Label-Efficient Dataset Pruning via Semi-Supervised Pseudo-Labeling

Yeseul Cho[1]*, **Baekrok Shin**[2], **Changmin Kang**[2], **Chulhee Yun**[2]
[1]Nota Inc., [2]Kim Jaechul Graduate School of AI, KAIST
yeseul.cho@nota.ai
{br.shin,cmkang8128,chulhee.yun}@kaist.ac.kr

Abstract

Dataset pruning reduces the storage and training costs of deep learning by selecting an informative subset from a large dataset. However, most existing pruning methods require fully labeled data, which limits their applicability in realistic settings where unlabeled data are abundant and annotation is costly. Recent label-free pruning methods address this issue, but they rely on features from pretrained models to estimate example difficulty. This dependence can be unreliable when the target dataset differs substantially from the pretraining distribution. We propose **SEMIPRUNE**, a label-efficient dataset pruning framework that uses a small randomly labeled subset and semi-supervised learning to generate pseudo-labels for unlabeled data, enabling existing supervised pruning methods to operate on the resulting pseudo-labeled training pool. We then estimate example difficulty from pseudo-label-induced training dynamics and select a coreset. By learning directly from the target dataset, our method better captures the target distribution and provides more reliable signals for difficulty estimation and coreset selection. We validate our approach on domain-specific, image-corrupted, and long-tailed datasets, where it achieves state-of-the-art performance among label-free and label-efficient baselines, while also demonstrating competitive performance on standard benchmarks. [2]

1 Introduction

Modern deep learning relies heavily on massive datasets, resulting in substantial storage and training costs (Achiam et al., 2023; Touvron et al., 2023). To address this challenge, dataset pruning aims to remove redundant examples while retaining the most informative subset of the dataset, namely the *coreset*. Existing pruning methods have demonstrated strong performance when models are trained on the retained coreset (Coleman et al., 2020; Paul et al., 2021; Zheng et al., 2023; Maharana et al., 2024; He et al., 2024; Zhang et al., 2024; Cho et al., 2025). However, most of these methods require an initial training stage on fully labeled data. Since fully labeled datasets are limited in practice, and annotating all candidate examples solely for pruning can be prohibitively expensive, developing pruning methods under a limited annotation budget is highly desirable.

Several label-free methods (Sorscher et al., 2022; Zheng et al., 2025; Griffin et al., 2026) reduce this annotation cost by estimating example importance from features extracted by pretrained models. However, their reliability depends heavily on how well the pretrained feature geometry aligns with the target data. When the target distribution differs from that of the pretraining data, such as in domain-specific, image-corrupted, or class-imbalanced datasets, feature-based difficulty estimates may fail to reflect the true structure of the data.

*Work done at KAIST.
[2]Code is available at https://github.com/cyseul/SemiPrune.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

In contrast, semi-supervised learning (SSL) can effectively learn the distribution of a target dataset using only a small budget of labeled examples (Sohn et al., 2020; Wei et al., 2021). Leveraging this ability, we propose SEMIPRUNE, a framework for label-efficient dataset pruning that first generates pseudo-labels through SSL and then applies existing supervised pruning methods. Because the SSL model is trained directly on the target dataset, its pseudo-labels better reflect the target class structure and provide more reliable supervision for difficulty estimation. This approach requires only a small fraction of examples to be initially annotated, reducing the annotation cost of coreset selection relative to fully supervised pruning.

We evaluate our framework in settings where target-specific difficulty estimation is particularly important. These include domain-specific datasets such as Food-101 and SUN397, corrupted variants of CIFAR-100 and Tiny-ImageNet, and long-tailed datasets including Caltech-101 and a long-tailed variant of CIFAR-100. Across these challenging settings, our method achieves state-of-the-art performance among pruning baselines that use either no labels or only a small fraction of labeled data. We further evaluate our method on standard benchmarks, including CIFAR-10, CIFAR-100, and ImageNet-1K, showing that it remains competitive even when pretrained features are already well aligned with the target data.

2 Related Works

2.1 Dataset Pruning

Fully-labeled approaches. Supervised pruning methods typically assign importance scores to examples and retain those with the highest scores (Toneva et al., 2019; Coleman et al., 2020; Paul et al., 2021; Zhang et al., 2024). However, selecting only the top-k examples can reduce coverage of the data distribution, substantially degrading performance at high pruning ratios. To address this issue, several recent approaches have proposed sampling methods that can be combined with the scoring metric (Xia et al., 2023; Zheng et al., 2023; Maharana et al., 2024; Cho et al., 2025). Some methods instead formulate pruning as an optimization problem and solve it greedily under certain constraints (Borsos et al., 2020; Killamsetty et al., 2021a).

Label-free approaches. PROTOTYPICALITY (Sorscher et al., 2022) selects examples based on their distance to cluster centroids in the embedding space learned by SWaV (Caron et al., 2020). More recently, ELFS (Zheng et al., 2025) generates pseudo-labels through deep clustering on DINO embeddings (Adaloglou et al., 2023; Caron et al., 2021; Oquab et al., 2024), trains a scoring model on the pseudo-labeled pool, and selects a coreset using AUM-based double-end pruning. Another zero-shot pruning method, ZCORE (Griffin et al., 2026), constructs a zero-shot embedding space using a foundation model (Radford et al., 2021) and then iteratively samples lower-dimensional embedding subspaces to identify examples that provide broad coverage of features while reducing redundancy.

Few-label approaches. Recently, label-efficient pruning methods that require only a small fraction of labeled samples have also been proposed. SCORE EXTRAPOLATION (Schmidt et al., 2026) trains a scorer on a small labeled subset and extrapolates importance scores to the remaining examples using k-nearest neighbors and graph neural networks. Our framework instead uses the initial labels for SSL-based pseudo-label generation, enabling supervised scoring on the full pool.

2.2 Pseudo-labeling for Unlabeled Data

Pseudo-labeling extracts supervisory signals from unlabeled data, but its function differs depending on the available supervision. In fully unsupervised representation learning, pseudo-labels are often obtained through deep clustering, where examples are grouped in a learned feature space and the resulting cluster indices are used as training targets (Caron et al., 2018; Asano et al., 2020; Caron et al., 2020; Van Gansbeke et al., 2020). Recent deep clustering methods further show that utilizing rich pretrained representations from self-supervised ViTs and vision-language models such as CLIP (Radford et al., 2021), DINO (Caron et al., 2021), and DINOv2 (Oquab et al., 2023) substantially improves clustering quality on standard benchmarks (Adaloglou et al., 2023; Cai et al., 2023). However, because these assignments are produced without class annotations, they do not

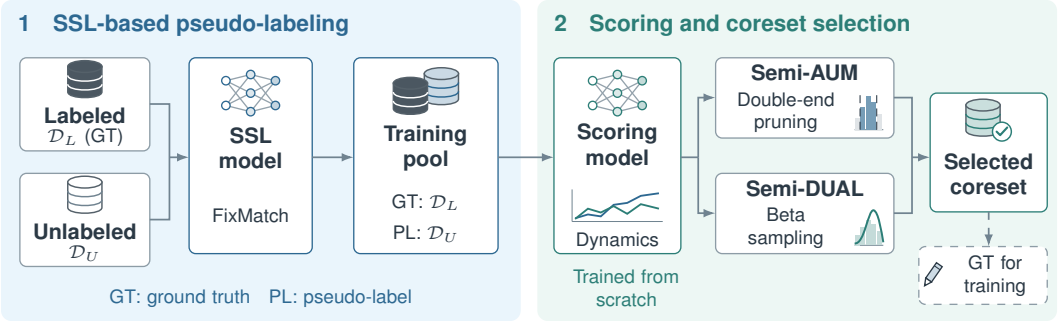


Figure 1: Overview of SEMIPRUNE. An SSL model uses a small labeled subset to generate pseudo-labels, retaining initial GT labels. A separate scorer uses the combined pool for AUM-based double-end pruning or DUAL-based Beta sampling. Selected examples receive GT labels for downstream training and evaluation.

necessarily correspond to semantic classes. Their primary role is instead to impose structure on the representation space by grouping examples with similar learned features.

In semi-supervised learning, pseudo-labeling serves a more class-directed purpose, as a small labeled subset anchors the predictions on unlabeled data. Lee (2013) introduces pseudo-labeling as a simple method that treats confident predictions on unlabeled examples as training targets. Modern consistency-based methods build on this idea by generating pseudo-labels from weakly augmented images and using them to supervise strongly augmented counterparts. FixMatch filters these predictions with a fixed high-confidence threshold (Sohn et al., 2020), while later methods refine this process through adaptive thresholding (Zhang et al., 2021; Wang et al., 2023). Since these pseudo-labels are guided by target-class supervision, they are more likely to approximate semantic classes and can propagate label information from labeled to unlabeled samples (Wei et al., 2021).

This distinction matters for dataset pruning because pseudo-labels provide the supervision used to estimate example difficulty. SSL offers a way to adapt this supervision to the target classes using a small labeled subset.

2.3 Orthogonal Works

Active learning also aims to reduce annotation cost, but iteratively acquires labels for informative samples through oracle queries during training (Ash et al., 2019; Killamsetty et al., 2021a; Hacohen et al., 2022). Consequently, its sample selection is closely tied to the current state of the model and proceeds progressively over the training process. In contrast, our method starts from a fixed, small randomly selected labeled subset, uses it to train a model under an SSL objective, and then constructs a coreset in a one-shot manner based on the pseudo-labels generated by that model.

Our work also differs from dynamic dataset pruning methods (Qin et al., 2024; Okanovic et al., 2024; Yuan et al., 2025; Wu et al., 2025). Dynamic pruning methods update the retained subset during training and maintain access to the full data pool (Qin et al., 2024; Okanovic et al., 2024; Yuan et al., 2025; Wu et al., 2025). We instead select a fixed coreset offline and train without revisiting discarded examples. Appendix C.2 compares these settings using InfoBatch and Partial Forward Blocking under matched optimizer updates.

We also distinguish our method from pruning approaches designed for SSL frameworks (Killamsetty et al., 2021b). Unlike these methods, which use the retained coreset within SSL training, we use SSL only to generate pseudo-labels and then train the model on the selected coreset under standard supervised learning.

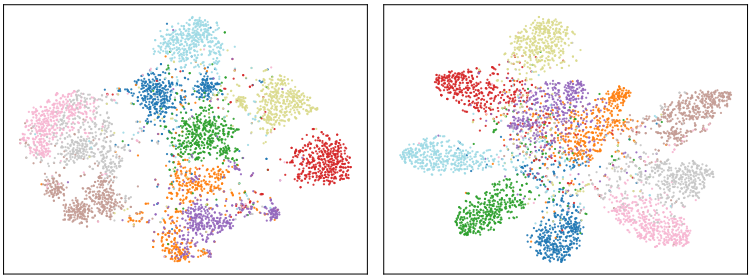


Figure 2: t-SNE visualization of Food-101 embeddings, colored by ground-truth labels. **(Left)** Embeddings from ImageNet-pretrained DINO. **(Right)** Embeddings from the SSL-trained model.

3 Methodology

3.1 Method Overview

Figure 1 summarizes the two-stage pipeline of SEMIPRUNE: SSL-based pseudo-label generation followed by supervised scoring and coreset selection. We train an SSL model using a small randomly labeled subset and the remaining unlabeled data. Its pseudo-labels, together with the retained ground-truth labels, supervise a separate model trained from scratch. We use this model's training dynamics to score examples and select a coreset.

Following ELFS (Zheng et al., 2025), we use pseudo-labels to enable supervised pruning without full-pool annotation. Our key distinction is to generate these labels through SSL on the target data, using a small labeled subset to guide target adaptation.

3.2 Limitations of Deep-clustering-based Pseudo-labeling

To motivate this choice of pseudo-label generator, we examine settings in which deep-clustering-based pseudo-labeling becomes unreliable. The deep clustering approach adopted in ELFS (Zheng et al., 2025) relies on features extracted from an ImageNet-pretrained foundation model (Oquab et al., 2024; Adaloglou et al., 2023). Therefore, its pseudo-label quality critically depends on whether the pretrained feature space is well aligned with the semantic structure of the target dataset.

First, when the target dataset is weakly aligned with the pretraining data, deep clustering may fail to recover semantically meaningful clusters. Rather than capturing the task-relevant class structure, it can group examples according to superficial visual similarity. This issue becomes more evident in domain-specific or fine-grained datasets such as Food-101, whose discriminative factors differ substantially from those captured by generic ImageNet-pretrained representations.

Figure 2 compares Food-101 embeddings from ImageNet-pretrained DINO (Caron et al., 2021) and a FixMatch-trained model (Sohn et al., 2020), using 500 examples from each of 10 randomly selected classes. The SSL-trained model exhibits clearer class separation despite using only a 10% initial label budget. This observation is consistent with the higher pseudo-label accuracy reported in Table 1.

Second, even when the target classes are reasonably aligned with the pretraining distribution, image corruptions can obscure class structure in pretrained representations, making clustering-based pseudo-labels less reliable. Figure 4 in Appendix A.1.1 compares the representations of the pretrained and SSL-trained models under corruption.

Third, class imbalance can distort feature-space clustering: broad majority-class regions may fragment, while sparse minority-class examples are absorbed into nearby clusters. The resulting pseudo-labels can misrepresent class identities and proportions, especially for underrepresented classes. SSL uses the labeled subset to guide predictions toward target classes and preserve class structure (Figure 6 and Table 1).

Across the evaluated settings, SSL-based pseudo-labeling achieves higher accuracy than deep clustering (Table 1), motivating its use as supervision for downstream pruning. These findings are consistent with theoretical work showing that SSL can exploit the underlying data distribution under suitable assumptions (Wei et al., 2021).

Table 1: Pseudo-label accuracy of deep clustering and SSL across diverse datasets. In dataset names, C denotes CIFAR and IN denotes ImageNet. For long-tailed datasets, we report balanced accuracy in parentheses. We adopt TEMI (Adaloglou et al., 2023) for deep clustering, and use a 10% initial label budget to train FixMatch (Sohn et al., 2020) as an SSL algorithm.

Method	Pseudo-label ACC (%)								
	Domain-specific		Image-corrupted		Long-tailed		Standard		
	Food	SUN	C-100	Tiny-IN	C-100	Caltech	C-10	C-100	IN-1K
Deep-Clust.	42.8	38.1	40.3	40.9	21.9 (18.9)	24.3 (12.7)	95.1	66.6	57.3
Semi-Sup.	61.7	49.2	65.4	58.0	65.4 (65.0)	52.6 (44.5)	95.4	74.1	64.0

3.3 SSL-based Pseudo-label Generation

We implement this pseudo-label generation stage using FixMatch (Sohn et al., 2020), training for up to 100 epochs on each dataset.[3] Given a small labeled subset $\mathcal{D}_L$ with $(\mathbf{x}_i, y_i) \in \mathcal{D}_L$ and an unlabeled pool $\mathcal{D}_U$ with $|\mathcal{D}_L| \ll |\mathcal{D}_U|$, FixMatch trains a model using the cross-entropy loss on $\mathcal{D}_L$ together with consistency regularization between weakly and strongly augmented views of unlabeled examples. Let ϕ denote the parameters of the FixMatch-trained model. For each unlabeled example $\mathbf{x}_i \in \mathcal{D}_U$, the model predicts a class distribution $p(\cdot \mid \mathbf{x}_i; \phi)$, and we assign the pseudo-label $\hat{y}_i = \arg\max_c p(c \mid \mathbf{x}_i; \phi)$. We then combine initially labeled examples, for which $\hat{y}_i = y_i$, with pseudo-labeled examples to form a pseudo-labeled training pool $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^N$, which is used in the subsequent score computation stage.

3.4 Dataset Pruning with Pseudo-label-based Scores

After constructing the pseudo-labeled training pool $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, \hat{y}_i)\}_{i=1}^N$, we train a supervised model on $\mathcal{D}_{\text{train}}$ from scratch and collect its prediction trajectories over T epochs. Let θ_t denote the model parameters at epoch t of supervised training, and let $p(c \mid \mathbf{x}_i; \theta_t)$ denote the predicted probability of class c for sample $\mathbf{x}_i$ under this model. These dynamics are used to compute example-wise pruning scores with two existing criteria: AUM with double-end pruning from Zheng et al. (2025) and DUAL with Beta sampling from Cho et al. (2025). We refer to their pseudo-label-based variants as **SEMI-AUM** and **SEMI-DUAL**, respectively.

3.4.1 SEMI-AUM with Double-end Pruning

We first follow the double-end pruning strategy introduced by ELFS (Zheng et al., 2025) using the AUM score (Pleiss et al., 2020). For each example $(\mathbf{x}_i, \hat{y}_i)$, the margin at epoch t is $M(\mathbf{x}_i, \hat{y}_i; \theta_t) := p(\hat{y}_i \mid \mathbf{x}_i; \theta_t) - \max_{c \neq \hat{y}_i} p(c \mid \mathbf{x}_i; \theta_t)$, and the AUM score is $\mathrm{AUM}(\mathbf{x}_i, \hat{y}_i) := \frac{1}{T}\sum_{t=1}^T M(\mathbf{x}_i, \hat{y}_i; \theta_t)$. Low-AUM examples are hard to learn, whereas high-AUM examples are easily learned and may be less informative. Following ELFS, we adopt a double-end pruning strategy to select examples from the middle of the AUM distribution using an interval $[\alpha, \alpha + \Delta]$, where Δ is fixed by the target coreset size and α is tuned for each Δ.

3.4.2 SEMI-DUAL with Beta Sampling

We also consider DUAL with Beta sampling, introduced by Cho et al. (2025). It computes an example score from early-training dynamics by combining prediction difficulty and prediction uncertainty. For a window of length J starting at epoch $k \in \{1, \ldots, T - J + 1\}$, we write $p_{i,t} := p(\hat{y}_i \mid \mathbf{x}_i; \theta_t)$ and define the average probability assigned to the pseudo-label and its standard deviation as

$$\bar{p}_{k,J} := \frac{1}{J}\sum_{j=0}^{J-1} p_{i,k+j}, \qquad \sigma_{k,J} := \sqrt{\frac{1}{J-1}\sum_{j=0}^{J-1}\left(p_{i,k+j} - \bar{p}_{k,J}\right)^2}.$$

Here, $1 - \bar{p}_{k,J}$ measures prediction difficulty, while $\sigma_{k,J}$ measures prediction uncertainty within the window. The original DUAL score within this window is calculated as $\mathrm{DUAL}_{k,J}(\mathbf{x}_i, \hat{y}_i) := (1 - \bar{p}_{k,J})\,\sigma_{k,J}$. To control the relative contribution of uncertainty under pseudo-label supervision,

[3] We shorten the original FixMatch schedule of about one million training iterations to reduce cost; Appendices E.2 and E.1 report pseudo-label generation time and end-to-end costs, respectively.

we introduce an exponent $\gamma > 0$: $\mathrm{DUAL}^{(\gamma)}_{k,J}(\mathbf{x}_i, \hat{y}_i) := (1 - \bar{p}_{k,J})\,\sigma^{\gamma}_{k,J}$, where $\gamma = 1$ recovers the original score. The final example score averages these window-level scores:

$$\mathrm{DUAL}^{(\gamma)}(\mathbf{x}_i, \hat{y}_i) := \frac{1}{T - J + 1} \sum_{k=1}^{T-J+1} \mathrm{DUAL}^{(\gamma)}_{k,J}(\mathbf{x}_i, \hat{y}_i).$$

We then select the coreset following the original DUAL+BETA procedure (Cho et al., 2025) to adjust the sampling distribution according to the pruning ratio. Let $r \in [0, 1]$ denote the pruning ratio and μ_D be the prediction mean of an example with a high DUAL score. For each r, the Beta parameters are defined as

$$\beta_r = C(1 - \mu_D)(1 - r^{c_D}), \qquad \alpha_r = C - \beta_r, \tag{1}$$

where $C > 0$ is a concentration constant and $c_D \geq 1$ controls the evolution rate of the distribution. Following the original DUAL setting, we keep C fixed at 16 and do not tune it. Because the mean of $\mathrm{Beta}(\alpha_r, \beta_r)$ is $\alpha_r/(\alpha_r + \beta_r)$, this choice shifts the sampling mean from μ_D at small pruning ratios toward 1 at large pruning ratios. Therefore, as pruning becomes more aggressive, the sampler increasingly favors easier examples.

3.4.3 Hyperparameter Search with Pseudo-labeled Validation Set

Following ELFS (Zheng et al., 2025), we use a 10% held-out validation split for hyperparameter search, where the validation labels are given by the generated pseudo-labels. For AUM with double-end pruning, we tune the cutoff ratio used to remove hard examples. For DUAL with Beta sampling, we fix the window length to $J = 10$ and tune the score-computation epoch T and the dataset-specific parameter c_D. We use $\gamma = 1$ except on CIFAR-10, CIFAR-100, and ImageNet, where it is tuned (Appendix B); Table 16 reports sensitivity to γ.

4 Experiments

4.1 Experimental Setup

Datasets. We primarily evaluate our framework under the challenging settings motivated in Section 3: domain mismatch (Food-101 (Bossard et al., 2014) and SUN397 (Xiao et al., 2010)), feature-space distortion (30% corrupted CIFAR-100 and Tiny-ImageNet following Xia et al. (2023)), and class imbalance (Caltech-101 (Fei-Fei et al., 2004) and long-tailed CIFAR-100 with an imbalance factor of 0.1). We additionally evaluate on standard benchmarks, including CIFAR-10/100 (Krizhevsky, 2009) and ImageNet-1K (Deng et al., 2009).

Baselines. We compare against RANDOM, PROTOTYPICALITY (Sorscher et al., 2022), ZCORE (Griffin et al., 2026), SCORE EXTRAPOLATION (Schmidt et al., 2026), and two ELFS variants: the default with DINO features and a self-trained SwAV encoder on the target data (Zheng et al., 2025). We evaluate two variants of our method: **SEMI-AUM+CUTOFF** and **SEMI-DUAL+BETA**. We report the original supervised DUAL+BETA (Cho et al., 2025) as a representative fully supervised method (denoted as FULLY SUPERVISED in all tables). Across all results, we highlight the best and second-best results among methods that use either no labels or only a small fraction of labeled data, excluding the fully supervised baseline from this ranking. The best result is shown in **bold**, and the second-best is underlined. Full experimental and baseline details are provided in Appendix B.

Evaluation protocol. Unless otherwise stated, the 10% label budget refers to the initial annotations used for SSL-based pseudo-label generation and coreset selection. For downstream evaluation, the selected coreset is annotated with ground-truth labels and a model is trained on it from scratch. Thus, this initial budget does not include all annotations required for final coreset training.

4.2 Coreset Performance across Diverse Settings

We evaluate SEMIPRUNE under the three challenging data regimes that motivate our framework: domain mismatch, image corruption, and class imbalance. We then consider standard benchmarks, where pretrained representations are expected to provide a stronger basis for feature-based label-free pruning methods.

4.2.1 Domain Mismatch

Table 2: Coreset performance on Food-101 and SUN397. We report the average test accuracy over five runs. The full-dataset test accuracies are 80.3% and 61.4%, respectively.

Label Usage	Pruning Rate	Food-101					SUN397				
		30%	50%	70%	80%	90%	30%	50%	70%	80%	90%
With Labels	FULLY SUPERVISED	80.2	77.5	71.3	67.2	55.3	60.7	56.1	50.0	49.3	40.4
Without Labels	RANDOM	76.6	72.9	63.9	55.5	37.7	57.5	52.9	42.8	41.0	30.4
	PROTOTYPICALITY	72.3	66.9	56.6	47.1	28.9	59.1	52.9	39.6	33.9	21.0
	ELFS (DINO)	77.2	73.3	65.6	61.1	49.2	60.6	**56.1**	46.7	43.7	33.7
	ELFS (SELF-ENCODER)	73.7	62.8	58.6	52.9	40.0	59.5	55.1	45.2	41.9	31.3
	ZCORE	76.7	72.1	63.4	56.3	37.8	59.4	55.0	44.7	42.0	29.9
With 10% Labels	SCORE EXTRAPOLATION	76.3	71.8	63.2	57.6	42.3	58.7	53.6	43.9	39.9	30.0
	SEMI-AUM+CUTOFF	**77.9**	**73.9**	**67.2**	**63.1**	51.2	**60.7**	55.8	**47.7**	**44.1**	**35.0**
	SEMI-DUAL+BETA	**77.9**	73.4	66.3	62.2	**51.8**	60.2	**56.1**	47.3	43.0	33.9

We first consider datasets whose visual characteristics differ substantially from ImageNet-1K, the dataset used to pretrain many off-the-shelf feature extractors. Under such domain mismatch, the effectiveness of ELFS (DINO) can be limited because its ImageNet-pretrained representation may not preserve the intrinsic geometry of the target dataset. As illustrated in Figure 2, semantically meaningful groups are not clearly separated, resulting in noisy pseudo-label assignments. These errors can in turn distort the training dynamics used for difficulty estimation and degrade coreset performance.

Using only a 10% labeled initial subset, our method generates target-adaptive pseudo-labels and consistently achieves the best coreset performance among all methods except the fully supervised reference. In contrast, SCORE EXTRAPOLATION yields substantially lower coreset performance despite using the same labeled subset. These results support the use of limited target-class supervision for generating pseudo-labels used in full-pool scoring.

ELFS (SELF-ENCODER) also underperforms ELFS (DINO) and sometimes random selection, indicating that self-supervised adaptation to the target data alone does not necessarily yield representations suitable for coreset selection. Other label-free methods, such as PROTOTYPICALITY and ZCORE, also depend heavily on pretrained representations and can perform below random selection when those representations are poorly aligned with the target data.

4.2.2 Image Corruption

Table 3: Coreset performance on 30% corrupted CIFAR-100 and Tiny-ImageNet. We report the average test accuracy over five runs. The full-dataset test accuracies are 73.7% and 44.3%, respectively.

Label Usage	Pruning Rate	CIFAR-100-C					Tiny-ImageNet-C				
		30%	50%	70%	80%	90%	30%	50%	70%	80%	90%
With Labels	FULLY SUPERVISED	73.6	71.1	65.2	61.1	52.6	43.7	41.3	35.5	31.4	24.2
Without Labels	RANDOM	69.8	65.4	56.7	49.7	35.8	38.9	34.4	27.1	22.6	17.0
	PROTOTYPICALITY	72.5	68.1	58.3	50.4	35.7	39.3	30.2	18.4	12.2	6.1
	ELFS (DINO)	73.1	68.7	61.1	56.6	44.9	40.9	37.4	32.5	29.4	22.7
	ELFS (SELF-ENCODER)	71.5	68.0	60.2	54.1	38.6	40.2	35.5	29.0	25.6	19.1
	ZCORE	68.4	65.7	57.2	54.5	41.3	37.8	33.6	28.4	25.1	18.7
With 10% Labels	SCORE EXTRAPOLATION	67.5	58.6	36.6	25.0	12.9	41.3	31.7	32.1	28.3	23.0
	SEMI-AUM+CUTOFF	73.1	70.0	**63.9**	**60.2**	**52.0**	40.9	37.7	33.4	**30.8**	**24.1**
	SEMI-DUAL+BETA	**73.3**	**70.4**	63.8	58.6	47.5	**42.0**	**38.5**	**33.8**	30.1	24.0

We next evaluate our framework on corrupted datasets, where visual corruptions distort the representation space and make coreset selection particularly challenging for methods that rely on fixed features. Corruption-induced artifacts can alter the geometry of examples in feature space, making it difficult for representation-based methods to preserve the underlying semantic class structure.

By learning directly from the corrupted target dataset with limited class supervision, SSL produces more accurate pseudo-labels than deep clustering (Table 1); Appendix A.1.1 further compares their

class-wise distributions. Using these pseudo-labels for scoring yields stronger coreset performance, with particularly large gains over ELFS (DINO) at high pruning ratios on CIFAR-100-C.

SCORE EXTRAPOLATION faces a different limitation in this setting. It trains a scoring model only on the initial labeled subset and propagates the resulting scores to unlabeled examples through a feature-based graph. When corruption makes neighborhood relationships in the feature space unreliable, this propagation can produce inaccurate importance estimates for the unlabeled pool.

4.2.3 Long-Tailed Distribution

Table 4: Coreset performance on CIFAR-100-LT with imbalance factor 0.1 and Caltech-101. For CIFAR-100, we use the original balanced test set. Five runs are averaged. The full-dataset test accuracies are 62.9% and 78.6%, respectively.

Label Usage	Pruning Rate	CIFAR-100-LT					Caltech-101				
		30%	50%	70%	80%	90%	30%	50%	70%	80%	90%
With Labels	FULLY SUPERVISED	58.1	54.9	43.5	35.9	27.1	79.2	72.4	59.7	54.9	42.7
Without Labels	RANDOM	32.9	18.8	8.3	5.4	3.2	75.9	65.1	53.5	50.5	40.3
	PROTOTYPICALITY	54.7	50.5	33.2	24.4	16.7	76.0	64.9	35.0	23.8	12.0
	ELFS (DINO)	49.5	49.1	43.7	34.2	25.1	75.7	68.1	53.8	47.0	40.0
	ELFS (SELF-ENCODER)	56.0	51.3	40.7	29.6	20.4	75.6	66.2	54.0	47.1	37.2
	ZCORE	55.5	51.5	38.2	32.0	19.8	71.5	65.8	56.3	51.3	38.5
With 10% Labels	SCORE EXTRAPOLATION	55.3	48.5	36.0	28.6	20.4	73.2	66.6	56.1	49.1	38.8
	SEMI-AUM+CUTOFF	50.4	46.5	**43.8**	34.4	27.0	78.4	70.6	58.5	53.1	45.7
	SEMI-DUAL+BETA	**58.3**	**53.6**	41.9	**35.1**	**27.5**	**79.0**	**70.7**	**58.8**	**53.9**	**46.8**

We next evaluate under class imbalance using CIFAR-100-LT with an imbalance factor of 0.1 and the naturally imbalanced Caltech-101 dataset. Unlike image corruption, class imbalance does not directly alter the pretrained feature extractor, but instead changes the sample distribution within the embedding space.

Under this imbalance, deep clustering may produce pseudo-labels whose class distribution deviates from the true long-tailed distribution. For example, clustering can split broad majority-class regions into multiple clusters while merging nearby minority-class regions. We further compare the class-wise distributions of generated pseudo-labels in Appendix A.1.2, which supports this distributional mismatch. In contrast, SSL-based pseudo-labeling uses a small labeled subset to anchor predictions to the target class space, allowing the resulting pseudo-labels to better reflect the underlying long-tailed distribution.

The benefit of these pseudo-labels nevertheless depends on the scoring criterion and pruning ratio. On CIFAR-100-LT, SEMI-AUM+CUTOFF trails ELFS (DINO) at 50% pruning, whereas SEMI-DUAL+BETA outperforms it at most evaluated pruning ratios. SEMI-DUAL+BETA also exceeds the fully supervised DUAL+BETA reference at 30% and 90% pruning on CIFAR-100-LT and at 90% pruning on Caltech-101. Additional CIFAR-100-LT results at an imbalance factor of 0.01 and scorer-choice discussion appear in Appendices C.1 and E.3, respectively.

4.2.4 Standard Benchmarks

Table 5: Coreset performance on standard benchmarks. The full-dataset test accuracies are 95.3%, 78.9%, and 73.6%, respectively. We report the average test accuracy over five runs for CIFAR datasets and the test accuracy from a single run for ImageNet.

Label Usage	Pruning Rate	CIFAR-10					CIFAR-100					ImageNet-1K				
		30%	50%	70%	80%	90%	30%	50%	70%	80%	90%	30%	50%	70%	80%	90%
With Labels	FULLY SUPERVISED	95.5	95.2	93.0	91.4	87.1	77.9	74.7	69.3	64.8	54.5	73.3	72.3	69.4	66.5	60.0
Without Labels	RANDOM	94.4	93.2	90.5	88.3	83.7	75.2	71.7	64.9	59.2	45.1	72.2	70.3	66.7	62.5	52.3
	PROTOTYPICALITY	95.1	93.9	87.2	79.7	59.1	75.5	69.1	53.2	39.1	19.6	70.9	60.8	54.6	41.9	30.6
	ELFS (DINO)	95.3	95.1	93.0	**91.3**	86.9	77.3	**73.8**	67.4	61.9	53.0	**73.5**	71.8	67.2	63.4	54.9
	ELFS (SELF-ENCODER)	94.8	93.5	91.2	89.5	84.5	75.5	71.7	64.8	59.5	46.9	73.2	71.4	66.8	62.7	53.4
	ZCORE	94.6	93.5	91.0	89.1	84.2	76.0	72.9	65.9	61.9	52.1	72.3	70.8	66.4	62.1	53.1
With 10% Labels	SCORE EXTRAPOLATION	95.3	94.6	91.9	89.4	81.5	75.5	70.8	61.9	53.8	34.9	68.9	65.2	59.1	55.2	53.1
	SEMI-AUM+CUTOFF	**95.4**	**95.2**	**93.1**	**91.3**	**87.2**	**77.4**	73.5	**67.7**	**63.1**	**53.9**	73.3	**72.1**	67.7	63.8	54.2
	SEMI-DUAL+BETA	**95.4**	95.0	92.7	90.9	86.2	77.1	**73.8**	67.3	62.9	53.1	72.5	71.4	**67.8**	**64.6**	**57.8**

Finally, we evaluate on standard benchmarks, where ImageNet-pretrained representations are expected to be more reliable and therefore provide a more favorable setting for ELFS (DINO). SEMIPRUNE remains competitive in these settings, with gains varying across scoring criteria and pruning ratios. For example, SEMI-DUAL+BETA exceeds ELFS (DINO) by 2.9 percentage points on ImageNet-1K at 90% pruning.

ELFS (SELF-ENCODER) still performs poorly, indicating that a self-trained encoder does not necessarily provide a reliable substitute for a strong pretrained representation. Under the same initial annotation budget, both SEMIPRUNE variants outperform SCORE EXTRAPOLATION at every evaluated pruning ratio on these standard benchmarks. These results show that SSL-based pseudo-labeling can effectively exploit limited target labels even when pretrained representations already provide a strong baseline.

4.3 Analysis and Ablations

4.3.1 Effect of the Initial Label Budget

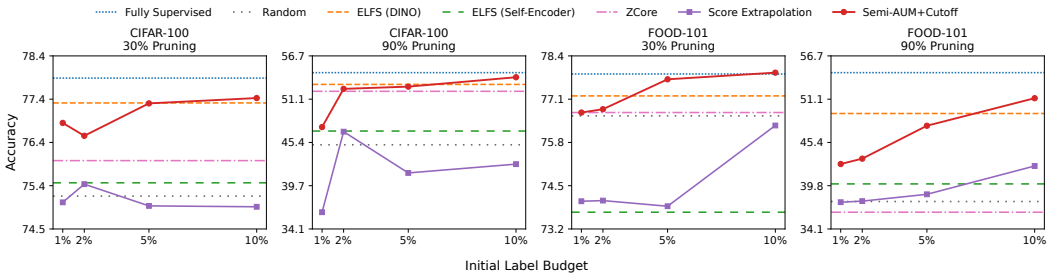


Figure 3: Effect of the initial label budget on coreset performance. We compare our method with SCORE EXTRAPOLATION under identical initial label budgets and include label-free baselines for reference.

We next examine the sensitivity of SEMIPRUNE to the amount of target-class supervision. Figure 3 compares SEMI-AUM+CUTOFF with SCORE EXTRAPOLATION under matched initial label budgets on CIFAR-100 and Food-101. SEMI-AUM+CUTOFF consistently outperforms SCORE EXTRAPOLATION across the evaluated budgets and can exceed ELFS (DINO) using only 5% labeled data. It also outperforms ELFS (SELF-ENCODER) with a 1% initial label budget. Additional pruning ratios are reported in Appendix D.1.

These results support the motivation in Section 3: limited target-class supervision helps adapt scoring when pretrained features are poorly aligned with the target data. The comparison with ELFS (SELF-ENCODER) further suggests that this supervision can be more effective than self-supervised target adaptation alone in the evaluated settings.

Additional analyses of the SSL algorithm and pseudo-label generator architecture are provided in Appendices D.2.1 and D.2.2, respectively.

4.3.2 Contribution of Training-Dynamics Scoring

To isolate the contribution of training-dynamics scoring, we compare SEMIPRUNE with an *SSL Confidence* baseline that ranks examples directly by the maximum confidence of the SSL pseudo-label generator. The baseline uses the same double-end selection and validation procedure as SEMI-AUM+CUTOFF, thereby isolating the choice of scoring signal.

Across diverse datasets and pruning ratios, at least one training-dynamics-based variant consistently outperforms direct SSL-confidence ranking.

SSL confidence reflects the pseudo-label generator's confidence in its own predictions, whereas SEMI-AUM and SEMI-DUAL capture how individual examples behave during the optimization of a separate scoring model. These results support training a separate scorer to capture signals beyond pseudo-label confidence. Appendix A.2 provides class-wise and controlled-noise analyses linking pseudo-label errors to score rankings, coreset selection, and downstream performance.

Table 6: Comparison of direct SSL-confidence ranking and training-dynamics scoring. Values are mean ± sample standard deviation.

Dataset	Method	30%	50%	70%	80%	90%
Food-101	SSL Confidence	77.2 ± 0.17	73.4 ± 0.18	66.3 ± 0.09	61.4 ± 0.20	50.0 ± 0.46
	SEMI-AUM	77.9 ± 0.09	73.9 ± 0.09	67.2 ± 0.21	63.1 ± 0.27	51.2 ± 0.36
	SEMI-DUAL	77.9 ± 0.16	73.4 ± 0.21	66.3 ± 0.14	62.2 ± 0.22	51.8 ± 0.32
CIFAR-100-C	SSL Confidence	69.5 ± 0.52	65.4 ± 0.52	56.7 ± 1.52	50.2 ± 1.05	37.0 ± 0.88
	SEMI-AUM	73.1 ± 0.19	70.0 ± 0.31	63.9 ± 0.59	60.2 ± 0.42	52.0 ± 0.31
	SEMI-DUAL	73.3 ± 0.28	70.4 ± 0.39	63.8 ± 0.76	58.6 ± 0.68	47.5 ± 0.81
CIFAR-100-LT	SSL Confidence	55.5 ± 0.86	52.0 ± 0.59	40.5 ± 1.80	34.1 ± 0.96	22.3 ± 1.38
	SEMI-AUM	50.4 ± 1.17	46.5 ± 0.58	43.8 ± 0.62	34.4 ± 0.82	27.0 ± 0.87
	SEMI-DUAL	58.3 ± 0.94	53.6 ± 0.73	41.9 ± 0.79	35.1 ± 0.91	27.5 ± 0.96
CIFAR-100	SSL Confidence	75.8 ± 0.12	72.4 ± 0.60	65.8 ± 0.44	59.7 ± 0.45	47.9 ± 0.83
	SEMI-AUM	77.4 ± 0.23	73.5 ± 0.10	67.7 ± 0.26	63.1 ± 0.21	53.9 ± 0.23
	SEMI-DUAL	77.1 ± 0.18	73.8 ± 0.27	67.3 ± 0.24	62.9 ± 0.31	53.1 ± 0.49

We additionally compare SSL-based and deep-clustering-based pseudo-labels under Forgetting, EL2N, and AUM in Appendix D.2.3.

5 Conclusion and Limitations

We propose SEMIPRUNE, an SSL-based framework that extends supervised training-dynamics-based pruning to limited-label settings. Using a small labeled subset, SEMIPRUNE generates target-adaptive pseudo-labels that enable existing pruning criteria to score the full data pool without full-pool annotation. This target-specific supervision is particularly useful when pretrained representations are poorly aligned with the target domain. Our results demonstrate improved coreset performance over label-free baselines while requiring substantially fewer annotations than fully supervised pruning. SEMIPRUNE also consistently outperforms SCORE EXTRAPOLATION under the same initial annotation budget, highlighting the effectiveness of SSL-based pseudo-labeling for leveraging limited labels in dataset pruning.

The main limitations of SEMIPRUNE are its reliance on an initial labeled subset and the additional computation required for SSL training (Appendix E.1). An insufficient label budget or poor class coverage in the labeled subset can compromise pseudo-label quality, affecting score rankings and coreset selection (Appendix A.2.2). Although our experiments demonstrate compatibility with both FixMatch and FlexMatch, broader evaluation across SSL methods and alternative proxy-label sources, including VLM-generated labels, remains future work (Appendix D.2.1).

Acknowledgments and Disclosure of Funding

This work was supported by two Institute of Information & Communications Technology Planning & Evaluation (IITP) grants (No. RS-2024-00457882 National AI Research Lab Project, 80%; No. RS-2019-II190075 Artificial Intelligence Graduate School Program (KAIST), 20%) funded by the Korean government (MSIT). We also thank Nota Inc. for supporting our participation in the conference.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Nikolas Adaloglou, Felix Michels, Hamza Kalisch, and Markus Kollmann. Exploring the limits of deep image clustering using pretrained models. *arXiv preprint arXiv:2303.17896*, 2023.

Abhinav Agarwalla. SwAV for CIFAR-10. `https://github.com/abhinavagarwalla/swav-cifar10`, 2020. GitHub repository, adapted from `https://github.com/facebookresearch/swav`.

Yuki M. Asano, Christian Rupprecht, and Andrea Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=Hyx-jyBFPr.

Jordan T Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2019.

Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 2020.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

Shaotian Cai, Liping Qiu, Xiaojun Chen, Qin Zhang, and Longteng Chen. Semantic-enhanced image clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pages 6869–6878, 2023.

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European conference on computer vision (ECCV)*, pages 132–149, 2018.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

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

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. URL https://openreview.net/forum?id=HJg2b0VYDr.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *2004 conference on computer vision and pattern recognition workshop*, pages 178–178. IEEE, 2004.

Brent A Griffin, Jacob Marks, and Jason J Corso. Zero-shot coreset selection via iterative subspace sampling. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2114–2124, 2026.

Guy Hacohen, Avihu Dekel, and Daphna Weinshall. Active learning on a budget: Opposite strategies suit high and low budgets. In *International Conference on Machine Learning*, pages 8175–8195. PMLR, 2022.

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*, pages 7713–7722, 2024.

Krishnateja Killamsetty, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021a.

Krishnateja Killamsetty, Xujiang Zhao, Feng Chen, and Rishabh Iyer. Retrieve: Coreset selection for efficient and robust semi-supervised learning. *Advances in neural information processing systems*, 34:14488–14501, 2021b.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `http://www.cs.toronto.edu/~kriz/cifar.html`.

Dong-Hyun Lee. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896. Atlanta, 2013.

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. URL `https://openreview.net/forum?id=thbtoAkCe9`.

Patrik Okanovic, Roger Waleffe, Vasilis Mageirakos, Konstantinos Nikolakakis, Amin Karbasi, Dionysios 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. URL `https://openreview.net/forum?id=JnRStoIuTe`.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL `https://openreview.net/forum?id=a68SUt6zFt`. Featured Certification.

Mansheej Paul, Surya Ganguli, and Gintare Karolina Dziugaite. Deep learning on a data diet: Finding important examples early in training. *Advances in neural information processing systems*, 34: 20596–20607, 2021.

Geoff Pleiss, Tianyi Zhang, Ethan Elenberg, and Kilian Q Weinberger. Identifying mislabeled data using the area under the margin ranking. *Advances in Neural Information Processing Systems*, 33: 17044–17056, 2020.

Ziheng Qin, Kai Wang, Zangwei Zheng, Jianyang Gu, Xiangyu Peng, Zhaopan Xu, Zhou Daquan, Lei Shang, Baigui Sun, Xuansong Xie, et al. Infobatch: Lossless training speed up by unbiased dynamic data pruning. In *International Conference on Learning Representations*, volume 2024, pages 39723–39744, 2024.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

Sebastian Schmidt, Prasanga Dhungel, Christoffer Löffler, Björn Nieth, Stephan Günnemann, and Leo Schwinn. Effective data pruning through score extrapolation, 2026. URL `https://openreview.net/forum?id=ycxzArIvgF`.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari Morcos. Beyond neural scaling laws: beating power law scaling via data pruning. *Advances in Neural Information Processing Systems*, 35:19523–19536, 2022.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J. Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=BJlxm30cKm.

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.

Wouter Van Gansbeke, Simon Vandenhende, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Scan: Learning to classify images without labels. In *European conference on computer vision*, pages 268–285. Springer, 2020.

Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. In *Eleventh International Conference on Learning Representations*. OpenReview. net, 2023.

Colin Wei, Kendrick Shen, Yining Chen, and Tengyu Ma. Theoretical analysis of self-training with deep networks on unlabeled data. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=rC8sJ4i6kaH.

Dongyue Wu, Zilin Guo, Jialong Zuo, Nong Sang, and Changxin Gao. Partial forward blocking: A novel data pruning paradigm for lossless training acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 319–328, 2025.

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. URL https://openreview.net/forum?id=7D5EECbOaf9.

Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 3485–3492. IEEE, 2010.

Suqin Yuan, Runqi Lin, Lei Feng, Bo Han, and Tongliang Liu. Instance-dependent early stopping. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=P42DbV2nuV.

Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in neural information processing systems*, 34:18408–18419, 2021.

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*, pages 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. URL https://openreview.net/forum?id=QwKvL6wC8Yi.

Haizhong Zheng, Elisa Tsai, Yifu Lu, Jiachen Sun, Brian R. Bartoldson, Bhavya Kailkhura, and Atul Prakash. ELFS: Label-free coreset selection with proxy training dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=yklJpvB7Dq.

Appendix Contents

A Additional Pseudo-Label Analysis **15**

A.1 Pseudo-Label Quality under Distribution Shift 15

A.1.1 Image Corruption 15

A.1.2 Long-Tailed Distributions 16

A.2 How Pseudo-Label Errors Affect Pruning 18

A.2.1 Class-wise Error Structure 18

A.2.2 Controlled Pseudo-Label Noise 19

B Experimental Details **19**

B.1 Baselines 19

B.2 Training Protocols 20

B.3 Selection and Search Hyperparameters 21

C Additional Evaluations and Comparisons **21**

C.1 Severe Class Imbalance 22

C.2 Comparison with Dynamic Pruning 22

C.2.1 Comparison Setting 22

C.2.2 Coreset Performance 22

D Additional Ablations **23**

D.1 Effect of the Initial Label Budget 23

D.2 Generality Across Design Choices 23

D.2.1 SSL Algorithm 23

D.2.2 Pseudo-Label Generator Architecture 24

D.2.3 Scoring Criterion 24

D.3 Full-Pipeline Seed Sensitivity 25

E Efficiency and Practical Considerations **25**

E.1 End-to-End Cost 25

E.2 Cost of Pseudo-Label Generation 26

E.3 Choosing Between SEMI-AUM+CUTOFF and SEMI-DUAL+BETA 27

A Additional Pseudo-Label Analysis

Beyond the domain-mismatch examples in the main text, we compare representations, pseudo-label quality, and assigned class distributions under image corruption and class imbalance. We then examine class-wise errors and their consequences for scoring and selection. Pseudo-label quality is measured against ground-truth labels using accuracy (ACC), adjusted random index (ARI), and normalized mutual information (NMI), reported in %.

A.1 Pseudo-Label Quality under Distribution Shift

A.1.1 Image Corruption

The embeddings of CIFAR-100-C in Figure 4 illustrate how corruption affects the representation space. Deep-clustering boundaries become less distinct than on clean data, whereas SSL better preserves separation between semantic classes.

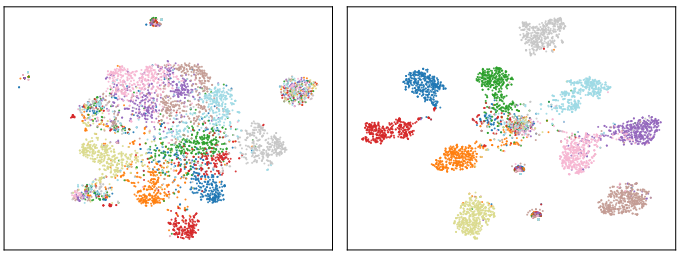


Figure 4: t-SNE visualization of corrupted CIFAR-100 embeddings, colored by ground-truth labels. **(Left)** Embeddings from ImageNet-pretrained DINO. **(Right)** Embeddings from the SSL-trained model.

Table 7 quantifies pseudo-label quality on CIFAR-100-C and Tiny-ImageNet-C. SSL outperforms deep clustering on all three reported metrics for both datasets.

Table 7: Pseudo-labeling quality on CIFAR-100-C and Tiny-ImageNet-C.

Method	CIFAR-100-C			Tiny-ImageNet-C		
	ACC (%)	ARI (%)	NMI (%)	ACC (%)	ARI (%)	NMI (%)
Deep Clustering	40.3	23.5	52.4	40.9	28.0	54.5
Semi-Supervised	65.4	36.9	61.2	58.0	35.1	56.3

Figure 5 complements these metrics with the class-wise distribution of assigned pseudo-labels, shown for both the full pool and the corrupted examples alone.

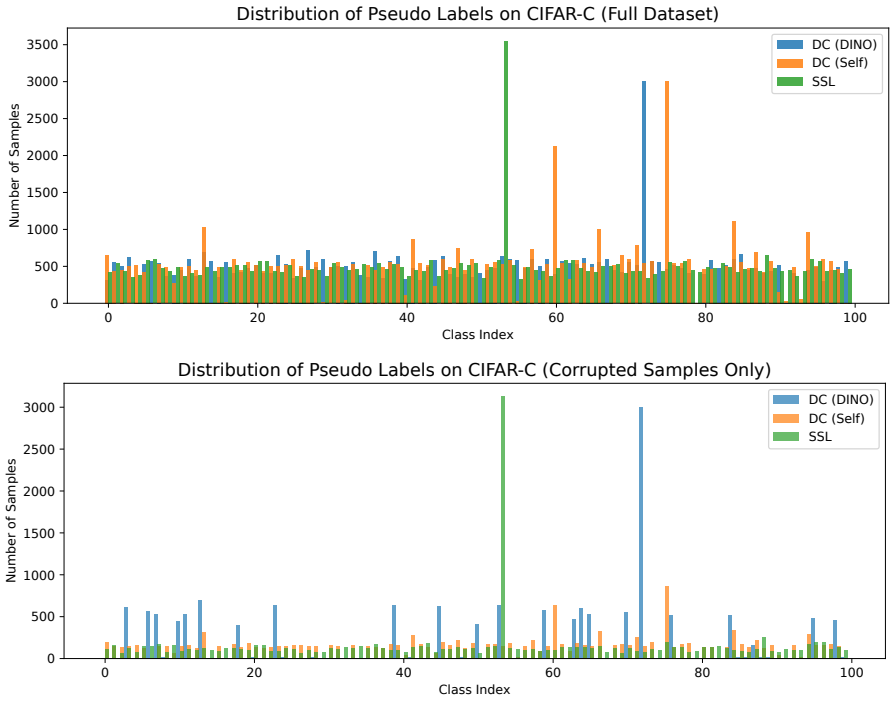


Figure 5: Class-wise distribution of assigned pseudo-labels. **(Top)** Full dataset. **(Bottom)** Corrupted samples only. Deep-Clustering does not distribute pseudo-label assignments evenly across classes, leaving some classes with no assigned samples at all, whereas Semi-Supervised Learning yields a relatively more balanced distribution. For corrupted samples, both DC and SSL assign a substantial portion of samples to particular classes. However, SSL maintains a relatively broad distribution across classes, while DC shows a stronger tendency to concentrate assignments on only a few specific classes.

A.1.2 Long-Tailed Distributions

We next examine Caltech-101 and CIFAR-100-LT. Figures 6 and 7 visualize the representations of CIFAR-100-LT with an imbalance factor of 0.1 and Caltech-101, respectively.

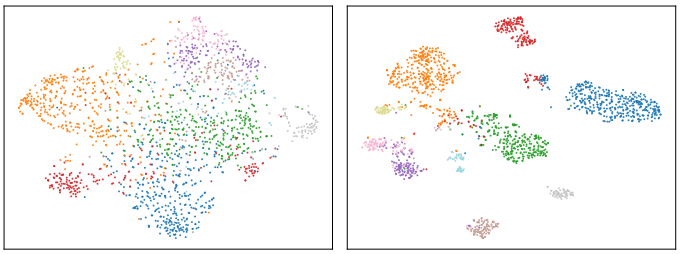


Figure 6: t-SNE visualization of long-tailed CIFAR-100 embeddings with an imbalance factor of 0.1, colored by ground-truth labels. **(Left)** Embeddings from ImageNet-pretrained DINO. **(Right)** Embeddings from the SSL-trained model.

Table 8 reports pseudo-label quality on Caltech-101 and CIFAR-100-LT under two imbalance settings.

Figures 8, 9, and 10 show the corresponding class-wise pseudo-label distributions. The coreset results for the more severe CIFAR-100-LT setting, with an imbalance factor of 0.01, are reported separately in Appendix C.1.

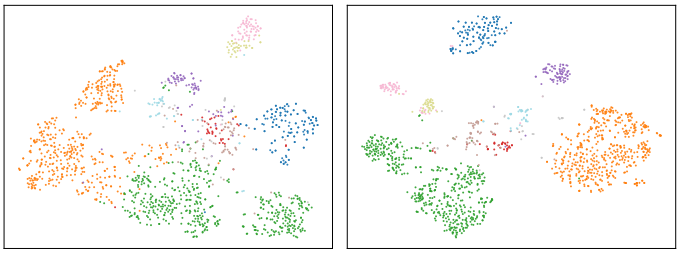


Figure 7: t-SNE visualization of Caltech-101 embeddings, colored by ground-truth labels. **(Left)** Embeddings from ImageNet-pretrained DINO. **(Right)** Embeddings from the SSL-trained model.

Table 8: Pseudo-labeling quality on Caltech-101 and CIFAR-100-LT under two imbalance settings.

Method	Caltech-101			CIFAR-100-LT, IF = 0.1			CIFAR-100-LT, IF = 0.01		
	ACC (%)	ARI (%)	NMI (%)	ACC (%)	ARI (%)	NMI (%)	ACC (%)	ARI (%)	NMI (%)
Deep Clustering	24.3 (12.7)	28.8	41.3	21.9 (18.9)	12.2	35.2	21.6 (15.3)	12.9	35.7
Semi-Supervised	52.6 (44.5)	69.6	65.9	65.4 (65.0)	48.4	67.1	61.2 (69.0)	45.3	63.3

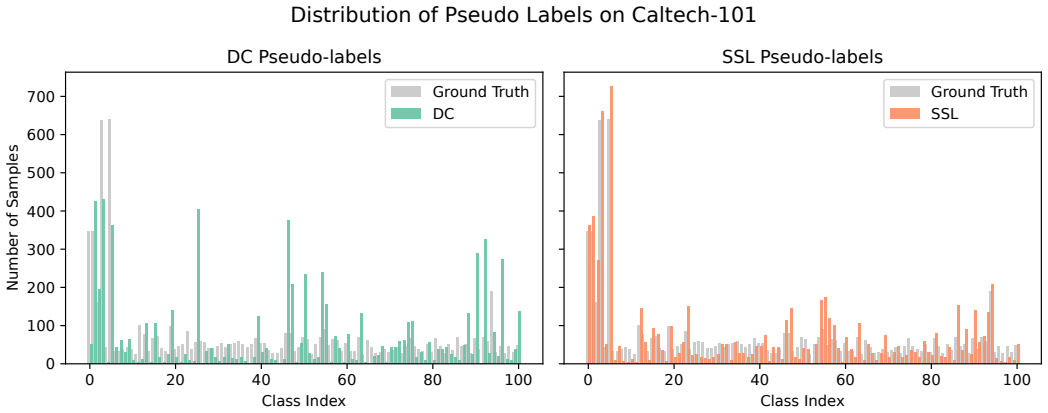


Figure 8: Class-wise distribution of assigned pseudo-labels on Caltech-101.

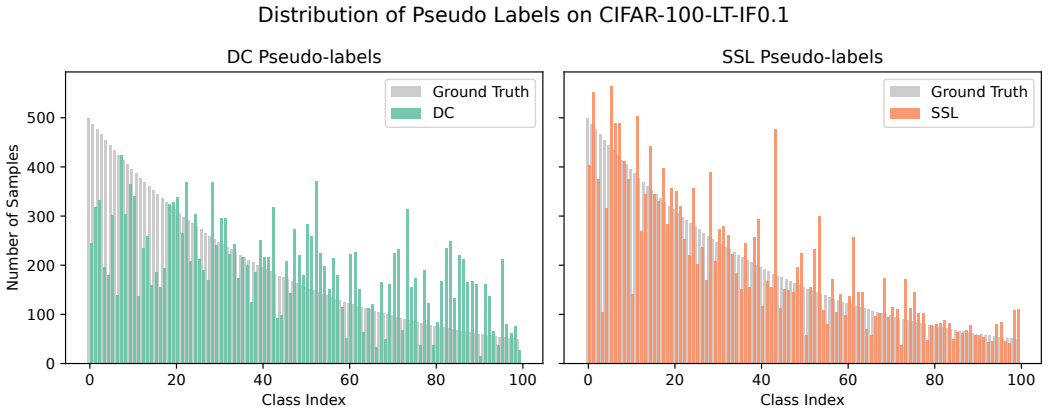


Figure 9: Class-wise distribution of assigned pseudo-labels on CIFAR-100-LT with an imbalance factor of 0.1.

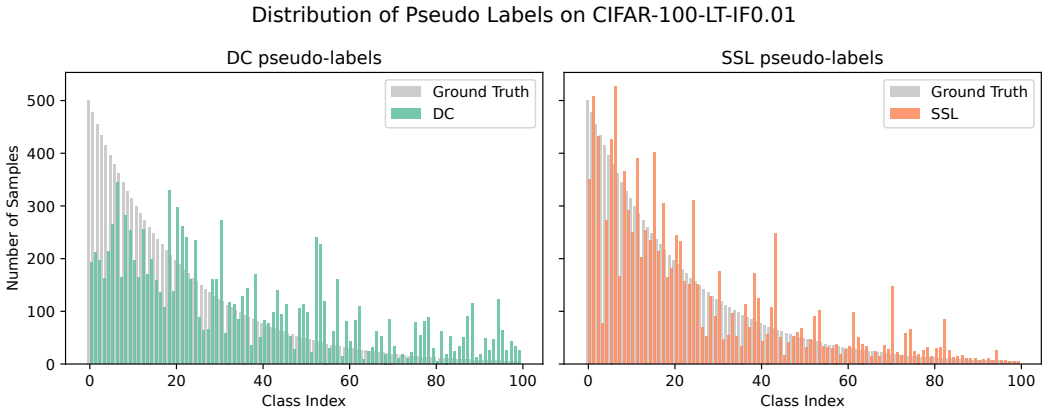


Figure 10: Class-wise distribution of assigned pseudo-labels on CIFAR-100-LT with an imbalance factor of 0.01.

A.2 How Pseudo-Label Errors Affect Pruning

A.2.1 Class-wise Error Structure

Mean pseudo-label accuracy can mask severe lower-tail errors. On SUN397, ELFS and FixMatch reach mean pseudo-label accuracies of 38.1% and 49.2%, but worst-class accuracies of 0.00% and 5.28%. We trace this structure through scoring and selection on Food-101 at a 50% pruning ratio.

Table 9: Class-wise pseudo-label quality traced through scoring and selection. Food-101, 50% pruning ratio.

Metric	ELFS-AUM	SEMI-AUM
Mean pseudo-label accuracy	42.81%	61.66%
Worst-class pseudo-label accuracy	0.40%	20.13%
Class-level Spearman ρ (PL accuracy vs. mean AUM rank)	0.893	0.968
Overlap with GT-selected subset	60.14%	68.26%
Coreset test accuracy	73.3%	73.9%

Class-wise pseudo-label accuracy is strongly associated with mean AUM rank, and correctly pseudo-labeled examples receive higher AUM scores than incorrectly labeled ones even within the same ground-truth class. SEMI-AUM also agrees more with ground-truth-based selection than ELFS-AUM: 68.26% versus 60.14% overlap, corresponding to Jaccard overlaps of 51.81% and 43.00%. The two selected subsets have similar ground-truth class-distribution total-variation distances (0.063 vs. 0.059), which suggests the difference arises from changes in score ordering and retained examples rather than from aggregate class balance.

On Food-101, these substantial differences in class-wise pseudo-label quality and selection agreement produce only a modest gap in overall accuracy (73.3% vs. 73.9%). To test whether the difference becomes more visible under class imbalance, we report lower-tail behavior on Caltech-101 at a 50% pruning ratio.

Table 10: Lower-tail pseudo-label and model accuracy under class imbalance. Caltech-101, 50% pruning ratio.

Method	Worst-class PL acc.	Coreset test acc.	Worst-class model acc.	Bottom-5 class mean acc.
ELFS-AUM	0.00%	68.1%	0.00%	1.90%
SEMI-AUM	8.00%	70.6%	2.78%	5.52%

On Caltech-101 the improvement in lower-tail pseudo-label quality is accompanied by better overall and lower-tail downstream performance, which suggests that the effect of pseudo-label-based scoring becomes more visible under class imbalance.

A.2.2 Controlled Pseudo-Label Noise

The preceding analyses describe errors produced by actual pseudo-label generators. To isolate how label errors propagate into scoring and selection, we run a controlled experiment on CIFAR-100 at a 50% pruning ratio. Starting from ground-truth labels, we randomly perturb labels to obtain target pseudo-label accuracies from 50% to 90%, and rerun the complete scoring, selection, and coreset-training pipeline at each level. Spearman correlation and subset overlap are measured against the corresponding fully supervised pruning run, which appears in the 100% column.

Table 11: Effect of controlled pseudo-label noise on score ordering, selected-subset membership, and downstream coreset accuracy. CIFAR-100, 50% pruning ratio. Columns denote the pseudo-label accuracy of the perturbed labels; the 100% column is the fully supervised reference run.

Method	Metric	50%	60%	70%	80%	90%	100% (GT)
SEMI-AUM	Spearman correlation	0.107	0.239	0.403	0.576	0.776	1.000
	Subset overlap	0.466	0.540	0.553	0.568	0.704	1.000
	Coreset accuracy	73.26	74.12	74.64	74.65	75.49	75.36
SEMI-DUAL	Spearman correlation	−0.083	−0.090	0.012	0.248	0.592	1.000
	Subset overlap	0.350	0.343	0.315	0.413	0.543	1.000
	Coreset accuracy	70.95	70.90	69.24	71.99	72.77	74.40

Despite some non-monotonicity, particularly for DUAL under severe noise, higher pseudo-label quality generally increases agreement with the ground-truth score ordering and the ground-truth selected subset, and also improves downstream coreset accuracy. Pseudo-label errors therefore do not act only through aggregate label accuracy: they reach the score ranking, the retained examples, and the final coreset performance.

Two points qualify this experiment. First, symmetric random corruption differs from the errors an SSL model actually makes: random corruption assigns incorrect labels independently of example difficulty, whereas SSL errors are structured by the model's representations and confidence. For the FixMatch pseudo-labels on CIFAR-100, the Spearman correlation between pseudo-label-based and ground-truth-based DUAL scores is 0.655 overall, and 0.816 among correctly pseudo-labeled examples. The weak correlations observed under severe random corruption therefore do not transfer directly to the practical SSL setting.

Second, SEMI-DUAL+BETA still selects a reasonable coreset even where the composite DUAL-score correlation is near zero. This is because Beta sampling does not draw sampling probabilities from the composite score ranking; it assigns them from the prediction mean. The Spearman correlation between the pseudo-label-based and ground-truth-based prediction mean remains high, at 0.850 and 0.898 for 50% and 60% label accuracy respectively. The axis that Beta sampling actually uses thus retains ground-truth-consistent structure even when the composite score does not.

B Experimental Details

We describe the compared methods, training protocols, and coreset-selection hyperparameters used in our experiments.

B.1 Baselines

RANDOM selects training examples uniformly at random. **FULLY SUPERVISED** uses ground-truth labels with DUAL+BETA, where DUAL estimates the example difficulty and uncertainty from early training stages, and Beta sampling adaptively selects a coreset according to the target pruning ratio.

PROTOTYPICALITY first clusters the data using k-means and measures each example's distance to its nearest cluster centroid; examples farther from centroids are treated as less prototypical and

are prioritized for selection. **ZCORE** is another label-free baseline that leverages CLIP embeddings and iteratively selects a subset by balancing representativeness and redundancy. **ELFS (DINO)** is a label-free selection baseline that applies the ELFS pipeline using DINO features. **ELFS (SELF-ENCODER)** is a variant of ELFS that pretrains an encoder directly on the target dataset from scratch before performing deep clustering; this setting tests whether learning target-specific representations can improve label-free selection in challenging scenarios. We follow the original ELFS and train the encoder with SwAV (Caron et al., 2020). For CIFAR datasets, we use a CIFAR-adapted SwAV implementation as a reference (Agarwalla, 2020). For the remaining datasets, we use the ImageNet training hyperparameters reported in the original SwAV paper.

SCORE EXTRAPOLATION assumes access to a small labeled subset, trains a supervised scorer on this subset, and extrapolates the resulting scores to the remaining unlabeled examples; we use the same initial label budget as our method for a fair comparison. We report two variants of our method. **SEMI-AUM+CUTOFF** uses SSL-generated pseudo-labels and applies AUM-based double-end pruning with a cutoff strategy, following the selection principle of ELFS. **SEMI-DUAL+BETA** uses the same DUAL+BETA selection rule as FULLY SUPERVISED.

B.2 Training Protocols

We use a unified training protocol across datasets, while adjusting the model architecture and batch size according to dataset scale. We generate pseudo-labels using FixMatch, implemented with the publicly available USB codebase[4]. The corresponding training hyperparameters are provided in Table 12.

For coreset selection, we first train a model on the full pseudo-labeled dataset, and then retrain the model from scratch on the selected coreset using the same base hyperparameters. For all datasets except ImageNet-1K, we halve the batch size at 80% pruning and halve it again at 90% pruning. We build our implementation on top of the publicly available DUAL and ELFS codebases[5]. Unless otherwise specified, we follow their original training, scoring, and coreset selection procedures, and extend them to support pseudo-labeled datasets and our experimental settings. The detailed training hyperparameters are summarized in Table 13. For the main multi-run results, pseudo-labels are fixed while scoring and downstream training seeds vary. Additional full-pipeline seed sensitivity results on CIFAR-100 are reported in Appendix D.3.

Table 12: Training hyperparameters for pseudo-label generation. The same settings for CIFAR-10 and CIFAR-100 are also applied to their corrupted and long-tailed variants, i.e., CIFAR-100-C, CIFAR-100-LT.

Dataset	Architecture	Epochs	Batch Size	U Ratio
CIFAR-10	WRN-28-2	100	64	7
CIFAR-100	WRN-28-8	100	64	7
Caltech-101	ResNet-50	100	64	1
Food-101	ResNet-50	50	64	1
SUN397	ResNet-50	50	64	1
Tiny-ImageNet	ResNet-50	100	128	1
ImageNet-1K	ResNet-50	100	128	1

Table 13: Training hyperparameters used for coreset selection. The same settings for CIFAR-10 and CIFAR-100 are also applied to their corrupted and long-tailed variants, i.e., CIFAR-100-C, CIFAR-100-LT. We use CIFAR-style ResNet for CIFAR-10 and CIFAR-100, and use ImageNet-style ResNet for the others.

Dataset	Architecture	Epochs	Batch Size	LR	Optimizer	Weight Decay
CIFAR-10 / CIFAR-100	ResNet18	200	128	0.1	SGD, momentum 0.9	5×10^{-4}
Food101 / SUN397 / Caltech101	ResNet18	200	128	0.1	SGD, momentum 0.9	1×10^{-4}
ImageNet-1K / Tiny-ImageNet	ResNet34	90	256	0.1	SGD, momentum 0.9	1×10^{-4}

[4] `https://github.com/microsoft/Semi-supervised-learning.git`

[5] `https://github.com/behaapyy/dual-pruning`; `https://github.com/eltsai/elfs.git`

B.3 Selection and Search Hyperparameters

Following Zheng et al. (2025), we use a 90/10 train-validation split of the pseudo-labeled dataset for hyperparameter search, evaluating validation accuracy against pseudo-labels. This split is used only for hyperparameter search; the scorer is trained on the full pseudo-labeled pool. Once the hyperparameters are fixed, we select the final coreset from the entire training pool, including the 10% validation subset. For SEMI-AUM+CUTOFF, we perform a grid search over the cutoff from 0 to r, where r is the pruning ratio, with a step size of 0.1. For SEMI-DUAL+BETA, we first fix T based on the best performance at a 30% pruning ratio, and then tune c_D with the selected T. For SEMI-DUAL+BETA, we additionally tune the uncertainty exponent γ, which reduces sensitivity to uncertainty estimates under noisy pseudo-labels. For CIFAR-10 and CIFAR-100, we search $\gamma \in \{0.5, 0.8, 1.0, 1.2\}$ and select $\gamma = 1.2$ based on validation performance; for ImageNet-1K, we evaluate $\gamma \in \{0.5, 1.0\}$ and select $\gamma = 0.5$.

Table 14: Cutoff ratios used for SEMI-AUM+CUTOFF. For each dataset and pruning ratio, we report the cutoff ratio used to remove low-AUM examples before selecting the final coreset.

Category	Dataset	30%	50%	70%	80%	90%
Domain-specific	Food-101	0.1	0.3	0.5	0.6	0.8
	SUN397	0.1	0.1	0.1	0.2	0.2
Image-corrupted	CIFAR-100	0.1	0.2	0.4	0.5	0.7
	Tiny-ImageNet	0.2	0.4	0.6	0.7	0.8
Long-tailed	CIFAR-100 (IF=0.1)	0.1	0.4	0.5	0.6	0.8
	Caltech-101	0.0	0.1	0.4	0.5	0.6
Standard	CIFAR-10	0.0	0.0	0.1	0.2	0.4
	CIFAR-100	0.1	0.2	0.4	0.6	0.7
	ImageNet-1K	0.1	0.3	0.5	0.5	0.7

Table 15: Hyperparameters used for SEMI-DUAL+BETA. We report (T, c_D), where T is the score-computation epoch and c_D is the Beta-sampling parameter reflecting dataset difficulty.

Domain-specific		Image-corrupted		Long-tailed		Standard		
Food	SUN	C-100	Tiny-IN	C-100	Caltech	C-10	C-100	IN-1K
$(30, 5)$	$(30, 5)$	$(70, 4)$	$(50, 6)$	$(30, 6)$	$(30, 5)$	$(30, 5.5)$	$(30, 4)$	$(60, 11)$

Sensitivity to the uncertainty exponent. Table 16 reports the effect of the selected uncertainty exponent on CIFAR-10 and CIFAR-100. The selected value changes average accuracy only modestly relative to $\gamma = 1$, with the largest differences occurring at 90% pruning.

Table 16: Effect of the uncertainty exponent γ on coreset accuracy. Parentheses give the change relative to $\gamma = 1$.

Dataset	γ	30%	50%	70%	80%	90%
CIFAR-10	1.0	95.4	95.1	92.6	90.8	85.8
	1.2	95.4 (+0.0)	95.0 (−0.1)	92.7 (+0.1)	90.9 (+0.1)	86.2 (+0.4)
CIFAR-100	1.0	77.2	73.8	67.2	62.7	52.3
	1.2	77.1 (−0.1)	73.8 (+0.0)	67.3 (+0.1)	62.9 (+0.2)	53.1 (+0.8)

C Additional Evaluations and Comparisons

We provide complementary evaluations under more severe class imbalance and against dynamic pruning methods.

C.1 Severe Class Imbalance

We extend the long-tailed evaluation to CIFAR-100-LT with an imbalance factor of 0.01. The balanced test set is unchanged, and the full training set achieves 41.7% test accuracy. Pseudo-label quality and class distributions for this setting are analyzed in Appendix A.1.2.

Table 17: Coreset performance on long-tailed CIFAR-100 with imbalance factor 0.01. We use the original CIFAR-100 test set, which is balanced. Results are averaged over five runs. The test accuracy for the full train dataset is 41.7%.

Label Usage	Method	30%	50%	70%	80%	90%
With Labels	FULLY SUPERVISED	37.0	32.3	21.9	19.4	15.4
Without Labels	RANDOM	22.9	11.5	5.9	4.0	2.5
	PROTOTYPICALITY	34.9	29.2	19.8	14.7	11.0
	ELFS (DINO)	35.0	**31.8**	24.0	19.9	12.4
	ELFS (SELF-ENCODER)	34.8	29.3	22.6	16.8	12.4
	ZCORE	32.5	28.8	20.0	16.1	12.6
With 10% Labels	SCORE EXTRAPOLATION	34.7	28.4	20.0	16.1	12.1
	SEMI-AUM+CUTOFF	33.8	30.6	**24.7**	**20.1**	**14.5**
	SEMI-DUAL+BETA	**38.0**	31.0	22.4	18.7	14.4

Table 17 shows that the relative performance of the two scoring criteria varies with the pruning ratio. SEMI-DUAL+BETA achieves the highest accuracy among the label-free and label-efficient methods at 30% pruning, whereas SEMI-AUM+CUTOFF performs best at 70–90% pruning. ELFS (DINO) remains strongest at 50% pruning. These results further illustrate that the preferred scoring criterion depends on the data regime and pruning ratio.

C.2 Comparison with Dynamic Pruning

C.2.1 Comparison Setting

Dynamic and online pruning methods such as InfoBatch (Qin et al., 2024) and Partial Forward Blocking (PFB) (Wu et al., 2025) also reduce the effective amount of data used for training, but they operate under a different data-access and supervision setting from ours. They prune online while retaining access to the full pool throughout training, so an example discarded at one stage can be reintroduced later. In this comparison, the dynamic baselines train on pseudo-labels from the initial pseudo-label pool. SEMIPRUNE uses this pool for scoring and selection, then trains on ground-truth labels for the selected coreset without revisiting the discarded data.

We match the number of optimizer updates and the number of retained examples per update. SEMIPRUNE trains its retained subset for 200 epochs. PFB enlarges its candidate batch so that, after online pruning, its selected batch size and its number of updates per epoch match the static setting. All methods consequently process the same total number of retained training examples. The downstream supervision and data-access settings differ, however: the dynamic baselines use pseudo-labels and retain full-pool access, whereas SEMIPRUNE uses ground-truth labels and a fixed coreset. The comparison therefore reflects the combined effects of selection, supervision, and data access.

C.2.2 Coreset Performance

Table 18 compares static and dynamic pruning under the matched update budget described above.

The results show a trade-off between the static and dynamic regimes rather than a uniform ordering. At moderate pruning ratios (30–50%) the retained subset is still large enough to preserve example coverage, and SEMIPRUNE benefits from acquiring ground-truth labels for the retained examples, outperforming both dynamic methods by a wide margin on CIFAR-100-C. Under aggressive pruning the picture reverses: repeatedly training on a single fixed subset limits example diversity, whereas PFB can distribute the same update budget over different examples drawn from the full pool. PFB is accordingly stronger at 90% pruning on both datasets. These results illustrate the trade-off between ground-truth supervision on a fixed coreset and pseudo-label supervision with continued access to the full pool.

Table 18: Static versus dynamic pruning under matched optimizer updates and matched retained examples per update. Dynamic baselines train on pseudo-labels; SEMIPRUNE trains on ground-truth labels for the selected coreset. The first row gives the PFB candidate/selected batch sizes used at each pruning ratio. Best result per column in bold.

Dataset	Method	30%	50%	70%	90%
PFB config	Candidate / selected batch	183 / 128	256 / 128	427 / 128	320 / 32
CIFAR-100	InfoBatch	72.78 ± 0.17	72.09 ± 0.46	70.70 ± 0.28	52.18 ± 2.35
	PFB	72.62 ± 0.20	72.45 ± 0.25	**71.43 ± 0.07**	**66.87 ± 0.83**
	SEMI-AUM	**77.40 ± 0.23**	73.50 ± 0.10	67.70 ± 0.26	53.90 ± 0.23
	SEMI-DUAL	77.10 ± 0.18	**73.80 ± 0.27**	67.30 ± 0.24	53.10 ± 0.49
CIFAR-100-C	InfoBatch	58.29 ± 0.80	56.66 ± 0.23	53.80 ± 0.83	31.56 ± 2.84
	PFB	58.23 ± 0.22	57.57 ± 0.14	55.78 ± 0.33	**55.30 ± 1.30**
	SEMI-AUM	73.10 ± 0.19	70.00 ± 0.31	**63.90 ± 0.59**	52.00 ± 0.31
	SEMI-DUAL	**73.30 ± 0.28**	**70.40 ± 0.39**	63.80 ± 0.76	47.50 ± 0.81

D Additional Ablations

We extend the initial-label-budget analysis to additional pruning ratios and examine the effects of the SSL algorithm, pseudo-label generator architecture, and scoring criterion.

D.1 Effect of the Initial Label Budget

We extend the initial-label-budget analysis in the main text to all evaluated pruning ratios on CIFAR-100 and Food-101 (Figure 11). Our SEMI-AUM consistently performs better than SCORE EXTRAPOLATION under the same initial annotation budget. With 5% annotated samples, SEMI-AUM can outperform the strongest label-free baseline, ELFS.

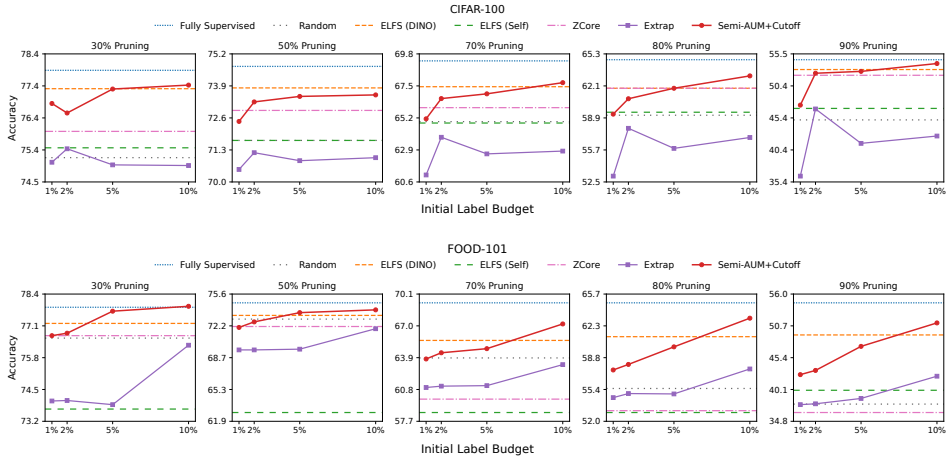


Figure 11: Effect of the initial label budget on coreset performance on CIFAR-100 (top) and Food-101 (bottom).

D.2 Generality Across Design Choices

D.2.1 SSL Algorithm

To test whether the results depend on the specific SSL algorithm, we replace FixMatch with FlexMatch on CIFAR-100 while holding the scoring and downstream protocols fixed.

FlexMatch raises pseudo-label accuracy from 74.1% to 77.8%. With FlexMatch, SEMI-AUM is comparable to or better than its FixMatch counterpart and continues to match or outperform ELFS at

Table 19: Effect of the SSL method used for pseudo-label generation. CIFAR-100, with scoring and downstream protocols held fixed.

Score	SSL method	PL acc.	30%	50%	70%	80%	90%
SEMI-AUM	FixMatch	74.1	77.4 ± 0.23	73.5 ± 0.10	67.7 ± 0.26	63.1 ± 0.21	53.9 ± 0.23
SEMI-AUM	FlexMatch	77.8	77.3 ± 0.23	74.0 ± 0.11	68.1 ± 0.24	63.2 ± 0.37	54.2 ± 0.69
SEMI-DUAL	FixMatch	74.1	77.1 ± 0.18	73.8 ± 0.27	67.3 ± 0.24	62.9 ± 0.31	53.1 ± 0.49
SEMI-DUAL	FlexMatch	77.8	77.2 ± 0.35	73.8 ± 0.30	67.3 ± 0.29	62.8 ± 0.37	52.7 ± 0.42

every pruning ratio. SEMI-DUAL stays comparable to both its FixMatch counterpart and ELFS, but does not improve consistently. Higher aggregate pseudo-label accuracy therefore does not translate into uniform gains for every scoring criterion, while the framework's effectiveness is not specific to FixMatch.

D.2.2 Pseudo-Label Generator Architecture

The pseudo-label generator and the scoring model play different roles. The SSL model is used only to produce proxy labels; its features, confidence values, and training dynamics are never consumed by SEMI-AUM or SEMI-DUAL. Following common SSL benchmark practice we use WRN-28-8 for CIFAR-100 and ResNet-50 for the larger datasets, rather than matching the generator to the downstream scoring model, which is trained from scratch on the proxy-labeled pool. Architectural matching is therefore not a requirement of the framework. We nevertheless evaluate a matched setting by replacing the WRN-28-8 FixMatch model with ResNet-18, the architecture also used for scoring and final subset training, holding the rest of the protocol fixed.

Table 20: Matching the pseudo-label generator architecture to the scoring and subset-training model. CIFAR-100.

PL generator	Method	PL acc.	30%	50%	70%	80%	90%
WRN-28-8	SEMI-AUM	74.1	77.4 ± 0.23	73.5 ± 0.10	67.7 ± 0.26	63.1 ± 0.21	53.9 ± 0.23
ResNet-18	SEMI-AUM	68.4	77.4 ± 0.05	73.4 ± 0.64	66.9 ± 0.78	61.0 ± 0.50	52.7 ± 2.25
WRN-28-8	SEMI-DUAL	74.1	77.1 ± 0.18	73.8 ± 0.27	67.3 ± 0.24	62.9 ± 0.31	53.1 ± 0.49
ResNet-18	SEMI-DUAL	68.4	77.3 ± 0.29	73.6 ± 0.42	66.2 ± 1.03	61.4 ± 0.82	51.5 ± 0.84

Matching does not give a consistent improvement here. The ResNet-18 generator lowers pseudo-label accuracy from 74.1% to 68.4%; coreset performance stays comparable at 30–50% pruning but declines more noticeably at higher ratios. This does not establish that architecture matching is generally irrelevant, but it shows no systematic benefit when matching reduces proxy-label quality.

D.2.3 Scoring Criterion

We finally examine whether the benefit of SSL-based pseudo-labeling extends beyond the scoring criteria used in our main variants. Specifically, we compare SSL-based and deep-clustering-based pseudo-labels under Forgetting, EL2N, and AUM.

Table 21: Pseudo-labeling effectiveness across scoring metrics. SSL-based pseudo-labels yield consistent gains on Food-101 and remain competitive on CIFAR-100.

Score	Pseudo-labels	Food-101					CIFAR-100				
		30%	50%	70%	80%	90%	30%	50%	70%	80%	90%
Forgetting	Deep Clust.	77.4	73.0	64.8	60.7	48.7	77.1	**74.2**	**67.5**	62.0	51.8
	Semi-sup.	**77.7**	**73.4**	**66.4**	**62.0**	**52.5**	**77.3**	73.6	66.9	**62.6**	**53.3**
EL2N	Deep Clust.	77.2	72.9	65.2	59.8	48.6	76.7	72.9	65.9	61.0	51.1
	Semi-sup.	**77.5**	**73.0**	**65.5**	**60.4**	**50.9**	**77.5**	**73.2**	**66.8**	**61.4**	**51.7**
AUM	Deep Clust.	77.2	73.3	65.6	61.1	49.2	77.3	**73.8**	67.4	61.9	53.1
	Semi-sup.	**77.9**	**73.9**	**67.2**	**63.1**	**51.2**	**77.4**	73.5	**67.7**	**63.1**	**53.9**

As shown in Table 21, SSL-based pseudo-labeling improves coreset accuracy over deep clustering across all evaluated scoring metrics and pruning ratios on Food-101. On CIFAR-100, the two approaches perform more comparably, with SSL-based pseudo-labeling achieving similar or better accuracy in most settings. The larger gains on Food-101 are consistent with the greater value of target-class supervision when pretrained representations are less aligned with the target data. These results show that the benefit of SSL-based pseudo-labeling generalizes across different training-dynamics-based scoring criteria.

D.3 Full-Pipeline Seed Sensitivity

On CIFAR-100 with a 10% initial label budget, we additionally repeat the full pipeline with five independent seeds. Each run independently resamples the initial labeled subset and repeats FixMatch training, scoring, coreset selection, and downstream training. Table 22 reports the resulting coreset test accuracy.

Table 22: Full-pipeline seed sensitivity on CIFAR-100. Values are coreset test accuracy (%), reported as mean $\pm$ sample standard deviation over five independent full-pipeline runs. Columns indicate pruning ratios.

Method	30%	50%	70%	80%	90%
SEMI-AUM	77.4 ± 0.15	73.6 ± 0.42	67.4 ± 0.54	62.3 ± 0.65	53.2 ± 0.50
SEMI-DUAL	77.2 ± 0.28	73.7 ± 0.30	67.0 ± 0.26	62.4 ± 0.56	52.9 ± 0.22

Across these five runs, pseudo-label accuracy is $74.04 \pm 0.13\%$, and downstream accuracy standard deviations are at most 0.65 percentage points. This additional analysis is limited to CIFAR-100; the main results use fixed pseudo-labels and vary scoring and downstream training seeds.

E Efficiency and Practical Considerations

We report end-to-end computational costs, examine the pseudo-label generation stage across datasets, and discuss the practical trade-offs between the two scoring variants.

E.1 End-to-End Cost

We first report the complete point-of-use cost, including the SSL stage required by SEMIPRUNE. The comparison covers proxy-label acquisition, dynamics collection, selection search, and final subset training. A downloaded DINO encoder has zero encoder-training cost at the point of use, so we separate ELFS (DINO) from ELFS (SELF-ENCODER), which trains SwAV on the target data. All timings were measured on NVIDIA RTX A6000 GPUs and are reported as elapsed wall-clock time. For Food-101, FixMatch proxy acquisition used four GPUs. The totals sum the elapsed times of the pipeline stages.

The score/search stage uses AUM with double-end pruning for ELFS and SEMI-AUM, and DUAL with Beta sampling for SEMI-DUAL. Dynamics collection trains a separate scoring model on the proxy-labeled pool; all methods use the same downstream architecture and the same final subset-training schedule. Compared with ELFS (DINO), SEMI-AUM and SEMI-DUAL require a 10% initial label budget and respectively add 12h 24m and 1h 48m of point-of-use time. We therefore do not claim universal compute savings.

Food-101 is a setting where the SSL stage is itself relatively expensive. For completeness we also report the end-to-end comparison on clean CIFAR-100, where the target distribution is well matched to the pretraining distribution.

The additional target-specific SSL cost is most justified when pretrained representations are poorly aligned with the target distribution. On a clean benchmark such as CIFAR-100 the accuracy gain relative to the additional cost is limited, which is consistent with the positioning of SEMIPRUNE as an annotation–accuracy–compute trade-off rather than a compute reduction.

Table 23: Stage-by-stage point-of-use cost on Food-101 at a 50% pruning ratio. † The downloaded DINO encoder's pretraining cost is excluded. ‡ Search over six cutoff ratios, $\{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$, including candidate-subset retraining. § Search over four c_D candidates, $\{3, 4, 5, 6\}$, including candidate-subset retraining.

Method	Initial labels	Proxy acquisition	Dynamics collection	Score/ search	Subset training	Total	Acc.
ELFS (DINO)	0%	1h 36m†	4h 29m (200 ep)	19h 36m‡	3h 16m	**28h 57m**	73.3%
ELFS (SELF-ENCODER)	0%	13h 17m	4h 29m (200 ep)	19h 36m‡	3h 16m	**40h 38m**	62.8%
ZCORE	0%	∼4m	–	∼1m	3h 16m	**3h 21m**	72.1%
SCORE EXTRAPOLATION	10%	–	∼55m (200 ep)	∼15m	3h 16m	**4h 26m**	71.8%
SEMI-AUM	10%	13h 42m	4h 47m (200 ep)	19h 36m‡	3h 16m	**41h 21m**	73.9%
SEMI-DUAL	10%	13h 42m	∼43m (30 ep)	13h 04m§	3h 16m	**30h 45m**	73.4%

Table 24: Stage-by-stage point-of-use cost on clean CIFAR-100 at a 50% pruning ratio. Footnote markers follow Table 23, except that § here denotes a search over four $c_D \in \{3, 4, 5, 6\}$ followed by $\gamma \in \{0.5, 0.8, 1.0, 1.2\}$, including candidate-subset retraining.

Method	Initial labels	Proxy acquisition	Dynamics collection	Score/ search	Subset training	Total	Acc.
ELFS	0%	30m†	45m (200 ep)	∼2h 15m‡	∼20m	∼**3h 50m**	73.8%
SEMI-AUM	10%	4h	45m (200 ep)	∼2h 15m‡	∼20m	∼**7h 20m**	73.5%
SEMI-DUAL	10%	4h	8m (30 ep)	∼3h 10m§	∼20m	∼**7h 38m**	73.8%

E.2 Cost of Pseudo-Label Generation

To examine the proxy-acquisition stage in more detail, we compare methods that require training on the target dataset for pseudo-label generation. All experiments were conducted on NVIDIA RTX A6000 GPUs. ELFS also considers a self-trained SwAV encoder, which requires target-dataset self-supervised training before clustering. In contrast, our method trains an SSL model directly on the target dataset using a small labeled subset and the remaining unlabeled examples.

Table 25: Computation time for target-dataset pseudo-label generation. We compare target-dataset self-supervised encoder training used by SwAV with SSL-based pseudo-labeling methods. We report representative GPU-heavy benchmarks and omit smaller auxiliary settings from this comparison.

Method	Dataset	Backbone	Epochs	GPUs	Wall-clock Time	GPU-hours
SwAV	CIFAR-100	ResNet-50	500	1	11.8 h	11.80
SwAV	Food-101	ResNet-50	200	4	12.06 h	48.24
SwAV	SUN397	ResNet-50	200	8	9.57 h	76.56
SwAV	Tiny-ImageNet	ResNet-50	200	4	30.40 h	121.59
SwAV	ImageNet-1K	ResNet-50	800	64	39.84 h	2549.76
FixMatch	CIFAR-100	WRN-28-8	100	1	4.00 h	4.00
FixMatch	Food-101	ResNet-50	50	4	13.65 h	54.60
FixMatch	SUN397	ResNet-50	50	4	13.82 h	55.28
FixMatch	Tiny-ImageNet	ResNet-50	100	4	7.3 h	29.28
FixMatch	ImageNet-1K	ResNet-50	100	8	32.88 h	263.04

E.3 Choosing Between SEMI-AUM+CUTOFF and SEMI-DUAL+BETA

SEMI-AUM+CUTOFF and SEMI-DUAL+BETA are reported as separate instantiations of the framework, not as a best-of-two result selected after observing test performance. We do not claim a universally validated *a priori* rule for choosing between them, but the observed pattern admits a consistent reading based on what a low score represents in each setting.

Under image corruption, severely degraded examples tend to keep low or negative margins over many epochs, so a margin-based criterion with a cutoff is well suited to identifying and removing persistently unreliable examples. In long-tailed data a low margin may instead belong to a valid tail-class example that is learned late because it is underrepresented, and a global AUM cutoff can then discard rare but informative examples. SEMI-DUAL+BETA captures variation in the target probability over training while emphasizing low-confidence examples, and Beta sampling supplements those examples with moderate-difficulty ones. This may help retain informative tail examples, but the relative advantage depends on the imbalance level and pruning ratio. For example, on CIFAR-100-LT with an imbalance factor of 0.01 (Table 17), SEMI-DUAL+BETA performs best among the label-free and label-efficient methods at 30% pruning, whereas SEMI-AUM+CUTOFF performs best at 70–90%. Thus, the reported long-tail results support a conditional empirical trend rather than a general preference for SEMI-DUAL+BETA across long-tailed datasets.

Deployment cost provides a second practical consideration. SEMI-DUAL+BETA requires less computation for dynamics collection in the reported comparisons, which can be useful when scoring is repeated. The end-to-end benefit also depends on pseudo-label generation and hyperparameter search: Appendix E.1 shows lower total cost for SEMI-DUAL+BETA than for SEMI-AUM+CUTOFF on Food-101, but not on CIFAR-100. SEMI-AUM+CUTOFF remains a reasonable choice when substantial image corruption is expected and additional computation is available.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: Our claims made in the abstract and introduction match experimental results in the later sections.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5 discusses the initial-label requirement, class-coverage sensitivity, additional SSL computation, and limits of generalization across pseudo-label generators; see also Appendices A.2.2, D.2.1, and E.1.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

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

Answer: [N/A]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The information needed to reproduce the experimental results is disclosed in Appendix B.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

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

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

Answer: [Yes]

Justification: All experiments use publicly available datasets, models, and codebases. The setup directly follows these public implementations. Appendix B specifies the choice of model, dataset, optimizers, and hyperparameters, which are sufficient to reproduce the experimental results. Our implementation and usage instructions are available at `https://github.com/cyseul/SemiPrune`; the experiments use the cited public datasets.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [Yes]

Justification: The experimental details are specified in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The main benchmark tables report mean downstream supervised training performance over five random seeds, using the same seeds across methods. Selected additional experiments report sample standard deviations, but the main benchmark tables omit error bars and we do not quantify variability across repeated SSL pseudo-label generation runs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide information on the computer resources in Appendix E.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The authors of this paper have reviewed the NeurIPS Code of Ethics. This paper conforms with the NeurIPS Code of Ethics in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

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

    Answer: [N/A]

    Justification: This work is purely experimental. The main focus is on the suggestion of new methodology without a direct relation to downstream applications.

    Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: This paper does not involve releasing any new data or models.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We use publicly available datasets, optimizers, and pretrained models. All of them are properly cited in the paper.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

    Answer: [No]

    Justification: We release research code with environment setup and usage instructions at `https://github.com/cyseul/SemiPrune`. A dedicated asset card and complete licensing documentation are not provided.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLM is used only for editing the paper.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.