Closed-Form Linear-Probe Dataset Distillation for Pre-trained Vision Models

Bincheng Peng [†]
Hokkaido University
peng@lmd.ist.hokudai.ac.jp

Guang Li [†,*]
Hokkaido University
guang@lmd.ist.hokudai.ac.jp

Ping Liu
University of Nevada, Reno
pingl@unr.edu

Takahiro Ogawa
Hokkaido University
ogawa@lmd.ist.hokudai.ac.jp

Miki Haseyama
Hokkaido University
mhaseyama@lmd.ist.hokudai.ac.jp

Abstract

Dataset distillation compresses a large training set into a small synthetic set that preserves downstream training utility. While most existing methods target training networks from scratch, modern visual transfer learning often uses frozen pre-trained encoders followed by lightweight linear probing. Existing distillation methods for this setting either rely on linear-probe gradient matching or use kernel and feature-regression formulations originally designed for from-scratch training. Frozen-feature linear probing admits a closed-form solution determined directly by the pre-trained features themselves, with no infinite-width approximation and no inner-loop trajectory. We propose Closed-Form Linear-Probe Dataset Distillation (CLP-DD), a bilevel formulation that computes the linear probe induced by the synthetic set with a sample-space kernel ridge solver. The synthetic images are then updated by evaluating this induced classifier on real features through a temperature-scaled softmax cross-entropy, where the classifier columns act as learned class anchors in feature space. We further show that the choice of outer objective is decisive: replacing the class-anchor objective with a standard MSE outer loss reduces average accuracy by 2.9 percentage points on ImageNet-100. On ImageNet-100 with four pre-trained backbones, CLP-DD substantially improves over LGM without DSA and approaches LGM with DSA at a fraction of the computational cost. On ImageNet-1K, CLP-DD achieves accuracy comparable to LGM with DSA; profiling with DINOv2 at IPC=1 shows roughly $14\times$ faster optimization and less than one-eighth of the peak GPU memory. The code is available at https://github.com/Guang000/CLP-DD.

1 Introduction

Large-scale pre-training has reshaped visual transfer learning. Modern pipelines commonly use powerful pre-trained vision models, such as CLIP [26], DINOv2 [25], MoCo-v3 [6], and EVA-02 [9], as frozen feature extractors followed by a lightweight linear classifier. In this regime, dataset distillation (DD) [36, 15, 38] seeks to compress a large training set into a small synthetic set that

[†] Equal contribution.
[*] Correspondence to: Guang Li <guang@lmd.ist.hokudai.ac.jp>.

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

preserves downstream linear-probe performance. As pre-trained backbones continue to grow, full-network distillation becomes increasingly expensive, making frozen-backbone linear probing a natural setting in which DD can scale. This setting inherits the strong representations of large pre-trained encoders while making the inner adaptation problem simple enough to admit structural acceleration.

Existing DD methods relevant to this setting fall into two broad families, neither of which fully exploits the structure available once the encoder is fixed. Gradient- and trajectory-matching methods match classifier gradients or training trajectories; the recent Linear Gradient Matching (LGM) [3] matches linear-probe gradients in the frozen-backbone setting, and obtains its strongest results when combined with Differentiable Siamese Augmentation (DSA) [39], at substantial computational cost. Kernel-based closed-form methods such as KIP [23] and FRePo [43] replace the inner trajectory with a kernel ridge regression solution. Both were developed primarily for from-scratch training: KIP uses NTK-based regression, whereas FRePo uses neural feature regression with a pool of evolving models. Once the encoder is fixed, however, the inner problem reduces to a ridge-regularized linear solve on pre-trained features: the relevant kernel is simply the empirical Gram matrix of frozen features, and no NTK approximation is required. This structural simplification suggests a more direct route to frozen-backbone DD than costly gradient matching or DSA-heavy optimization.

We propose Closed-Form Linear-Probe Dataset Distillation (CLP-DD), a bilevel framework that turns this structural simplification into a concrete distillation algorithm. The inner adaptation problem in our setting is structurally simpler than in from-scratch DD: it is a strongly convex ridge regression on fixed features rather than a non-convex deep-network optimization, and it has a single closed-form minimizer that does not depend on initialization, learning rate, or the number of inner steps. In the low-IPC regime where the number of synthetic samples N is much smaller than the feature dimension d, this minimizer can equivalently be written as a sample-space kernel ridge regression, replacing a $d \times d$ inverse with an $N \times N$ inverse while preserving differentiability through the solve end-to-end.

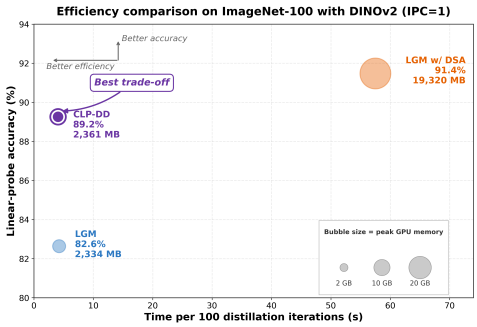


Figure 1: Accuracy versus efficiency on ImageNet-100 with DINOv2 at IPC=1. CLP-DD attains accuracy close to LGM with DSA at the runtime and memory of LGM without DSA. Bubble area denotes peak GPU memory.

Building on this closed-form inner step, we further design the outer objective to align synthetic-image optimization with linear-probe transfer. We find that pairing this solver with the standard ridge MSE loss, which corresponds to a frozen-backbone analogue of KIP, underperforms the class-anchor outer objective. The bottleneck is that MSE penalizes squared deviation from one-hot targets uniformly across class dimensions, whereas the linear-probe transfer task only requires correct class discrimination on real features. CLP-DD therefore evaluates the induced classifier on a small class-balanced batch of real features through a temperature-scaled softmax cross-entropy, where the columns of the closed-form classifier act as learned class anchors in feature space. This aligns the synthetic-image gradient with the actual linear-probe evaluation objective rather than an MSE surrogate, while requiring neither trajectory unrolling nor heavy differentiable augmentation like DSA.

To validate this design, we evaluate CLP-DD on ImageNet-100 and ImageNet-1K with four pre-trained backbones at IPC=1, demonstrating strong performance without DSA and accuracy comparable to LGM with DSA on ImageNet-1K. As shown in Figure 1, CLP-DD attains this accuracy with the runtime and memory of DSA-free LGM, about $14\times$ faster and using less than one-eighth of the peak GPU memory of DSA-based LGM, providing a favorable accuracy–efficiency balance on the frontier of low-IPC frozen-backbone dataset distillation.

Our contributions are summarized as follows:

- We identify a structural opportunity in frozen-backbone dataset distillation: frozen-feature linear probing admits an exact closed-form solution determined directly by the pre-trained features. In the low-IPC regime, this solution can be computed in sample-space kernel form, avoiding both inner-loop unrolling and NTK-based kernel approximation.

- We identify a discriminative-outer-objective gap for closed-form distillation with frozen backbones. Pairing the closed-form inner solver with a temperature-scaled softmax cross-entropy objective treats its classifier columns as class anchors and improves average accuracy over an MSE outer loss, a frozen-backbone analogue of KIP.
- We propose CLP-DD, which combines the sample-space closed-form solver with the discriminative outer objective. On ImageNet-100 and ImageNet-1K with four pre-trained backbones, CLP-DD improves average accuracy over real-image baselines and DSA-free LGM, and achieves accuracy comparable to LGM with DSA on ImageNet-1K. DINOv2 profiling at IPC=1 shows roughly $14\times$ faster optimization and less than one-eighth of the peak GPU memory of DSA-based LGM.

2 Related Work

Dataset Distillation. Dataset distillation (DD) aims to compress a large training set into a small synthetic set that preserves downstream training utility [36]. Existing methods typically optimize synthetic data by matching gradients [41, 22, 2], training trajectories [4, 7, 10, 14, 42], feature distributions [35, 40, 13, 31, 28, 18, 19], generative processes [16, 17, 32, 37, 5], synthetic-data parameterizations [12, 8, 11, 34], or kernel-based objectives [24, 23, 43, 20]. These approaches have substantially advanced DD, but most of them are designed for training networks from scratch. As a result, they often require repeated inner-loop optimization, long-horizon meta-gradient computation, or strong differentiable augmentation. In contrast, we study DD in the frozen-backbone linear-probing regime, where the encoder is fixed, and the inner adaptation reduces to a much simpler convex problem with an exact closed-form solution.

Dataset Distillation for Pre-trained Vision Models. The closest work to ours is Linear Gradient Matching (LGM) [3], which formulates DD for pre-trained self-supervised vision models. LGM optimizes synthetic images by matching gradients induced by a linear probe on frozen features, providing a strong benchmark for low-IPC distillation in this setting. However, it requires differentiation through the gradient-matching objective and achieves its best performance when combined with Differentiable Siamese Augmentation (DSA) [39], which substantially increases the computational cost. CLP-DD addresses the same frozen-backbone setting from a complementary perspective. Instead of matching classifier gradients, it directly computes the linear classifier induced by the synthetic set through a closed-form linear-probe solver, and optimizes the synthetic images using a discriminative outer objective evaluated on real features.

Implicit and Closed-form Optimization. Meta-learning and DD methods often obtain meta-gradients by differentiating through an inner optimization process, which becomes costly when the inner trajectory is long [43]. Prior work has therefore explored implicit differentiation and equilibrium formulations to avoid explicit unrolling [27, 1]. CLP-DD follows the same high-level goal of removing expensive inner-loop unrolling, but differs in a key respect: it does not approximate the solution of a non-convex training process. Since the only trainable component in our setting is a ridge-regularized linear classifier over fixed features, the inner problem admits an exact ridge-regression solution. In the low-IPC regime, this solution can be equivalently expressed in sample-space kernel form, allowing linear-probe adaptation to be implemented as a differentiable algebraic solve rather than an unrolled training trajectory.

3 Method

We consider dataset distillation in the frozen-backbone linear-probing regime. Let $\phi(\cdot)$ denote a pre-trained feature extractor whose parameters are fixed throughout distillation. Given a synthetic image set $\mathcal{S}_{\mathrm{syn}}$ with N samples, we denote its feature matrix by

$$X_{\mathrm{syn}} = \phi(\mathcal{S}_{\mathrm{syn}}) \in \mathbb{R}^{N\times d}, \tag{1}$$

where d is the feature dimension. The corresponding one-hot label matrix is $Y_{\mathrm{syn}} \in \mathbb{R}^{N\times C}$, where C is the number of classes. We use $\mathcal{S}_{\mathrm{real}}$ to denote real training samples used to define the outer distillation objective, and write $X_{\mathrm{real}} = \phi(\mathcal{S}_{\mathrm{real}})$. The downstream model is a linear probe $W \in \mathbb{R}^{d\times C}$ trained on top of the frozen features.

Our goal is to optimize $\mathcal{S}_{\text{syn}}$ such that a linear classifier trained on its frozen features performs well on real data. Instead of differentiating through a sequence of linear-probe updates, we exploit the fact that the ridge-regularized linear-probe problem admits a closed-form solution. We refer to the resulting framework as Closed-Form Linear-Probe Dataset Distillation (CLP-DD). Detailed derivations corresponding to the following components are provided in Appendix A.

3.1 Overview

CLP-DD separates closed-form inner adaptation from discriminative outer optimization. The inner problem models linear-probe adaptation on synthetic features with a ridge-regularized quadratic objective, which admits a closed-form solution. The outer problem then updates the synthetic images by evaluating the resulting classifier on real training samples. This separation keeps the inner adaptation analytically tractable while using real data to provide a discriminative optimization signal.

Concretely, each CLP-DD iteration consists of three steps. First, we extract frozen features for the current synthetic images. Second, we compute the linear-probe classifier induced by the synthetic set using a sample-space closed-form solver. Third, we evaluate an outer class-anchor loss on real features and backpropagate through the solver to update the synthetic images. Algorithm 1 summarizes the overall procedure.

3.2 Closed-Form Linear-Probe Solver

The inner optimization is defined on the synthetic features. We consider the ridge-regularized quadratic objective

$$\mathcal{L}_{\text{in}}(W) = \frac{1}{2}\|X_{\text{syn}}W - Y_{\text{syn}}\|_F^2 + \frac{\lambda}{2}\|W\|_F^2, \tag{2}$$

where $\lambda > 0$ is the ridge coefficient. Since the encoder is fixed, the only trainable component in the inner problem is a linear classifier over frozen features. The objective is therefore strongly convex and admits the standard ridge-regression solution

$$W^* = (X_{\text{syn}}^\top X_{\text{syn}} + \lambda I_d)^{-1} X_{\text{syn}}^\top Y_{\text{syn}}. \tag{3}$$

Thus, linear-probe adaptation on the synthetic set can be computed directly, without differentiating through an unrolled sequence of classifier updates.

For completeness, Eq. (3) can also be viewed as the steady state of gradient descent on Eq. (2). Defining

$$H = X_{\text{syn}}^\top X_{\text{syn}} + \lambda I_d, \qquad c = X_{\text{syn}}^\top Y_{\text{syn}}, \tag{4}$$

gradient descent with step size $\eta > 0$ follows

$$W_{t+1} = W_t - \eta(HW_t - c) = (I - \eta H)W_t + \eta c. \tag{5}$$

Under the standard spectral-radius condition $\rho(I - \eta H) < 1$, this recursion converges to the unique fixed point satisfying $HW^* = c$, which recovers Eq. (3). We provide the derivation in Appendix A.2.

3.3 Sample-Space Kernel Reformulation

Directly computing Eq. (3) requires solving a $d \times d$ linear system. In the low-IPC regime, the number of synthetic samples is much smaller than the feature dimension, i.e., $N \ll d$. We therefore use the equivalent sample-space kernel form

$$W^* = X_{\text{syn}}^\top (X_{\text{syn}} X_{\text{syn}}^\top + \lambda I_N)^{-1} Y_{\text{syn}}. \tag{6}$$

The equivalence follows from the standard push-through identity:

$$(X_{\text{syn}}^\top X_{\text{syn}} + \lambda I_d)^{-1} X_{\text{syn}}^\top = X_{\text{syn}}^\top (X_{\text{syn}} X_{\text{syn}}^\top + \lambda I_N)^{-1}, \tag{7}$$

where $\lambda > 0$ ensures that the regularized systems are positive definite and invertible. Let

$$K_{\text{syn}} = X_{\text{syn}} X_{\text{syn}}^\top \in \mathbb{R}^{N \times N}. \tag{8}$$

Then the solver only needs to form the synthetic sample kernel and solve an $N \times N$ linear system, where $N = C \times \text{IPC}$. This shifts the computational bottleneck from the feature dimension to the number of synthetic samples, making the formulation well-suited to extreme low-IPC distillation. Importantly, this closed-form step removes the dependence on an unrolled classifier-training horizon, although gradients still need to be backpropagated through the frozen encoder to update the synthetic pixels. When N is not smaller than d, the primal form in Eq. (3) can be used instead without changing the induced classifier. We provide the derivation of the equivalence in Appendix A.3.

Algorithm 1 Closed-Form Linear-Probe Dataset Distillation (CLP-DD)

Require: Frozen feature extractor ϕ; synthetic images $\mathcal{S}_{\text{syn}}$ and labels Y_{syn}; real training samples $\mathcal{S}_{\text{real}}$; ridge coefficient λ; temperature τ
Ensure: Optimized synthetic set $\mathcal{S}_{\text{syn}}$
for $t = 1, \ldots, 4{,}000$ **do**
  Extract synthetic features: $X_{\text{syn}} \leftarrow \phi(\mathcal{S}_{\text{syn}})$
  Sample a class-balanced real batch from $\mathcal{S}_{\text{real}}$ and extract features: $X_{\text{real}} \leftarrow \phi(\mathcal{S}_{\text{real}})$
  Form the synthetic kernel matrix: $K_{\text{syn}} \leftarrow X_{\text{syn}} X_{\text{syn}}^\top$
  Solve the closed-form linear probe: $W^* \leftarrow X_{\text{syn}}^\top (K_{\text{syn}} + \lambda I_N)^{-1} Y_{\text{syn}}$
  Compute the outer loss $\mathcal{L}_{\text{meta}}$ on X_{real} using Eq. (9)
  Backpropagate through the closed-form solve to obtain $\nabla_{\mathcal{S}_{\text{syn}}} \mathcal{L}_{\text{meta}}$
  Update the synthetic images $\mathcal{S}_{\text{syn}}$
end for
return $\mathcal{S}_{\text{syn}}$

3.4 Outer Class-Anchor Objective

The closed-form solver specifies how a linear probe is induced by the current synthetic features. To optimize the synthetic images, we further evaluate this induced classifier on real training features and use the resulting discriminative signal as the outer objective.

Given W^*, the outer loss is

$$\mathcal{L}_{\text{meta}} = -\frac{1}{M} \sum_{i=1}^{M} \log \frac{\exp(x_{\text{real},i}^\top w_{y_i}^* / \tau)}{\sum_{j=1}^{C} \exp(x_{\text{real},i}^\top w_j^* / \tau)}, \tag{9}$$

where M is the number of real samples in the outer batch, i.e., $M = CB$ for a class-balanced batch with B real samples per class, τ is a temperature parameter, and w_j^* denotes the j-th column of W^*.

We refer to Eq. (9) as the class-anchor cross-entropy objective. It treats the columns of the closed-form classifier as class anchors in the frozen feature space and encourages real samples to score highly against the anchor of their ground-truth class while being separated from competing class anchors. Compared with a squared loss on one-hot targets, this objective directly optimizes the softmax decision rule used by the final linear probe, and therefore provides a discriminative outer signal more aligned with transfer accuracy. The gradient of this objective with respect to the closed-form classifier is derived in Appendix A.5.

3.5 Differentiating Through the Solver

CLP-DD updates the synthetic images by differentiating the outer loss through the closed-form solver. From Eq. (6), let

$$A = K_{\text{syn}} + \lambda I_N. \tag{10}$$

Then

$$W^* = X_{\text{syn}}^\top A^{-1} Y_{\text{syn}}. \tag{11}$$

The gradient of $\mathcal{L}_{\text{meta}}$ with respect to X_{syn} is obtained by differentiating through both $X_{\text{syn}}^\top$ and A^{-1}. The derivative of the inverse follows

$$\mathrm{d}(A^{-1}) = -A^{-1} \mathrm{d}A A^{-1}. \tag{12}$$

Since A depends on X_{syn} through $K_{\text{syn}} = X_{\text{syn}} X_{\text{syn}}^\top$, gradients from the outer loss propagate to the synthetic features and then to the synthetic images through the frozen feature extractor. In practice, this differentiation is implemented with automatic differentiation through the linear solve, avoiding the need to store or backpropagate through an unrolled inner trajectory. Detailed derivations for differentiating through the closed-form solver and propagating the gradient to synthetic images are provided in Appendices A.4 and A.6.

Table 1: Quantitative performance comparison on ImageNet-100 and ImageNet-1K at IPC=1. We report the mean and standard deviation over three repeated runs. Bold indicates the best DSA-free low-IPC result on ImageNet-100 and the best low-IPC result on ImageNet-1K; full-data results are references.

Dataset	Method	Backbone Architectures				Average
		DINOv2	MoCo-v3	CLIP	EVA-02	
ImageNet-100	Neighbor	86.0±0.2	77.1±0.2	67.8±0.2	78.8±0.1	77.4±0.2
	Centroid	86.9±0.3	77.7±0.1	77.1±0.1	80.9±0.2	80.6±0.2
	Random	74.8±2.6	61.3±2.6	57.6±1.5	64.4±2.5	64.5±2.5
	k-center	**90.2**±0.2	78.6±0.6	75.8±0.2	81.1±0.1	81.4±0.3
	LFP	88.4±0.2	81.7±0.1	78.3±0.3	85.3±0.8	83.4±0.4
	LGM	82.6±0.2	59.5±0.5	58.4±0.3	73.9±0.2	68.6±0.4
	CLP-DD (Ours)	89.2±0.1	**83.9**±0.1	**79.2**±0.1	**87.6**±0.0	**85.0**±0.1
	LGM w/ DSA	91.4±0.1	83.3±0.1	84.6±0.1	88.6±0.1	87.0±0.1
	Full dataset	95.1±0.1	89.3±0.2	92.4±0.0	94.0±0.1	92.7±0.1
ImageNet-1K	Neighbor	67.7±0.1	56.3±0.1	38.7±0.1	50.0±0.1	53.2±0.1
	Centroid	69.5±0.0	57.4±0.1	53.8±0.1	58.0±0.0	59.6±0.1
	Random	50.3±0.5	38.7±0.5	31.7±0.4	37.7±0.6	39.6±0.5
	LGM w/ DSA	75.0±0.1	63.2±0.0	**62.9**±0.0	70.3±0.1	67.8±0.1
	CLP-DD (Ours)	**75.6**±0.1	**63.4**±0.1	62.4±0.1	**71.2**±0.1	**68.2**±0.1
	Full dataset	83.0±0.1	76.5±0.0	78.7±0.0	81.7±0.0	80.0±0.1

4 Experiments

We evaluate **CLP-DD** on ImageNet-100 [33] and ImageNet-1K [29]. We use four representative pre-trained vision backbones: CLIP [26], DINOv2 [25], EVA-02 [9], and MoCo-v3 [6]. Unless otherwise specified, all experiments are conducted at a resolution of 224×224 using the ViT-B variant of each backbone. We distill each dataset for a fixed $4,000$ iterations, without validation- or test-based checkpoint selection, and evaluate the final distilled images by training a randomly initialized linear classifier to convergence on top of the frozen features. Accuracy results are reported as mean $\pm$ standard deviation over three independent runs unless indicated otherwise.

At each distillation iteration, we sample a class-balanced real batch with $B = 4$ samples per class to define the outer objective. We analyze the effect of B in the ablation study. Unless otherwise specified, we set the ridge coefficient to $\lambda = 0.1$ and the temperature to $\tau = 0.07$, and analyze their sensitivity in the ablation study.

By default, CLP-DD uses standard image augmentations, including random cropping and noise injection, and does not use Differentiable Siamese Augmentation (DSA) [39]. For controlled evaluation, we extract features from the final layer of each backbone and use the same linear-probe training protocol across all methods and baselines. Additional implementation details are provided in Appendix B.

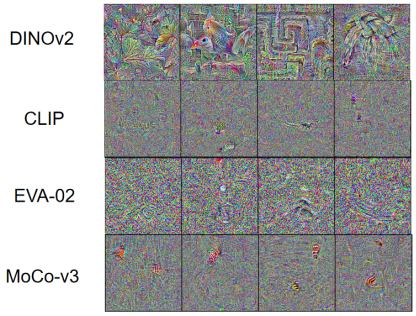


Figure 2: Examples of IPC=1 images distilled by CLP-DD on ImageNet-100 using the CLIP, DINOv2, MoCo-v3 and EVA-02 backbone.

Figure 2 shows examples of images distilled by CLP-DD with the CLIP, DINOv2, MoCo-v3, and EVA-02 backbones. Rows correspond to different backbones, and columns correspond to the first four ImageNet-100 categories. As is common in pixel-space dataset distillation, the distilled images are highly textured and do not resemble natural images. This is expected because CLP-DD optimizes synthetic images for frozen-feature linear-probe performance rather than visual realism. We therefore use downstream linear-probe accuracy as the primary evaluation metric. Additional distilled images are provided in Appendix F.

Table 2: Quantitative performance comparison on Spawrious and WaterBirds at IPC=1. We report the mean and standard deviation over three repeated runs. Bold indicates the best DSA-free low-IPC result; gray rows are DSA-based references.

Dataset	Method	Backbone Architectures				Average
		DINOv2	MoCo-v3	CLIP	EVA-02	
Spawrious	Neighbor	76.9±1.2	33.0±3.4	41.7±4.6	**40.7**±2.7	48.1±3.0
	Centroid	**80.2**±2.7	30.5±1.6	44.8±5.0	38.1±3.3	**48.4**±3.2
	Random	68.1±8.1	31.8±2.5	**46.2**±4.4	33.3±5.3	44.9±5.1
	LGM	57.2±6.8	22.6±4.3	26.1±6.7	28.7±2.1	33.6±5.0
	CLP-DD (Ours)	78.3±2.2	**33.2**±2.7	42.7±4.8	37.4±3.5	47.9±3.3
	LGM w/ DSA	80.8±3.1	32.7±2.3	43.1±6.4	36.5±3.1	48.3±3.7
	Full dataset	78.1±0.1	35.8±0.0	50.4±0.3	50.1±0.4	53.6±0.2
WaterBirds	Neighbor	67.3±5.8	45.2±11.4	74.6±6.5	74.3±3.3	65.3±6.8
	Centroid	65.7±3.4	62.0±5.2	69.8±8.6	64.6±7.9	65.5±6.3
	Random	57.1±8.6	67.4±5.7	71.9±4.5	59.8±13.3	64.0±8.0
	LGM	58.3±3.2	63.4±1.6	64.1±2.2	64.8±1.8	62.6±2.2
	CLP-DD (Ours)	**80.0**±2.3	**78.5**±0.4	**75.4**±0.3	**79.6**±0.4	**78.3**±0.8
	LGM w/ DSA	82.1±2.9	77.8±0.0	77.9±0.2	78.0±0.8	79.0±1.0
	Full dataset	95.5±0.1	74.3±0.1	86.0±0.1	90.4±0.2	86.5±0.1

4.1 Standard ImageNet Evaluation

We evaluate each distilled set by training a linear classifier on the synthetic images and reporting accuracy on the corresponding ImageNet validation split, using the same protocol for all real-image baselines. We compare with three IPC=1 real-image baselines: Random selects one image per class uniformly at random; Centroid selects the image closest to the class-wise mean feature; and Neighbor selects the real image closest to the corresponding CLP-DD synthetic image in feature space. On ImageNet-100, we additionally compare with class-wise k-center selection and Learned Feature Prototypes (LFP). For LGM, we report both the DSA-free and DSA-based variants when available.

Table 1 compares CLP-DD with real-image baselines and LGM on ImageNet-100 and ImageNet-1K at IPC=1. CLP-DD consistently outperforms Random, Centroid, and Neighbor across all four backbones on both datasets, improving the four-backbone average over k-center by 3.6 points and over LFP by 1.6 points on ImageNet-100, and over the strongest real-image baseline by 8.6 percentage points on ImageNet-1K. k-center is stronger on DINOv2 (90.2% versus 89.2%), so the ImageNet-100 advantage is not uniform across backbones. Compared with LGM, CLP-DD substantially improves over the DSA-free variant on ImageNet-100, from 68.6% to 85.0%. Compared with LGM w/ DSA, CLP-DD trails by 2.0 points on ImageNet-100 (85.0% vs. 87.0%) and is comparable on ImageNet-1K (68.2% versus 67.8%); the small per-backbone differences are interpreted descriptively. These results position CLP-DD as an efficient DSA-free alternative to linear gradient matching.

4.2 Auxiliary Evaluation under Background Shift

We further evaluate CLP-DD on Spawrious and WaterBirds, following the protocol of LGM. Unlike standard ImageNet evaluation, these benchmarks test performance under background shift: class labels are spuriously correlated with background attributes during training, while evaluation is conducted under shifted or adversarial background configurations.

Table 2 summarizes the results. On Spawrious [21], CLP-DD achieves 47.9% average accuracy, competitive with Centroid (48.4%), Neighbor (48.1%), and LGM w/ DSA (48.3%), while substantially outperforming DSA-free LGM (33.6%). All low-IPC methods, including the strongest real-image baselines, remain below the full-dataset upper bound of 53.6%, indicating that severe background shift remains challenging in this regime. On WaterBirds [30], CLP-DD obtains 78.3% average accuracy, close to LGM w/ DSA at 79.0%, and even achieves better results on MoCo-v3 and EVA-02 compared to LGM w/ DSA.

4.3 IPC Scaling

Table 3 reports IPC scaling on ImageNet-100 with DINOv2. CLP-DD outperforms both Centroid and LGM at each tested IPC. The gains over LGM are 6.6, 5.0, and 4.0 percentage points at IPC= 1, 3, and 5, respectively, without DSA. Appendix C.3 reports additional results.

Table 3: IPC scaling on ImageNet-100 with DINOv2.

Method	IPC		
	1	3	5
Centroid	86.9±0.3	89.3±0.2	90.2±0.2
LGM	82.6±0.2	86.3±0.2	87.8±0.3
CLP-DD	**89.2**±0.1	**91.3**±0.1	**91.8**±0.0

4.4 Efficiency Analysis

Table 4 compares the empirical efficiency of CLP-DD against LGM baselines on a single NVIDIA RTX 5090 GPU. All methods are evaluated with the same DINOv2 backbone and image resolution. CLP-DD has a similar time and memory footprint to LGM without DSA, while achieving substantially higher accuracy. In contrast, LGM with DSA obtains the highest accuracy in this setting, but requires substantially more memory and longer runtime due to the additional augmentation pipeline. Time measures 100 complete distillation iterations, including feature extraction, augmentation, loss computation, backward propagation, image updates, and the differentiable solve where applicable. One-time setup and final evaluation are excluded for all methods. CLP-DD approaches the accuracy of LGM with DSA at a cost close to DSA-free LGM.

Table 4: ImageNet-100/DINOv2 efficiency at IPC=1. Time is seconds per 100 full iterations; memory is peak MB.

Method	Time	Memory	Acc.
LGM w/ DSA	57.49s	19,320MB	91.4±0.1
LGM	4.25s	2,334MB	82.6±0.2
CLP-DD	4.07s	2,361MB	89.2±0.1

Figure 3 compares the convergence behavior of CLP-DD with LGM variants under the same distillation budget. CLP-DD reaches high accuracy in fewer iterations than DSA-free LGM and maintains a clear advantage throughout most of the optimization process. LGM w/ DSA also achieves strong final accuracy, but as shown in Table 4, this comes with substantially higher runtime and memory cost. These results suggest that the closed-form linear-probe solver provides a more direct optimization signal than linear gradient matching in the DSA-free setting. The mild late-stage accuracy decline occurs during outer image optimization, since the inner ridge solution is exact. Its mechanism remains unresolved; training-only stopping criteria or additional regularization are future work.

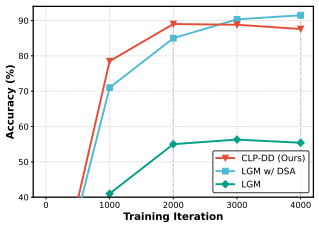


Figure 3: Convergence comparison on ImageNet-100 with DINOv2 at IPC=1.

4.5 Ablation Study

Table 5: Ablation of the outer objective on ImageNet-100 at IPC=1. "w/o CA" replaces the class-anchor objective with an MSE objective.

Method	DINOv2	MoCo-v3	CLIP	EVA-02	Average
LGM	82.6±0.2	59.5±0.5	58.4±0.3	73.9±0.2	68.6±0.4
CLP-DD w/o CA	88.3±0.0	82.1±0.1	74.1±0.1	84.1±0.1	82.1±0.1
CLP-DD (Ours)	**89.2**±0.1	**83.9**±0.1	**79.2**±0.1	**87.6**±0.0	**85.0**±0.1

Effectiveness of the Class-Anchor Outer Objective. We ablate the outer objective on ImageNet-100 at IPC=1 by replacing the temperature-scaled class-anchor objective with an ordinary MSE objective. As shown in Table 5, the class-anchor objective improves the average accuracy from 82.1% to 85.0%, with consistent gains across all four backbones. The largest improvement is observed on CLIP, where the accuracy increases from 74.1% to 79.2%.

These results show that the outer objective contributes beyond the closed-form inner solver. While the sample-space solver computes the linear probe induced by the current synthetic set, the outer objective determines how the synthetic images are updated so that the induced classifier transfers to real samples. Compared with ordinary MSE, the class-anchor objective directly optimizes relative class discrimination in the frozen feature space, which appears more suitable for linear-probe dataset distillation.

Effect of Real Batch Size (B). Table 6 varies the number B of real samples per class on ImageNet-100 with DINOv2. Accuracy is 88.0%, 88.9%, 89.2%, and 89.3% at $B = 1$, 2, 4, and 8, respectively.

Table 6: Ablation on real samples per class (B).

B	**Acc (%)**
1	88.0±0.1
2	88.9±0.2
4	**89.2**±0.1
8	89.3±0.0

Each outer batch contains CB real samples across C classes, with the synthetic set size fixed throughout the sweep. Successive doublings of B improve accuracy by 0.9, 0.3, and 0.1 percentage points, respectively. We therefore use $B = 4$ by default, retaining most of the observed gain while using half as many real samples per iteration as $B = 8$.

Sensitivity Analysis on Hyperparameters. We analyze the sensitivity of CLP-DD to the temperature τ and the ridge coefficient λ on ImageNet-100 with the MoCo-v3 backbone at IPC$= 1$. We vary one hyperparameter at a time while keeping the other fixed to its default value.

Table 7: Sensitivity analysis of τ and λ on ImageNet-100 with MoCo-v3 at IPC=1.

τ	**Acc (%)**	λ	**Acc (%)**
0.01	69.7±0.3	0.05	79.3±0.2
0.05	78.1±0.2	0.07	81.3±0.2
0.07	**83.9**±0.1	**0.10**	**83.9**±0.1
0.10	83.2±0.0	0.50	76.2±0.1

As shown in Table 7, CLP-DD achieves the best accuracy among the tested temperatures at $\tau = 0.07$. A smaller temperature, such as $\tau = 0.01$, makes the softmax distribution overly sharp and leads to a noticeable accuracy drop, while $\tau = 0.10$ remains competitive but slightly worse. For the ridge coefficient, $\lambda = 0.10$ yields the best result. Smaller values provide weaker regularization and may amplify noise in the induced classifier, whereas larger values over-regularize W^* and reduce its discriminative capacity. Overall, CLP-DD remains stable across a moderate range of τ and λ, with performance degradation mainly observed under overly sharp temperatures or excessive ridge regularization.

The sweeps in Appendix C.6 show that $\tau = 0.07$ and $\lambda = 0.10$ also give the highest mean accuracy among the tested values on DINOv2, CLIP, and EVA-02. Increasing λ from 0.10 to 0.20 lowers accuracy by 1.6–2.6 percentage points across the four backbones.

Effect of Standard Augmentation. Cropping and noise improve accuracy from 87.9% to 89.2%, a gain of 1.3 percentage points (Table 8). Without augmentation, CLP-DD still exceeds DSA-free LGM (82.6%) by 5.3 points, indicating that augmentation alone cannot explain the improvement. Standard augmentation therefore complements the proposed formulation, while a substantial advantage remains when it is removed. Nevertheless, this ablation does not fully disentangle the contributions of the closed-form inner learner, class-anchor outer objective, and augmentation to the gap between CLP-DD and LGM. Further augmentation and stability results are provided in Appendix C.

Table 8: ImageNet-100/DINOv2 augmentation ablation at IPC=1. Mean ± std., three runs.

CLP-DD setting	Acc. (%)
No augmentation	87.9 ± 0.8
Cropping w/ noise	89.2 ± 0.1

Across three runs, cropping and noise also reduce the observed standard deviation from 0.8 to 0.1 percentage points. Adding DSA ($m = 1$) to the default augmentation yields 89.3% accuracy, with no reported standard deviation (Table 15). Relative to the default, this 0.1-point difference comes with an increase from 4.07 s to 9.39 s per 100 distillation iterations and from 2,361 MB to 4,011 MB in peak GPU memory.

4.6 Cross-Backbone Transfer and ImageNet-1K Efficiency

Cross-backbone transfer. Under a matched protocol, we test DINOv2-distilled ImageNet-100 sets (IPC$= 1$) on CLIP, MoCo-v3, and EVA-02. In Table 9, CLP-DD gains 7.1 and 4.2 points over frozen-feature KIP and FRePo. MSE lowers the unseen mean by 13.4 points. Accuracy drops from 89.2% on DINOv2 to 50.9% on unseen backbones, reflecting representation dependence.

Table 9: ImageNet-100 transfer from DINOv2 at IPC=1.

Method	Source	Unseen
Frozen-feature KIP	84.5 ± 0.3	43.8
Frozen-feature FRePo	86.3 ± 0.2	46.7
CLP-DD w/ MSE	88.3 ± 0.0	37.5
CLP-DD	$\mathbf{89.2 \pm 0.1}$	**50.9**

ImageNet-1K efficiency. Figure 4 shows that CLP-DD attains $75.6 \pm 0.1\%$ accuracy at 46.33 s per 100 iterations and 11,924 MB peak memory. All methods use matched hardware, backbone, and batch settings. Timings cover complete distillation iterations and exclude setup and final evaluation. Relative to LGM with DSA ($m = 3$), it provides a $13.9\times$ speedup and 87.6% lower peak memory with comparable accuracy. Against DSA-free LGM, it gains 12.4 accuracy points for 11.6% more time and 17.7% more memory. These results highlight the practical value of CLP-DD for large-scale frozen-backbone adaptation when both high linear-probe accuracy and limited computational resources are important.

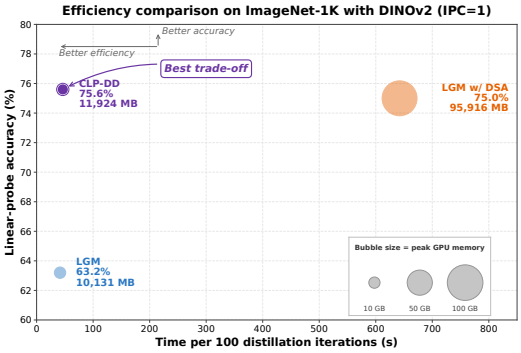


Figure 4: ImageNet-1K/DINOv2, IPC=1, on a 96 GB RTX PRO 6000. Bubble area: peak memory. Accuracy: three-run mean. Time excludes setup and evaluation.

5 Discussion

CLP-DD targets low-IPC adaptation to a specified frozen encoder. Centroid avoids iterative distillation and becomes competitive at higher IPC, trailing CLP-DD by 0.9 points on DINOv2 at IPC=10 (Appendix C.3). CLP-DD saves computation relative to LGM with DSA, while its advantage over DSA-free LGM is primarily accuracy at comparable or moderately higher cost.

The transfer results support the class-anchor objective beyond the source encoder, but the substantial matched-to-unseen-backbone drop limits portability. Same-backbone adaptation remains the setting with the strongest evidence; the full matrix and target-wise ablation are provided in Appendices C.1 and C.2. On Spawrious, CLP-DD is competitive with real-image selection (47.9% versus 48.4% for Centroid), showing that stronger in-distribution linear probing does not ensure robustness to background shift. Distillation may retain biases of the frozen representation, motivating complementary encoders and robustness-oriented objectives.

Broader probe losses, nonlinear heads, and partial fine-tuning remain untested. Positive ridge regularization guarantees invertibility, while acceptable finite-precision behavior is supported only over the tested range (Appendix C.7). The mild late-stage decline remains unexplained. Final CLP-DD accuracies use a fixed 4,000-iteration budget without validation- or test-based checkpoint selection; training-only stopping and additional regularization remain future directions. With three runs, small accuracy differences warrant descriptive interpretation rather than significance claims.

6 Conclusion

In this work, we introduced Closed-Form Linear-Probe Dataset Distillation (CLP-DD), a distillation framework tailored to frozen-backbone linear probing with pre-trained vision models. CLP-DD exploits the closed-form structure of ridge-regularized linear-probe adaptation in sample-space kernel form, enabling efficient low-IPC distillation without unrolled inner-loop optimization. Combined with a temperature-scaled class-anchor cross-entropy objective, it updates synthetic images to induce discriminative classifiers in the frozen feature space. Experiments on ImageNet-100 and ImageNet-1K with multiple pre-trained backbones show that CLP-DD improves average accuracy over real-image baselines at IPC=1 and serves as an efficient DSA-free alternative to LGM, while remaining competitive on spurious-correlation benchmarks. Limitations and broader impacts are discussed in Appendices D and E.

Acknowledgments

This research was supported in part by JSPS KAKENHI Grant Numbers JP24K02942 and JP25K21218.

References

[1] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, pages 688–699, 2019.

[2] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020.

[3] George Cazenavette, Antonio Torralba, and Vincent Sitzmann. Dataset distillation for pre-trained self-supervised vision models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

[4] 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 (CVPR)*, pages 4750–4759, 2022.

[5] Jeffrey A. Chan-Santiago and Mubarak Shah. Learnability-guided diffusion for dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 41657–41666, 2026.

[6] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9640–9649, 2021.

[7] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590, 2023.

[8] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[9] Yuxin Fang, Quan Sun, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. EVA-02: A visual representation for neon genesis. *arXiv preprint arXiv:2303.11331*, 2023.

[10] Ziyao Guo, Kai Wang, George Cazenavette, Hui Li, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *International Conference on Learning Representations*, 2024.

[11] Chenyang Jiang, Zhengcen Li, Hang Zhao, Qiben Shan, Shaocong Wu, and Jingyong Su. Beyond pixels: Efficient dataset distillation via sparse gaussian representation. *arXiv preprint arXiv:2509.26219*, 2025.

[12] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 11102–11118, 2022.

[13] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364, 2022.

[14] Yongmin Lee and Hye Won Chung. SelMatch: Effectively scaling up dataset distillation via selection-based initialization and partial updates by trajectory matching. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 26546–26567, 2024.

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

[16] Longzhen Li, Guang Li, Ren Togo, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Generative dataset distillation: Balancing global structure and local details. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 7664–7671, 2024.

[17] Longzhen Li, Guang Li, Ren Togo, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Generative dataset distillation based on self-knowledge distillation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2025.

[18] Wenyuan Li, Guang Li, Keisuke Maeda, Takahiro Ogawa, and Miki Haseyama. Hyperbolic dataset distillation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[19] Haoyang Liu, Yijiang Li, Tiancheng Xing, Peiran Wang, Vibhu Dalal, Luwei Li, Jingrui He, and Haohan Wang. Dataset distillation via the wasserstein metric. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

[20] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[21] Aengus Lynch, Gbètondji J-S Dovonon, Jean Kaddour, and Ricardo Silva. Spawrious: A benchmark for fine control of spurious correlation biases. *arXiv preprint arXiv:2303.05470*, 2023.

[22] Dmitry Medvedev and Alexander D'yakonov. Learning to generate synthetic training data using gradient matching and implicit differentiation. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 138–150, 2021.

[23] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations (ICLR)*, 2021.

[24] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021.

[25] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

[26] 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 (ICML)*, pages 8748–8763. PMLR, 2021.

[27] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. Meta-learning with implicit gradients. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 32, pages 113–124, 2019.

[28] Fengli Ran, Xiao Pu, Bo Liu, Xiuli Bi, and Bin Xiao. TGDD: Trajectory guided dataset distillation with balanced distribution. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 15662–15670, 2026.

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[30] Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. In *International Conference on Learning Representations (ICLR)*, 2020.

[31] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023.

[32] Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. D^4M: Dataset distillation via disentangled diffusion model. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5809–5818, 2024.

[33] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European Conference on Computer Vision (ECCV)*, pages 776–794. Springer, 2020.

[34] Linh-Tam Tran and Sung-Ho Bae. Post training quantization for efficient dataset condensation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 25992–26000, 2026.

[35] 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 (CVPR)*, pages 12196–12205, 2022.

[36] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

[37] Linfeng Ye, Shayan Mohajer Hamidi, Guang Li, Takahiro Ogawa, Miki Haseyama, and Konstantinos N. Plataniotis. Information-guided diffusion sampling for dataset distillation. In *Advances in Neural Information Processing Systems Workshops*, 2025.

[38] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):150–170, 2024.

[39] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685, 2021.

[40] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023.

[41] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations (ICLR)*, 2021.

[42] Wenliang Zhong, Haoyu Tang, Qinghai Zheng, Mingzhu Xu, Yupeng Hu, and Weili Guan. Towards stable and storage-efficient dataset distillation: Matching convexified trajectory. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 25581–25589, 2025.

[43] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 9813–9827, 2022.

A Detailed Mathematical Derivations

This appendix provides the derivations used in Closed-Form Linear-Probe Dataset Distillation (CLP-DD). We first derive the closed-form ridge solution and its steady-state interpretation under gradient descent. We then present the sample-space kernel reformulation, the differential through the closed-form solver, the gradient of the outer class-anchor objective, and the final gradient flow to the synthetic images.

A.1 Notation and Ridge Inner Problem

For compactness in the appendix, we drop the 'syn' subscript

$$X := X_{\text{syn}} \in \mathbb{R}^{N\times d}, \qquad Y := Y_{\text{syn}} \in \mathbb{R}^{N\times C}, \tag{13}$$

where N is the number of synthetic samples, d is the feature dimension, and C is the number of classes. The linear probe is denoted by $W \in \mathbb{R}^{d\times C}$.

The ridge-regularized inner objective is

$$\mathcal{L}_{\text{in}}(W) = \frac{1}{2}\|XW - Y\|_F^2 + \frac{\lambda}{2}\|W\|_F^2, \qquad \lambda > 0. \tag{14}$$

Using standard matrix calculus, its gradient is

$$\nabla_W \mathcal{L}_{\text{in}}(W) = (X^\top X + \lambda I_d)W - X^\top Y. \tag{15}$$

Defining

$$H = X^\top X + \lambda I_d, \qquad c = X^\top Y, \tag{16}$$

we can write Eq. (15) as

$$\nabla_W \mathcal{L}_{\text{in}}(W) = HW - c. \tag{17}$$

Since $\lambda > 0$, H is symmetric positive definite and hence invertible. Setting the gradient to zero gives the standard ridge-regression solution

$$W^* = H^{-1}c = (X^\top X + \lambda I_d)^{-1}X^\top Y. \tag{18}$$

A.2 Steady-State View of Ridge-Regression Linear Probing

The closed-form solution in Eq. (18) can also be viewed as the steady state of gradient descent on the ridge-regularized linear-probe objective. Gradient descent with step size $\eta > 0$ follows

$$W_{t+1} = W_t - \eta(HW_t - c) = (I_d - \eta H)W_t + \eta c. \tag{19}$$

Let

$$B = I_d - \eta H. \tag{20}$$

Starting from $W_0 = \mathbf{0}$, unrolling the recursion gives

$$W_t = \eta \sum_{k=0}^{t-1} B^k c. \tag{21}$$

Using the finite matrix geometric series and $I_d - B = \eta H$, we obtain

$$W_t = \eta(I_d - B^t)(I_d - B)^{-1}c = (I_d - B^t)H^{-1}c. \tag{22}$$

If the step size satisfies the standard stability condition

$$0 < \eta < \frac{2}{\mu_{\max}(H)}, \tag{23}$$

then $\rho(B) < 1$ and $B^t \to 0$. Therefore,

$$\lim_{t\to\infty} W_t = H^{-1}c = (X^\top X + \lambda I_d)^{-1}X^\top Y. \tag{24}$$

Thus, the unrolled linear-probe dynamics converge to the same classifier as the closed-form ridge solution used by CLP-DD.

A.3 Sample-Space Kernel Reformulation

We next derive the sample-space form of the ridge solution. For $\lambda > 0$, both $X^\top X + \lambda I_d$ and $XX^\top + \lambda I_N$ are invertible. We have

$$(X^\top X + \lambda I_d)X^\top = X^\top (XX^\top + \lambda I_N). \tag{25}$$

Multiplying the two sides on the left by $(X^\top X + \lambda I_d)^{-1}$ and on the right by $(XX^\top + \lambda I_N)^{-1}$ gives the standard push-through identity

$$(X^\top X + \lambda I_d)^{-1}X^\top = X^\top (XX^\top + \lambda I_N)^{-1}. \tag{26}$$

Substituting Eq. (26) into Eq. (18) yields

$$W^* = X^\top (XX^\top + \lambda I_N)^{-1}Y. \tag{27}$$

Let

$$K = XX^\top, \qquad A = K + \lambda I_N. \tag{28}$$

Then the sample-space solver used in CLP-DD is

$$W^* = X^\top A^{-1}Y. \tag{29}$$

This replaces a $d \times d$ linear system with an $N \times N$ system, which is suitable for the low-IPC regime where the number of synthetic samples is small. When N is not smaller than d, the primal form in Eq. (18) can be used instead without changing the induced classifier.

A.4 Differentiating Through the Closed-Form Solver

We now derive the differential through the sample-space solver. This derivation clarifies how the outer objective depends on the synthetic features.

Let

$$A = XX^\top + \lambda I_N, \qquad S = A^{-1}, \qquad P = SY = A^{-1}Y. \tag{30}$$

The kernel-space solution is

$$W^* = X^\top P. \tag{31}$$

For fixed Y, its differential with respect to X is

$$\mathrm{d}W^* = \mathrm{d}X^\top P - X^\top S\mathrm{d}XX^\top P - X^\top SX\mathrm{d}X^\top P. \tag{32}$$

This follows by differentiating $W^* = X^\top P$:

$$\mathrm{d}W^* = \mathrm{d}X^\top P + X^\top \mathrm{d}P. \tag{33}$$

Since $P = A^{-1}Y$ and Y is fixed,

$$\mathrm{d}P = \mathrm{d}(A^{-1})Y = -A^{-1}(\mathrm{d}A)A^{-1}Y = -S(\mathrm{d}A)P, \tag{34}$$

where

$$\mathrm{d}(A^{-1}) = -A^{-1}(\mathrm{d}A)A^{-1}. \tag{35}$$

Moreover,

$$\mathrm{d}A = \mathrm{d}XX^\top + X\mathrm{d}X^\top. \tag{36}$$

Substituting these expressions into $\mathrm{d}W^*$ gives Eq. (32).

Let

$$G = \frac{\partial \mathcal{L}_{\mathrm{meta}}}{\partial W^*} \in \mathbb{R}^{d\times C} \tag{37}$$

be the upstream gradient from the outer objective. Then

$$\mathrm{d}\mathcal{L}_{\mathrm{meta}} = \mathrm{tr}(G^\top \mathrm{d}W^*). \tag{38}$$

Using Eq. (32) and collecting terms in the form $\mathrm{tr}((\nabla_X \mathcal{L}_{\mathrm{meta}})^\top \mathrm{d}X)$ gives

$$\nabla_X \mathcal{L}_{\mathrm{meta}} = PG^\top - SXGP^\top X - PG^\top X^\top SX. \tag{39}$$

In practice, CLP-DD computes this gradient using automatic differentiation through a linear solve, but Eq. (39) makes explicit how the synthetic features influence the outer objective through the closed-form classifier.

A.5 Outer Objective Gradient

The outer objective is a temperature-scaled cross-entropy loss over the closed-form classifier:

$$\mathcal{L}_{\text{meta}} = -\frac{1}{M}\sum_{i=1}^{M}\log\frac{\exp(x_{\text{real},i}^{\top}w_{y_i}^{*}/\tau)}{\sum_{j=1}^{C}\exp(x_{\text{real},i}^{\top}w_{j}^{*}/\tau)}. \tag{40}$$

Let

$$X_{\text{real}} \in \mathbb{R}^{M\times d}, \qquad T \in \mathbb{R}^{M\times C} \tag{41}$$

denote the real feature matrix and the corresponding one-hot label matrix. Define

$$Z = \frac{1}{\tau}X_{\text{real}}W^{*}, \qquad \Pi = \text{softmax}(Z), \tag{42}$$

where the softmax is applied row-wise over the C classes. The loss can be written as

$$\mathcal{L}_{\text{meta}} = -\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{C}T_{ij}\log\Pi_{ij}. \tag{43}$$

Using the standard softmax-cross-entropy derivative,

$$\frac{\partial\mathcal{L}_{\text{meta}}}{\partial Z} = \frac{1}{M}(\Pi - T). \tag{44}$$

Since $Z = \tau^{-1}X_{\text{real}}W^{*}$, the gradient with respect to the closed-form classifier is

$$G = \frac{\partial\mathcal{L}_{\text{meta}}}{\partial W^{*}} = \frac{1}{M\tau}X_{\text{real}}^{\top}(\Pi - T). \tag{45}$$

Substituting Eq. (45) into Eq. (39) gives the gradient of the outer objective with respect to the synthetic features.

A.6 Gradient Flow to Synthetic Images

The synthetic features are produced by the frozen backbone:

$$X_{\text{syn}} = \phi(\mathcal{S}_{\text{syn}}). \tag{46}$$

Although the parameters of ϕ are fixed, the synthetic images remain learnable. For the n-th synthetic image s_n, with feature $x_n = \phi(s_n)$, the gradient is

$$\nabla_{s_n}\mathcal{L}_{\text{meta}} = J_{\phi}(s_n)^{\top}\nabla_{x_n}\mathcal{L}_{\text{meta}}, \tag{47}$$

where

$$J_{\phi}(s_n) = \frac{\partial\phi(s_n)}{\partial s_n} \tag{48}$$

is the Jacobian of the frozen feature extractor with respect to its input. Stacking all the synthetic samples gives

$$\nabla_{\mathcal{S}_{\text{syn}}}\mathcal{L}_{\text{meta}} = \left(\frac{\partial X_{\text{syn}}}{\partial\mathcal{S}_{\text{syn}}}\right)^{\top}\nabla_{X_{\text{syn}}}\mathcal{L}_{\text{meta}}. \tag{49}$$

Thus, CLP-DD avoids differentiating through an unrolled inner trajectory and instead propagates gradients through the closed-form linear-probe solution and the frozen feature extractor.

B Implementation Details

This section provides additional implementation details on datasets, backbone models, distillation settings, evaluation protocols, and data augmentations.

B.1 Datasets

We use the same four datasets and splits as LGM [3].

ImageNet-1K. We use the ILSVRC 2012 classification release [29], with the official training split for distillation and the validation split for evaluation. The ILSVRC image terms limit use to non-commercial research and education.

ImageNet-100. We restrict both ILSVRC 2012 splits to the 100 WordNet synsets from CMC [33], using the class list in the LGM implementation. The subset is subject to the same ILSVRC image terms.

Spawrious. We use the official One-to-One Hard split (`o2o_hard`) [21], distributed as `spawrious224__o2o_hard.tar.gz`. The training split pairs four dog breeds with different backgrounds, and the official test split changes those pairings. The upstream README lists CC BY 4.0. The repository LICENSE file is CC0-1.0.

WaterBirds. We use Waterbirds v1.0 [30] through the WILDS loader, with the official training and test splits defined in `metadata.csv`. The images combine CUB-200-2011 birds with Places backgrounds. The dataset documentation limits use to non-commercial research and education. The WILDS software license is MIT.

B.2 Backbone Models

We use the four ViT-B checkpoints from the LGM implementation [3]. We extract final-layer features and freeze the backbone parameters throughout distillation and evaluation.

CLIP. We use OpenAI `ViT-B/16` [26] from the MIT-licensed CLIP repository.

DINOv2. We use Torch Hub `dinov2_vitb14` [25]. The official release licenses the code and weights under Apache-2.0.

EVA-02. We use `eva02_base_patch14_224.mim_in22k` [9]. The checkpoint card lists MIT, and the `timm` library uses Apache-2.0.

MoCo-v3. We use `nyu-visionx/moco-v3-vit-b` [6]. The upstream Meta repository uses CC BY-NC 4.0. The Hugging Face distribution does not specify a separate license.

B.3 Distillation Settings

Synthetic images are randomly initialized and optimized for $4,000$ distillation iterations. Unless otherwise specified, the image resolution is fixed to 224×224, the ridge coefficient is set to $\lambda = 0.1$, and the temperature in the outer objective is set to $\tau = 0.07$. At each distillation iteration, we sample a class-balanced real batch with $B = 4$ real samples per class to compute the outer loss.

The synthetic images are optimized using Adam with an initial learning rate of 0.05 and a cosine annealing schedule. Gradients are backpropagated through the frozen feature extractor and the closed-form linear-probe solver, while the backbone parameters remain fixed. Unless otherwise specified, the batch size for linear-probe training is set to 256.

The experiments use NVIDIA RTX 5090 GPUs and NVIDIA RTX Pro 6000 GPUs with 96GB memory. ImageNet-100 efficiency profiling is conducted on an RTX 5090, while the additional higher-IPC accuracy experiments and the ImageNet-1K efficiency profile use an RTX Pro 6000. Within each timing comparison, all methods use the same GPU model, backbone, image resolution, and batch configuration; runtime is not compared across GPU models.

B.4 Evaluation Protocol

After distillation, we evaluate the synthetic set by training a randomly initialized linear classifier on top of frozen backbone features. The linear classifier is trained only on the distilled images and evaluated on the ImageNet validation splits or on the official test splits for Spawrious and WaterBirds. Unless otherwise specified, the classifier is trained for 500 epochs using Adam with a learning rate of 0.01.

For real-image baselines, we use the same frozen backbone and the same linear-probe evaluation protocol. This ensures that the comparison reflects the quality of the distilled or selected images, rather than differences in classifier training or feature extraction.

B.5 Augmentation

By default, CLP-DD uses standard image augmentations, including random cropping and noise injection, while keeping the spatial resolution fixed at 224×224. CLP-DD does not use Differentiable Siamese Augmentation (DSA) by default. DSA is used only for explicitly identified LGM configurations and the optional CLP-DD augmentation ablation.

C Additional Experimental Results

This section provides the additional cross-backbone, IPC-scaling, efficiency, augmentation, hyperparameter-sensitivity, and numerical-stability results. Accuracy results are reported as mean $\pm$ standard deviation over three independent runs unless indicated otherwise.

C.1 Cross-Backbone Transfer

Table 10 evaluates each ImageNet-100 synthetic set at IPC=1 with all four frozen backbones. Diagonal entries use the same backbone for distillation and evaluation; the unseen average excludes the diagonal. The unseen averages of 50.9–52.3% are substantially below matched-backbone accuracy, demonstrating limited transfer and substantial source-representation dependence rather than backbone invariance.

Table 10: Cross-backbone transfer on ImageNet-100 at IPC=1. Rows identify the distillation backbone and columns the evaluation backbone. Individual accuracies are mean $\pm$ standard deviation over three runs; unseen averages report the mean over the three off-diagonal backbones.

Source	DINOv2	MoCo-v3	CLIP	EVA-02	Unseen avg.
DINOv2	$\mathbf{89.2 \pm 0.1}$	52.0 ± 0.5	46.8 ± 0.6	54.0 ± 0.4	50.9
MoCo-v3	57.2 ± 0.2	$\mathbf{83.9 \pm 0.1}$	45.6 ± 0.3	54.1 ± 0.1	52.3
CLIP	53.8 ± 0.1	46.5 ± 0.1	$\mathbf{79.2 \pm 0.1}$	55.3 ± 0.3	51.9
EVA-02	57.0 ± 0.1	48.4 ± 0.1	51.2 ± 0.2	$\mathbf{87.6 \pm 0.0}$	52.2

C.2 Controlled Cross-Backbone Comparisons

With DINOv2 as the source, Table 9 compares methods under the same fixed features, resolution, IPC, optimization budget, and evaluation protocol. CLP-DD improves the unseen average by 7.1 points over frozen-feature KIP and 4.2 points over frozen-feature FRePo. FRePo's evolving model pool is inapplicable to the fixed-encoder setting. Replacing the class-anchor objective with MSE reduces unseen accuracy by 13.4 points, providing additional evidence for the classification-oriented outer loss. Same-backbone frozen-feature adaptation remains the primary use case.

Table 11 gives the individual target-backbone results and the reported three-run unseen averages for the objective ablation. This changes the distillation-stage objective, rather than the downstream evaluation objective, and therefore does not establish robustness to substantially different probe losses or nonlinear heads.

the

Table 11: Outer-objective ablation for transfer from DINOv2 on ImageNet-100 at IPC=1. Accuracy (%) is mean $\pm$ standard deviation over three runs; the unseen average excludes DINOv2.

Outer objective	DINOv2	MoCo-v3	CLIP	EVA-02	Unseen avg.
MSE	88.3 ± 0.0	31.3 ± 0.6	39.6 ± 0.4	41.6 ± 0.3	37.5 ± 0.4
Class-anchor	$\mathbf{89.2 \pm 0.1}$	$\mathbf{52.0 \pm 0.5}$	$\mathbf{46.8 \pm 0.6}$	$\mathbf{54.0 \pm 0.4}$	$\mathbf{50.9 \pm 0.5}$

C.3 Extended IPC Scaling

Table 12 extends the real-image-selection comparison to IPC$= \{1, 2, 3, 5, 10\}$ using DINOv2. CLP-DD remains more accurate than Neighbor, Centroid, Random, and DSA-free LGM at every tested budget. The margin over Centroid decreases from 2.3 to 0.9 points as IPC grows from 1 to 10. Thus, the largest advantage is at low IPC; the 0.9-point gain at IPC=10 requires 4,000 distillation iterations, whereas Centroid requires no iterative distillation.

Table 12: Extended IPC scaling on ImageNet-100 with DINOv2. Accuracy (%) is the mean $\pm$ standard deviation over three runs. LGM denotes the DSA-free variant.

Method	IPC=1	IPC=2	IPC=3	IPC=5	IPC=10
Neighbor	86.0 ± 0.2	87.8 ± 0.3	88.7 ± 0.2	89.7 ± 0.2	90.9 ± 0.2
Centroid	86.9 ± 0.3	88.5 ± 0.2	89.3 ± 0.2	90.2 ± 0.2	91.2 ± 0.1
Random	74.8 ± 2.6	79.0 ± 2.4	81.8 ± 1.4	84.2 ± 1.3	87.4 ± 1.9
LGM	82.6 ± 0.2	85.0 ± 0.3	86.3 ± 0.2	87.8 ± 0.3	89.6 ± 0.4
CLP-DD	$\mathbf{89.2 \pm 0.1}$	$\mathbf{90.7 \pm 0.1}$	$\mathbf{91.3 \pm 0.1}$	$\mathbf{91.8 \pm 0.0}$	$\mathbf{92.1 \pm 0.1}$

Table 13 repeats the IPC$= \{1, 5, 10\}$ comparison with MoCo-v3. The margin over Centroid also narrows, from 6.2 to 3.8 points, but remains larger than on DINOv2. The diminishing-margin trend is shared by the two tested backbones, while the size of the improvement depends on the representation.

Table 13: IPC scaling on ImageNet-100 with MoCo-v3. Accuracy (%) is the mean $\pm$ standard deviation over three runs. LGM denotes the DSA-free variant.

Method	IPC=1	IPC=5	IPC=10
Centroid	77.7 ± 0.1	81.7 ± 0.2	82.9 ± 0.2
LGM	59.5 ± 0.5	68.3 ± 0.5	73.3 ± 0.4
CLP-DD	$\mathbf{83.9 \pm 0.1}$	$\mathbf{86.4 \pm 0.1}$	$\mathbf{86.7 \pm 0.1}$

C.4 ImageNet-1K Efficiency

Table 14 extends profiling to ImageNet-1K with DINOv2 at IPC=1 on a 96 GB NVIDIA RTX PRO 6000. All methods use identical hardware and batch configuration within this comparison. LGM with DSA uses $m = 3$ augmentation replicas. Time measures 100 complete distillation iterations, including method-specific feature extraction, augmentation, loss computation, backward propagation, synthetic-image updates, and the differentiable solve where applicable. Peak memory covers the same computation. One-time setup and final linear-probe evaluation are excluded for all methods.

Table 14: ImageNet-1K/DINOv2 efficiency at IPC=1 on a 96 GB RTX PRO 6000. Time is seconds per 100 complete distillation iterations, peak memory is in MB, and accuracy (%) is the mean $\pm$ standard deviation over three runs.

Method	Time / 100 iters	Peak memory	Accuracy
LGM w/ DSA ($m = 3$)	642.14	95,916	75.0 ± 0.1
LGM	41.51	10,131	63.2 ± 0.6
CLP-DD	46.33	11,924	$\mathbf{75.6 \pm 0.1}$

CLP-DD is 13.9$\times$ faster than LGM with DSA and uses 12.4% of its peak memory, while accuracy is comparable. Against DSA-free LGM, CLP-DD gains 12.4 accuracy points at 11.6% more runtime and 17.7% more peak memory. These results establish the accuracy–efficiency trade-off for this configuration; higher-IPC ImageNet-1K scaling remains untested. Absolute times are not compared with the ImageNet-100 RTX 5090 profile because the GPU models differ.

Extrapolating the measured optimization cost to 4,000 iterations gives about 1,853 s for CLP-DD, 1,660 s for DSA-free LGM, and 25,686 s for LGM with DSA. These are estimates of the iterative optimization stage, excluding setup and evaluation, rather than measured end-to-end runtimes.

C.5 Augmentation Ablations

Table 15 evaluates standard image augmentation and optional DSA on ImageNet-100 with DINOv2 at IPC=1. Removing cropping and noise reduces accuracy from $89.2 \pm 0.1\%$ to $87.9 \pm 0.8\%$, still 5.3 points above DSA-free LGM. Adding DSA with multiplicity $m = 1$ to CLP-DD yields 89.3% accuracy, with no standard deviation reported for that configuration.

Table 15: Augmentation ablations on ImageNet-100/DINOv2 at IPC=1. Accuracy (%) is the mean $\pm$ standard deviation over three runs where available; the DSA accuracy has no reported standard deviation. Time is seconds per 100 complete iterations, and peak memory is in MB. A dash denotes an unreported profiling measurement.

CLP-DD setting	Accuracy	Time / 100 iters	Peak memory
Without image augmentation	87.9 ± 0.8	–	–
Default: cropping and noise	89.2 ± 0.1	4.07	2,361
Default w/ DSA ($m = 1$)	89.3 ± 0.1	9.39	4,011

Standard augmentation contributes 1.3 points, but does not alone explain CLP-DD's advantage over DSA-free LGM. Adding DSA changes accuracy by only 0.1 points while increasing runtime by $2.31\times$ and peak memory by 1,650 MB. Under this tested configuration, CLP-DD therefore achieves strong performance without requiring DSA. Together with the MSE-objective ablation, these results support the role of the proposed objective, but provide only a partial decomposition: they do not uniquely attribute the remaining gap to LGM with DSA to any single component.

C.6 Sensitivity Across Backbones

We vary one hyperparameter at a time on ImageNet-100 at IPC=1, fixing the other at its default. Tables 16 and 17 report the complete four-backbone sweeps, including the additional intermediate ridge coefficient $\lambda = 0.20$. The same defaults, $\tau = 0.07$ and $\lambda = 0.10$, perform best among all tested values for every backbone. Accuracy decreases already at $\lambda = 0.20$, with reductions of 2.1, 2.6, 1.6, and 1.7 points on DINOv2, MoCo-v3, CLIP, and EVA-02, respectively. No backbone-specific choice of the defaults is used.

Table 16: Temperature sensitivity on ImageNet-100 at IPC=1 with $\lambda = 0.10$. Accuracy (%) is mean $\pm$ standard deviation from the three-run sensitivity sweep.

Backbone	$\tau = 0.01$	$\tau = 0.05$	$\tau = 0.07$	$\tau = 0.10$
DINOv2	80.8 ± 0.3	86.7 ± 0.2	$\mathbf{89.2 \pm 0.1}$	88.8 ± 0.1
MoCo-v3	69.7 ± 0.3	78.1 ± 0.2	$\mathbf{83.9 \pm 0.1}$	83.2 ± 0.0
CLIP	62.5 ± 0.4	73.9 ± 0.3	$\mathbf{79.2 \pm 0.2}$	78.5 ± 0.2
EVA-02	75.9 ± 0.3	83.9 ± 0.2	$\mathbf{87.6 \pm 0.1}$	87.0 ± 0.1

Table 17: Ridge-coefficient sensitivity on ImageNet-100 at IPC=1 with $\tau = 0.07$. Accuracy (%) is the mean $\pm$ standard deviation from the three-run sensitivity sweep; the $\lambda = 0.20$ column reports the added intermediate setting.

Backbone	$\lambda = 0.05$	$\lambda = 0.07$	$\lambda = 0.10$	$\lambda = 0.20$	$\lambda = 0.50$
DINOv2	86.8 ± 0.2	88.1 ± 0.1	$\mathbf{89.2 \pm 0.1}$	87.1 ± 0.2	84.6 ± 0.2
MoCo-v3	79.3 ± 0.2	81.3 ± 0.2	$\mathbf{83.9 \pm 0.1}$	81.3 ± 0.2	76.2 ± 0.1
CLIP	74.8 ± 0.3	77.1 ± 0.2	$\mathbf{79.2 \pm 0.2}$	77.6 ± 0.3	70.9 ± 0.4
EVA-02	83.8 ± 0.2	85.9 ± 0.2	$\mathbf{87.6 \pm 0.1}$	85.9 ± 0.2	80.9 ± 0.3

C.7 Numerical Stability

We examine four backbones, three seeds, and IPC= $\{1, 2, 5, 10\}$, for a total of 48 runs. For the regularized Gram matrix $A = X_{\text{syn}} X_{\text{syn}}^{\top} + \lambda I$, the median condition number increases from 7.4×10^1

at IPC=1 to 7.6×10^2 at IPC=10; the largest observed condition number is 6.4×10^3. The maximum relative solve residual remains below 10^{-4}, and FP32 and FP64 evaluation accuracies differ by at most 0.14 percentage points. We observe no solver failures, NaNs, or infinite values. Thus, $\lambda > 0$ guarantees positive definiteness, while the finite-precision behavior is supported empirically over the tested range rather than by a general guarantee of good conditioning.

D Limitations and Future Work

CLP-DD is designed for the low-IPC frozen-backbone linear-probing regime, where the goal is to obtain an efficient synthetic set for a fixed pre-trained representation. Accordingly, its behavior is naturally influenced by the representation quality and feature geometry of the chosen backbone. This setting enables an exact closed-form inner solver and strong efficiency, but also makes the distilled images backbone-specific. The cross-backbone evaluation demonstrates limited transfer, but the substantial gap to matched-backbone performance confirms strong source-representation dependence; the primary setting remains adaptation to a given frozen encoder. The current evaluation transfers distilled images to a separately initialized, iteratively trained linear classifier, but does not establish robustness to substantially different probe losses, regularization choices, or nonlinear heads. In addition, while the sample-space solver is well-suited to low-IPC distillation, larger synthetic sets may benefit from primal, block-wise, or approximate solvers. CLP-DD is competitive with real-image selection under severe background shift, rather than consistently superior, and may inherit biases of the frozen representation. Future work will explore scalable solvers, multi-backbone objectives, broader downstream probes, and robustness-oriented outer objectives.

E Broader Impact

CLP-DD aims to reduce the computational and storage cost of adapting pre-trained vision models by replacing large training sets with compact synthetic sets. This can make frozen-backbone transfer learning more accessible in settings with limited computation, storage, or data-management resources. Since CLP-DD operates on fixed visual representations, the generated synthetic images reflect information encoded by the source data and the chosen backbone. As with prior dataset distillation methods, responsible use therefore requires appropriate validation for the target application, especially under distribution shift or bias-sensitive scenarios. CLP-DD does not introduce a generative model or a new data collection pipeline, and its primary impact is to improve the efficiency of existing transfer-learning workflows.

F Additional Distilled Images

This section provides additional qualitative visualizations of different datasets and backbones.

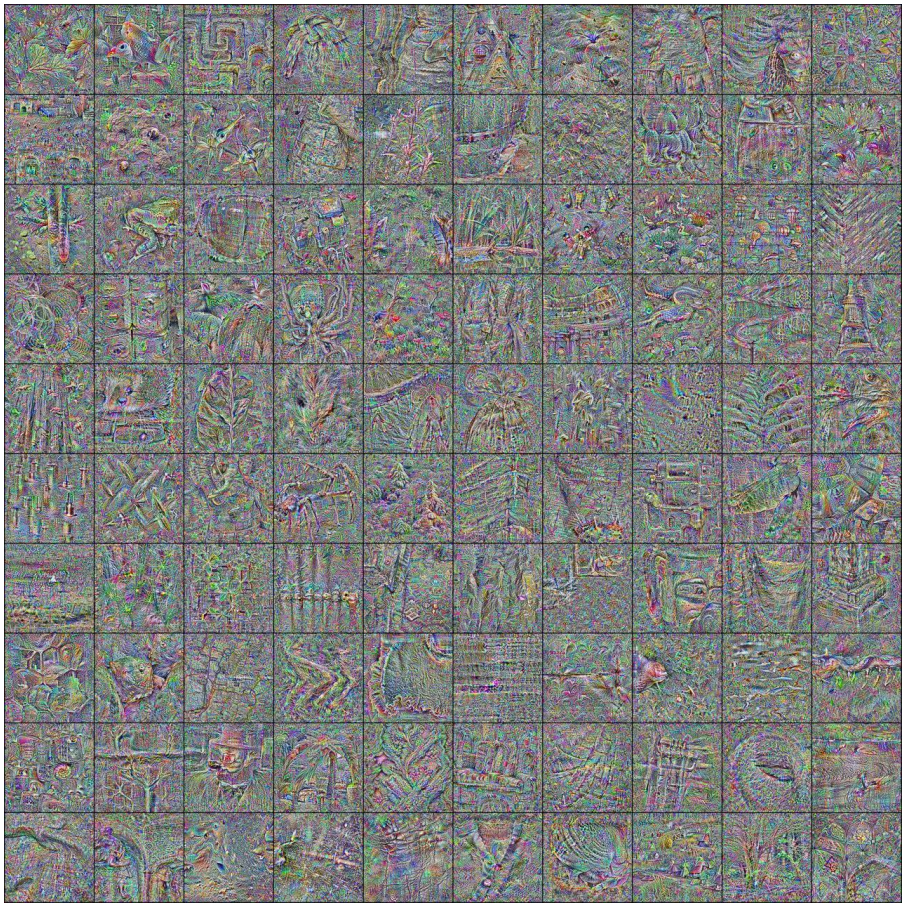


Figure 5: Additional examples of images distilled by CLP-DD on ImageNet-100 using the DINOv2 backbone.

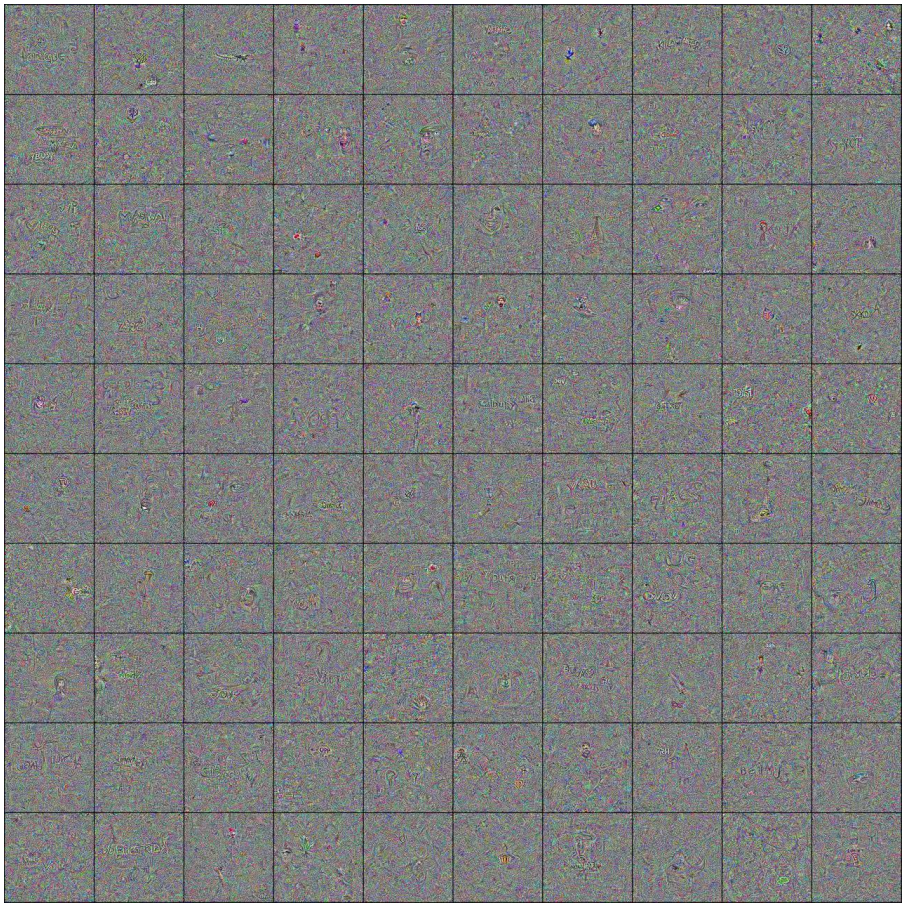


Figure 6: Additional examples of images distilled by CLP-DD on ImageNet-100 using the CLIP backbone.

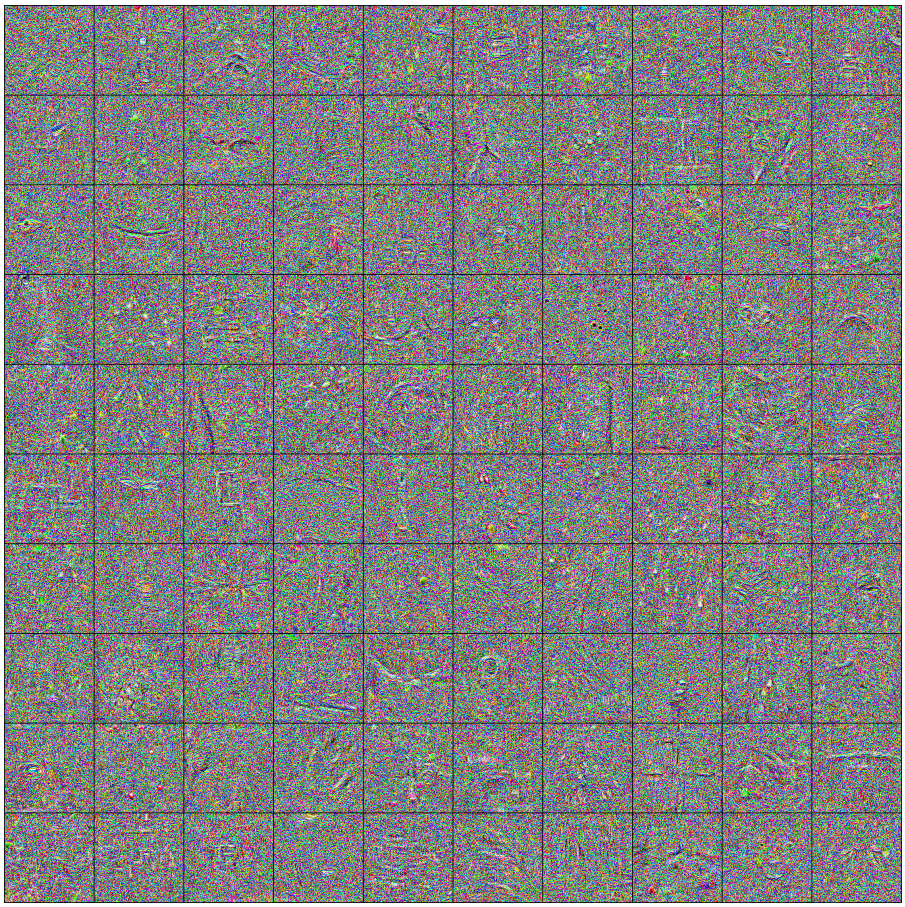


Figure 7: Additional examples of images distilled by CLP-DD on ImageNet-100 using the EVA-02 backbone.

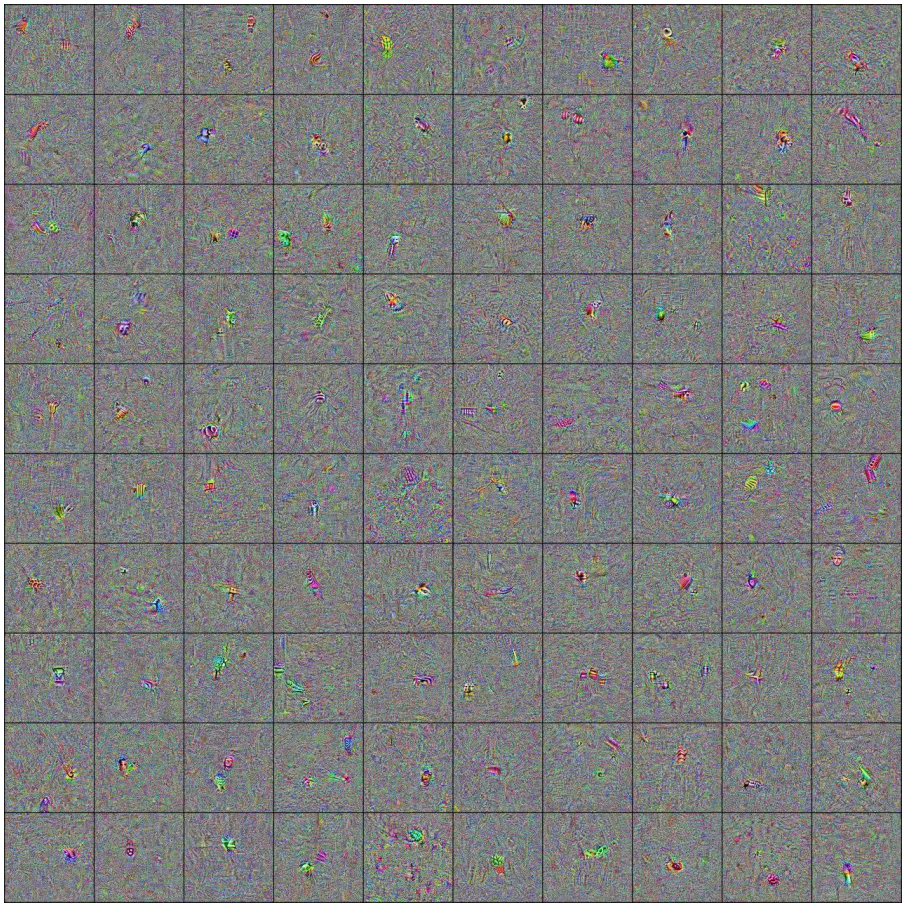


Figure 8: Additional examples of images distilled by CLP-DD on ImageNet-100 using the MoCo-v3 backbone.

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: The abstract and introduction clearly state the main research claims. The theoretical basis is provided in Section 3 and Appendix A, while the claimed experimental results in Section 4 support performance improvements.

   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: We discuss the limitations of the proposed framework in the conclusion section and provide additional details in Appendix D.

   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: [Yes]

Justification: We provide complete explanations and proofs in Section 3 of the main text and Appendix A.

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 proposed framework is fully described in Section 3 of the manuscript, while Section 4 and Appendix B present the detailed experimental configurations.

   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: [No]

Justification: We provide pseudocode and implementation details in the main paper and Appendix B. Complete reproduction scripts and configurations are not included in the paper package.

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 detailed experimental setup, including hyperparameters, is fully described in Section 4 of the main text and Appendix B.

Guidelines:

- The answer [Yes] 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: [Yes]

Justification: Accuracy tables report mean and standard deviation over three repeated runs where available; profiling measurements and aggregate transfer averages are identified separately.

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: In Section 4 and Appendix B.3, we specify the computing resources used in our experiments.

   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 have carefully reviewed the NeurIPS Code of Ethics and conducted the research in accordance with it.

   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: [Yes]

    Justification: In this study, we propose a novel perspective on dataset distillation and discuss its broader impact in Appendix E.

- reducing computational resource requirements for deep learning.
- decreasing energy consumption associated with large-scale training.

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 study proposes a dataset distillation framework whose risks are not substantially higher than those of prior dataset distillation methods. The framework does not include a generative component, and the potential for misuse of the research results is limited. Therefore, we consider this item not applicable to our study.

    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: The datasets, models, and other resources used in this paper have been properly cited.

    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: [N/A]

    Justification: This paper does not introduce a new benchmark dataset. Implementation details are provided in Appendix B.

    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 study does not involve crowdsourcing experiments 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 study does not involve human-subject research or crowdsourcing and therefore does not require IRB approval.

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: This study does not involve large language models (LLMs) in its core methodology, data processing, or experimental design. The research is based on dataset distillation techniques without any LLM-related components. Therefore, no declaration of LLM usage is required for this submission.

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.