DISC: DATASET DISTILLATION FOR SUB-1 IPC LEARNING UNDER IMPLICIT SHARED REPRESENTATIONS WITH CORESETS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distilling high-resolution datasets under tight storage budgets requires synthetic representations that are both compact and useful for downstream learning. We introduce ***D**ataset Distillation for Sub-1 IPC Learning under **I**mplicit **S**hared Representations with **C**oresets* (DISC), a framework for the sub-1 images-per-class (IPC) regime, where the storage budget is smaller than that required for 1 IPC. Building on the storage efficiency of implicit neural representations (INRs), DISC further reduces per-instance storage costs by sharing a coordinate encoder across synthetic images while retaining lightweight instance-specific decoders. This allows more synthetic instances to be stored within the same byte budget. To translate this additional capacity into effective training data, DISC couples parameter sharing with a two-stage coreset initialization: median-distance samples pretrain the shared encoder on non-outlier variation, and prototype-nearest samples warm up the encoder and decoders with representative examples before distillation. On six ImageNet subsets at a 0.5 IPC-equivalent budget, DISC improves mean Top-1 accuracy over DDiF by 4.6 and 5.4 percentage points at 128×128 and 256×256, respectively, under distribution matching. Further improvements under gradient matching and in average cross-architecture performance demonstrate the effectiveness of combining shared representations with stage-specific initialization for a better accuracy–storage trade-off in highly compressed dataset distillation.

1 INTRODUCTION

In recent years, the imbalance between data volume and computational resources has created a bottleneck in the scalability and practicality of deep learning models (Strubell et al., 2020). In response, dataset distillation has emerged as a promising solution for efficient learning (Wang et al., 2018). The core concept of dataset distillation is to create a concise synthetic dataset that retains the essential knowledge of a large-scale dataset (Shin et al., 2025; Zhao et al., 2021; Wang et al., 2025). This approach enables efficient model training without relying on a full-scale dataset, thereby reducing storage costs and improving training efficiency (Yu et al., 2024).

Gradient Matching (DC) (Zhao et al., 2021), Distribution Matching (DM) (Zhao & Bilen, 2023), and Trajectory Matching (Cazenavette et al., 2022) are widely used objectives for dataset distillation. A crucial challenge remains the effective distillation of high-dimensional, high-resolution datasets (Lei & Tao, 2023). These foundational methods directly optimize in pixel space, which scales poorly with resolution due to the curse of discretization (Mescheder, 2020). Furthermore, recent studies have primarily focused on achieving high performance with relatively large budgets, such as 10 or 50 IPC (Wang et al., 2025; Zou et al., 2025). Consequently, the sub-1 IPC regime, which directly tests the objective of maximizing data compression, remains largely underexplored, especially for high-resolution data.

To overcome the poor scaling of pixel-space optimization, recent work parameterizes synthetic data as compact codes that are decoded into images. Neural fields are a natural fit for high-resolution synthetic images: DDiF (Shin et al., 2025) represents each image with an independent sinusoidal representation network (Sitzmann et al., 2020), storing it in far fewer parameters than its pixels. In the sub-1 IPC regime, how the budget is allocated becomes the central question. Each independently

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

stored field carries a fixed cost, which caps the number of synthetic instances that fit within the budget; sharing part of the representation across instances lowers this marginal cost so that more instances fit. The remaining challenge is to choose and initialize those additional instances so that the larger count translates into a better distilled dataset.

We introduce ***D**ataset Distillation for Sub-1 IPC Learning under **I**mplicit **S**hared Representations with **C**oresets* (DISC), a co-design of storage allocation and instance selection. Following the encoder-sharing structure of STRAINER (Vyas et al., 2024), DISC stores one encoder and an individual lightweight decoder per synthetic image. At 128×128, an additional DISC decoder costs 1,932 bytes versus 3,852 bytes for a DDiF field, allowing 50 rather than 25 instances per class at the same 0.5 IPC budget.

DISC pairs this allocation with a two-stage coreset pipeline. Median-distance samples pretrain the shared encoder on non-outlier within-class variation, and prototype-nearest samples provide representative targets for encoder–decoder warmup before distillation. This stage-specific selection turns the additional instance capacity into accuracy under small budgets. On six ImageNet subsets under DM at 0.5 IPC, DISC reaches 52.4% average Top-1 accuracy compared with DDiF's 47.8%. Across the full budget sweep, DISC degrades gracefully from IPC 1.0 to 0.1 and stays ahead of DDiF at every sub-1 budget under both objectives.

Our main contributions are summarized as follows:

- We introduce DISC, a storage-efficient framework for sub-1 IPC dataset distillation. Building on implicit neural representations, DISC shares a coordinate encoder across synthetic images while retaining lightweight instance-specific decoders, fitting more synthetic instances within the same byte budget.
- We develop a two-stage coreset initialization tailored to shared neural fields: median-distance samples pretrain the shared encoder, and prototype-nearest samples warm up the encoder and decoders. This stage-specific initialization translates the added instance capacity into more effective synthetic data.
- Across ImageNet subsets, two resolutions (DM), and both DM and DC, DISC degrades gracefully as the budget shrinks and improves the low-budget accuracy–storage trade-off.

2 Related work

2.1 Implicit neural representations (INRs)

Implicit Neural Representations (INRs) model data as continuous functions using coordinate-based Multi-layer Perceptrons (MLPs), offering a resolution-independent and memory-efficient approach. However, traditional INRs using ReLU activation functions exhibit a spectral bias, prioritizing low-frequency components during training (Essakine et al., 2024). SIREN (Sitzmann et al., 2020) mitigates this bias by employing sinusoidal activations $\sigma(x) = \sin(\omega x)$, allowing effective modeling of high-frequency content.

2.1.1 STRAINER.

STRAINER (Vyas et al., 2024) shares encoder layers across multiple signals while retaining signal-specific decoders, and studies transfer and reconstruction of unseen signals. DISC brings this encoder-sharing structure into fixed-budget dataset distillation, where sharing reduces the marginal storage cost of each synthetic instance. DISC co-designs compressed instance allocation and coreset-guided initialization: the shared encoder expands the number of decoders that fit within the budget, and class-wise coresets populate this capacity before distillation.

2.2 Dataset distillation

Dataset Distillation (Wang et al., 2018) aims to synthesize a small data summary from a large dataset to reduce computational costs, with primary methods including Gradient Matching (Zhao et al., 2021), Distribution Matching (Zhao & Bilen, 2023), and Trajectory Matching (Cazenavette et al., 2022). While effective, these techniques face scalability challenges with high-resolution images and

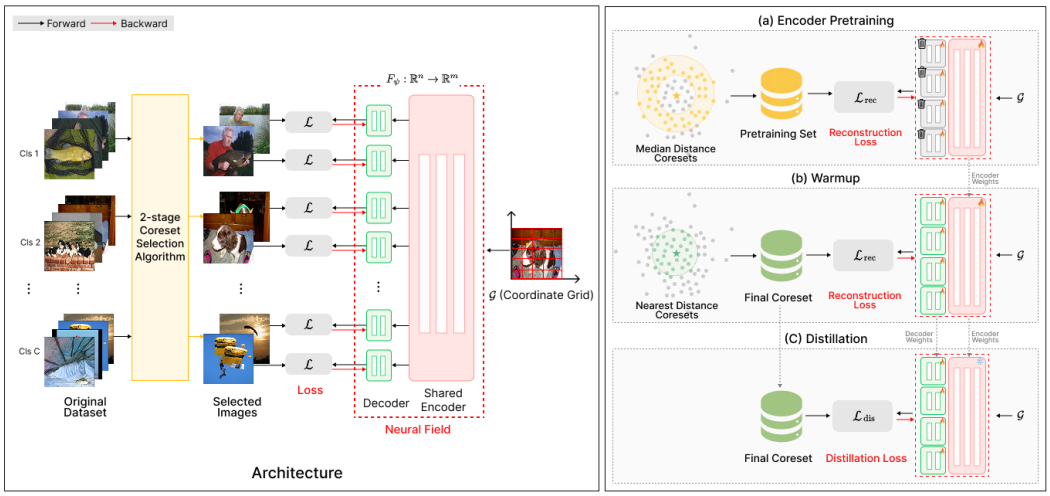


Figure 1: Overview of the DISC framework. One shared encoder lowers the storage cost of each additional synthetic instance. (a) Encoder pretraining: median-distance coreset samples are reconstructed with temporary instance-specific decoders. (b) Warmup: the transferred encoder and the budgeted per-instance decoders are jointly fit to prototype-nearest coreset samples. (c) Distillation: the encoder is fixed and only decoders are optimized with distribution or gradient matching. The two coreset stages initialize the instance capacity made available by compression.

in extremely low-budget (sub-1 IPC) scenarios where optimizing in pixel space is inefficient. This has established the efficient parameterization of synthetic data as a critical research direction to overcome such limitations.

2.2.1 Distilling Dataset into Neural Field (DDiF).

DDiF (Shin et al., 2025) stores synthetic images as independent SIREN neural fields and demonstrates the value of continuous image parameterization for high-resolution distillation. DISC builds on this representation-space perspective by sharing parameters across instances to lower their marginal storage cost. Its two-stage coreset pipeline then gives encoder pretraining and warmup stage-specific reconstruction targets, coupling higher compression with informed initialization in the sub-1 IPC regime.

2.2.2 Poster Dataset Distillation (PoDD).

PoDD (Shul et al., 2025) compresses a dataset into a single poster image and enables sub-1 IPC distillation by allowing multiple classes to share pixels, with their spatial arrangement guided by semantic similarity. Its unrolled optimization retains the poster, differentiable parameter updates, and optimizer state, making its running-time overhead substantial at high resolution, as discussed in Appendix C.2.

3 Methodology

3.1 Budget definition

Conventional dataset distillation methods typically specify the data budget using Images Per Class (IPC), storing synthetic images explicitly in pixel space. In contrast, parameterization-based approaches, including DISC, represent synthetic data using neural parameters, allowing flexible and fine-grained control over capacity under extremely low budgets. We define the total parameter budget as:

$$B_{\text{total}} = C \times r^2 \times \text{IPC} \times K, \tag{1}$$

where C is the number of channels, r is the image resolution, and K is the number of classes. Given this budget, we allocate parameters to a shared encoder (P_{enc}) and lightweight decoders (P_{dec} each),

yielding:

$$N_{\text{dec}} = \left\lfloor \frac{B_{\text{total}} - P_{\text{enc}}}{P_{\text{dec}}} \right\rfloor, \quad N_{\text{dec/class}} = \left\lfloor \frac{N_{\text{dec}}}{K} \right\rfloor. \tag{2}$$

For comparison, an independent-field method can store at most $\lfloor B_{\text{total}}/P_{\text{field}} \rfloor$ instances, where P_{field} is the parameter count of one field. With 32-bit parameters at 128×128, a DISC decoder costs 1,932 bytes and a DDiF field 3,852 bytes, while the DISC encoder incurs a one-time 5,280-byte cost (Appendix A.6.3). Thus the shared allocation admits roughly twice as many instances within the same budget; the coreset stages below determine which images initialize that capacity.

3.2 Neural field

DDiF (Shin et al., 2025) represents each synthetic image as a coordinate-to-RGB SIREN field. DISC retains this neural-field representation but stores the coordinate encoder once and pays only for another lightweight decoder when adding an image. This changes the instance count available at a given byte budget.

3.3 Implicit shared representations architecture

Building upon this neural field formulation, DISC instantiates a shared-encoder INR parameterization for dataset distillation. The architecture follows the encoder-sharing principle of STRAINER (Vyas et al., 2024). As illustrated in Figure 1, one encoder is stored across all classes and each synthetic instance has its own lightweight decoder. This factorization reduces the marginal parameter cost and expands the number of instances available under the fixed budget. The two-stage coreset pipeline then supplies stage-specific reconstruction targets for pretraining and warmup, coupling compression with informed initialization.

By default, the shared encoder has three hidden layers and each decoder has one (details in Appendix A.5).

3.4 Two-stage coreset selection

To use the larger instance allocation, DISC applies a **two-stage coreset selection** strategy before distillation. Intuitively, we (i) expose the shared encoder to samples around the *median* prototype distance to encourage broad intra-class coverage, and then (ii) warm up the neural fields using *representative* samples near the class prototype to obtain a high-fidelity reconstruction initialization.

3.4.1 Distance in feature space.

We define a class-wise distance metric in a fixed feature space induced by a feature extractor $\Phi(\cdot)$. Detailed specifications of the feature extractor and preprocessing recipes are provided in Appendix A.3. Given a class-k dataset $\mathcal{X}^{(k)} = \{x_i^{(k)}\}_{i=1}^{n_k}$, we compute the class prototype (centroid)

$$\mu_k \;=\; \frac{1}{n_k} \sum_{i=1}^{n_k} \Phi(x_i^{(k)}), \tag{3}$$

and measure the prototype distance of each sample as

$$d_i^{(k)} \;=\; \left\| \Phi(x_i^{(k)}) - \mu_k \right\|_2. \tag{4}$$

We use the same distance definition for both stages; the difference lies only in the selection rule (median-targeted vs. prototype-nearest).

3.4.2 Pretraining: median-distance coreset selection.

For each class k, we select a subset $\mathcal{S}_{\text{med}}^{(k)} \subset \mathcal{X}^{(k)}$ by targeting the *median* distance of the prototype distribution $\{d_i^{(k)}\}_{i=1}^{n_k}$. Motivated by observations that samples of *medium difficulty* can provide a favorable trade-off between fidelity and learnability (Xia et al., 2022; Shen et al., 2025; Chen et al., 2025), we prioritize samples whose $d_i^{(k)}$ lie around the median, avoiding both extremes: highly

distinct outliers (large $d_i^{(k)}$) and overly redundant, near-prototype samples (small $d_i^{(k)}$). Formally, letting $\mathrm{med}_k = \mathrm{median}(\{d_i^{(k)}\}_{i=1}^{n_k})$, we select

$$\mathcal{S}_{\text{med}}^{(k)} \;=\; \arg\min_{\substack{\mathcal{S}\subset\mathcal{X}^{(k)} \\ |\mathcal{S}|=M_{\text{med}}}} \sum_{x_i^{(k)}\in\mathcal{S}} \left| d_i^{(k)} - \mathrm{med}_k \right|. \tag{5}$$

We use $\mathcal{S}_{\text{med}}^{(k)}$ to pretrain the encoder on non-outlier within-class variation before initializing the final set of decoders.

3.4.3 Final coreset: prototype-nearest selection (warmup).

For each class k, we also select a subset $\mathcal{S}_{\text{near}}^{(k)} \subset \mathcal{X}^{(k)}$ using the same prototype distance $\{d_i^{(k)}\}$, but with a prototype-nearest rule. These representative instances near the class prototype provide a stable reconstruction signal for warmup. Formally, we select the nearest samples to the prototype:

$$\mathcal{S}_{\text{near}}^{(k)} \;=\; \arg\min_{\substack{\mathcal{S}\subset\mathcal{X}^{(k)} \\ |\mathcal{S}|=N_{\text{dec/class}}}} \sum_{x_i^{(k)}\in\mathcal{S}} d_i^{(k)}, \tag{6}$$

i.e., the $N_{\text{dec/class}}$ samples with the smallest $d_i^{(k)}$. We use $\mathcal{S}_{\text{near}}^{(k)}$ to warm up the neural fields (shared encoder + instance decoders) prior to distillation.

3.5 Training procedure & objective

DISC follows three conceptual steps: **(1) Encoder pretraining**, **(2) Warmup**, and **(3) Distillation**. In Step (1), we expose the encoder to samples around the *median* distance of the prototype distribution, covering non-outlier within-class variation. In Step (2), we switch to samples that are *nearest to the class prototype* to obtain a stable, representative reconstruction initialization. Finally, Step (3) performs dataset distillation by matching training signals between real and synthetic data.

3.5.1 Step 1: encoder pretraining with median-distance coreset.

We first pretrain the shared encoder using the median-distance coreset from Section 3.4. Let $\mathcal{S}_{\text{med}}^{(k)} = \{\tilde{x}_j^{(k)}\}_{j=1}^{M_{\text{med}}}$ denote the selected images for class k, with ground-truth pixel grids $v_j^{(k)} \in \mathbb{R}^{C\times r\times r}$ and a fixed coordinate grid $\mathcal{G}$. We initialize a set of instance-specific decoders $\{\psi_j^{(k)}\}$ (used only during encoder pretraining) and optimize the reconstruction objective:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\Psi_{\text{med}}|} \sum_{k=1}^{K} \sum_{j=1}^{M_{\text{med}}} \left\| F_{\theta_{\text{enc}},\,\psi_j^{(k)}}(\mathcal{G}) - v_j^{(k)} \right\|_2^2, \tag{7}$$

where θ_{enc} denotes the shared encoder parameters and $|\Psi_{\text{med}}| = K\,M_{\text{med}}$. This step initializes the shared parameters from a coreset chosen for non-outlier within-class variation.

3.5.2 Step 2: warmup with prototype-nearest coreset (encoder transfer).

Next, we select a prototype-nearest coreset $\mathcal{S}_{\text{near}}^{(k)} = \{\bar{x}_i^{(k)}\}_{i=1}^{N_{\text{dec/class}}}$ (Section 3.4). We then construct the neural fields for distillation by:

1. **Transferring** the shared encoder weights θ_{enc} obtained from **Step (1)**, and
2. **Initializing** per-instance decoders $\{\psi_i^{(k)}\}$ corresponding to samples in $\mathcal{S}_{\text{near}}^{(k)}$.

We perform warmup via reconstruction:

$$\mathcal{L}_{\text{rec}} = \frac{1}{|\Psi|} \sum_{k=1}^{K} \sum_{i=1}^{N_{\text{dec/class}}} \left\| F_{\theta_{\text{enc}},\,\psi_i^{(k)}}(\mathcal{G}) - u_i^{(k)} \right\|_2^2, \tag{8}$$

where $u_i^{(k)} \in \mathbb{R}^{C\times r\times r}$ denotes the ground-truth pixel grid of $\bar{x}_i^{(k)}$, $|\Psi| = K\,N_{\text{dec/class}}$, and $F_{\theta_{\text{enc}},\,\psi_i^{(k)}}$ denotes the neural field output. Adam updates both θ_{enc} and the decoders to obtain a high-PSNR initialization.

3.5.3 STEP 3: DISTILLATION VIA GENERALIZED MATCHING.

After warmup, DISC freezes the encoder and optimizes the decoders by matching real and synthetic training signals on a randomly initialized network f_θ. We employ two standard objectives, Distribution Matching (DM) and Gradient Matching (DC):

$$\mathcal{L}_{\text{dis}} = \begin{cases} \mathcal{L}_{\text{DM}}, & \text{(Distribution Matching; DM)} \\ \mathcal{L}_{\text{DC}}, & \text{(Gradient Matching; DC).} \end{cases} \tag{9}$$

Distribution matching (DM). DM aligns the class-conditional feature statistics of real and synthetic data. Let $\phi_\theta(\cdot)$ denote an intermediate feature extractor induced by f_θ (e.g., the penultimate-layer embedding), and let $\text{Aug}(\cdot)$ be an augmentation operator applied identically to both real and synthetic samples. Then DM minimizes the discrepancy between class-wise mean embeddings:

$$\begin{aligned} \mathcal{L}_{\text{DM}} = \sum_{k=1}^{K} \Big\| & \mathbb{E}[\phi_\theta(\text{Aug}(x_s)) \mid y = k] \\ & - \mathbb{E}[\phi_\theta(\text{Aug}(x_r)) \mid y = k] \Big\|_2^2. \end{aligned} \tag{10}$$

Gradient matching (DC). Given class-wise real and synthetic mini-batches $(x_r^{(k)}, y^{(k)})$ and $(x_s^{(k)}, y^{(k)})$, DC matches the gradients of the classification loss $\ell(\cdot, \cdot)$ w.r.t. the network parameters θ:

$$\mathcal{L}_{\text{DC}} = \sum_{k=1}^{K} D\Big(\nabla_\theta \ell\big(f_\theta(x_s^{(k)}), y^{(k)}\big),\ \nabla_\theta \ell\big(f_\theta(x_r^{(k)}), y^{(k)}\big)\Big), \tag{11}$$

where D sums one minus the cosine similarity over output units and layers (Zhao et al., 2021).

In both objectives, $\mathcal{L}_{\text{dis}}$ is evaluated under random initializations $\theta \sim \mathcal{P}_{\text{init}}$.

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Most experiments are conducted on high-resolution ImageNet-Subsets (Cazenavette et al., 2022; 2023). We use 128×128 images for the main sub-1 IPC comparisons and additionally report DM results at 256×256 to evaluate DISC at higher resolution. To further evaluate performance in the low-resolution setting, we additionally employ CIFAR-10 (Krizhevsky & Hinton, 2009) (32×32) and compare our method against PoDD (Shul et al., 2025), a baseline specifically designed for sub-1 IPC. We train DISC with both DM and DC losses. Unless otherwise specified, we employ DM as the default objective, including for the CIFAR-10 experiments, and use DC to verify robustness across distillation losses. Each distilled set is evaluated by training a conventional Convolutional Network classifier from random initialization: ConvNetD5 for ImageNet-Subsets at 128×128, ConvNetD6 for the 256×256 DM comparison in Table 1, and ConvNetD3 for CIFAR-10. The resulting performance is compared against the state-of-the-art parameterization-based method DDiF (Shin et al., 2025), as well as other recent baselines. Full synthesis cost is reported in Appendix C.1.

4.2 MAIN RESULTS

Table 1 compares DISC with representative parameterization-based dataset distillation baselines under DM at two resolutions of ImageNet-Subsets. DISC achieves the best average performance at both 128×128 and 256×256, and its advantage over DDiF grows as the budget shrinks: at 128×128 the margin rises from +0.5pp at IPC 1.0 to +4.6pp at 0.5 and +6.3pp at 0.25, and at 256×256 from +1.2pp to +5.4pp and +14.7pp. At IPC 0.25 and 128×128, DISC already exceeds DDiF at twice the budget (48.9% vs. 47.8% at IPC 0.5). These gains reflect the co-design: at a fixed byte budget DISC stores twice as many synthetic instances, and the coreset pipeline initializes this additional capacity with informative targets.

Table 1: Main performance comparison under Distribution Matching (DM) at 128×128 and 256×256 resolutions. Results marked with † are reproduced using the official implementation. Best and second-best markings are determined within the IPC 1.0 setting for each resolution.

Method	IPC	128 × 128							256 × 256						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
Vanilla	1	30.4	20.7	20.4	–	20.1	26.6	23.6	32.1	20.0	19.5	33.4	21.2	27.6	25.6
IDC	1	48.3	27.0	29.9	–	30.5	38.8	34.9	53.7	30.2	33.1	52.2	34.6	47.0	41.8
FreD	1	56.2	31.0	33.4	–	33.3	42.7	39.3	54.2	31.2	32.5	49.1	34.0	43.1	40.7
GLaD	1	32.2	21.2	21.8	–	22.3	27.6	25.0	–	–	–	–	–	–	–
H-GLaD	1	34.8	23.9	24.4	–	24.2	29.5	27.4	–	–	–	–	–	–	–
DDiF	1	69.2	42.0	45.3	62.4†	**45.8**	64.6	54.9	67.8	39.6	43.2	63.1	44.8	**67.0**	54.3
	0.5	63.2†	34.4†	38.4†	55.6†	39.3†	55.7†	47.8	61.4†	31.9†	38.0†	56.2†	38.9†	54.8†	46.9
	0.25	57.4†	31.2†	35.4†	50.7†	34.0†	46.9†	42.6	42.6†	23.6†	25.5†	42.0†	24.5†	32.4†	31.8
DISC (Ours)	1	**69.6**	**42.4**	**45.5**	**63.9**	44.0	**67.0**	**55.4**	**68.6**	**43.8**	**44.1**	**64.1**	**46.3**	65.8	**55.5**
	0.5	66.4	38.8	43.7	60.0	41.8	63.4	52.4	66.1	40.3	41.1	59.8	42.6	63.6	52.3
	0.25	61.6	37.2	40.6	57.3	39.6	56.9	48.9	61.2	33.2	37.8	52.8	38.8	55.0	46.5
Entire dataset $\mathcal{T}$		87.4	67.0	63.9	84.4	66.7	87.5	76.2	92.5	80.1	70.2	90.5	72.2	93.2	83.1

4.3 Universality to distillation loss

Following the evaluation style of DDiF, we examine whether the complete DISC pipeline remains effective when the distillation objective changes. Table 2 reports DC results on ImageNet-Subsets at 128×128. DISC outperforms DDiF at every budget, by +4.0pp, +7.0pp, and +6.2pp at IPC 1.0, 0.5, and 0.25. With the DM results in Table 1, this shows that DISC's advantage is not tied to one objective.

Table 2: Universality to distillation loss: performance under Gradient Matching (DC) on ImageNet-Subsets (128×128). Results marked with † are reproduced using the official implementation. Best and second-best markings are determined within the IPC 1.0 setting.

Method	IPC	DC						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
Vanilla	1	34.2	22.5	21.0	–	22.0	32.0	26.3
IDC	1	45.4	25.5	26.8	–	25.3	34.6	31.5
FreD	1	49.1	26.1	30.0	–	28.7	39.7	34.7
GLaD	1	35.4	22.3	20.7	–	22.6	33.8	27.0
H-GLaD	1	36.9	24.0	22.4	–	24.1	35.3	28.5
DDiF	1	61.2	35.2	37.8	56.0†	39.1	54.3	47.3
	0.5	56.1†	31.6†	32.8†	48.7†	32.6†	40.4†	40.4
	0.25	49.0†	28.0†	30.7†	45.4†	29.2†	35.4†	36.3
DISC (Ours)	1	**66.1**	**41.4**	**41.6**	**60.2**	**41.5**	**56.7**	**51.3**
	0.5	61.1	38.5	40.5	56.6	38.1	49.7	47.4
	0.25	56.5	35.4	34.4	51.8	34.7	42.3	42.5
Entire dataset $\mathcal{T}$		87.4	67.0	63.9	84.4	66.7	87.5	76.2

4.4 Cross-architecture generalization

We evaluate the cross-architecture generalization of DISC by distilling data with a standard ConvNetD5 and evaluating it on various unseen architectures: AlexNet, VGG11, ResNet18, and ViT. Table 3 reports the average test accuracy across these four architectures. DISC has higher average accuracy than DDiF at every budget, and the margin grows from +1.0pp at IPC 1.0 to +3.4pp at 0.5 and +5.4pp at 0.25. The data

Table 3: Cross-architecture DM results on ImageNet-Subsets (128×128). Test accuracy (%) averaged over AlexNet, VGG11, ResNet18, and ViT. Bold denotes the best within each IPC. † indicates results reproduced with the official implementation.

Method	IPC	Subset						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DDiF	1.0	60.0†	31.8†	38.4†	55.4†	**35.6**†	59.3†	46.8†
	0.5	53.3†	27.8†	34.4†	49.6†	30.5†	52.0†	41.3†
	0.25	45.1†	22.6†	30.4†	42.7†	26.6†	42.5†	35.0†
DISC (Ours)	1.0	**62.1**	**35.0**	**40.2**	**56.7**	32.1	**60.8**	**47.8**
	0.5	**57.3**	**31.6**	**38.2**	**52.6**	**30.9**	**57.7**	**44.7**
	0.25	**50.9**	**27.1**	**35.3**	**49.8**	**28.6**	**50.7**	**40.4**

distilled by DISC therefore transfers to unseen architectures, with the largest gains in the most compressed setting.

4.5 Cross-resolution generalization

We evaluate cross-resolution generalization on ImageNet-Nette by distilling at 128×128 with ConvNetD5 and evaluating at 256×256 (ConvNetD6) and 512×512 (ConvNetD7). Detailed settings and hyperparameters are provided in Appendices A.4 and A.5. Figure 2 illustrates the test accuracy and the performance degradation relative to the original resolution.

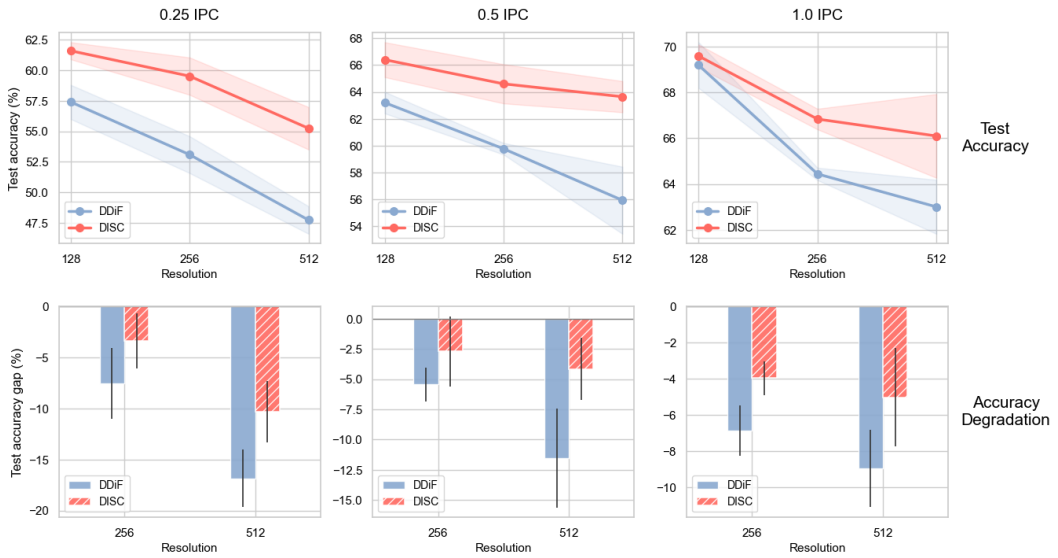


Figure 2: Cross-resolution evaluation on ImageNet-Nette with DM loss. Synthetic data are distilled with ConvNetD5 at 128×128 and evaluated with ConvNetD6 at 256×256 and ConvNetD7 at 512×512. The top row compares test accuracy and the bottom row reports degradation relative to 128×128. DISC has higher accuracy and lower relative degradation than reproduced DDiF results at IPC 0.25, 0.5, and 1.0.

DISC outperforms DDiF across the tested resolutions and IPC budgets. As shown in Figure 2, DISC also exhibits less relative accuracy degradation, defined as $(Acc_{\text{org}} - Acc_{\text{test}})/Acc_{\text{org}}$, as the evaluation resolution increases. The data distilled by DISC therefore generalizes to higher resolutions more robustly than DDiF.

Table 4: Six-subset average Top-1 accuracy (%) on ImageNet-Subsets (128×128) as the budget decreases to IPC 0.1. The DDiF entries at IPC 1.0 use published values except for the reproduced Yellow subset; all other DDiF entries are reproduced with the official implementation. Per-subset results for every budget from 1.0 to 0.1 are given in Appendix C.

Loss	Method	1.0	0.8	0.6	0.4	0.2	0.1
DM	DDiF	54.9	50.5	49.0	43.0	40.7	36.5
	DISC (Ours)	**55.4**	**53.2**	**52.6**	**50.3**	**43.7**	**40.6**
DC	DDiF	47.3	42.4	41.5	36.5	35.8	32.7
	DISC (Ours)	**51.3**	**48.8**	**47.5**	**44.4**	**37.9**	**34.7**

4.6 Budget scaling

Table 4 extends the budget sweep from IPC 1.0 down to 0.1. Under both objectives, DISC stays ahead of DDiF at every budget and degrades gradually, reaching 40.6% (DM) and 34.7% (DC) at IPC 0.1. The per-subset results for every budget, together with the full degradation curves, are given in Appendix C.

4.7 Impact of coreset selection strategy

We analyze the impact of different coreset selection strategies on the final distillation performance. Note that the Random strategy selects a coreset purely at random, independent of the feature extractor $\Phi(\cdot)$. Table 5 compares available single-stage selection methods with two-stage selection strategies

Table 5: Coreset selection ablation on ImageNet-Subsets (128×128) at IPC 1.0 under DM. Results are mean $\pm$ std accuracy (%); † marks reproduced DDiF results. DDiF has no encoder-pretraining stage (–).

Selection stage	Subset selection	Average Accuracy (%)	
		DDiF	DISC (Ours)
1-stage	Random	54.9 ±0.5	48.9 ±0.5
	Farthest	51.0† ±0.5	48.3 ±0.4
	Median	53.9† ±0.6	51.9 ±0.5
	Proto-Nearest	53.1† ±0.4	50.1 ±0.4
2-stage	Random + Random	–	52.0 ±0.5
	Proto-Nearest + Median	–	52.5 ±0.5
	Random + Proto-Nearest	–	54.0 ±0.4
	Farthest + Proto-Nearest	–	54.0 ±0.4
	Median + Proto-Nearest	–	**55.4** ±0.3

that combine an encoder-pretraining set and a final warmup set. The results show that median-distance pretraining followed by prototype-nearest warmup yields the best average accuracy of all strategies we evaluate. A fully random two-stage pipeline reaches 52.0%, while the informed pairing reaches 55.4%; reversing the selection order reaches 52.5%. Thus the selection rules and their assignment to stages matter for using the budgeted decoder capacity. Median-distance samples expose the encoder to non-outlier variation, while prototype-nearest samples provide representative warmup targets. A reduced set of controls at IPC 0.5 shows a consistent trend: adding each component in turn raises accuracy (DDiF 47.8 $\rightarrow$ DDiF with prototype-nearest initialization 48.7 $\rightarrow$ DISC with random selection 50.8 $\rightarrow$ the full method 52.4), so both the larger allocation and informed selection contribute. Per-subset results and additional ablations are provided in Appendix C.3.2.

4.8 Performance on low-resolution dataset

Table 6: Top-1 accuracy (%) on CIFAR-10 (32×32) under sub-1 IPC settings, comparing DISC with DDiF (Shin et al., 2025) and PoDD (Shul et al., 2025). †Not reported in the original paper; obtained by running the official PoDD implementation. At IPC 0.1, the poster budget equals a single 32×32 image.

IPC	CIFAR-10		
	DDiF	PoDD	DISC (Ours)
0.5	46.9 ±0.2	49.5 ±0.5	**50.2** ±0.5
0.4	44.7 ±0.5	47.1 ±0.4	**47.9** ±0.4
0.3	42.0 ±0.1	42.3 ±0.3	**46.5** ±0.3
0.2	39.3 ±0.7	31.9† ±0.5	**42.6** ±0.2
0.1	34.2 ±0.9	11.1† ±1.1	**38.2** ±0.7

We extend our evaluation to the low-resolution CIFAR-10 (32×32) dataset to benchmark DISC against PoDD (Shul et al., 2025), one of the few prior methods designed specifically for the sub-1 IPC setting. We also include DDiF as the closest INR-based baseline at the same low-resolution storage budget. We focus on budgets from IPC 0.5 down to 0.1, where low resolution and a tight byte budget together form the most extreme compression condition. Table 6 presents the results. DISC outperforms both reproduced DDiF and PoDD at every budget. The PoDD entries at IPC 0.3–0.5 are the reported values, which we reproduced within their reported standard deviations using the official implementation; since IPC 0.2 and 0.1 are not reported, we obtained these entries with the same implementation and protocol, changing only the poster size to match the budget.

5 Conclusion & Limitations

We introduced ***D**ataset Distillation for Sub-1 IPC Learning under **I**mplicit **S**hared Representations with **C**oresets* (DISC) for dataset distillation under sub-1 IPC storage budgets. Sharing the encoder reduces the marginal cost of a synthetic instance, while median-distance pretraining and prototype-nearest warmup help turn the resulting larger instance allocation into accuracy. Across ImageNet subsets at two resolutions, two distillation objectives, and CIFAR-10, DISC improves the accuracy–budget trade-off over prior parameterization methods, with larger gains below IPC 1.0 than at IPC 1.0. Our evaluation covers a limited set of datasets and backbones, and a formal account of the allocation–selection interaction remains open.

AI USE STATEMENT

In this work, we did not use generative AI tools for any tasks requiring disclosure, including generating synthetic datasets, developing theoretical models or conceptual frameworks, formulating mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, cleaning or reformatting datasets, or interpreting results. Tasks related to providing ingredients for or writing proofs, translation, and qualitative or thematic data analysis were not applicable to this work.

We used generative AI tools only to edit the manuscript for readability and tone. All AI-assisted edits were reviewed by the authors and checked against the original text to ensure that the intended technical meaning and factual content were preserved.

We take responsibility for the final content of this work, including any text, claims, or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

Section 3 describes the method, including the budget definition, the shared-encoder parameterization, the two-stage coreset selection, and the training objectives, and Appendix B gives pseudocode for encoder pretraining, warmup, and distillation. Appendix A describes the baselines and datasets, the feature extractor used for coreset selection (Appendix A.3), the training and evaluation protocol (Appendix A.4), the main hyperparameters (Appendix A.5), and the parameter budgeting that determines the number of decoders per class (Appendix A.6). Per-subset results with standard deviations for the tabulated comparisons are given in Appendix C; the cross-resolution study (Figure 2) covers ImageNet-Nette only, and its shaded bands show the spread over evaluation classifiers.

REFERENCES

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*, pp. 4750–4759, 2022.

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

Yudong Chen, Xuwei Xu, Frank de Hoog, Jiajun Liu, and Sen Wang. Medium-difficulty samples constitute smoothed decision boundary for knowledge distillation on pruned datasets. In *The Thirteenth International Conference on Learning Representations*, 2025.

Amer Essakine, Yanqi Cheng, Chun-Wun Cheng, Lipei Zhang, Zhongying Deng, Lei Zhu, Carola-Bibiane Schönlieb, and Angelica I Aviles-Rivero. Where do we stand with implicit neural representations? a technical and performance survey. *arXiv preprint arXiv:2411.03688*, 2024.

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 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 11102–11118. PMLR, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL `https://arxiv.org/abs/1412.6980`.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`.

Shiye Lei and Dacheng Tao. A comprehensive survey of dataset distillation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1):17–32, 2023.

Lars Morten Mescheder. *Stability and Expressiveness of Deep Generative Models*. PhD thesis, Eberhard Karls Universität Tübingen, Tübingen, Germany, May 2020. URL `http://hdl.handle.net/10900/106074`. Day of Oral Examination: 2020-05-06.

Zhiqiang Shen, Ammar Sherif, Zeyuan Yin, and Shitong Shao. Delt: A simple diversity-driven earlylate training for dataset distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 4797–4806, 2025.

Donghyeok Shin, Seungjae Shin, and Il-Chul Moon. Frequency domain-based dataset distillation. *Advances in Neural Information Processing Systems*, 36:70033–70044, 2023.

Donghyeok Shin, HeeSun Bae, Gyuwon Sim, Wanmo Kang, and Il-Chul Moon. Distilling dataset into neural field. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Asaf Shul, Eliahu Horwitz, and Yedid Hoshen. Distilling datasets into less than one image. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856.

Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, volume 33, pp. 7462–7473. Curran Associates, Inc., 2020.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for modern deep learning research. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34, pp. 13693–13696. AAAI Press, 2020. doi: 10.1609/aaai.v34i09.7123. URL `https://ojs.aaai.org/index.php/AAAI/article/view/7123`.

Kushal Kardam Vyas, Imtiaz Humayun, Aniket Dashpute, Richard Baraniuk, Ashok Veeraraghavan, and Guha Balakrishnan. Learning transferable features for implicit neural representations. *Advances in Neural Information Processing Systems*, 37:42268–42291, 2024.

Haoxuan Wang, Zhenghao Zhao, Junyi Wu, Yuzhang Shang, Gaowen Liu, and Yan Yan. Cao_2: Rectifying inconsistencies in diffusion-based dataset distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

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

Xiaobo Xia, Jiale Liu, Jun Yu, Xu Shen, Bo Han, and Tongliang Liu. Moderate coreset: A universal method of data selection for real-world data-efficient deep learning. In *The Eleventh International Conference on Learning Representations*, 2022.

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. doi: 10.1109/TPAMI.2023.3323376.

Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 6514–6523, 2023.

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

Xinhao Zhong, Hao Fang, Bin Chen, Xulin Gu, Meikang Qiu, Shuhan Qi, and Shu-Tao Xia. Hierarchical features matter: A deep exploration of progressive parameterization method for dataset distillation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 30462–30471, 2025.

Yawen Zou, Guang Li, Duo Su, Zi Wang, Jun Yu, and Chao Zhang. Dataset distillation via vision-language category prototype. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

A IMPLEMENTATION SETTINGS

A.1 BASELINES

Since our main focus is on the parameterization in dataset distillation, we compare with static decoding: IDC (Kim et al., 2022) and FreD (Shin et al., 2023); and deep generative priors: GLaD (Cazenavette et al., 2023) and H-GLaD (Zhong et al., 2025). Most baseline results, including Vanilla, are directly quoted from the DDiF (Shin et al., 2025) paper. However, for DDiF, we report the results reproduced via the official implementation for the Yellow subset and sub-1 IPC settings. Any other reproduced baseline results are explicitly marked. For completeness, we additionally compare against PoDD (Shul et al., 2025) on CIFAR-10. Table 6 reports this low-resolution comparison, while Appendix C.2 provides a separate ImageNet-Nette study at 128×128, including measured accuracy, time, and memory.

A.2 DATASETS

We use CIFAR-10 (Krizhevsky & Hinton, 2009) (32×32) and ImageNet-Subsets (Cazenavette et al., 2022; 2023) for our experiments. ImageNet-Subsets are evaluated at 128×128 for the main sub-1 IPC and cross-loss experiments, and at 256×256 for the DM resolution-scaling comparison in the main paper. CIFAR-10 is additionally used to compare our approach with PoDD, enabling a direct performance evaluation against this representative sub-1 IPC baseline.

A.3 FEATURE EXTRACTOR FOR CORESET SELECTION

To perform coreset selection in a semantically meaningful space as described in Section 3.4, we first pretrain a feature extractor $\Phi(\cdot)$ on the full real training set and use its intermediate embedding for the selection process. We employ the same ConvNet backbone family as used in the corresponding distillation process (i.e., ConvNetD3 for CIFAR-10, ConvNetD5 for ImageNet-Subsets at 128×128, and ConvNetD6 for the 256×256 DM comparison), and train it for 100 epochs with a batch size of 256 using SGD with an initial learning rate of 0.01, momentum of 0.9, and weight decay of 5×10^{-4}. The learning rate is decayed by a factor of 10 once at the midpoint of training. No additional data augmentation, DiffAugment, Mixup, or CutMix is used in this pretraining stage.

We also considered several alternatives for constructing $\Phi(\cdot)$, including using an ImageNet-1K pretrained model (e.g., ResNet-18) and distilling knowledge from ResNet-18 into the ConvNetD5 feature extractor or its resolution-matched ConvNet variant. In our preliminary trials, these variants did not yield consistent or meaningful improvements for coreset selection, while introducing additional hyperparameters and implementation complexity. We therefore use the simpler resolution-matched ConvNet feature extractor throughout the reported experiments.

After pretraining, all real samples are forwarded through the feature extractor, and the feature vector returned by the embedding layer is extracted in mini-batches. Each feature vector is then ℓ_2-normalized so that distance computations are comparable across samples and classes.

A.4 TRAINING, OPTIMIZATION, AND EVALUATION SETTINGS

For DISC, we perform a 2,000-iteration encoder pretraining followed by a 3,000-iteration warmup (5,000 initialization iterations in total); DDiF uses its default 5,000-iteration initialization. Then, we run 20,000 iterations for Distribution Matching (DM) and 1,000 iterations for Gradient Matching (DC), following DDiF (Shin et al., 2025). By default, we use DM as our dataset distillation loss. We employ Convolutional Network (ConvNet) as the backbone model, using a 3-layer Convolutional Network (ConvNetD3) for CIFAR-10 (32×32), a 5-layer Convolutional Network (ConvNetD5) for ImageNet-Subsets at 128×128, and a 6-layer Convolutional Network (ConvNetD6) for the 256×256 DM comparison in Table 1. For this 256×256 DM setting, we use INR layer size 40; the 128×128 ImageNet-Subsets experiments use layer size 20. We use the Adam optimizer (Kingma & Ba, 2017) for all updates. Unless otherwise stated, we use the **All** encoder-sharing configuration by default in all experiments, where a single encoder is shared by *all* classes. For a detailed ablation study on the impact of different sharing strategies, please refer to Section C.9. For evaluation, we follow the conventional protocol of prior work (Shin et al., 2023; Cazenavette et al., 2023; Shin et al., 2025):

train five randomly initialized classifiers on the optimized synthetic dataset and report Top-1 accuracy. Each synthetic dataset is obtained from a single distillation run, and the reported standard deviations are computed over the five evaluation classifiers (eight evaluation networks for PoDD, following its official protocol). Average columns report the propagated standard deviation $\sqrt{\sum_k \sigma_k^2}/K$ over the K subsets, which reflects evaluation-classifier variation rather than variation across distillation runs. Note that the 256×256 entries in Table 1 are separately distilled and evaluated with ConvNetD6. In contrast, for the cross-resolution generalization experiments, we evaluate synthetic data distilled at 128×128 resolution (using ConvNetD5) on higher-resolution architectures: ConvNetD6 for 256×256 and ConvNetD7 for 512×512.

A.5 Hyperparameter settings

A.5.1 Implicit shared representations.

Table 7 summarizes the hyperparameters for the Implicit Shared Representations. While the specific architecture of the Implicit Shared Representations varies by dataset, the general hyperparameter settings remain consistent across methods.

In this context, *layer size* refers to the hidden width of the Implicit Neural Representations, i.e., the number of units in each hidden layer. This hyperparameter controls the representational capacity of both the shared encoder and the instance-specific decoders. More importantly for budgeting, once the layer size is fixed for a dataset and resolution setting, the number of parameters contributed by each additional hidden layer is also fixed. Accordingly, the total parameter count is determined by the number of hidden layers together with the mandatory boundary layers that connect the input and output spaces. For ImageNet-Subsets, we use layer size 20 at 128×128 and layer size 40 at 256×256. The resulting budget formulation is detailed in Appendix A.6.

Table 7: Implicit Shared Representations settings for each dataset. Hidden Layers is the total per instance, $E + D$; Shared Encoder Layers is E, and each decoder has the remaining D (one by default).

Dataset	Hidden Layers	Shared Encoder Layers	Layer Size
ImageNet-Subsets (128×128)	4	3	20
ImageNet-Subsets (256×256)	4	3	40
CIFAR-10	3	2	6

A.5.2 Distribution Matching (DM).

Table 8: Model and optimization settings for DM experiments.

Dataset	Batch size	Learning rate (Neural Field)
ImageNet (128^2)	64	5e-5
ImageNet (256^2)	64	1e-5
CIFAR-10	256	5e-4

Table 9: Model and optimization settings for DC experiments.

Dataset	Batch size	Learning rate (Neural Field)
ImageNet (128^2)	64	5e-5

Table 8 summarizes the optimization settings for the DM experiments. We use a single NVIDIA RTX 3090 GPU and run the distillation for 20,000 iterations.

A.5.3 Gradient Matching (DC).

Table 9 summarizes the optimization settings for the DC experiments. For DISC, we use a single NVIDIA RTX 3090 GPU and run the distillation for 1,000 iterations.

Table 10: Per-dataset parameter atoms used for budgeting. $b_{\text{enc}}/b_{\text{dec}}$: mandatory boundary-layer cost. Totals are $P_{\text{enc}} = b_{\text{enc}} + p_{\text{enc}} \times E$ and $P_{\text{dec}} = b_{\text{dec}} + p_{\text{dec}} \times D$.

Dataset	b_{enc}	p_{enc}	b_{dec}	p_{dec}
ImageNet-Subsets (128×128)	60	420	63	420
ImageNet-Subsets (256×256)	120	1,640	123	1,640
CIFAR-10	18	42	21	42

Table 11: Number of decoders per class ($N_{\text{dec/class}}$), architecture settings, and total parameter budgets (B_{total}) for ImageNet-Subsets at 128×128 resolution in DDiF and DISC (Ours) across different IPCs. Hidden layers count width-to-width layers, excluding the input and output layers; DDiF's $L = 3$ (Shin et al., 2025) corresponds to 2 here. For DISC, the four hidden layers split into $E = 3$ shared-encoder layers and $D = 1$ decoder layer (Appendix A.6).

IPC	Method	$N_{\text{dec/class}}$	Hidden Layers	Shared Encoder Layers	B_{total}
1	DDiF	51	2	-	491,520
	DISC (Ours)	101	4	3	
0.5	DDiF	25	2	-	245,760
	DISC (Ours)	50	4	3	
0.25	DDiF	12	2	-	122,880
	DISC (Ours)	25	4	3	
0.1	DDiF	5	2	-	49,152
	DISC (Ours)	9	4	3	

A.6 Parameter budgeting

Table 10 decomposes the parameter count of the shared encoder and decoder into dataset-specific atoms. Here, b_{enc} denotes the mandatory boundary-layer cost of the encoder, namely the parameter cost required to map the input coordinate representation into the encoder hidden space. Likewise, b_{dec} denotes the mandatory boundary-layer cost of the decoder, i.e., the parameter cost required to map the decoder hidden representation to the RGB output. The symbols p_{enc} and p_{dec} represent the incremental parameter cost contributed by one additional hidden layer in the encoder and decoder, respectively. Since the layer size is fixed for each dataset, each extra hidden layer adds a fixed number of parameters, which is exactly what p_{enc} and p_{dec} capture.

Using these atoms, the total parameter counts of one shared encoder and one decoder are written as $P_{\text{enc}} = b_{\text{enc}} + p_{\text{enc}} \times E$ and $P_{\text{dec}} = b_{\text{dec}} + p_{\text{dec}} \times D$, where E and D denote the numbers of hidden layers used in the encoder and decoder, respectively. Thus, P_{enc} is the total number of trainable parameters allocated to the shared encoder, and P_{dec} is the total number of trainable parameters allocated to a single instance-specific decoder.

For example, on ImageNet-Subsets at 128×128, a 1-layer shared encoder has $P_{\text{enc}} = 60 + 420 \times 1 = 480$ parameters, whereas on CIFAR-10 (32×32) it becomes $18 + 42 \times 1 = 60$. These quantities are then used in the overall budget calculation B_{total} and in deriving the number of decoders per class, $N_{\text{dec/class}}$.

A.6.1 ImageNet-Subsets.

We evaluate DISC on ImageNet-Subsets (Cazenavette et al., 2022; 2023) to validate its performance in large-scale, high-dimensional scenarios where parameter efficiency is crucial. Table 11 compares decoder count, architecture settings, and total parameters between DDiF and DISC at 128×128 across different IPCs.

Table 12: Number of decoders per class ($N_{\text{dec/class}}$), architecture settings, and total parameter budgets (B_{total}) for CIFAR-10 in DISC (Ours) across different IPCs. Layer counts follow Table 11.

IPC	Method	$N_{\text{dec/class}}$	Hidden Layers	Shared Encoder	B_{total}
0.5	**DISC (Ours)**	24	3	2	15,360
0.4	**DISC (Ours)**	19	3	2	12,288
0.3	**DISC (Ours)**	14	3	2	9,216
0.2	**DISC (Ours)**	9	3	2	6,144
0.1	**DISC (Ours)**	4	3	2	3,072

A.6.2 CIFAR-10.

In addition to ImageNet-Subsets, we evaluate DISC on CIFAR-10 (Krizhevsky & Hinton, 2009) (32×32) under the sub-1 IPC storage budgets reported by PoDD (Shul et al., 2025). Table 12 presents the decoder count and parameter budgets across different IPCs.

A.6.3 Raw storage budgets.

The IPC axis used throughout the paper is a storage budget expressed in parameter slots: $B_{\text{total}}(\text{IPC}) = C \times r^2 \times \text{IPC} \times K$ (Eq. 1), exactly the number of values occupied by the corresponding raw images. Every method stores 32-bit floating-point parameters counted against the same B_{total}, so IPC converts to raw bytes by a factor of four. For ImageNet-Subsets at 128×128 (10 classes), IPC 1.0 therefore corresponds to 491,520 values (1,966,080 B, ≈2.0 MB), and IPC 0.5 / 0.25 / 0.1 to 983,040 B / 491,520 B / 196,608 B; at 256×256 the IPC 1.0 budget is 1,966,080 values (7,864,320 B). For CIFAR-10, IPC 0.5 / 0.4 / 0.3 / 0.2 / 0.1 correspond to 61,440 B / 49,152 B / 36,864 B / 24,576 B / 12,288 B. In byte terms, one DDiF field costs 3,852 B (963 parameters) whereas one DISC decoder costs 1,932 B (483 parameters), with the shared encoder costing a one-time 5,280 B (1,320 parameters, 0.27% of the IPC 1.0 budget); this pricing is what lets DISC allocate roughly twice as many instances per class within the same byte budget (Table 11). PoDD's poster on ImageNet-Nette at 128×128 holds exactly 491,520 values as well, so the comparison in Appendix C.2 is budget-matched by construction.

B Algorithm

Algorithms 1–3. We illustrate the training pipeline of DISC as described in Section 3: **(1)** encoder pretraining using a median-distance coreset ($\mathcal{S}_{\text{med}}$) to cover non-outlier within-class variation, **(2)** warmup of the encoder and decoders on a prototype-nearest coreset ($\mathcal{S}_{\text{near}}$) via reconstruction, and **(3)** dataset distillation via matching loss (DM/DC). Notably, the shared encoder parameters are frozen after the warmup and remain fixed during the distillation.

Algorithm 1 Pretraining. Encoder Pretraining with Median-Distance Coreset

Input: Real dataset $\mathcal{X} = \bigcup_{k=1}^{K} \mathcal{X}^{(k)}$; Number of classes K; Feature extractor $\Phi(\cdot)$; Median-coreset size M_{med} per class; Pretraining iterations T_{pre}; Learning rate η_{pre}; Fixed coordinate grid $\mathcal{G}$.

Output: Pretrained shared encoder parameters θ^0_{enc}.

for $k = 1$ **to** K **do**
  Compute class prototype μ_k and distances $d_i^{(k)} = \|\Phi(x_i^{(k)}) - \mu_k\|_2$.
  Select median-distance coreset $\mathcal{S}_{\text{med}}^{(k)} \subset \mathcal{X}^{(k)}$ of size M_{med} (see Eq. (5)).
end for
Form the pretraining set $\mathcal{S}_{\text{med}} \leftarrow \bigcup_{k=1}^{K} \mathcal{S}_{\text{med}}^{(k)}$.
Instantiate a temporary shared INR with $|\mathcal{S}_{\text{med}}|$ decoders: $(\theta_{\text{enc}}, \{\psi_j\}_{j=1}^{|\mathcal{S}_{\text{med}}|})$.
for $e = 1$ **to** T_{pre} **do**
  Compute reconstruction loss $\mathcal{L}_{\text{rec}}$ on $\mathcal{S}_{\text{med}}$ (MSE over all decoders).
  Update all temporary parameters via Adam with lr η_{pre}.
end for
Extract and save encoder weights: $\theta^0_{\text{enc}} \leftarrow \theta_{\text{enc}}$.

Algorithm 2 Final Coreset. Warmup Initialization with Prototype-Nearest Coreset

Input: Pretrained encoder θ^0_{enc} (from Pretraining); Real dataset $\mathcal{X}$; Number of classes K; Feature extractor $\Phi(\cdot)$; Decoders per class $N_{\text{dec/class}}$; Warmup iterations T_{warm}; Learning rate η_{warm}; Fixed coordinate grid $\mathcal{G}$.

Output: Warmed-up neural field parameters $\Psi^0 = \{\theta_{\text{enc}}, \{\psi_j^{(k)}\}\}$.

for $k = 1$ **to** K **do**
  Compute prototype distances $\{d_i^{(k)}\}$ in feature space.
  Select prototype-nearest coreset $\mathcal{S}_{\text{near}}^{(k)} \subset \mathcal{X}^{(k)}$ of size $N_{\text{dec/class}}$ (see Eq. (6)).
end for
Form the initialization set $\mathcal{S}_{\text{near}} \leftarrow \bigcup_{k=1}^{K} \mathcal{S}_{\text{near}}^{(k)}$.
Construct the DISC architecture with $K \cdot N_{\text{dec/class}}$ decoders.
Initialize encoder with θ^0_{enc} and decoders randomly.
for $e = 1$ **to** T_{warm} **do**
  Compute reconstruction loss $\mathcal{L}_{\text{rec}}$ on $\mathcal{S}_{\text{near}}$ ▷ Eq. (8)
  Update encoder and decoders via Adam with lr η_{warm}.
end for
$\Psi^0 \leftarrow \{\theta_{\text{enc}}, \{\psi_j^{(k)}\}\}$.
Freeze encoder parameters θ_{enc} for the subsequent distillation stage.

Algorithm 3 Distillation. Matching Distillation (Optimizing Decoders Only)

Input: Warmed-up parameters Ψ^0 (with frozen encoder); Real dataset $\mathcal{X}$; Total iterations T_{max}; Outer-loop steps K_{out}; Inner-loop steps K_{in}; Real batch size B_{real}; Synthetic batch size B_{syn}; Fixed coordinate grid $\mathcal{G}$.
Output: Optimized synthetic parameters Ψ^*; Synthetic dataset $\mathcal{S}_{\text{syn}}$.

for $t = 1$ **to** T_{max} **do**
  Sample a randomly initialized target network f_θ and optimizer.
  for $o = 1$ **to** K_{out} **do**
    Sample real batch $(x_r, y_r) \sim \mathcal{X}$ of size B_{real}.
    Generate synthetic batch (x_s, y_s) from Ψ^0 evaluated on $\mathcal{G}$ (size B_{syn}).
    Compute matching loss $\mathcal{L}_{\text{dis}}$ (e.g., DM or DC) ▷ Eq. (9)
    Update active parameters (decoders $\{\psi_j^{(k)}\}$) via Adam.
    if $o < K_{\text{out}}$ **then**
      Update f_θ for K_{in} steps on the current synthetic data.
    end if
  end for
end for
$\Psi^* \leftarrow \Psi^0$
$\mathcal{S}_{\text{syn}} \leftarrow \{(F_{\theta_{\text{enc}},\, \psi_j^{(k)}}(\mathcal{G}),\, y^{(k)})\}_{k,j}$

C Additional experimental results

C.1 Synthesis time and memory cost

Table 13: Synthesis cost measured on ImageNet-Nette at 128×128 and IPC 1.0, with every pipeline stage accounted for and both methods run under the identical protocol and seed on the same node. **Top**: per-stage wall-clock under DM (20,000 iterations). The feature extractor is a one-time, per-dataset cost reused across all IPC settings, objectives, and seeds ($\geq$10 runs per subset in this paper; amortized $\leq$0.34 h per run). **Bottom**: DC (1,000 iterations); DISC backpropagates the matching loss per class, which keeps its peak memory within a single 24 GB card; the official DDiF implementation uses two GPUs at this resolution, and the same restructuring could also be applied to it.

DM, 128×128	**DISC**	**DDiF**
Feature extractor (one-time)	3.36 h	–
Coreset selection	0.03 h	–
Encoder pretraining	0.20 h	–
Warmup / SIREN init	0.31 h	1.20 h
Distillation	3.36 h	4.21 h
Hardware	1× RTX 3090	1× RTX 3090
Synthesis (init + distill)	**3.90 h**	5.41 h
Peak memory (distill)	4.8 GB	4.5 GB
DC, 128×128	**DISC**	**DDiF**
Hardware	1× RTX 3090	2× RTX 3090
Synthesis (init + distill)	**1.25 h**	1.69 h
Peak memory (distill)	7.2 GB	19.7 GB

All measurements in this subsection use ImageNet-Nette at IPC 1.0, at 128×128 unless noted otherwise. We report the full cost of *synthesizing* the distilled dataset, accounting for every pipeline stage, and treat downstream evaluation separately (see below). Table 13 summarizes wall-clock time and peak memory, with both methods instrumented under the identical protocol and seed on the same node.

Under DM, DISC completes initialization and distillation in 3.90 h against DDiF's 5.41 h ($1.39\times$ faster; the ratio is 1.35–$1.41\times$ across the three node types we measured). The gap comes from initialization: DDiF fits an independent SIREN per instance (1.20 h), whereas DISC warms up its decoders together with an already pretrained shared encoder (0.31 h, with the encoder pretraining itself costing 0.20 h). The feature extractor used for coreset selection is a one-time, per-dataset cost: it is reused across all IPC settings, both objectives, and all seeds ($\geq$10 runs per subset in this paper), so its amortized cost is at most 0.34 h per run; even a fully cold-start single run (7.26 h) is only moderately above DDiF, and cumulative DISC time is lower from the third run on a dataset. Peak distillation memory is comparable (4.8 vs. 4.5 GB), and both fit well within a single 24 GB GPU.

Under DC, DISC backpropagates the matching loss per class instead of accumulating a single autograd graph over all classes, which keeps its peak memory at 7.2 GB. DISC thus completes DC synthesis in 1.25 h on a single RTX 3090, while the official DDiF implementation takes 1.69 h on two GPUs. This difference reflects implementation rather than parameterization: an accumulated-graph implementation of either method exceeds a single 24 GB card at this resolution, and the same per-class restructuring could equally be applied to DDiF; we report DDiF with its official implementation.

At 256×256 (DM, ConvNetD6, layer size 40), both methods run on a single RTX 3090. DISC initializes in 0.76 h (encoder pretraining 0.12 h and warmup 0.64 h) against 1.15 h for DDiF's per-instance SIREN fitting, and distills at 2.34 versus 2.60 s per iteration, with peak distillation memory of 19.5 versus 20.2 GB.

Evaluation cost scales linearly with the number of decoded images and is therefore reported separately from synthesis: DISC decodes 1,010 images versus DDiF's 510 at the same byte budget (this $2\times$ is precisely the parameter-efficiency claim), so training a downstream classifier on the decoded set costs

proportionally more. This is a property of evaluating more images under a fixed protocol, not of the synthesis procedure.

C.2 High-resolution comparison with PoDD

Beyond the CIFAR-10 comparison in the main paper, we study PoDD (Shul et al., 2025) at higher resolution using its provided implementation on ImageNet-Nette. In short, under its official configuration, PoDD's unrolled optimization incurs a substantial cost at high resolution: at 128×128 it nearly fills a 48 GB GPU and its distillation updates alone take about $30\times$ DISC's full cold-start synthesis time. At 128×128, we completed the 6,000-epoch schedule with an unroll window of 30 and a total window of 80. The $640 \times 256 \times 3$ poster contains 491,520 values, exactly matching the total storage budget assigned to IPC 1.0 in our tables. We report this study separately because it also measures the computation and memory required to scale PoDD beyond CIFAR-10. At 256×256, with the poster scaled to the IPC 1.0 budget and the same configuration, PoDD runs out of memory even on a single 80 GB GPU.

Memory. PoDD's unrolled optimization keeps the poster, differentiable parameter updates, and optimizer state resident. The measured per-process peak at 128×128 was 48,328 MiB (47.2 GiB, 98.4% of the usable capacity) on a 48 GB RTX A6000. Two independent measurements of the same configuration on 80 GB H100s reached 53,956 and 54,246 MiB (52.7 and 53.0 GiB).

Time and accuracy. The completed schedule consumed 360.3 GPU-hours (15.0 days) on the A6000, including PoDD's scheduled evaluation blocks. At the measured training-step rate, the distillation updates alone account for approximately 215 GPU-hours (9.0 days), about $30\times$ DISC's 7.26-hour cold-start synthesis cost. PoDD's best three-classifier block mean was 55.7% at epoch 2,900, and its mean over the final 20 evaluation blocks was 53.1%. DISC reaches 69.6% under DM at the same byte budget and on the same dataset. The PoDD figure selects the best of 480 three-classifier evaluation blocks, whereas the DISC result is a five-classifier mean, so these accuracy values provide a measured high-resolution reference rather than a paired protocol comparison. The tabulated PoDD comparison therefore remains the CIFAR-10 result in the main paper.

C.3 Detailed ablations of coreset selection and instance allocation

C.3.1 Coreset selection at IPC 1.0.

Table 14 expands the coreset ablation of the main paper with per-subset results and two additional two-stage variants, *Median + Random* and *Median + Farthest*, which keep median-distance pretraining and change only the warmup set. With the pretraining set fixed, prototype-nearest warmup (55.4%) outperforms random (53.0%) and farthest (47.8%) warmup, so the choice of warmup samples matters on its own. We further detail the two controls summarized in the main paper.

First, a fully uninformed pipeline (*Random + Random*), which keeps the two-stage structure but selects both the encoder-pretraining set and the warmup set at random, reaches an average of 52.0%, essentially the same as the best single-stage DISC variant (51.9%). Informed selection contributes a further +3.4pp within the two-stage pipeline (52.0 $\rightarrow$ 55.4).

Second, swapping the roles of the two selection rules (*Proto-Nearest + Median*: prototype-nearest samples for encoder pretraining, median-distance samples for warmup) drops the average from 55.4% to 52.5%, close to the fully random pipeline (52.0%). The assignment of rules to stages therefore matters: the best configuration uses median-distance samples for pretraining and prototype-nearest samples for warmup.

C.3.2 Allocation and selection at IPC 0.5.

Table 15 repeats a subset of the coreset and allocation ablation at IPC 0.5, the center of the sub-1 regime this paper targets, adding the components of our pipeline one at a time at an identical byte budget and protocol. The resulting ladder is monotone: DDiF 47.8 $\rightarrow$ DDiF with prototype-nearest initialization 48.7 $\rightarrow$ DISC with fully random selection 50.8 $\rightarrow$ random pretraining with prototype-nearest warmup 51.6 $\rightarrow$ the full method 52.4. Three observations follow. First, prototype-nearest selection transfers to DDiF but adds only +0.9pp in this control. Second, without informed selection, the shared allocation alone scores +3.0pp above DDiF at the same bytes, reflecting the larger instance

Table 14: Detailed ablation results on coreset subset selection strategies at IPC 1.0 across ImageNet-Subsets (128×128). Results are reported as mean $\pm$ std accuracy (%) over the five evaluation classifiers. † marks reproduced DDiF results; on a row label it applies to the whole row.

Selection stage	Method	Subset selection	Subset						
			Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
1-stage	DDiF	Random	69.2 ±1.0	42.0 ±0.4	45.3 ±1.8	62.4† ±1.6	45.8 ±1.1	64.6 ±1.1	54.9 ±0.5
		Farthest†	65.8 ±0.9	37.2 ±0.5	40.6 ±1.2	61.4 ±1.8	41.2 ±1.1	59.8 ±0.8	51.0 ±0.5
		Median†	69.5 ±1.8	39.7 ±1.3	42.9 ±1.5	62.1 ±0.7	44.3 ±2.1	64.8 ±1.5	53.9 ±0.6
		Proto-Nearest†	67.1 ±0.7	38.5 ±0.6	42.2 ±1.8	62.1 ±1.1	44.8 ±1.2	64.0 ±0.3	53.1 ±0.4
	DISC (Ours)	Random	63.2 ±0.7	38.5 ±1.2	37.6 ±1.6	57.0 ±1.6	37.8 ±0.7	59.1 ±0.9	48.9 ±0.5
		Farthest	62.0 ±0.9	35.3 ±1.7	38.6 ±0.6	57.5 ±0.5	37.2 ±0.5	59.1 ±0.3	48.3 ±0.4
		Median	65.8 ±0.3	38.6 ±1.2	41.0 ±1.9	59.7 ±1.5	41.2 ±0.9	65.0 ±1.0	51.9 ±0.5
		Proto-Nearest	64.1 ±0.4	37.8 ±1.3	42.9 ±0.2	57.5 ±1.7	38.6 ±1.2	59.4 ±0.9	50.1 ±0.4
2-stage	**DISC (Ours)**	Random + Proto-Nearest	68.3 ±1.0	41.4 ±1.3	43.9 ±1.2	63.6 ±0.5	41.6 ±1.0	64.9 ±1.0	54.0 ±0.4
		Farthest + Proto-Nearest	69.8 ±0.9	40.7 ±1.4	42.5 ±1.0	64.0 ±1.4	41.0 ±1.1	66.2 ±0.4	54.0 ±0.4
		Median + Proto-Nearest	69.6 ±0.5	42.4 ±0.5	45.5 ±0.6	63.9 ±0.5	44.0 ±1.1	67.0 ±1.1	**55.4** ±0.3
		Median + Random	66.9 ±0.7	38.8 ±1.3	42.3 ±1.4	63.2 ±1.0	41.5 ±1.3	65.1 ±1.2	53.0 ±0.5
		Median + Farthest	63.0 ±0.5	36.5 ±1.7	37.6 ±0.9	56.4 ±1.3	36.4 ±1.3	57.0 ±1.4	47.8 ±0.5
		Random + Random	68.0 ±0.8	39.5 ±1.8	40.6 ±1.7	60.3 ±0.9	39.4 ±1.6	64.5 ±0.9	52.0 ±0.5
		Proto-Nearest + Median	67.1 ±0.9	38.4 ±1.7	42.9 ±0.9	61.4 ±0.9	39.6 ±1.5	65.6 ±0.8	52.5 ±0.5

Table 15: Component attribution at IPC 0.5 under DM on ImageNet-Subsets (128×128), with identical parameter budget and protocol in every row. Results are reported as mean $\pm$ std accuracy (%) over the five evaluation classifiers. The DDiF Random and DISC Median + Proto-Nearest rows are the corresponding main-table cells; since DDiF has no encoder-pretraining stage, only its warmup initialization can be varied. The *Encoder control* row is not a DISC configuration: it keeps DISC's 50 decoders per class and prototype-nearest warmup but replaces the trained shared encoder with a randomly initialized, frozen one, to test whether the additional decoders help without a learned encoder.

Method	Subset selection	Subset						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DDiF	Random	63.2 ±0.8	34.4 ±2.0	38.4 ±0.6	55.6 ±1.0	39.3 ±1.7	55.7 ±1.0	47.8 ±0.5
	Proto-Nearest	63.1 ±1.5	36.8 ±2.1	37.6 ±1.3	56.3 ±0.6	41.6 ±0.8	56.5 ±1.3	48.7 ±0.6
DISC (Ours)	Random + Random	65.0 ±0.5	37.2 ±2.1	40.0 ±1.2	59.8 ±1.4	39.4 ±1.7	63.2 ±1.1	50.8 ±0.6
	Random + Proto-Nearest	65.9 ±1.2	37.3 ±1.2	42.8 ±1.5	**60.2** ±0.8	40.8 ±1.2	62.5 ±0.7	51.6 ±0.5
	Median + Proto-Nearest	**66.4** ±1.3	**38.8** ±1.2	**43.7** ±1.8	60.0 ±1.0	**41.8** ±2.3	**63.4** ±1.8	**52.4** ±0.7
Encoder control	Untrained encoder (frozen) + Proto-Nearest	58.7 ±1.0	33.0 ±1.0	38.4 ±1.0	53.1 ±1.4	37.0 ±0.8	54.1 ±0.7	45.7 ±0.4

count it affords. Third, informed two-stage selection contributes a further +1.6pp with the same DISC allocation and decoder count. A further control isolates the encoder: at the same IPC 0.5 budget, replacing the trained shared encoder with a randomly initialized, frozen one (still 50 decoders per

class with prototype-nearest warmup; *Encoder control* in Table 15) lowers accuracy from 52.4% to 45.7%, below DDiF's 47.8%, so the additional instances pay off only when the shared encoder is learned. At IPC 1.0 the fully random DISC configuration scores 52.0 against DDiF's 54.9 (Table 14), so packing more instances alone does not guarantee a gain. The relative contribution of allocation and selection therefore depends on the budget.

C.4 Budget scaling and degradation

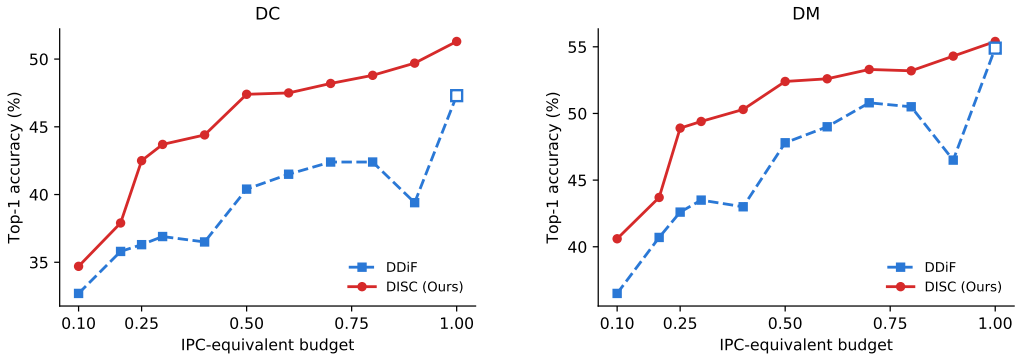


Figure 3: Six-subset mean Top-1 accuracy of DDiF and DISC on ImageNet-Subsets (128×128) against the IPC-equivalent byte budget, from IPC 0.1 up to 1.0, using DC (left) and DM (right). Values are taken from the main and fine-grained tables. The DDiF point at IPC 1.0, marked with the open symbol, uses published values except for the reproduced Yellow subset. Decoder counts convert to IPC via the parameter atoms of Appendix A.6.

We analyze how both methods behave as the byte budget scales, sweeping the IPC axis from 1.0 down to 0.1 under both DC and DM. Figure 3 plots the full trend (Table 4 in the main paper summarizes representative budgets), with 0.1 increments below IPC 1.0 plus the IPC 0.25 point; the complete per-step tables with standard deviations are given in Tables 16 and 17.

Three trends emerge. First, DISC is above DDiF at every plotted budget under both losses, and its average margin over the sub-1 budgets exceeds its margin at IPC 1.0 (DM: $+4.6$ vs. $+0.5$pp; DC: $+6.0$ vs. $+4.0$pp; Table 18). Second, DISC degrades gracefully below IPC 1.0: it loses only 1.1pp (DM) and 1.6pp (DC) at IPC 0.9 and then declines smoothly down to IPC 0.1, keeping a clear margin over DDiF throughout the sub-1 regime. Third, the advantage also extends above IPC 1.0: DISC at IPC 1.0 already exceeds DDiF operating at roughly twice the budget under both losses (Appendix C.8). The advantage of the co-design is therefore larger on average in the sub-1 regime this paper targets than at IPC 1.0, but is not confined to that regime.

Tables 16 and 17 give the per-subset results behind Figure 3. DISC has the higher average accuracy at every IPC level under both losses (e.g., 40.6% vs. 36.5% under DM at IPC 0.1), and under DC it surpasses DDiF on *every* subset at *every* IPC level.

Table 16: Top-1 accuracy (%) of DDiF and DISC on six ImageNet-Subsets (128×128) as the IPC budget decreases from 1.0 to 0.1, with the training loss fixed to DM. Bold highlights the better result for each (IPC, subset) pair; the rightmost column gives the average across subsets with propagated standard deviation.

IPC	Method	Subset						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Average
1.0	DDiF	69.2 ±1.0	42.0 ±0.4	45.3 ±1.8	62.4 ±1.6	**45.8** ±1.1	64.6 ±1.1	54.9 ±0.5
	DISC (Ours)	**69.6** ±0.5	**42.4** ±0.5	**45.5** ±0.6	**63.9** ±0.5	44.0 ±1.1	**67.0** ±1.1	**55.4** ±0.3
0.9	DDiF	63.0 ±1.0	30.3 ±1.7	38.1 ±1.7	55.7 ±2.2	36.8 ±1.5	55.2 ±1.5	46.5 ±0.7
	DISC (Ours)	**69.8** ±0.9	**40.2** ±1.4	**44.7** ±2.1	**63.9** ±1.3	**42.2** ±1.8	**65.1** ±0.8	**54.3** ±0.6
0.8	DDiF	65.4 ±1.2	37.9 ±2.1	40.6 ±1.3	58.9 ±1.4	40.7 ±1.4	59.6 ±1.3	50.5 ±0.6
	DISC (Ours)	**67.8** ±1.1	**38.9** ±2.4	**44.9** ±0.9	**62.1** ±0.7	**41.1** ±0.6	**64.7** ±1.1	**53.2** ±0.5
0.7	DDiF	65.1 ±1.1	38.2 ±1.6	42.3 ±0.4	58.7 ±1.2	40.5 ±0.7	60.0 ±1.0	50.8 ±0.4
	DISC (Ours)	**67.9** ±1.3	**39.5** ±1.2	**43.2** ±1.8	**61.9** ±0.6	**42.2** ±0.6	**65.0** ±0.5	**53.3** ±0.5
0.6	DDiF	62.2 ±0.8	35.1 ±0.6	41.1 ±0.9	57.9 ±0.4	**40.4** ±1.4	57.3 ±0.9	49.0 ±0.4
	DISC (Ours)	**67.0** ±0.8	**39.9** ±1.0	**42.5** ±1.0	**60.8** ±1.5	**40.4** ±0.6	**65.0** ±1.1	**52.6** ±0.4
0.5	DDiF	63.2 ±0.8	34.4 ±2.0	38.4 ±0.6	55.6 ±1.0	39.3 ±1.7	55.7 ±1.0	47.8 ±0.5
	DISC (Ours)	**66.4** ±1.3	**38.8** ±1.2	**43.7** ±1.8	**60.0** ±1.0	**41.8** ±2.3	**63.4** ±1.8	**52.4** ±0.7
0.4	DDiF	58.9 ±0.8	30.8 ±1.4	33.5 ±1.6	51.9 ±0.7	34.0 ±0.9	48.7 ±1.2	43.0 ±0.5
	DISC (Ours)	**65.1** ±1.1	**36.2** ±0.5	**41.8** ±1.0	**58.8** ±0.6	**39.4** ±2.2	**60.2** ±1.3	**50.3** ±0.5
0.3	DDiF	58.9 ±1.1	31.0 ±0.3	37.0 ±1.7	51.9 ±0.6	34.0 ±1.2	48.5 ±0.5	43.5 ±0.4
	DISC (Ours)	**64.1** ±0.6	**36.4** ±0.7	**40.6** ±1.2	**58.2** ±1.0	**38.7** ±0.7	**58.7** ±0.7	**49.4** ±0.3
0.2	DDiF	54.8 ±1.6	28.4 ±0.7	35.3 ±1.0	49.4 ±1.7	31.8 ±1.5	44.6 ±1.2	40.7 ±0.5
	DISC (Ours)	**59.1** ±1.1	**31.3** ±1.5	**36.7** ±1.2	**50.9** ±1.3	**33.4** ±2.0	**50.8** ±1.3	**43.7** ±0.6
0.1	DDiF	50.6 ±0.9	28.4 ±1.5	30.0 ±1.1	45.2 ±1.2	28.4 ±1.2	36.7 ±1.3	36.5 ±0.5
	DISC (Ours)	**54.8** ±1.4	**29.3** ±1.5	**33.8** ±1.8	**48.7** ±1.1	**31.8** ±1.3	**45.1** ±1.5	**40.6** ±0.6

Table 17: Top-1 accuracy (%) of DDiF and DISC on six ImageNet-Subsets (128×128) as the IPC budget decreases from 1.0 to 0.1, with the training loss fixed to DC. Bold highlights the better result for each (IPC, subset) pair; the rightmost column gives the average across subsets with propagated standard deviation.

IPC	Method	Subset						
		Nette	Woof	Fruit	Yellow	Meow	Squawk	Average
1.0	DDiF	61.2 ±1.0	35.2 ±1.7	37.8 ±1.1	56.0 ±1.0	39.1 ±1.3	54.3 ±1.0	47.3 ±0.5
	DISC (Ours)	**66.1** ±0.6	**41.4** ±0.9	**41.6** ±0.8	**60.2** ±1.5	**41.5** ±0.6	**56.7** ±0.5	**51.3** ±0.4
0.9	DDiF	54.9 ±1.6	27.8 ±0.3	32.3 ±1.0	50.0 ±1.5	31.0 ±1.0	40.2 ±1.6	39.4 ±0.5
	DISC (Ours)	**65.2** ±1.4	**39.6** ±0.8	**41.2** ±0.8	**58.4** ±1.0	**40.0** ±1.8	**53.6** ±1.2	**49.7** ±0.5
0.8	DDiF	60.5 ±0.6	32.5 ±0.7	33.2 ±1.1	52.2 ±2.1	34.7 ±1.1	41.4 ±1.4	42.4 ±0.5
	DISC (Ours)	**63.5** ±1.0	**40.2** ±1.6	**39.9** ±0.6	**58.2** ±0.6	**38.5** ±1.1	**52.7** ±0.8	**48.8** ±0.4
0.7	DDiF	59.8 ±0.7	31.8 ±1.1	35.0 ±1.2	51.1 ±0.9	33.8 ±0.6	42.6 ±0.8	42.4 ±0.4
	DISC (Ours)	**62.9** ±0.8	**39.0** ±0.6	**40.6** ±0.9	**56.7** ±0.8	**38.0** ±1.4	**52.0** ±1.4	**48.2** ±0.4
0.6	DDiF	57.6 ±0.8	30.3 ±1.6	34.4 ±1.4	52.7 ±1.0	34.5 ±0.6	39.7 ±0.6	41.5 ±0.4
	DISC (Ours)	**62.5** ±1.0	**36.7** ±1.5	**39.6** ±1.0	**56.4** ±1.0	**39.1** ±0.3	**50.8** ±0.8	**47.5** ±0.4
0.5	DDiF	56.1 ±1.2	31.6 ±1.7	32.8 ±1.0	48.7 ±0.7	32.6 ±0.8	40.4 ±0.3	40.4 ±0.4
	DISC (Ours)	**61.1** ±1.2	**38.5** ±0.6	**40.5** ±0.7	**56.6** ±1.2	**38.1** ±1.5	**49.7** ±0.9	**47.4** ±0.4
0.4	DDiF	49.5 ±1.8	28.8 ±1.1	28.2 ±1.7	47.3 ±1.3	30.1 ±0.5	35.2 ±0.8	36.5 ±0.5
	DISC (Ours)	**57.8** ±0.8	**36.4** ±2.1	**36.8** ±0.8	**52.8** ±0.5	**36.5** ±1.0	**46.2** ±1.1	**44.4** ±0.5
0.3	DDiF	48.9 ±1.0	30.2 ±0.8	29.5 ±1.3	47.7 ±0.3	29.1 ±1.2	35.9 ±0.3	36.9 ±0.4
	DISC (Ours)	**58.2** ±1.2	**36.7** ±0.9	**37.1** ±0.8	**52.0** ±1.2	**34.0** ±1.3	**44.0** ±1.4	**43.7** ±0.5
0.2	DDiF	48.3 ±0.7	28.2 ±1.6	30.1 ±1.0	46.7 ±0.3	28.5 ±0.9	33.2 ±0.8	35.8 ±0.4
	DISC (Ours)	**52.3** ±2.1	**28.4** ±0.8	**31.2** ±1.1	**47.4** ±1.3	**29.5** ±0.8	**38.4** ±0.6	**37.9** ±0.5
0.1	DDiF	42.3 ±0.6	24.2 ±0.9	25.7 ±1.7	44.7 ±0.8	28.3 ±1.0	31.0 ±1.0	32.7 ±0.4
	DISC (Ours)	**45.2** ±1.2	**27.4** ±1.0	**27.7** ±0.7	**45.1** ±0.7	**28.9** ±1.1	**33.7** ±0.5	**34.7** ±0.4

C.5 PAIRED SIGNIFICANCE TESTS ACROSS SUBSETS

Table 18: Paired significance tests of the DISC−DDiF margin across the six ImageNet-Subsets at each IPC level, computed from the fine-grained results in Tables 16 and 17 (pairing unit = subset, $n = 6$). We report the mean margin Δ (pp), its 95% confidence interval, the two-sided paired t-test p-value ($\mathrm{df} = 5$), the number of subsets where DISC is ahead, and the exact two-sided sign-test p-value (whose floor at $n = 6$ is 0.031). The pooled sub-1 row aggregates all 54 cells with IPC < 1.0 as a descriptive summary.

DM						DC					
IPC	Δ (pp)	**95% CI**	p (t)	**Wins**	p (sign)	**IPC**	Δ (pp)	**95% CI**	p (t)	**Wins**	p (sign)
1.0	+0.52	[−0.97, +2.00]	0.412	5/6	0.219	1.0	+3.98	[+2.44, +5.53]	0.0012	6/6	0.031
0.9	+7.80	[+5.85, +9.75]	0.0001	6/6	0.031	0.9	+10.30	[+8.25, +12.35]	<0.0001	6/6	0.031
0.8	+2.73	[+0.81, +4.66]	0.015	6/6	0.031	0.8	+6.42	[+3.29, +9.54]	0.0032	6/6	0.031
0.7	+2.48	[+0.89, +4.07]	0.010	6/6	0.031	0.7	+5.85	[+3.51, +8.19]	0.0014	6/6	0.031
0.6	+3.60	[+0.71, +6.49]	0.024	5/5*	0.063	0.6	+5.98	[+3.20, +8.77]	0.0027	6/6	0.031
0.5	+4.58	[+2.67, +6.49]	0.0016	6/6	0.031	0.5	+7.05	[+5.37, +8.73]	0.0001	6/6	0.031
0.4	+7.28	[+4.83, +9.73]	0.0006	6/6	0.031	0.4	+7.90	[+5.89, +9.91]	0.0002	6/6	0.031
0.3	+5.90	[+3.50, +8.30]	0.0015	6/6	0.031	0.3	+6.78	[+4.76, +8.80]	0.0003	6/6	0.031
0.2	+2.98	[+0.95, +5.01]	0.013	6/6	0.031	0.2	+2.03	[−0.12, +4.18]	0.059	6/6	0.031
0.1	+4.03	[+1.48, +6.59]	0.010	6/6	0.031	0.1	+1.97	[+0.70, +3.23]	0.010	6/6	0.031
sub-1	+4.60	–	$\sim 10^{-17}$	53/53*	$\sim 10^{-16}$	sub-1	+6.03	–	$\sim 10^{-19}$	54/54	$\sim 10^{-16}$

* One exact tie (excluded from the win count and the sign test).

To assess whether the margins in the fine-grained evaluation above are consistent rather than driven by a few favorable subsets, we run paired tests at every IPC level. The pairing unit is the subset: for each (IPC, loss) we form $\Delta = \mathrm{DISC} - \mathrm{DDiF}$ on each of the six subsets ($n = 6$) and test whether the mean margin differs from zero with a two-sided paired t-test, complemented by an exact two-sided sign test that ignores magnitudes entirely. Pairing removes the large between-subset difficulty variation, so the test asks precisely the robustness question: is the margin consistent across subsets?

Table 18 summarizes the results. **At every sub-1 IPC level and under both losses, DISC is ahead of DDiF on every subset except for a single tie**: under DM, the paired t-test gives $p \leq 0.024$ at all nine sub-1 levels; under DC, $p \leq 0.011$ at eight of nine, and the remaining level (IPC 0.2, $p = 0.059$) is not significant under the t-test although DISC is ahead on all six subsets. Pooled over the 54 sub-1 cells per loss, DISC wins 53/53 non-tied cells under DM and 54/54 under DC (sign-test $p \approx 10^{-16}$ each). At IPC 1.0 the picture is loss-dependent: the DC margin (+3.98pp) is significant, while the DM margin (+0.52pp, $p = 0.412$) is not; the consistent advantage lies in the sub-1 regime that DISC targets.

Three notes on interpretation. First, each table cell is a single distillation run, and the $\pm$ values in the tables are the spread over the five evaluation classifiers; the paired tests therefore address consistency across subsets, not across distillation seeds. Second, the pooled rows reuse subsets across IPC levels and are reported as a descriptive summary; the per-IPC tests are the primary evidence. Third, an exact Wilcoxon signed-rank test reaches the same conclusion at the 0.05 level except at DM IPC 0.6, where the tie leaves five pairs and the smallest attainable p is 0.0625, and at DC IPC 0.2, where it gives $p = 0.031$.

C.6 DETAILED RESULTS FOR MAIN COMPARISONS

Table 19, Table 20, and Table 21 provide detailed numerical results including standard deviations for the DM main comparison at 128×128 and 256×256 and the DC cross-loss comparison at 128×128, respectively.

Table 19: Full performance comparison on DM at 128×128 with standard deviations. The rightmost column gives the average across subsets with propagated standard deviation.

Method	IPC	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DDiF	1	69.2 ±1.0	42.0 ±0.4	45.3 ±1.8	62.4 ±1.6	**45.8** ±1.1	64.6 ±1.1	54.9 ±0.5
	0.5	63.2 ±0.8	34.4 ±2.0	38.4 ±0.6	55.6 ±1.0	39.3 ±1.7	55.7 ±1.0	47.8 ±0.5
	0.25	57.4 ±1.4	31.2 ±0.6	35.4 ±1.2	50.7 ±1.4	34.0 ±0.4	46.9 ±1.8	42.6 ±0.5
DISC	1	**69.6** ±0.5	**42.4** ±0.5	**45.5** ±0.6	**63.9** ±0.5	44.0 ±1.1	**67.0** ±1.1	**55.4** ±0.3
	0.5	66.4 ±1.3	38.8 ±1.2	43.7 ±1.8	60.0 ±1.0	41.8 ±2.3	63.4 ±1.8	52.4 ±0.7
	0.25	61.6 ±0.7	37.2 ±1.9	40.6 ±1.4	57.3 ±1.0	39.6 ±0.8	56.9 ±1.3	48.9 ±0.5

Table 20: Full performance comparison on DM at 256×256 resolution with standard deviations. The rightmost column gives the average across subsets with propagated standard deviation.

Method	IPC	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DDiF	1	67.8 ±1.0	39.6 ±1.6	43.2 ±1.7	63.1 ±0.8	44.8 ±1.1	**67.0** ±0.9	54.3 ±0.5
	0.5	61.4 ±1.8	31.9 ±1.0	38.0 ±1.6	56.2 ±1.4	38.9 ±1.8	54.8 ±0.3	46.9 ±0.6
	0.25	42.6 ±0.6	23.6 ±2.2	25.5 ±1.1	42.0 ±1.5	24.5 ±1.4	32.4 ±1.7	31.8 ±0.6
DISC	1	**68.6** ±1.3	**43.8** ±1.6	**44.1** ±1.0	**64.1** ±1.0	**46.3** ±1.3	65.8 ±1.0	**55.5** ±0.5
	0.5	66.1 ±1.1	40.3 ±1.2	41.1 ±0.5	59.8 ±1.4	42.6 ±1.0	63.6 ±0.5	52.3 ±0.4
	0.25	61.2 ±0.5	33.2 ±1.4	37.8 ±1.0	52.8 ±1.8	38.8 ±0.9	55.0 ±1.7	46.5 ±0.5

Table 21: Full performance comparison on DC at 128×128 with standard deviations. The rightmost column gives the average across subsets with propagated standard deviation.

Method	IPC	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DDiF	1	61.2 ±1.0	35.2 ±1.7	37.8 ±1.1	56.0 ±1.0	39.1 ±1.3	54.3 ±1.0	47.3 ±0.5
	0.5	56.1 ±1.2	31.6 ±1.7	32.8 ±1.0	48.7 ±0.7	32.6 ±0.8	40.4 ±0.3	40.4 ±0.4
	0.25	49.0 ±1.2	28.0 ±1.2	30.7 ±0.9	45.4 ±1.1	29.2 ±1.2	35.4 ±1.3	36.3 ±0.5
DISC	1	**66.1** ±0.6	**41.4** ±0.9	**41.6** ±0.8	**60.2** ±1.5	**41.5** ±0.6	**56.7** ±0.5	**51.3** ±0.4
	0.5	61.1 ±1.2	38.5 ±0.6	40.5 ±0.7	56.6 ±1.2	38.1 ±1.5	49.7 ±0.9	47.4 ±0.4
	0.25	56.5 ±1.2	35.4 ±1.7	34.4 ±1.6	51.8 ±0.8	34.7 ±1.4	42.3 ±1.0	42.5 ±0.5

C.7 DETAILED RESULTS FOR CROSS-ARCHITECTURE GENERALIZATION

This section details the cross-architecture results of the main paper. While the main paper primarily focuses on the Distribution Matching (DM) loss, we extend our evaluation to include the Gradient Matching (DC) loss to further validate the robustness of our method. We compare DISC against the state-of-the-art method, DDiF (Shin et al., 2025), across various architectures including AlexNet, VGG11, ResNet18, and ViT. Note that the results for DDiF are reproduced using the official implementation. As shown in Table 22 and Table 23, DISC achieves higher subset-average accuracy than DDiF in 23 of the 24 (loss, IPC, architecture) combinations; the only exception is ResNet18 under DM at IPC 1.0. Under DC, DISC leads on every architecture at every budget, and under both losses the margin is larger in the sub-1 regime than at IPC 1.0.

Table 22: Detailed cross-architecture generalization results on ImageNet-Subsets (128×128) across IPC 1.0, 0.5, and 0.25 under Distribution Matching (DM). The DDiF results are reproduced using the official implementation. The rightmost column gives the average across subsets with propagated standard deviation.

IPC	Test Net	Method	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg
1.0	AlexNet	DDiF	63.5 ±0.6	30.6 ±0.9	35.1 ±0.9	53.6 ±1.0	**34.4** ±0.9	59.3 ±1.3	46.1 ±0.4
		DISC (Ours)	**68.4** ±0.7	**36.8** ±0.5	**39.4** ±1.2	**54.9** ±1.7	33.4 ±1.7	**62.5** ±1.0	**49.2** ±0.5
	VGG11	DDiF	67.5 ±1.8	40.5 ±0.7	42.8 ±0.8	64.1 ±0.6	**41.2** ±1.8	67.4 ±0.7	53.9 ±0.5
		DISC (Ours)	**69.6** ±0.3	**44.6** ±1.3	**43.5** ±1.3	**64.9** ±1.0	39.2 ±0.9	**68.2** ±0.8	**55.0** ±0.4
	ResNet18	DDiF	**62.5** ±1.7	31.5 ±1.4	38.3 ±0.9	**60.3** ±2.0	**36.8** ±1.3	**61.9** ±0.6	**48.6** ±0.6
		DISC (Ours)	61.4 ±0.5	**33.4** ±0.6	**38.5** ±1.7	57.1 ±1.6	26.2 ±2.1	61.4 ±0.7	46.3 ±0.6
	ViT	DDiF	46.4 ±0.9	24.4 ±1.4	37.4 ±1.1	43.7 ±1.5	**29.9** ±1.3	48.7 ±0.4	38.4 ±0.5
		DISC (Ours)	**49.0** ±1.0	**25.0** ±1.0	**39.2** ±1.2	**49.8** ±1.5	29.4 ±1.3	**51.0** ±0.8	**40.6** ±0.5
0.5	AlexNet	DDiF	56.5 ±1.1	27.4 ±0.9	33.1 ±1.3	47.5 ±1.2	28.7 ±0.4	53.1 ±1.1	41.1 ±0.4
		DISC (Ours)	**63.6** ±1.1	**34.0** ±1.2	**36.8** ±0.6	**49.4** ±1.0	**31.5** ±2.0	**58.4** ±0.5	**45.6** ±0.5
	VGG11	DDiF	63.0 ±0.5	32.5 ±1.1	37.6 ±1.0	58.2 ±0.6	36.0 ±1.1	59.0 ±0.9	47.7 ±0.4
		DISC (Ours)	**64.7** ±0.7	**38.8** ±0.7	**41.0** ±0.8	**60.2** ±0.6	**37.8** ±2.2	**64.6** ±0.9	**51.2** ±0.5
	ResNet18	DDiF	53.5 ±1.1	26.1 ±1.3	31.9 ±1.6	53.0 ±0.7	**32.3** ±1.7	52.3 ±0.4	41.5 ±0.5
		DISC (Ours)	**57.0** ±1.6	**30.5** ±1.5	**37.5** ±0.4	**53.5** ±0.8	28.6 ±4.2	**58.2** ±1.5	**44.2** ±0.8
	ViT	DDiF	40.2 ±1.0	**25.3** ±0.6	34.8 ±2.0	39.7 ±1.0	24.8 ±2.3	43.4 ±1.4	34.7 ±0.6
		DISC (Ours)	**44.0** ±1.5	23.2 ±0.7	**37.5** ±0.7	**47.4** ±0.8	**25.5** ±1.7	**49.5** ±1.5	**37.9** ±0.5
0.25	AlexNet	DDiF	47.2 ±1.5	22.7 ±2.0	28.9 ±0.9	42.2 ±2.5	**30.4** ±1.7	41.8 ±1.4	35.5 ±0.7
		DISC (Ours)	**55.4** ±0.9	**29.0** ±0.9	**33.0** ±0.7	**48.3** ±0.8	28.1 ±0.8	**50.0** ±0.9	**40.6** ±0.3
	VGG11	DDiF	54.1 ±1.8	23.9 ±0.8	35.1 ±0.7	50.8 ±1.1	29.9 ±0.9	45.4 ±1.0	39.9 ±0.5
		DISC (Ours)	**59.5** ±0.6	**30.6** ±0.7	**38.9** ±0.8	**57.6** ±0.6	**32.4** ±0.9	**55.5** ±1.3	**45.8** ±0.3
	ResNet18	DDiF	42.6 ±0.5	22.1 ±1.0	29.4 ±1.6	42.2 ±1.1	24.7 ±0.9	43.6 ±1.3	34.1 ±0.5
		DISC (Ours)	**47.3** ±0.5	**25.8** ±0.9	**34.7** ±0.9	**49.2** ±1.4	**29.9** ±0.7	**53.6** ±1.1	**40.1** ±0.4
	ViT	DDiF	36.3 ±1.5	21.7 ±0.8	28.2 ±1.7	35.7 ±1.4	21.4 ±1.0	39.2 ±1.0	30.4 ±0.5
		DISC (Ours)	**41.3** ±1.1	**22.8** ±1.7	**34.7** ±0.5	**44.2** ±0.8	**24.0** ±0.7	**43.5** ±0.9	**35.1** ±0.4

Table 23: Detailed cross-architecture generalization results on ImageNet-Subsets (128×128) across IPC 1.0, 0.5, and 0.25 under Gradient Matching (DC). The DDiF results are reproduced using the official implementation. The rightmost column gives the average across subsets with propagated standard deviation.

IPC	Test Net	Method	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg
1.0	AlexNet	DDiF	61.4 ±0.7	30.6 ±1.2	35.4 ±1.1	53.4 ±0.8	34.0 ±0.4	51.8 ±1.2	44.4 ±0.4
		DISC (Ours)	**67.2** ±1.1	**39.3** ±0.7	**40.4** ±0.7	**55.5** ±1.1	**35.4** ±1.5	**55.3** ±1.8	**48.9** ±0.5
	VGG11	DDiF	63.2 ±0.8	38.0 ±1.0	**42.3** ±0.7	59.4 ±1.5	**44.0** ±1.6	54.8 ±1.2	50.3 ±0.5
		DISC (Ours)	**67.7** ±0.8	**46.0** ±1.0	41.6 ±0.5	**62.3** ±1.2	41.2 ±1.1	**62.2** ±1.0	**53.5** ±0.4
	ResNet18	DDiF	60.6 ±1.1	32.9 ±1.4	38.8 ±1.3	54.8 ±1.1	**38.8** ±1.5	55.3 ±1.1	46.9 ±0.5
		DISC (Ours)	**64.6** ±0.6	**40.4** ±0.9	**42.7** ±1.6	**60.9** ±0.5	35.7 ±1.5	**58.8** ±1.1	**50.5** ±0.5
	ViT	DDiF	44.7 ±0.5	21.8 ±1.2	36.4 ±1.2	42.2 ±1.1	25.0 ±1.2	46.2 ±2.0	36.1 ±0.5
		DISC (Ours)	**52.9** ±0.9	**30.2** ±1.6	**39.8** ±1.0	**49.8** ±1.9	**31.5** ±1.0	**47.7** ±1.1	**42.0** ±0.5
0.5	AlexNet	DDiF	52.2 ±0.6	27.7 ±0.8	33.6 ±0.8	44.2 ±0.7	31.3 ±1.8	41.6 ±1.0	38.4 ±0.4
		DISC (Ours)	**62.1** ±0.8	**35.1** ±0.9	**39.0** ±0.5	**53.0** ±1.4	**34.0** ±0.8	**49.4** ±0.7	**45.4** ±0.4
	VGG11	DDiF	56.9 ±1.1	34.7 ±0.8	38.9 ±1.5	52.3 ±0.7	**39.3** ±1.2	47.2 ±1.1	44.9 ±0.4
		DISC (Ours)	**63.8** ±0.4	**40.7** ±1.4	**40.8** ±1.0	**59.5** ±0.6	39.2 ±0.8	**56.3** ±0.7	**50.1** ±0.4
	ResNet18	DDiF	51.2 ±1.3	29.4 ±1.5	35.0 ±0.8	46.2 ±1.3	33.0 ±2.0	48.8 ±1.7	40.6 ±0.6
		DISC (Ours)	**58.5** ±2.0	**35.2** ±0.9	**37.4** ±1.4	**56.6** ±0.7	**35.2** ±1.3	**54.7** ±1.9	**46.3** ±0.6
	ViT	DDiF	41.2 ±0.4	20.4 ±1.0	34.2 ±1.8	40.1 ±1.2	21.2 ±0.8	37.0 ±1.1	32.4 ±0.5
		DISC (Ours)	**46.9** ±1.1	**27.3** ±0.9	**37.4** ±1.3	**45.9** ±0.8	**28.3** ±1.5	**44.2** ±1.9	**38.3** ±0.5
0.25	AlexNet	DDiF	46.1 ±0.9	25.5 ±1.1	32.3 ±0.8	39.8 ±0.9	25.3 ±1.3	31.5 ±0.7	33.4 ±0.4
		DISC (Ours)	**53.0** ±1.2	**29.8** ±0.8	**35.4** ±1.1	**44.4** ±1.1	**32.4** ±2.3	**41.8** ±0.7	**39.5** ±0.5
	VGG11	DDiF	52.2 ±0.8	31.1 ±1.1	38.3 ±1.6	49.9 ±1.1	36.8 ±0.9	42.5 ±0.9	41.8 ±0.4
		DISC (Ours)	**59.6** ±0.8	**35.2** ±1.7	**39.5** ±1.4	**56.3** ±0.7	**37.9** ±0.8	**48.7** ±0.5	**46.2** ±0.4
	ResNet18	DDiF	41.8 ±1.5	24.4 ±1.2	32.2 ±1.0	40.4 ±0.8	28.0 ±1.3	40.5 ±0.5	34.6 ±0.4
		DISC (Ours)	**53.7** ±0.8	**30.4** ±1.1	**35.2** ±1.7	**52.1** ±1.3	**31.8** ±1.1	**47.2** ±0.9	**41.7** ±0.5
	ViT	DDiF	37.7 ±2.1	21.4 ±2.0	31.2 ±0.8	35.2 ±1.0	18.4 ±1.4	32.2 ±1.4	29.4 ±0.6
		DISC (Ours)	**44.5** ±0.5	**24.2** ±0.9	**33.7** ±0.4	**41.8** ±1.5	**24.4** ±1.0	**39.6** ±1.1	**34.7** ±0.4

C.8 MATCHED INSTANCE-COUNT ANALYSIS

Table 24: Matched instance-count comparison on ImageNet-Subsets (128×128). DISC columns are its main-comparison results at IPC 0.5 (50 instances/class) and IPC 1.0 (101 instances/class). DDiF at 51 instances/class reports its published IPC 1.0 results (Yellow reproduced); DDiF at 101 instances/class (IPC $\approx$1.98) is a same-protocol reproduction. † marks results reproduced with the official implementation, as in Table 1. At both points DISC uses about half of DDiF's byte budget. Each cell reports the mean $\pm$ std over 5 evaluation networks; bold marks the better method within each column group.

		$\sim$51 inst./class (DDiF 51 @ IPC 1.0 / DISC 50 @ IPC 0.5)		**101 inst./class** (DDiF IPC $\approx$1.98 / DISC IPC 1.0)	
Loss	**Subset**	**DDiF**	**DISC (Ours)**	**DDiF†**	**DISC (Ours)**
DC	Nette	**61.2** $\pm$1.0	61.1 $\pm$1.2	64.8 $\pm$0.5	**66.1** $\pm$0.6
	Woof	35.2 $\pm$1.7	**38.5** $\pm$0.6	36.0 $\pm$1.2	**41.4** $\pm$0.9
	Fruit	37.8 $\pm$1.1	**40.5** $\pm$0.7	38.6 $\pm$1.7	**41.6** $\pm$0.8
	Yellow	56.0† $\pm$1.0	**56.6** $\pm$1.2	59.1 $\pm$1.4	**60.2** $\pm$1.5
	Meow	**39.1** $\pm$1.3	38.1 $\pm$1.5	39.4 $\pm$1.0	**41.5** $\pm$0.6
	Squawk	**54.3** $\pm$1.0	49.7 $\pm$0.9	52.6 $\pm$1.6	**56.7** $\pm$0.5
	Avg.	47.3 $\pm$0.5	**47.4** $\pm$0.4	48.4 $\pm$0.5	**51.3** $\pm$0.4
DM	Nette	**69.2** $\pm$1.0	66.4 $\pm$1.3	69.5 $\pm$1.0	**69.6** $\pm$0.5
	Woof	**42.0** $\pm$0.4	38.8 $\pm$1.2	38.5 $\pm$1.5	**42.4** $\pm$0.5
	Fruit	**45.3** $\pm$1.8	43.7 $\pm$1.8	43.1 $\pm$1.6	**45.5** $\pm$0.6
	Yellow	**62.4**† $\pm$1.6	60.0 $\pm$1.0	**64.7** $\pm$1.5	63.9 $\pm$0.5
	Meow	**45.8** $\pm$1.1	41.8 $\pm$2.3	43.9 $\pm$1.4	**44.0** $\pm$1.1
	Squawk	**64.6** $\pm$1.1	63.4 $\pm$1.8	65.0 $\pm$1.5	**67.0** $\pm$1.1
	Avg.	**54.9** $\pm$0.5	52.4 $\pm$0.7	54.1 $\pm$0.6	**55.4** $\pm$0.3

A near-matched-count comparison distinguishes the storage advantage from per-instance performance. Table 24 compares DDiF at 51 and 101 instances per class against DISC at its native IPC 0.5 (50 instances, i.e., decoders) and IPC 1.0 (101 instances). The DISC columns are the values reported in the main comparison; the DDiF cells at 51 instances are its published IPC 1.0 values (Yellow is reproduced, as it is not reported), and those at 101 instances are same-protocol reproductions with the official implementation, since this configuration is not reported. Because the two DDiF columns come from different sources (published results versus same-protocol reproductions), they are not directly comparable to each other, and the difference between them should not be read as the effect of doubling DDiF's instance count; each DDiF column is compared with the DISC column beside it. At the low-budget point, DISC uses its headline IPC 0.5 result, with one fewer instance and half the bytes. At the low-budget point (DDiF 51 instances at IPC 1.0 vs DISC 50 instances at IPC 0.5), DISC matches DDiF under DC (47.4 vs 47.3 on average, ahead on three of six subsets), while under DM it trails by 2.5pp on average. Thus, at half the byte budget and with one fewer instance, the shared parameterization is on par per instance under DC and somewhat below under DM; its advantage comes from how many instances the budget affords rather than from each instance being stronger. At 101 instances per class, the picture changes. This is DISC's native IPC 1.0 configuration, whereas hosting 101 independent fields costs DDiF roughly twice the bytes (IPC $\approx$1.98). Even against this doubled-budget baseline, DISC leads in eleven of twelve cells (DC +2.9pp on average, six of six; DM +1.3pp, five of six). The budget side of the co-design prices an additional instance at 1,932 B instead of 3,852 B, so the same IPC 1.0 budget affords 101 instances per class instead of 51. The two-stage coreset puts this capacity to use (Appendix C.3.2); the gains scale with the budget-driven instance count while the shared parameterization stays competitive per instance.

C.9 ENCODER SHARING STRATEGY

We investigate three encoder-sharing scopes under a fixed IPC 1.0 budget, with the *full two-stage coreset pipeline applied to every variant*: **(i) All**, where a single encoder is shared by *all* classes; **(ii) Each**, which assigns one dedicated encoder to *each* class; and **(iii) Group**, where an encoder is shared within small *groups* of instances inside a class. To compare the allocations induced by sharing scope, all three variants use the identical canonical network configuration (the encoder depth, decoder depth, and layer size of the default **All** setting); the per-class decoder count then follows from allocating the same total parameter budget to each scope, yielding 101 decoders per class for *All*, 99 for *Each*, and 70 for *Group*. The experimental results comparing these scopes under both DC and DM are presented in Table 25.

Table 25: Top-1 accuracy (%) of DISC under three encoder-sharing scopes on ImageNet-Subsets (128×128) at IPC = 1.0. All variants use the full two-stage coreset pipeline and the identical canonical network configuration (encoder depth, decoder depth, and layer size fixed to those of **All**); only the sharing scope differs, and "$N_{\mathrm{dec/class}}$" is the number of decoders (i.e., images) per class obtained by allocating the same parameter budget to each scope. Bold marks the best score within each column and loss.

Loss	Sharing	$N_{\mathrm{dec/class}}$	Subset						
			Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DC	Each	99	58.7	31.1	35.7	53.0	31.8	46.7	42.8
	Group	70	61.4	36.9	40.2	58.5	40.2	48.8	47.7
	All	101	**66.1**	**41.4**	**41.6**	**60.2**	**41.5**	**56.7**	**51.3**
DM	Each	99	50.2	23.8	35.6	46.1	25.6	49.0	38.4
	Group	70	61.8	38.8	43.1	58.2	38.7	62.7	50.6
	All	101	**69.6**	**42.4**	**45.5**	**63.9**	**44.0**	**67.0**	**55.4**

The fully shared **All** configuration wins in every subset under both objectives, averaging 51.3% (DC) and 55.4% (DM), followed by *Group* (47.7 / 50.6) and *Each* (42.8 / 38.4). Two observations stand out. First, **All** performs best at a matched byte budget and canonical network configuration: it spends little of the budget on its single encoder and makes nearly all of it available to decoders. Since each scope yields a different decoder count, this compares the practical allocations the scopes afford. Second, *Each* performs poorly under DM (12.2pp below *Group* on average) even though its 99 decoders per class are close to **All**'s 101; the difference is therefore not explained by count alone. We therefore use the **All** configuration throughout the paper.

Detailed configurations for the three sharing scopes, including the number of hidden layers and the resulting decoder counts, are provided in Table 26.

Table 26: Configurations for encoder sharing and their induced decoder counts. "Group Size" is instances per encoder group in the Group scheme; $N_{\mathrm{dec/class}}$ is decoders per class under a fixed budget. Hidden Layers $= E + D$ with $E = 3$ shared-encoder layers.

Sharing Type	Hidden Layers	Shared Encoder Layers	Group Size	$N_{\mathrm{dec/class}}$
All	4	3	–	101
Each	4	3	–	99
Group	4	3	10	70

C.10 ENCODER FREEZING STRATEGY

Our default DISC pipeline freezes the shared encoder after the reconstruction warmup stage and optimizes only the instance-specific decoders during the subsequent distillation stage. This design retains the encoder initialization learned during pretraining and warmup while allowing each decoder to adapt to the matching objective. To verify this choice, we compare the default frozen-encoder

setting with a no-freeze variant, where the shared encoder remains trainable during DM distillation. As shown in Table 27, keeping the encoder trainable reduces the average accuracy from 55.4% to 51.6%. The degradation is consistent across all six ImageNet-Subsets, so we freeze the encoder during distillation in all main experiments.

Table 27: Ablation on freezing the shared encoder under Distribution Matching (DM) on ImageNet-Subsets (128×128) at IPC 1.0. The frozen-encoder row corresponds to DISC in Table 1; the no-freeze variant keeps the shared encoder trainable during distillation. The rightmost column gives the average across subsets with propagated standard deviation.

Method	IPC	Nette	Woof	Fruit	Yellow	Meow	Squawk	Avg.
DISC (Frozen encoder)	1.0	**69.6** ±0.5	**42.4** ±0.5	**45.5** ±0.6	**63.9** ±0.5	**44.0** ±1.1	**67.0** ±1.1	**55.4** ±0.3
DISC (No freeze)	1.0	65.6 ±1.3	38.5 ±0.8	42.7 ±0.7	59.3 ±1.5	40.6 ±1.1	63.0 ±1.2	51.6 ±0.5

C.11 Additional visualizations

Figure 4 and Figure 5 visualize the synthetic images obtained after the distillation process is complete.

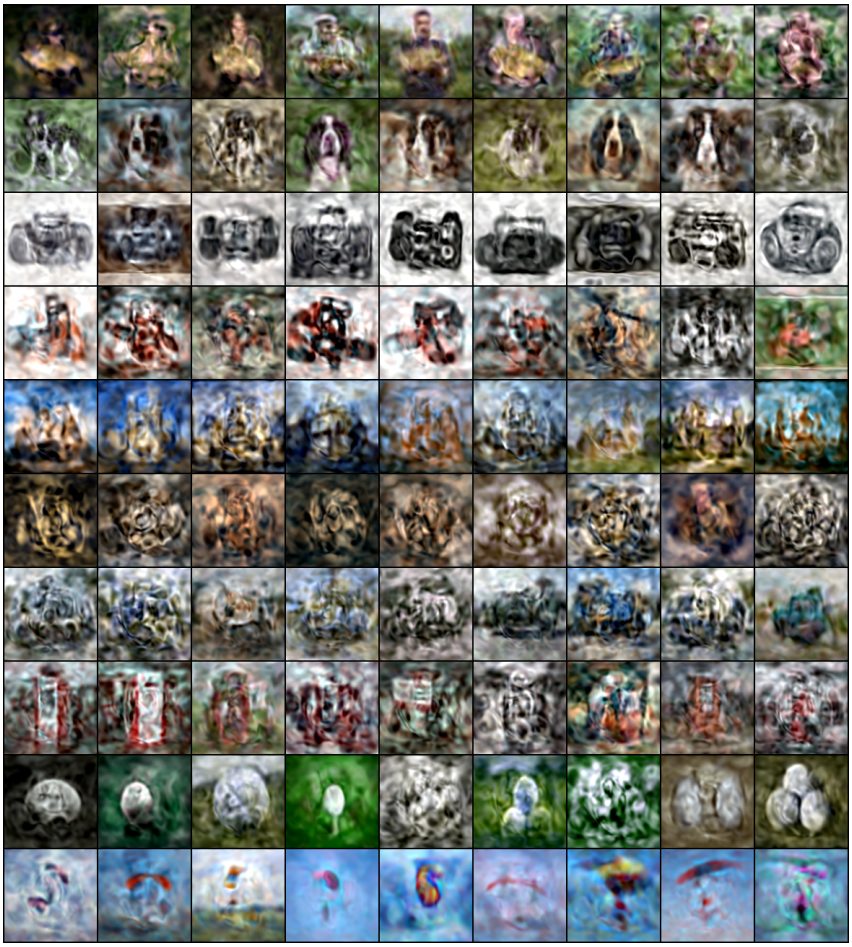


Figure 4: Synthetic Nette images at IPC 0.1 under Gradient Matching (DC).

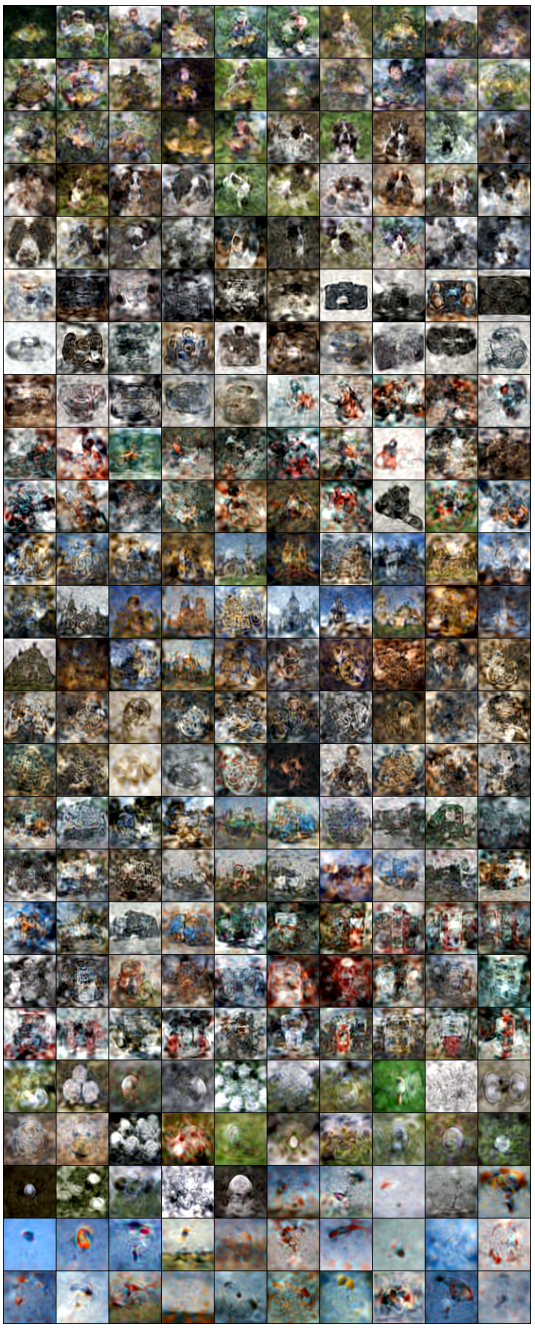


Figure 5: Synthetic Nette images at IPC 0.5 under Distribution Matching (DM).