected subset, our method

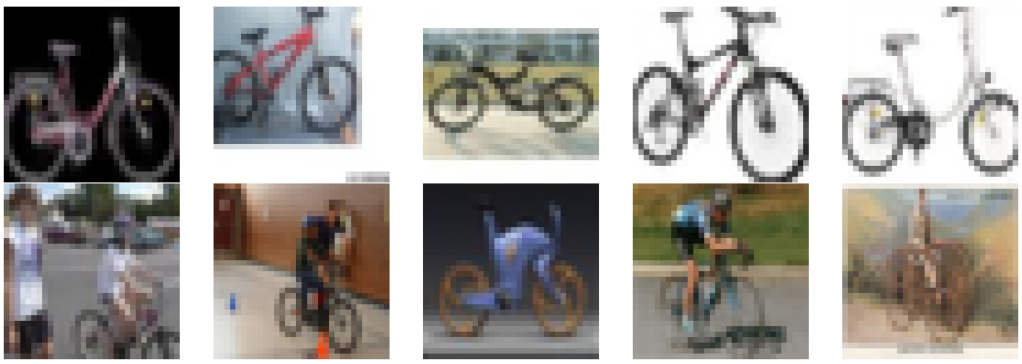


Figure 11: Visualization of the images of class 8 with the highest representativeness and diversity scores. The top ones are of the highest representativeness score, while the bottom images display high-diversity samples.

Table 9: **Architectural generalization and cross-dataset transferability.** Left: accuracy (%) across different model architectures on CIFAR10/100 when training with our method using a 70% per-epoch selection ratio. Right: transferability of the SAE scoring model from ImageNet-1K to CIFAR-10.

(a) Different architectures

Method	CIFAR10		CIFAR100		
	R-18	R-50	R-18	R-50	VGG-16
Full	96.1	**95.8**	78.2	79.1	73.9
Ours	**96.1**	95.6	**78.4**	**79.1**	**74.0**

(b) Cross-dataset transferability

Training Protocol	Selection Ratio per Epoch (%)			
	70	50	30	Full
I-1K → C-10	96.1	95.6	**95.7**	96.1
C-10 → C-10	96.1	95.6	95.3	96.1

improves the final accuracy from **71.0** to **72.5**, indicating that the scoring-and-rotation principle transfers beyond vision classification to pure textual tasks.

Medical image classification on BRISC. We further evaluate the framework on the BRISC (Fateh et al., 2026) four-class medical classification benchmark using VGG16. Under the default setting with 50% selected data, our method reaches **98.3%** accuracy, outperforming both the **96.9%** full-data baseline and a 50% random selection baseline. This result suggests that the proposed framework remains effective even under substantial domain shift.

Language modeling on WikiText-2. To extend the evaluation to a standard next-token prediction setting, we fine-tune GPT-2 (Radford et al., 2019) on WikiText-2(Merity et al., 2016). To remain consistent with the pure-text setting, we train the sparse-unit probe on RoBERTa features. Using only **60%** of the training data selected by our representativeness score, our method achieves a test loss of **3.26**, a perplexity of **26.1**, and a token accuracy of **0.40**, compared with **3.24**, **25.6**, and **0.40** under full-data fine-tuning, respectively. These results provide initial evidence that the framework can extend to standard language modeling with only a small performance gap under data reduction.

Generative multimodal fine-tuning on AQUA. We additionally evaluate the framework on a generative multimodal fine-tuning task, AQUA (Garcia et al., 2020), using LLaVA-v1.5 (Liu et al., 2024) LoRA. Under the default setting with 50% selected data for 6 epochs, our method achieves **34.4** ROUGE-L F1, remaining close to the **34.9** obtained by full-data training under the same epoch budget. Moreover, full-data training for only 3 epochs reaches **33.46**, showing that our selected-data training converges faster under matched training steps. This result indicates that the proposed framework is not restricted to discriminative classification tasks.

C.7 Additional Generalization Results Across Backbones and Datasets

Our framework is designed to be model-agnostic: representativeness and diversity are computed once in a plug-in feature space with an SAE-based sparse-unit probe, while online training only updates the usage-frequency penalty and the scheduler weight. We therefore further examine its transferability across model families and datasets.

Cross-backbone transferability. Table 9 and Figure 4b show that the same scoring module remains effective across different downstream architectures, including CNNs and vision transformers. Under the same selection ratio, the method consistently yields competitive accuracy–efficiency trade-offs and remains close to full-data training across architectures. This suggests that emphasizing dataset-level coverage of high-frequency factors together with process-level rotation provides stable selection signals that transfer across model families.

Cross-dataset transferability. We also test whether the sparse-unit probe trained on one dataset can be reused on another. Specifically, we reuse the SAE scoring model trained on ImageNet-1K to score CIFAR-10 and compare it against a probe trained directly on CIFAR-10. As shown in the right panel of Table 9, the ImageNet-trained probe yields performance comparable to the CIFAR-trained probe. This suggests that the sparse-unit probe captures reusable factor structure rather than a narrow dataset-specific heuristic.

Interpretation. Taken together, these results indicate that the framework generalizes at two levels: first, across downstream model families, since the selection scores are not tied to architecture-specific gradients or losses; and second, across datasets, since the sparse-unit probe appears to capture shared factor structure that can be reused beyond the original training set.

C.8 Robustness to Plug-in Feature Quality

Since our default instantiation uses CLIP as the plug-in feature extractor, a natural concern is whether the effectiveness of the framework depends on CLIP having strong task-specific zero-shot performance. Figure 4a provides evidence against this interpretation.

Tiny-ImageNet with VGG16. On Tiny-ImageNet, CLIP zero-shot classification achieves only **0.45%** accuracy. Despite this extremely weak task-specific zero-shot performance, full-data training reaches **59.4%**, and our method reaches **59.2%**, within **0.2** points of full data. This suggests that the plug-in feature space remains useful for factor-based sample ranking even when it is not directly predictive for the downstream task.

BRISC medical classification. A similar phenomenon appears on the BRISC medical classification benchmark, which exhibits substantial domain shift relative to CLIP's pretraining distribution. Here CLIP zero-shot accuracy is only **22.1%**, while full-data fine-tuning reaches **96.9%**, and our method with **50%** selected data further improves to **98.3%**. Again, effective data selection is possible even when the plug-in feature extractor itself is weak as a zero-shot solver.

Interpretation. These results support our interpretation that the plug-in feature space is used as a *factor-analysis probe* for sample ranking, rather than as a direct zero-shot predictor for the downstream task. In other words, the framework does not require the plug-in encoder to solve the task well by itself; it only requires the encoder to provide a broadly useful representation space in which high-frequency and rare factors can be meaningfully identified.

C.9 Robustness to Learning-Rate Schedulers

We report results under three widely used learning-rate schedulers: MultiStepLR, CosineAnnealingLR, and OneCycleLR.

MultiStepLR. We follow the standard configuration in Yang et al. (2025a), dividing the learning rate by 5 at epochs $\{60, 120, 180\}$.

Table 10: **Top-1 accuracy (%)** and training time (GPU hours) on CIFAR-100 (ResNet-18) under different learning-rate schedulers. ΔAcc denotes the accuracy difference relative to full-data training under the same scheduler. Lower time is better.

LR Scheduler	Method	Acc	ΔAcc	Time
MultiStep	Full	77.3	–	0.7
	Ours	**77.6**	+0.3	**0.4**
OneCycle	Full	78.2	–	0.7
	Ours	**78.2**	+0.0	**0.4**
CosAnneal	Full	78.2	–	0.7
	Ours	**78.4**	+0.2	**0.4**

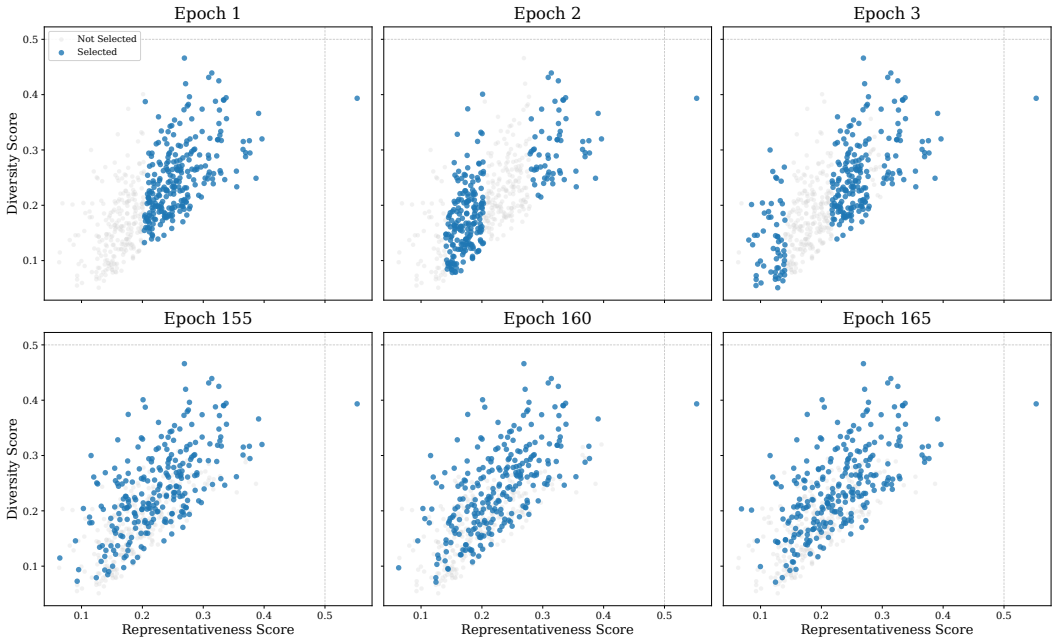


Figure 12: In the early stages of training, samples with high representativeness scores are primarily selected, while in the middle and late stages, samples are chosen by comprehensively considering both representativeness and diversity scores. The continuous epoch sampling visualization in the figure also intuitively demonstrates how our usage-frequency penalty functions.

OneCycleLR. We follow the setup in Qin et al. (2024) and use the same total training budget.

Results. As shown in Table 10, our method maintains comparable (often slightly higher) accuracy to full-data training across all schedules, while consistently reducing training time. These results indicate that the efficiency gains are stable under different learning-rate schedules and require no schedule-specific tuning.

D Additional Visualization

D.1 Selection Dynamics Visualization

To gain deeper insight into our semantics-aware data selection, we conduct three complementary visual analyses.

First, temporal sampling dynamics Figure 12 track how samples from class 77 are selected across epochs. Early epochs (1–80) predominantly choose images with a high representativeness score, while later epochs (120–180) increasingly incorporate challenging cases manifesting high diversity.

This evolution validates our design: the framework smoothly transitions from representativeness learning to boundary exploration. Simultaneously, our usage-based penalties play a crucial role across different stages. As shown in Figure 12, our sampling pattern evolves with the sampling epoch, shifting from representation-based scoring to a balanced approach that integrates both representation and diversity scores. Concurrently, penalties introduce distinct samples within adjacent epochs to ensure sampling coverage, thereby safeguarding process-level diversity.

Secondly, our score distribution analysis reveals the unique characteristics of each metric. To eliminate dimensional differences, we normalized both scores using the min-max method before applying them to the data selection process. The figure below illustrates the distribution of our two scores. As seen, both scores exhibit near-normal distributions with means clustered around 0.2. This ensures a relatively unified scoring system, preventing any single score with an excessively large absolute value from disproportionately influencing the overall composite score.

D.2 Qualitative Examples

Analysis in Figure 11 of high-scoring samples revealed the semantic meanings underlying the ratings. The top five images showcase high-representativeness score samples, while the bottom images display high-diversity score samples. The most representative bicycle images consistently feature a frontal perspective, clear outlines, simple backgrounds, and unobstructed views of the entire vehicle, the core characteristics of this category. Conversely, the most diverse samples exhibit semantic variability: obscured bicycles, cyclists on bikes, blurred images, and complex backgrounds. These are not outliers but marginally valuable cases that optimize decision boundaries.

E Related Work

Data Selection. Data selection aims to reduce training cost by identifying informative subsets while preserving model performance. Existing methods are commonly categorized into *static* and *dynamic* strategies according to whether the selected subset is fixed before training or updated during training.

Static selection methods select a subset once before training. Representative directions include coreset formulations, geometry-based coverage, diversity-preserved heuristics, and importance-based scoring. For example, K-Center (Sener & Savarese, 2018) selects samples by minimizing the maximum distance from each data point to its nearest selected center, and Herding (Welling, 2009) selects samples whose feature centroid approximates that of the full dataset, with more recent coreset formulations scaling such coverage ideas to data-efficient training of large networks (Mirzasoleiman et al., 2020). Importance-based scoring instead keeps the highest-value examples under criteria such as loss or gradient statistics (Paul et al., 2021), self-supervised difficulty scores (Sorscher et al., 2022), prediction entropy (Coleman et al., 2020), or closeness to feature medians (Xia et al., 2023). Other methods jointly consider multiple criteria such as representativeness, diversity, importance, and difficulty. G-DIG (Pan et al., 2024) combines gradient-based quality estimation with diversity control, $\mathbb{D}^2$ (Maharana et al., 2024) balances diversity and difficulty via message passing, and diversity-driven selection methods encourage broader coverage in representation or activation spaces (Yang et al., 2025b; Agarwal et al., 2020). CLIP-based data selection (Yang et al., 2025a) further shows that a strong pretrained feature space can improve the robustness and transferability of selection scores. InfoMax (Tan et al., 2025) formulates data selection as a discrete optimization problem that maximizes information while reducing redundancy, thereby balancing individual sample value and pairwise overlap. Coverage-centric coresets avoid the collapse of importance-based selection onto a few classes under aggressive pruning rates (Zheng et al., 2023), while trajectory-based scoring derives per-sample importance from training dynamics, e.g., via correlation of loss trajectories (CLD) (Nagaraj et al., 2025), prediction-uncertainty variation (Dyn-Unc) (He et al., 2024), temporal dual-depth contribution scoring (TDDS) (Zhang et al., 2024), and early-training difficulty combined with prediction uncertainty (DUAL) (Cho et al., 2025). These methods mainly construct a *fixed* subset; although they address the accuracy drop at high pruning rates, they commit to a single view of the data distribution across the whole training run.

Dynamic selection methods update the training subset along the optimization process and often achieve a better trade-off between training acceleration and final accuracy. Many dynamic methods rely on model-dependent feedback signals. UCB and ϵ-Greedy (Raju et al., 2021) select examples

based on uncertainty or difficulty estimated from the current model state, while InfoBatch (Qin et al., 2024) dynamically selects high-loss samples to accelerate training. Remarkably, RS2 (Okanovic et al., 2024) shows that simply resampling a random subset every epoch is already a strong baseline, highlighting the importance of process-level re-exposure over static selection, and RCAP (Hassan et al., 2025) further regularizes model-retraining dynamics during dynamic resampling. Such instantaneous feedback can identify examples that are currently informative for the model, but it may also amplify noisy or unstable signals, especially when high-loss or hard samples are repeatedly selected across epochs. This can lead to sample monopoly and induce a biased effective training distribution.

Our work differs from prior static and dynamic selection methods along two axes. First, we reformulate representativeness and diversity at the *factor* level: weighted coverage of high-frequency dataset-level factors versus exposure to rare ones, rather than via local geometric proxies or instantaneous model-dependent feedback. Second, we treat diversity as a *process-level* constraint over the training trajectory and organize the selection process itself: a curriculum scheduler shifts emphasis from high-frequency to rare factors, and a usage-frequency penalty prevents sample monopoly across epochs. This design yields a model-agnostic dynamic selection framework: representativeness and diversity scores are computed offline in a plug-in feature space, while online training only updates the scheduler and usage-frequency penalty. It is complementary to training-dynamics scoring methods (CLD, Dyn-Unc, TDDS, DUAL), which derive importance from the proxy training trajectory, whereas we derive it from a pretrained representation reused throughout training; and unlike RS2, which randomizes re-exposure, we guide it by semantic factor priorities.

Sparse Autoencoders. Sparse autoencoders (SAEs) are unsupervised representation learners that impose explicit sparsity constraints, such as ℓ_1 regularization or hard top-k activation, on hidden units (Gao et al., 2025). By reconstructing inputs using only a small subset of active neurons, SAEs encourage localized and disentangled feature bases, where individual units can correspond to coherent semantic concepts. Since the introduction of k-sparse autoencoders (Makhzani & Frey, 2013), SAEs have been used in representation learning, denoising, anomaly detection, and more recently, large-model interpretability. Recent studies show that sparse decomposition of model activations can reveal interpretable semantic features in language models (Huben et al., 2024) and multimodal models (Lou et al., 2025).

SAE-based data selection has also begun to receive attention. For example, prior work applies SAEs to language data selection by encouraging selected subsets to activate diverse sparse units (Yang et al., 2025b). SAE-V (Lou et al., 2025) extends SAE-based analysis to multimodal representations and studies image-text alignment through sparse features. In contrast, our use of SAEs is not to directly maximize subset-level activation diversity alone. Instead, we use sparse units as a probe for dataset-level factor statistics: high-frequency units are used to define weighted representativeness, while rare units support process-level diversity. This sparse factorization provides a simple and interpretable interface for constructing offline scores that can be reused across downstream models.