Rethinking Dataset Distillation: Hard Truths About Soft Labels

Supplementary Material

1. Details on Methods and their loss objectives

1.1. Large-scale methods

Early dataset distillation methods [1, 26, 27, 29, 31] relied on a bi-level optimization framework, where the distilled set is optimized in an outer loop while a model is trained on this distilled set in an inner loop. This process is computationally expensive, as it requires repeated model training for every update of the distilled dataset, making it particularly challenging to scale to larger settings. SRe2L [28] addressed this limitation by decoupling this optimization process of the distilled set from the model training. This approach eliminates the need for an inner loop during training, as the dataset is now synthesized by matching certain statistical properties of a pre-trained teacher model. This decoupling enables it to scale to larger datasets like ImageNet-1K [6] and beyond. Many subsequent works, like EDC [22], DWA [7], G-VBSM [21], etc. have adopted a similar approach by building on top of this framework. We briefly describe such techniques below for ease of reference and clarity:

SRe2L [28] performs distillation by optimizing two loss objectives: (1) the standard Cross-entropy loss, and (2) another loss which matches the BatchNorm-mean and BatchNorm-variance of the teacher model which is pre-trained on the original dataset. Following is the formulation for the latter part:

$$\begin{aligned}\mathcal{L}_{\text{BN-SRe2L}}(\tilde{x}) = &\underbrace{\sum_{l} \left\|\mu_l(\tilde{x}) - \mathbf{BN}_l^{\text{RM}}\right\|_2}_{\text{BN-Mean loss}} \\ &+ \underbrace{\sum_{l} \left\|\sigma_l^2(\tilde{x}) - \mathbf{BN}_l^{\text{RV}}\right\|_2}_{\text{BN-Var loss}}\end{aligned} \tag{1}$$

Here, l is the index of the Batch Normalization (BN) layer, $\mu_l(\tilde{x})$ and $\sigma_l^2(\tilde{x})$ represent the mean and variance of the model trained on distilled set, respectively. $\mathbf{BN}_l^{\text{RM}}$ and $\mathbf{BN}_l^{\text{RV}}$ denote the running mean and running variance statistics of the pre-trained model at the l-th layer.

DWA [7] (Directed Weight Adjustment) aims to address the diversity limitations of SRe2L by introducing *weight perturbations* in the model during matching along with a separate hyperparameter $\boldsymbol{\lambda_{var}}$ for the variance matching loss term:

$$\begin{aligned}\mathcal{L}_{\text{BN-DWA}}(\tilde{x}) = &\text{ BN-Mean loss}(\tilde{x}) \\ &+ \boldsymbol{\lambda_{var}} \cdot \text{BN-Var loss}(\tilde{x})\end{aligned} \tag{2}$$

RDED [24] creates a synthetic dataset by extracting and concatenating the most "impactful" patches (where the impact is measured using a scoring mechanism using the output of a pretrained teacher model) from all the images of a particular class. Therefore, even though RDED is considered a "synthesis" method by the research community, but in its core essence, it's a technique which groups together informative patches *of real images* to construct a compact set.

D4M [23] is a generative distribution matching method that utilizes a pre-trained Latent Diffusion Model (LDM) [20] to generate synthetic images. Multiple prototypes are created for each class using K-means clustering in the latent space, which are then denoised using the pre-trained LDM model before passing them through a pre-trained decoder to produce synthetic images.

Minimax Diffusion [9] incorporates diffusion-transformer (DiT) based generative models to create a distilled set. They do so by finetuning a pretrained DiT model separately for each subset of classes in order to learn their data distribution. This helps ammortize the generation cost for higher IPCs by simply performing inference with the finetuned model multiple times for class-conditional generation. Their loss objective involves three loss terms:

(1) Standard *diffusion loss* $\mathcal{L}_{\text{simple}} = ||\epsilon_\theta(\mathbf{z_t},\ \mathbf{c}) - \epsilon||_2^2$ of predicting the ground truth noise ϵ using the noise prediction network ϵ_θ at a time step $\mathbf{t}$;

(2) a *minimax loss* $\mathcal{L}_r = \arg\max_\theta \min_{m\ \in\ [N_m]} \cos(\hat{\mathbf{z}}_\theta(\mathbf{z_t},\ \mathbf{c}), \mathbf{z}_m)$, which pulls close together the predicted embedding $\hat{\mathbf{z}}_\theta$ to the *least* similar sample $\mathbf{z}_m$ stored in the memory bank $\mathcal{M} = \{\mathbf{z}_m\}_{m=1}^{N_m}$ to ensure representative alignment; and

(3) a *sample diversity loss* $\mathcal{L}_d = \arg\min_\theta \max_{d\ \in\ [N_D]} \cos(\hat{\mathbf{z}}_\theta(\mathbf{z_t},\ \mathbf{c}), \mathbf{z}_d)$, which has an opposite optimization target compared to the minimax loss $\mathcal{L}_r$, in where the predicted embedding $\hat{\mathbf{z}}_\theta$ is pushed away from the *most* similar one stored in another memory bank $\mathcal{D} = \{\mathbf{z}_d\}_{d=1}^{N_D}$.

Thus, the overall loss term becomes:

$$\mathcal{L}_{\text{Minimax}} = \mathcal{L}_{\text{simple}} \ +\ \lambda_r\, \mathcal{L}_r \ +\ \lambda_d\, \mathcal{L}_d \tag{3}$$

where λ_r and λ_d are weighting hyper-parameters.

While these methods utilize various (complex) techniques for image synthesis itself, they share a common strategy of employing multiple soft labels for every augmented image for downstream training of a student model. These soft labels provide a more detailed supervisory signal to guide the student model's training and have been shown to significantly improve the downstream performance. Apart from this, there are generative-model based distillation

methods like D4M [23] and GLaD [2] who perform distillation in the latent space and generates distilled images uisng pretrained decoders. For more details, we refer the reader to Li et al. [16].

1.2. Small-scale DD methods

Small-scale dataset distillation methods synthesize compact datasets through a bi-level optimization framework to enable efficient model training with fewer samples. However, scaling these methods to larger datasets such as ImageNet-1K and to deeper architectures like ResNet-18 remains challenging due to their substantial computational cost. In general, methods in this regime fall into three categories: trajectory matching (TM [1] and DATM [11]) , distribution matching (DM [29]), and gradient matching (DC [30]). We briefly describe these methods and their loss objectives below:

TM [1] aims to synthesize datasets by aligning the optimization trajectories of models trained on real and synthetic data. Specifically, given a real dataset $\mathcal{D}_{\text{train}}$ and a synthetic dataset $\mathcal{S}$, TM updates $\mathcal{S}$ to minimize a trajectory matching loss. This loss compares the model parameters obtained from training on $\mathcal{S}$ for N steps to those obtained from training on $\mathcal{D}_{\text{train}}$ for M steps, starting from a shared initialization point θ_t:

$$\mathcal{L}_{TM}(\mathcal{S}, \mathcal{D}_{\text{train}}) = \frac{\left\| \hat{\theta}_{t+N} - \theta_{t+M} \right\|_2^2}{\|\theta_t - \theta_{t+M}\|_2^2} \tag{4}$$

where θ_t denotes the model parameters at step t obtained from training on the real dataset from random initialization, and $\hat{\theta}_{t+N}$ represents the parameters after training on the synthetic dataset for N steps starting from θ_t. This objective encourages synthetic data to induce training dynamics similar to that of real data over a fixed training horizon. TM also uses an upper bound T^+ on the sampling range of t, so that only model parameters within this range are used for matching. In the original work, TM uses 100 different expert trajectories for synthesis. These expert trajectories are obtained by training on the real dataset with a different random initialization and saving intermediate checkpoints along the way.

DATM [11] improves upon Trajectory Matching (TM) with two main modifications: (1) First, it sets both a lower bound T^- and an upper bound T^+ on the sampling range for t, so that only parameters between these bounds are used for matching; and (2) second, it includes soft labels in the distilled data, which are also optimized during the distillation process.

DM [29] learns a distilled dataset by directly matching the output features of real and synthetic samples, where the features are extracted from a model with randomly initialized weights. The loss objective is to minimize the difference between the average feature representations of real and synthetic data under various augmentations. This is formalized as following:

$$\min_S \mathbb{E}_{v\sim\mathcal{P}_v,\, \omega\sim\Omega} \left\| \frac{1}{|T|}\sum_{i=1}^{|T|} \psi_v(A(x_i,\omega)) - \frac{1}{|S|}\sum_{i=1}^{|S|} \psi_v(A(\tilde{x}_i,\omega)) \right\|^2 \tag{5}$$

where ψ_v is a function mapping inputs to a randomly initialized model's feature space, ω is a data augmentation parameter sampled from a distribution Ω, and $A(\cdot,\omega)$ denotes the augmented version of an input. T is the set of real samples, and S is the synthetic dataset being optimized.

DC [30] matches the gradients of the loss function computed on real and synthetic data with respect to the network parameters θ. The synthetic dataset S and the model parameters θ are optimized in an alternating manner, typically under a bilevel optimization framework. The DC loss objective is formulated as follows:

$$S^* = \arg\min_S \mathop{\mathbb{E}}_{\theta_0\sim\mathcal{P}_{\theta_0}} \left[\sum_{t=0}^{T-1} \mathcal{D}(\nabla_\theta \mathcal{L}_S(\theta_t), \nabla_\theta \mathcal{L}_T(\theta_t)) \right] \quad \text{subject to} \quad \theta_{t+1} = \text{opt\_alg}_\theta(\mathcal{L}_S(\theta_t), \varsigma_\theta, \eta_\theta)\,. \tag{6}$$

where $\mathcal{P}_{\theta_0}$ is the distribution over network initializations, T is the number of outer-loop iterations used to update the synthetic data, ς_θ is the number of inner-loop optimization steps, η_θ is the learning rate for the model parameters, and $\mathcal{D}(\cdot,\cdot)$ measures the distance between real and synthetic gradients.

1.3. Coreset selection

Similar to dataset distillation, coreset selection aims to identify a small, informative subset of the original dataset that, when used for training, achieves performance comparable to training on the full dataset. Broadly, coreset methods can be grouped into the following categories:

1) *Diversity-Based Selection*: These methods aim to select subsets that span the data distribution well, encouraging representativeness and coverage of the input space. Example methods are Facility Location, K-Centers, etc. Facility Location uses pairwise similarity to select diverse examples that "cover" the dataset, whereas K-Centers selects points that are maximally distant from each other or represent cluster centers.

2) *Importance or Influence-Based Selection*: These approaches estimate the influence or importance of each data point based on its effect on model training or generalization. For instance, Craig [18] selects points to approximate the gradient of the full dataset. Glister [14], on the other

Table 1. **Hyperparameter settings for evaluating performance of distilled set**. Below hyperparameters are used to train a student model on a distilled set to evaluate its generalization performance on downstream tasks. KL-Div: Kullback-Leibler Divergence; T: Temperature of KL-Div loss; RRC: Random Resized Crop, HFlip: Random Horizontal Flip, AdamW: Adam optimizer with decoupled Weight decay, Cosine: Cosine annealing; DSA: Differentiable Siamese Augmentation.

Hyperparameter	**Large-scale (ImageNet-1K)**		**Small-scale (Tiny-IN / CIFAR-100)**	
	Hard Label (HL)	**Soft Label (SL / SL+KD)**	**HL setting**	**SL setting**
Epochs	200	200	300	300
Loss function	Cross-Entropy (CE)	KL-Div (T=20)	Cross-Entropy (CE)	KL-Div (T=20)
Optimizer	SGD	AdamW	SGD	AdamW
Learning rate	0.5	1e-3	1e-2	1e-3
Scheduler	Cosine	Cosine	StepLR@151	Cosine
Batch size	50 (IPC 10), 128 (IPC 50), 200 (IPC 100)	128	256	256
Augmentations	RRC + HFlip (+ PatchShuffle aug. for RDED-type sets)	RRC + HFlip (+ PatchShuffle aug. for RDED-type sets) (+ Cutmix aug. for SL+KD)	DSA	DSA
Other details	–	Warm-up of 5 epochs	–	–

hand, uses bilevel optimization to select subsets that maximize validation performance.

3) *Uncertainty and Informativeness-Based Selection*: These methods select examples that are likely to be the most informative or uncertain, often inspired by active learning. One example in this category is Entropy or Least Confidence, which selects examples with high predictive uncertainty. Another example is Margin Sampling, which picks points close to the decision boundary.

4) *Learning Dynamics and score-based Selection*: Approaches in this category leverage model training behavior to identify informative or difficult examples. For example, Forgetting [25] tracks how often samples are forgotten during training, whereas GraNd / EL2N [19] ranks samples using the gradient / error norm of the loss over a few training epochs.

DeepCore [10] provides a comprehensive benchmarking of these methods across different datasets and tasks, highlighting their strengths and limitations in practical settings.

2. Details on Training and Hyperparameters

We describe relevant hyperparameters for evaluation of both small-scale and large-scale DD methods in this section. Note that the hyperparameters for the teacher training and synthesis stage vary depending on the approach utilized. As we perform synthesis of any distilled set using their official codebase and recommended hyper-parameter settings (and use pretrained distilled sets when available), we mainly provide the details of the student training on distilled sets, which are summarized in Tab. 1. EDC [22] explores the design space of large-scale dataset distillation methods in SL+KD regime and highlights the issue of sub-optimal hyper-parameter settings in training a student model on a distilled set. Subsequently, it proposes a series of hyperparameter changes to extract the best performance out of a synthesized dataset. For instance, it proposes optimal learning rates, batch sizes and scheduler which substantially improves student performance on various downstream tasks. Therefore, we adopt EDC's hyperparameter configuration of large-scale methods and use it in the SL / SL+KD setting to enable an optimal comparison of different methods. For the HL setting, we tune the optimal hyper-parameters for coresets as well as for the distilled sets around the settings taken from DeepCore [10]. Additionally, we tune for the training batch sizes as per RDED [24].

Table 2. **Hyperparameters used for DATM synthesis.**

N	M	T^-	T^+	Synthetic Batch Size	Learning Rate (Label / Pixels)
80	2	20	35	250	10 / 100

DATM / TM hyperparameters. We also specify hyperparameters for DATM / TM synthesis experiments on TinyImageNet for DCS. We use the DATM [11] variant of TM loss for analysis, which sets a lower bound T^- along with an upper bound T^+ on the sampling range for the starting checkpoint for matching trajectories. Recall from Sec. 1.2 of the supplementary that the TM loss has two additional hyperparameters M and N. For our loss objective analysis experiments (DCS), we set $N = 80$ and $M = 2$. For synthesis, we use 100 expert models, each trained for 50 epochs. Remaining hyperparameters are summarized in Tab. 2.

3. Fitting scaling law equation for label regimes

To quantify the interplay between dataset quality, size, and compute, we adopt the data-aware scaling law introduced by Goyal et al. [8]. The predicted performance at epoch k is modeled as

$$y_k = a \cdot n_1^{b_1} \prod_{j=2}^{k} \left(\frac{n_j}{n_{j-1}} \right)^{b_j} + d, \tag{7}$$

$$b_{k+1} = b \cdot \left(\frac{1}{2} \right)^{\frac{k}{\tau}} = b \cdot \delta^k \tag{8}$$

Here, a is a normalizing factor; b_j governs diminishing gains as more data is seen and also reflects the intrinsic utility of the data pool, with lower b indicating higher utility; δ captures the decay in data utility with repetition; d represents the irreducible loss that cannot be further reduced; and τ indicates how the utility of a sample decays slowly with repetition.

We fit this scaling law to performance curves of IPC 100 subsets of random and EL2N-Easy under two label regimes: SL+KD and HL. In the SL+KD regime, we find that both the subsets, which vary highly in their underlying quality, exhibit nearly identical scaling behavior, with $b \approx 0.05$ and $\delta = 2.8947$. In contrast, the HL regime reveals a strong dependence on data quality: random subsets yield $b = -0.1859$ and $\delta = 5.2632$, while EL2N subsets exhibit $b = -0.1558$ and $\delta = 1.9474$, reflecting slower saturation of informative samples.

Implication. These scaling dynamics results indicate that the label regime of SL+KD, where abundant soft labels from a teacher model is used, the supervision nullifies the effect of data utility on downstream performance. This is in stark contrast to the hard-label (HL) regime, where the contribution of sample quality continues to strongly influence the model performance as evidenced by differing values of b and δ.

4. Derivation of EL2N-SL Score

Assumption 1 (Per-logit gradient orthogonality [19])**.** *For a model at time t and input x, let $\psi_t^{(k)}(x) = \nabla_{w_t} f_t^{(k)}(x)$ be the parameter gradient of the k-th logit. We assume*

$$\langle \psi_t^{(k)}(x), \psi_t^{(j)}(x) \rangle \approx 0 \quad (k \neq j), \quad \|\psi_t^{(k)}(x)\|_2 \approx c, \ \forall k,$$

for some constant $c > 0$ independent of k.

Lemma 1. *Let $p(w_t, x)$ be the temperature-T softmax of logits $f(w_t, x)$,*

$$p_i = \frac{\exp(f_i/T)}{\sum_j \exp(f_j/T)},$$

and let q be a target (soft) distribution. For notation brevity, let $f_t := f(w_t, .)$. Under Assumption 1, the GraNd score

$$\chi_t(x, q) = \mathbb{E} \left\| \sum_{k=1}^{C} \left(\nabla_{f^{(k)}} \ell(f_t(x), q) \right)^T \psi_t^{(k)}(x) \right\|_2$$

satisfies

$$\chi_t(x, q) \approx \frac{c}{T} \, \mathbb{E} \|p(w_t, x) - q(w_t, x)\|_2,$$

hence the EL2N-SL score may be defined (up to a constant) as

$$\text{EL2N-SL}(x) = \frac{1}{T} \, \mathbb{E} \|p(w_t, x) - q(w_t, x)\|_2.$$

Proof. Differentiate the temperature-scaled softmax to get

$$\nabla_{f_k} p_i = \frac{1}{T} p_i (\delta_{ik} - p_k).$$

Thus, for $\mathrm{KL}(q \| p)$ loss, the per-logit gradient is

$$\nabla_{f^{(k)}} \ell(f_t(x), q) \ = \ \frac{1}{T} \left(p_k - q_k \right),$$

Substitute into the GraNd definition $\chi_t(x, q)$,

$$\begin{aligned} \chi_t(x, q) &= \mathbb{E} \left\| \sum_{k=1}^{C} \frac{1}{T} (p_k - q_k) \, \psi_t^{(k)}(x) \right\|_2 \\ &= \frac{1}{T} \, \mathbb{E} \left\| \sum_{k=1}^{C} (p_k - q_k) \, \psi_t^{(k)}(x) \right\|_2, \end{aligned}$$

Under Assumption 1, the cross-logit gradient terms vanish and as each $\|\psi_t^{(k)}\|_2 \approx c$, therefore

$$\begin{aligned} \left\| \sum_k (p_k - q_k) \psi_t^{(k)} \right\|_2^2 &= \sum_k (p_k - q_k)^2 \|\psi_t^{(k)}\|_2^2 \\ &\approx c^2 \sum_k (p_k - q_k)^2 = c^2 \|p - q\|_2^2. \end{aligned}$$

Taking square-root and expectation gives the claim:

$$\chi_t(x, q) = \frac{c}{T} \, \mathbb{E} \|p - q\|_2.$$

Absorbing the constant c into normalization yields the stated EL2N-SL definition. □

5. Role of teacher strength in SL+KD saturation

In order to ascertain the role of teacher strength in the SL+KD saturation results observed in the main paper, we take two additional teachers: (a) a MobileNet-V2 (MNV2) model and (b) a weaker RN18 teacher and use them to obtains soft labels for the the student model (ResNet-18). The results, shown in Fig. 1, demonstrate that our key observation still holds in the SL+KD regime across teachers of different strengths, i.e., a saturation in model performance across subsets of varying data quality and size.

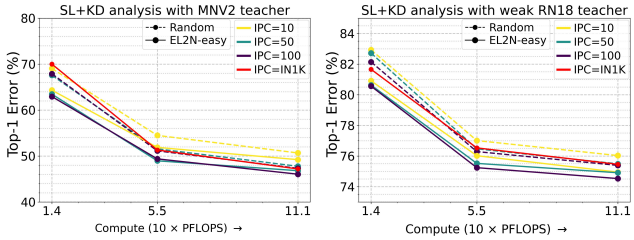

Figure 1. **Role of teacher strength in SL+KD saturation.** We perform scaling analysis in the SL+KD regime with different teachers: (a) MobileNet-V2 (MNV2) teacher, and (b) a *weak* ResNet-18 (RN18) teacher, in order to ascertain the robustness of our observations in the SL+KD regime with varying strength of the teacher models. From the results, we find that, indeed, our observations do hold strong despite varying teacher signals: that *despite varying data quality and size, performance saturation is inevitable given a fixed compute budget*.

6. Additional analysis for small-scale methods

6.1. CIFAR-100 Evaluation

In this section, we report the results of small-scale DD methods on CIFAR-100 dataset under the fixed HL and SL settings. The three methods considered in this evaluation[1] are DC [30], DM [29] and TM [1]. We baseline all the considered small-scale methods against standard coreset techniques (K-Centers [10] and random real subset), which consists of real images and do not involve any costly synthesis process. We adopt the standard setup provided by DCBench [4] for our evaluation. The model architecture is ConvNet-D3, and we compare performance for both IPC 10 and IPC 50. The results are summarized in Tab. 3 . The hyperparameters details are provided in Sec. 2 of the supplementary.

While TM [1] exhibits a *clear advantage over coreset baselines in the HL setting on CIFAR-100*, this advantage substantially diminishes once we transition to the SL regime—*even at low IPC*. For example, although TM exceeds K-Centers by approximately 8–13% under HL supervision, it no longer maintains this margin when soft labels are fixed. These results reinforce the pattern observed in Sec. 3.2 of the main paper: *subset quality, which is highly influential in the HL setting, plays a far more limited role in the SL regime*, even on a different dataset such as CIFAR-100.

[1]We exclude DATM [11] from this evaluation due to a significant discrepancy between the performance we observed (specifically, substantially lower accuracy) when running the official DATM code on their provided distilled CIFAR-100 sets and the results reported in their paper. This issue has also been identified by other researchers, for instance, https://github.com/NUS-HPC-AI-Lab/DATM/issues/17. In contrast, the results for DATM on TinyImageNet are consistent with the reported numbers in their paper. Therefore, we report DATM results only for TinyImageNet.

Table 3. **Performance comparison of small-scale DD methods with coresets on CIFAR-100 in HL and SL setting**. Model architecture is ConvNet-D3. The substantial performance gap in the HL setting closes when trained with fixed soft labels.

Method	Hard Label (HL)		Fixed Soft Label (SL)	
	IPC 10	IPC 50	IPC 10	IPC 50
Dataset Distillation				
DM [29]	29.23 ± 0.26	42.32 ± 0.37	26.13 ± 0.10	43.46 ± 0.18
DC [30]	28.42 ± 0.29	30.56 ± 0.56	23.54 ± 0.31	33.46 ± 0.38
TM [1]	**38.18** ± **0.42**	**46.32** ± **0.26**	**37.60** ± **0.25**	**46.26** ± **0.30**
Coreset Selection				
Random Real	18.64 ± 0.25	34.66 ± 0.41	33.43 ± 0.18	45.39 ± 0.23
K-centers	25.04 ± 0.30	38.64 ± 0.43	34.70 ± 0.27	**46.24** ± **0.12**

Table 4. **CIFAR-100 Cross-Architecture Transfer performance comparison of small-scale DD methods with coresets in HL and SL setting**. Model architecture is ConvNet-D3. The substantial performance gap in the HL setting closes when trained with fixed soft labels.

Method	Avg. Transfer (HL)		Avg. Transfer (SL)	
	IPC 10	IPC 50	IPC 10	IPC 50
Dataset Distillation				
DM [29]	12.44	23.59	15.69	33.51
DC [30]	11.86	14.10	12.39	21.35
TM [1]	**15.62**	**30.29**	**23.64**	36.36
Coreset Selection				
Random Real	10.39	21.86	20.36	35.88
K-centers	14.41	24.86	**22.99**	**37.03**

6.2. Cross-Arch. Transfer: TinyIN and CIFAR-100

Following the DCBench [4] setup, we evaluate three model architectures – MLP (411K), ResNet-18 (11M) and ResNet-152 (60M), and report their average transfer accuracy under the sub-heading "Avg. Transfer" in Tab. 4 and Tab. 5. When evaluated under the fixed soft-label (SL) setting, coreset methods like K-Centers perform competitively and on-par with state-of-the-art methods like TM and DATM.

7. Additional results on DCS

7.1. Small-scale methods

TM [1]. In this section, we further examine the behavior of the TM loss objective on TinyImageNet to further validate our observations discussed in Sec. 4.1 of the main paper. For this, we use two model architectures: ConvNet-D4 and ResNet-18. We use the DATM [11] variant of TM loss for analysis, which sets a lower bound T^- along with an upper bound T^+ on the sampling range for the starting checkpoint for matching trajectories. Recall from Sec. 1.2

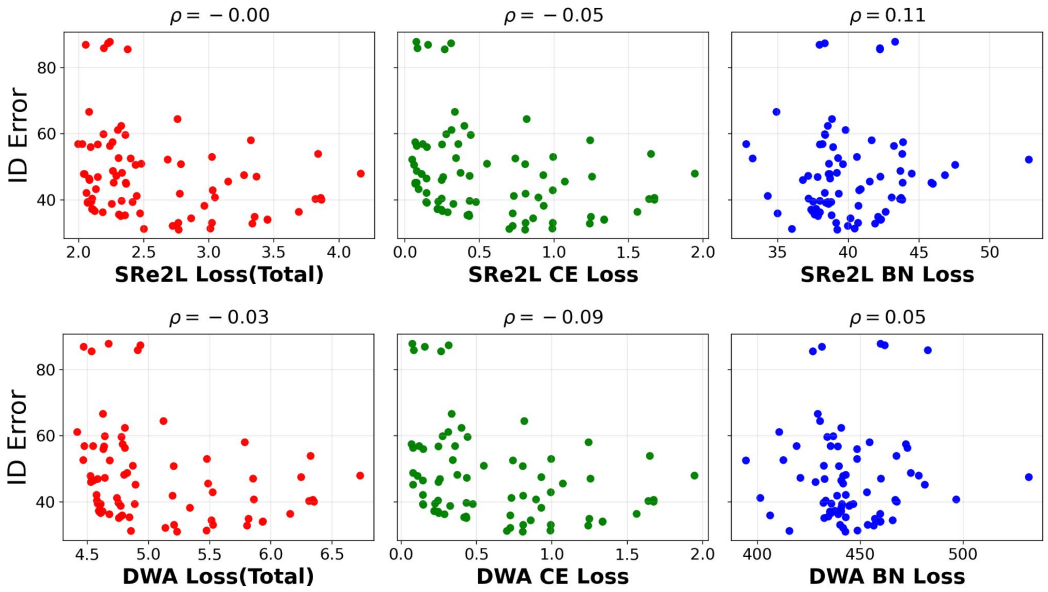

Figure 2. **Correlation analysis of distillation loss objectives on ImageNet-1K**. We compute the proposed DCS score (see Sec-4 of the main paper) for SRe2L and DWA across multiple IPC settings and data subsets (each data point in the plot represents IPC-subset combination, see Sec. 7.2 for details and discussion). One can observe a mis-alignment between these distillation objectives and their generalization performance, with either zero or negative Spearman correlation after adjusting for the bias of size of subsets.

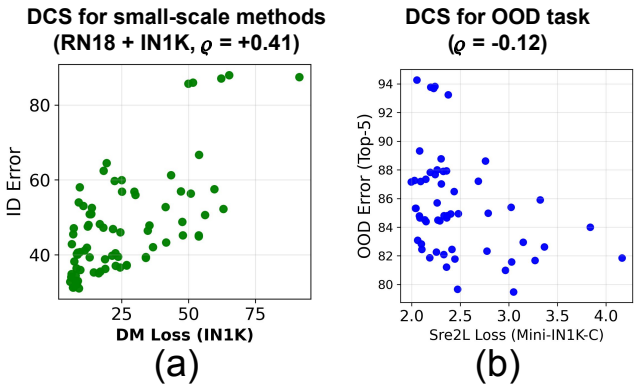

Figure 3. **DCS Additional results**. **(a) DCS on small-scale method DM**. We use DCS to plot the correlation of DM [29] loss objective with ID generalization error and find better-than-TM but modest correlation of $\rho = 0.41$. **(b) DCS robustness across tasks.** We also evaluate the robustness of DCS for an OOD task different than the ID task for SRe2L [28] on Mini-ImageNet-C dataset, and find that the DCS outputs a similar correlation score across both ID and OOD task ($\rho \sim -0.12$).

of the supplementary that the TM loss has two additional hyperparameters M and N. For our loss objective analysis experiments, we set $N = 80$ and $M = 2$. For both architectures, we visualize two things: (1) The first shows the TM loss variation as a function of the starting expert checkpoint. To obtain the final TM loss value, we average the loss values across T^- and T^+ epochs ($T^- = 20$ and $T^+ = 35$ here); (2) we also make a scatter plot of these averaged loss values against the In-Distribution (ID) accuracy of the considered subsets for both IPC 10 and IPC 50 (specifically, we consider the six methods discussed in Sec. 3.2 of the main paper).

From the Fig. 4(a), one can observe that for small models like ConvNet-D4, there is a reasonable variation in the TM Loss for different subsets, which reflect the different generalization performances of those sets. However, for more deeper and larger architectures like ResNet-18, the loss exhibits almost no discernible variation, with its value remaining nearly constant at around 0.805 for all the considered subsets. Further, as shown in Fig. 3(a) of the main paper, the scatter plot of averaged TM Loss for various methods reveal no correlation with in-domain generalization performance of these subsets.

To further ascertain the above observations, we perform the actual synthesis process using the DATM method on TinyImageNet for both ConvNet-D4 and ResNet-18, and track its loss training dynamics. For training, we set N=20 and M=2, based on the maximum available GPU memory for ResNet-18. We include the plots for IPC=50 (Fig. 4(b)) in the supplementary for completeness. The plots for IPC=10 were included in Sec. 4.1 of the main paper.

In these plots, we track both the "Grand Loss" (TM matching loss across multiple trajectores) as well as the accuracy of the distilled set as one synthesizes for more iter-

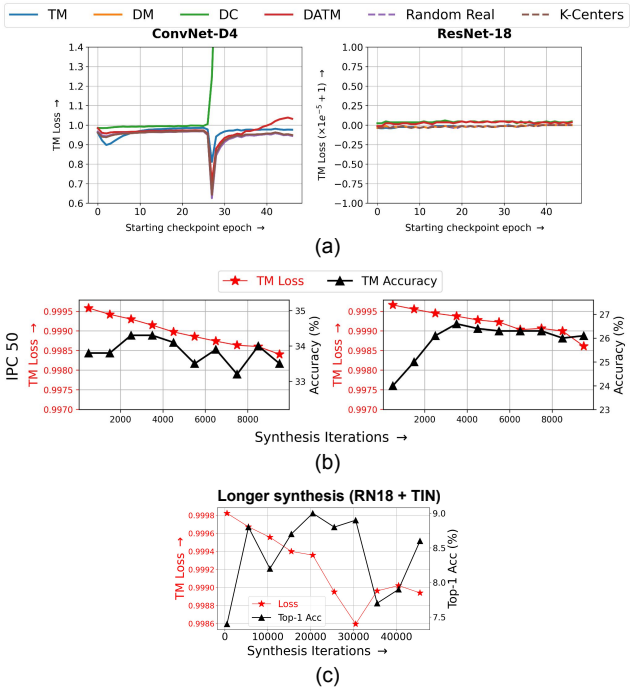

Figure 4. **(a) Analysis of TM Loss objective behavior for different synthesis methods on TinyImageNet.** We calculate the TM Loss for various distilled sets with trajectories starting from different training epochs for both ConvNet-D4 and ResNet-18 model. For deeper and larger architectures like ResNet-18, TM Loss fails to capture any meaningful variation, settling around a constant value of ~ 0.806 regardless of the method considered. Note that DC loss suddenly shoots up around the dip point (epoch $27-28$) for ConvNet-D4, and hence we clip the loss range to display meaningful variations. **(b) Training dynamics of DATM synthesis on TinyImageNet.** We track DATM synthesis for ConvNet-D4 (left) and ResNet-18 (right) at IPC 50. Both TM loss (red curve) and accuracy (**black** curve) show minimal change – loss varies only in the 3rd decimal place, and accuracy improves by just $2-3\%$ over its initial accuracy. **(c) Longer Synthesis with DATM loss.** We ascertain if the training dynamics of DATM loss presented in (b) is because of slower convergence. For this, we perform $5\times$ longer synthesis with the DATM loss than in (b) to check for convergence, and observe no observable trend (increasing Acc or decreasing Loss) in the curves, validating our observation that the TM loss objective indeed does not scale well to larger settings.

ations. Note that despite $10k$ synthesis iterations, one can hardly observe any significant change in both loss and accuracy, with loss value changing in 3rd decimal place and accuracy improving only by $2-3\%$ over the base accuracy, which is already relatively high due to initialization of the synthesis set with correctly classified real samples. To confirm that this is not a convergence issue, we also perform a $5\times$ longer synthesis with the same loss (i.e., $50k$ iterations), and plot its loss and accuracy trend in Fig. 4(c). The lack of any consistent trend in the accuracy confirms our

Table 5. **TinyImageNet Cross-Architecture Transfer performance comparison of small-scale DD methods with coresets in HL and SL setting**. Model architecture is ConvNet-D4. The substantial performance gap in the HL setting closes when trained with fixed soft labels.

Method	Avg. Transfer (HL)		Avg. Transfer (SL)	
	IPC 10	IPC 50	IPC 10	IPC 50
Dataset Distillation				
DM [29]	2.96	6.71	5.03	19.64
DC [30]	3.60	4.41	2.49	3.48
TM [1]	**5.50**	**10.26**	**10.27**	20.32
DATM [11]	5.22	10.12	8.93	**23.83**
Coreset Selection				
Random Real	2.49	6.17	9.28	22.91
K-centers	3.91	7.80	**10.51**	**23.53**

observation that using TM-based loss objective for synthesis on larger models and datasets is not useful, as it's unable to capture any meaningful changes in the model paramters relevant for downstream generalization.

DM [29]. We evaluate the distribution-matching (DM) objective using DCS on the ResNet-18 model with ImageNet-1K (IN1K) dataset, and plot its correlation in Fig. 3(a). We find that, although DM has a much better correlation than TM with downstream generalization, it's still a modest correlation ($\rho = 0.41$) and not very strong. Moreover, we also observe that DM exhibits greater variation in both loss and downstream performance than TM during the synthesis stage.

7.2. Large-scale methods: SRe2L [28] and DWA [7]

We employ DCS to evaluate two large-scale distillation objectives: SRe2L [28] and DWA [7], which match BatchNorm statistics of a teacher model as a proxy for distribution alignment. A more detailed description of these loss objectives along with their equation is included in Sec-1 of the supplementary material. We conduct this analysis across seven different IPC values ($10-700$) with a diverse set of data subsets S_j coming from four coreset methods (EL2N [19], EL2N-pareto fractions (see Sec. 3.1 of the main paper), CAL [17] and GraphCut [13]). For each IPC, we intentionally exclude extremely poor-performing subsets to ensure that the evaluation focuses on whether the loss objectives can meaningfully differentiate among the strongest candidate subsets at that IPC. In addition to these plausible subsets, we also include *adversarial degenerate subsets* by repeating the elements of the subset twice, allowing us to test whether the loss objectives can correctly identify these pathological cases wherein, repetition of the data results in redundancy and a good distillation loss objective should penalize that in favour of diversity and good dis-

tribution coverage. All of these subset variants combine to 158 data points per objective for correlation evaluation. We choose ResNet-18 and ImageNet-1K dataset as the evaluation benchmark and report Spearman correlation coefficient for each objective after accounting for the confounding effect of subset size, which can otherwise bias the correlation estimates.

The results are shown as scatter plots in Fig. 2. From the figure, it is evident that there is a significant mis / non-alignment between the distillation objectives and generalization performance, with SRe2L and DWA exhibiting *no correlation*, indicating that lower loss values in these objectives have no effect on better downstream generalization.

7.3. Robustness of DCS

To evaluate whether DCS remains informative beyond in-distribution settings, we provide additional results on the out-of-distribution (OOD) generalization task. Specifically, we compute DCS for SRe2L on Mini-ImageNet-C and present the corresponding results in Fig. 3(b). From the figure, it is evident that there is a significant misalignment between the the SRe2L objective and OOD generalization performance as well, which is captured accurately by DCS ($\rho = -0.12$).

This finding is further corroborated by the poor OOD performance of SRe2L at IPC 50, which achieves only 3.56% accuracy compared to 11.72% for a class-balanced random subset. These results suggest that DCS serves as a reliable indicator of dataset quality across both in-distribution and out-of-distribution evaluation regimes, making it a broadly applicable metric for assessing DD methods.

8. Additional results on the proposed method

As shown in the main paper, soft-label regimes (SL and SL+KD) suffer from the undesirable property that *data quality ceases to matter*, limiting the value of quality pruning metrics. Therefore, we focus our additional analyses on the HL regime, where sample quality remains influential, and evaluate both CAD-Prune and CA2D in greater depth.

8.1. Performance comparison with recent coreset and DD methods

Large-scale methods. We provide comparisons of our proposed DD method (CA2D) with some of the recently proposed large-scale DD techniques [5] for completeness[2]. Note that these methods may have used potentially different evaluation settings to assess the performance of their distilled sets, including evaluation under a different label regime, or using different training hyper-parameters like no. of epochs for student training. Therefore, we take their publicly available distilled sets and evaluate them under the HL regime in our evaluation setup to maintain a fair comparison. The results for the DD method comparison are shown in Tab. 6, where it is evident that the proposed technique CA2D outpeforms its baselines on IPC 50 [3].

Table 6. **Performance comparison of proposed methods with recently proposed DD techniques.** We add comparison against FAD-RM [5] here for completeness, demonstrating that CA2D outperforms it and other DD baselines at IPC 50. See Sec. 8.1 for the discussion.

IN1K + HL	SRe2L	D4M	RDED	FAD-RM [5]	CA2D (Ours)
IPC 50	9.79	21.56	38.49	21.21	**41.56**

Table 7. **Performance comparison of proposed methods with recently proposed coreset techniques.** We add comparison against Dyn-Unc [12] and DUAL [3] here for completeness, demonstrating that the proposed methods performs on-par or better compared to these baselines. See Sec. 8.1 for discussion.

Method	**IPC 10**	**IPC 50**	**IPC 100**
Dyn-Unc [12]	12.15 $\pm$ 0.22	33.03 $\pm$ 0.02	42.01 $\pm$ 0.10
DUAL [3]	**15.65** $\pm$ **0.17**	38.28 $\pm$ 0.09	46.22 $\pm$ 0.10
CAD-Prune (Ours)	12.57 $\pm$ 0.03	40.21 $\pm$ 0.15	**47.40** $\pm$ **0.20**
CA2D (Ours)	**15.25** $\pm$ **0.24**	**41.72** $\pm$ **0.40**	46.32 $\pm$ 0.10

We also compare CA2D and CAD-Prune with some latest state-of-the-art coreset methods such as Dyn-Unc [12] and DUAL [3], and report the results in Tab. 7, where we can observe on-par or better performance of the proposed methods against these baselines.

Small-scale methods. Small-scale result comparison of the proposed methods with existing popular baselines are provided in Tab. 8 (setting: ConvD3 arch. + CIFAR-100). As expected, due to the very low-resolution nature of the images in CIFAR-100 dataset, the performance is expectedly weaker.

8.2. Cross-Architecture Transfer

We perform subset synthesis / selection using ResNet-18 model architecture and then train a ResNet-50 / ResNet-101 model on the obtained subset. Results are presented in Tab. 9 for both IPC 50 and 100. The proposed compute-aware pruning method CAD-Prune matches the performance of the compute-intensive sliding window method EL2N-Best [15, 19] on larger architectures and across IPC values while being more efficient. The proposed DD method CA2D also outperforms RDED [24] across all settings, demonstrating the benefit of a optimal set selection to

[2]Note that we only consider papers which have been peer-reviewed and published at major AI conference venues.

[3]Note that we only consider IPC 50 for comparison due to unavailability of distilled sets for Cui et al. [5] for any other IPC.

Table 8. **Performance comparison of proposed methods with SOTA samll-scale DD and coreset techniques on CIFAR-100 in the HL setting.** Model architecture is ConvNet-D3. Due to the very low-resolution nature of the setup (CIFAR-100), performance of the propoesed techniques is expectedly lower.

Method	IPC 10	IPC 50
TM [1]	**38.18** ± **0.42**	**46.32** ± **0.26**
CA2D (factor=1)	20.95 ± 0.15	34.37 ± 0.11
Random Real	18.64 ± 0.25	34.66 ± 0.41
K-centers	**25.04** ± **0.30**	**38.64** ± **0.43**
CAD-Prune	22.95 ± 0.03	37.87 ± 0.16

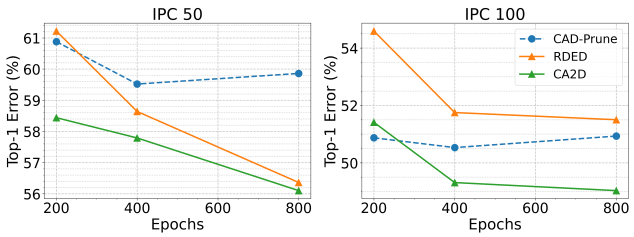

Figure 5. **Convergence analysis of DD methods vs coresets.** We plot downstream student performance (Top-1 Error) as a function of training epochs. One can observe that performance keeps improving for distilled sets (solid line) with longer training, while it saturates for coresets (dashed line), indicating the existence of *compression-extraction trade-off* in training on distilled set.

construct the distilled set.

8.3. Convergence Analysis with longer training

The goal of dataset distillation is to compress a large dataset into a small synthetic one on which a student model trained under a chosen label regime (HL / SL / SL+KD) can still achieve strong downstream performance. We observe a potential artifact of this compression process on the performance of the distilled sets: *Compressing rich feature information into fewer synthetic samples proportionately increases the optimization effort required to extract them in the first place*. We term this as *compression-extraction trade-off* in dataset distillation.

We demonstrate this in Fig. 5, where we show that, unlike coreset methods (like the proposed CAD-Prune), where performance saturation happens with longer training (indicating convergence in the HL regime), distilled sets like RDED and CA2D continue to improve with extended training, especially for smaller IPC settings, indicating that new information is still being extracted from the distilled samples. This analysis provides two key observations pertinent to dataset distillation research, where there exists an inherent trade-off in compression vs information extraction from a synthesized dataset:

- A certain minimum amount of compute budget is necessary to extract full information from a compressed set, and
- This extraction process is atleast 2× slower in comparison to coresets, where the underlying samples are from the real data distribution with no additional compression.

Table 9. **ImageNet-1K Cross-Arch. Transfer performance of proposed methods CAD-Prune and CA2D in HL setting.** Architecture used to perform selection / synthesis is ResNet-18. Student training is for 200 epochs. The proposed compute-aware pruning method CAD-Prune matches the expensive sliding window selection method EL2N-Best while being much more efficient. Similarly, the proposed CA2D built using CAD-Prune outperforms RDED across IPC and architectures. RN50=ResNet-50, RN101=ResNet-101.

Method	IPC 50		IPC 100	
	RN50	RN101	RN50	RN101
Coreset Selection (224 × 224)				
Random Real	32.70	31.97	45.20	47.01
EL2N-Best	**40.12**	**41.18**	51.24	**52.11**
CAD-Prune (Ours)	**40.37**	**41.34**	**52.00**	49.94
Dataset Distillation (224 × 224)				
RDED	37.64	36.60	47.71	49.11
CA2D (Ours)	**38.15**	**42.16**	**50.41**	**52.47**

8.4. Results on Higher IPC settings

Although evaluation results in dataset distillation is reported at typical IPC values of 10-100, coreset methods report performance values across the IPC range, focusing on the lossless setting which lean towards higher IPC values. To validate the robustness of our proposed compute-aware pruning method CAD-Prune in such IPC ranges, we evaluate it on IPC values of 200 and 500, and report its performance in Tab. 10. One can observe that the selection mechanism in CAD-Prune clearly maintains its effectiveness in choosing the best scoring subset for the given downstream compute budget by matching the best results obtainable for that IPC value (exhaustive sliding window approach), as shown by the EL2N-Best results.

8.5. Choice of the scoring checkpoint

Recall that the main hypothesis behind our proposed CAD-Prune pruning method is that, given a downstream compute budget for training a student model (e.g., 200 epochs on IPC 50), we select scores from a full-dataset training checkpoint whose training duration is compute-aligned with the given downstream budget. Doing so ensures we pick samples that are actually being actively learned when trained for the given budget. To validate that such a compute-aligned checkpoint selection is optimal for a given budget, we perform an ablation, wherein, we select two checkpoints: (1)

Table 10. **Performance comparison of proposed pruning method CAD-Prune with EL2N-Best at high IPC settings**. We demonstrate that the proposed compute-aware pruning method CAD-Prune works on-par (or better) w.r.t a compute-heavy sliding window method of "EL2N-Best" on IPC values beyond the ranges evaluated in DD literature (and more typical of coreset literature).

Method	IPC 200	IPC 500
EL2N-Best	56.74 ± 0.03	64.07 ± 0.04
CAD-Prune (Ours)	**57.07** ± **0.04**	**65.46** ± **0.19**

Table 11. **Performance comparison of coresets with different scoring checkpoints for pruning**. Coresets are chosen based on scores from different checkpoints. Depending on a downstream compute budget (e.g., 200 epochs of IPC 50), choosing a compute-aligned training checkpoint ("Compute-aware") performs much superior to early checkpoint of a longer training.

Scoring checkpoint	IPC 50	IPC 100
Early of full-training	30.95 ± 0.35	39.94 ± 0.25
Compute-aware	**40.21** ± **0.15** (+ 9.3)	**47.40** ± **0.20** (+ 7.5)
EL2N-Best	38.44 ± 0.30	**47.76** ± **0.43**

early checkpoint (matching downstream compute budget) from a longer training, and (2) final model with compute-aligned full-training on the full dataset; and select coresets based on the scores obtained from these checkpoints, and train a downstream model and evaluate it. The results are reported in Tab. 11, where we can find that compute-aware (full-training) checkpoint performs much better compared to the early chosen checkpoint from a longer training run, and it is on-par / slightly better than the exhaustive sliding-window approach of finding the best score range for a given IPC (EL2N-Best).

9. Image Visualization

Finally, we provide visualization of synthetic / selected images of various methods considered in the work for qualitative comparison, including our proposed pruning method CAD-Prune and the proposed DD method CA2D. They are shown in Fig. 6 – Fig. 10 for different classes of ImageNet-1K. Each row corresponds to one coreset / DD method, and we display five images per method for a particular class.

Figure 6. **Class: Bakery** in ImageNet-1K.

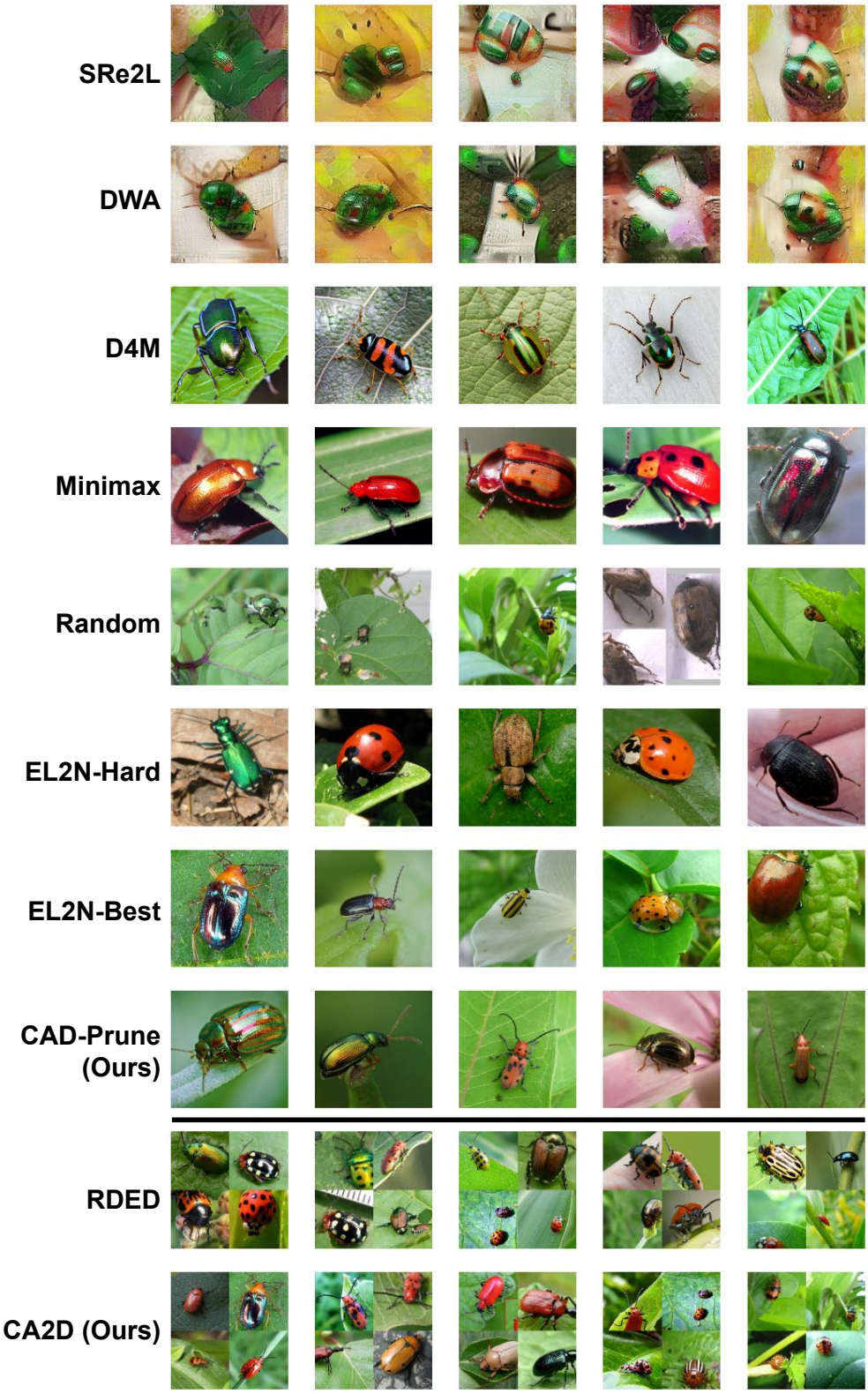

Figure 7. **Class: Leaf Beetle** in ImageNet-1K.

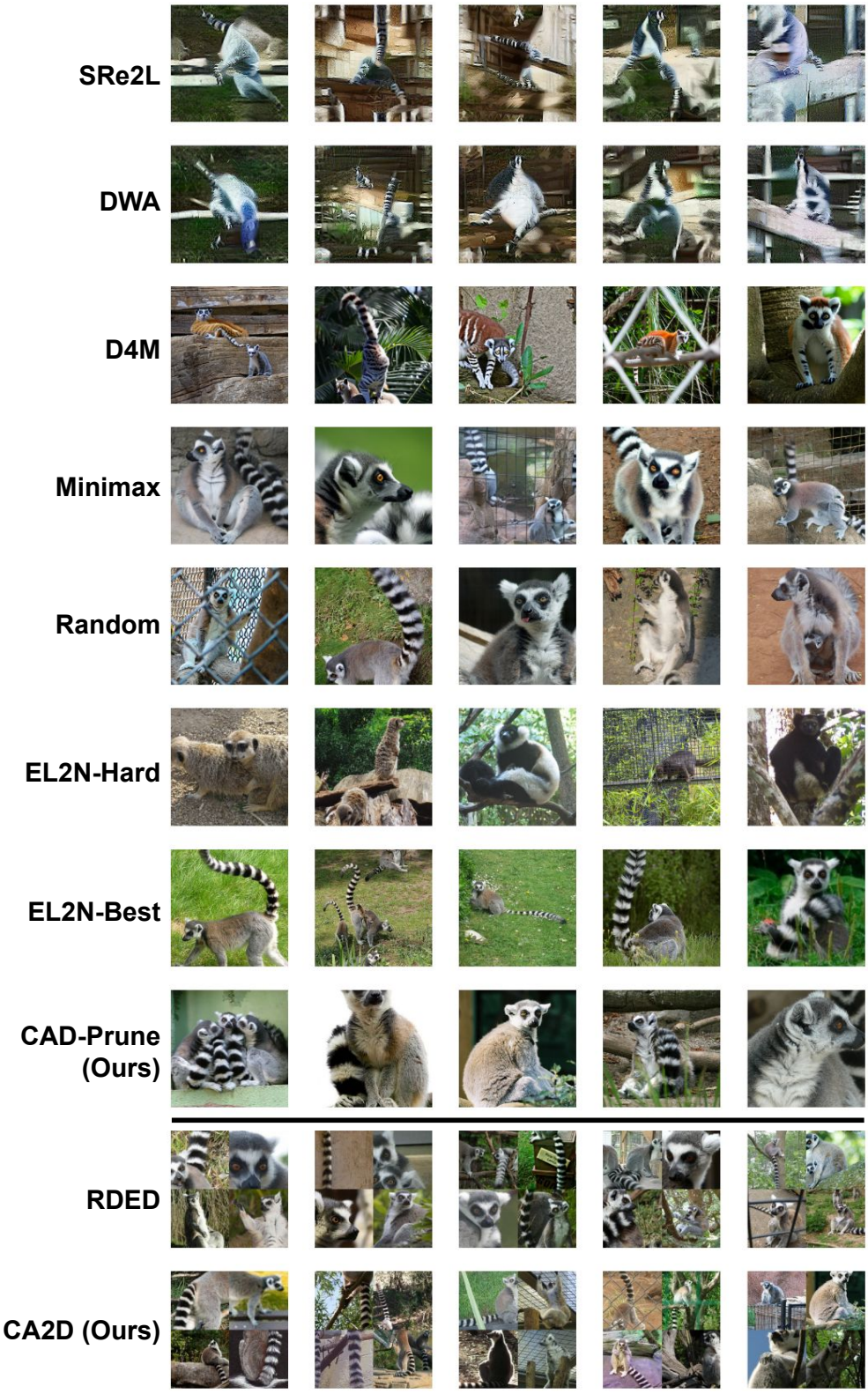

Figure 8. **Class: Madagascar Cat** in ImageNet-1K.

Figure 9. **Class: Boston Bull** in ImageNet-1K.

Figure 10. **Class: Garbage Truck** in ImageNet-1K.

References

[1] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 1, 2, 5, 7, 9

[2] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023. 2

[3] Yeseul Cho, Baekrok Shin, Changmin Kang, and Chulhee Yun. Lightweight dataset pruning without full training via example difficulty and prediction uncertainty. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 10602–10643. PMLR, 2025. 8

[4] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. *Advances in Neural Information Processing Systems*, 35:810–822, 2022. 5

[5] Jiacheng Cui, Xinyue Bi, Yaxin Luo, Xiaohan Zhao, Jiacheng Liu, and Zhiqiang Shen. Fast and accurate data residual matching for dataset distillation. In *Advances in Neural Information Processing Systems*, 2025. 8

[6] 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. 1

[7] Jiawei Du, Juncheng Hu, Wenxin Huang, Joey Tianyi Zhou, et al. Diversity-driven synthesis: Enhancing dataset distillation through directed weight adjustment. *Advances in neural information processing systems*, 37:119443–119465, 2024. 1, 7

[8] Sachin Goyal, Pratyush Maini, Zachary C Lipton, Aditi Raghunathan, and J Zico Kolter. Scaling laws for data filtering–data curation cannot be compute agnostic. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22702–22711, 2024. 4

[9] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[10] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022. 3, 5

[11] Ziyao Guo, Kai Wang, George Cazenavette, HUI LI, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2024. 2, 3, 5, 7

[12] 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 (CVPR) Workshops*, 2024. 8

[13] Rishabh Iyer, Ninad Khargoankar, Jeff Bilmes, and Himanshu Asanani. Submodular combinatorial information measures with applications in machine learning. In *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, pages 722–754. PMLR, 2021. 7

[14] Krishnateja Killamsetty, Durga Sivasubramanian, Ganesh Ramakrishnan, and Rishabh Iyer. Glister: Generalization based data subset selection for efficient and robust learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 8110–8118, 2021. 2

[15] Yongmin Lee and Hye Won Chung. Selmatch: Effectively scaling up dataset distillation via selection-based initialization and partial updates by trajectory matching. In *Forty-first International Conference on Machine Learning*, 2024. 8

[16] Guang Li, Bo Zhao, and Tongzhou Wang. Awesome dataset distillation. `https://github.com/Guang000/Awesome-Dataset-Distillation`, 2022. 2

[17] Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021. 7

[18] Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020. 2

[19] 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. 3, 4, 7, 8

[20] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 1

[21] Shitong Shao, Zeyuan Yin, Muxin Zhou, Xindong Zhang, and Zhiqiang Shen. Generalized large-scale data condensation via various backbone and statistical matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16709–16718, 2024. 1

[22] Shitong Shao, Zikai Zhou, Huanran Chen, and Zhiqiang Shen. Elucidating the design space of dataset condensation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 3

[23] Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. Dˆ 4: Dataset distillation via disentangled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5809–5818, 2024. 1, 2

[24] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9390–9399, 2024. 1, 3, 8

[25] 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. 3

[26] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and

Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022. 1

[27] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. Dataset distillation, 2020. 1

[28] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36:73582–73603, 2023. 1, 6, 7

[29] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023. 1, 2, 5, 6, 7

[30] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 2, 5, 7

[31] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 1