Learnability-Guided Diffusion for Dataset Distillation

Supplementary Material

This supplementary material is organized as follows: Sec. A.1 presents generalizability experiments across different diffusion models. Sec. A.2 provides detailed cross-increment complementarity analysis demonstrating the reduced redundancy of our method. Sec. A.3 examines the effect of different dataset initialization strategies. Sec. A.4 presents comprehensive hyperparameter sensitivity analysis. Sec. A.5 analyzes the impact of batch size for learnability ranking. Sec. A.6 provides systematic ablation studies of each component. Sec. B details the incremental learning algorithm. Sec. C provides complete implementation details for reproducibility. Sec. D presents qualitative visualizations of generated samples across different methods.

A. Extended Experimental Results

This section provides comprehensive experimental analyses, including generalizability studies, redundancy analysis, ablation studies, and hyperparameter sensitivity analysis.

A.1. Generalizability Across Base Diffusion Models

To validate that our learnability-guided framework generalizes beyond DiT [23], we apply our method to Minimax Diffusion [10], a diffusion model fine-tuned for dataset distillation to encourage both diversity and representativeness. We compare three configurations: vanilla Minimax, Minimax with IGD guidance [5], and Minimax with our learnability-guided approach (Minimax+Ours) on ImageNette and ImageWoof at 50 and 100 IPC across three architectures.

Our method consistently improves upon both the Minimax baseline and Minimax-IGD across all architectures, datasets, and IPC budgets, as shown in Tab. 5. On both ImageNette and ImageWoof, Minimax+Ours matches or exceeds the strongest baseline in almost every setting. These consistent improvements demonstrate that our learnability-guided synthesis framework is architecture-agnostic and can be seamlessly integrated with different generative priors to enhance dataset distillation quality.

Table 5. Comparison across distilled IPC budgets with Minimax Diffusion [10] on Nette and Woof Mean±std accuracy; best per row is shown in **bold**. Full represents training with the entire dataset.

Dataset	Model	IPC	Minimax[10]	Minimax-IGD[5]	Minimax+Ours	Full
Nette	ConvNet-6	50	76.9±0.9	**80.6±0.8**	77.8±1.0	94.3±0.5
		100	81.1±0.3	**85.1±0.5**	**85.1±0.7**	
	ResNetAP-10	50	78.2±0.7	81.5±0.3	**82.2±0.4**	94.6±0.5
		100	81.3±0.9	85.6±0.3	**86.4±0.4**	
	ResNet-18	50	78.1±0.6	82.2±0.4	**82.5±0.4**	95.3±0.6
		100	81.3±0.7	85.3±1.0	**87.2±0.8**	
Woof	ConvNet-6	50	50.7±1.8	54.6±1.3	**56.3±1.4**	85.9±0.4
		100	57.1±1.9	61.3±0.9	**62.6±1.0**	
	ResNetAP-10	50	59.8±0.8	62.5±1.4	**63.7±1.1**	87.2±0.6
		100	66.8±1.2	68.3±0.6	**70.0±0.5**	
	ResNet-18	50	60.5±0.5	63.4±0.6	**66.2±0.6**	89.0±0.6
		100	67.4±0.7	70.5±0.8	**72.1±0.4**	

A.2. Cross-Increment Complementarity Analysis

To quantify sample redundancy in distilled datasets, we partition each 50 IPC dataset into $K = 5$ disjoint 10 IPC increments and measure cross-increment learning dynamics. Specifically, we train a ResNet-AP-10 model on the first increment $\mathcal{I}_1$ (10 IPC) until convergence, then evaluate classification errors on subsequent increments $\mathcal{I}_{t+1}$ for $t \in \{1, 2, 3, 4\}$, without any additional training. High error counts indicate complementary information (the model has not learned the new increment's signals), while low errors indicate redundancy (overlapping information has already been captured).

Fig. 8 shows results on ImageNette. DiT exhibits severe redundancy with only 0–8 errors across increments (92–100% accuracy), indicating that a model trained on $\mathcal{I}_1$ already captures most of the training signal from remaining increments. IGD improves slightly with 2–16 errors (84–98% accuracy) but substantial overlap remains. In contrast, our method maintains 22–88 errors (12–78% accuracy), confirming that each increment introduces substantial new learning signals. Notably, our method shows a decreasing trend in errors across increments ($\mathcal{I}_2$: 88 errors, $\mathcal{I}_3$: 58 errors, $\mathcal{I}_4$: 31 errors, $\mathcal{I}_5$: 22 errors), suggesting that finding hard samples becomes progressively more challenging as the model improves; a natural consequence of curriculum learning where easier patterns are learned first. Even at $\mathcal{I}_5$ (the 40–50 IPC increment), our method produces 22 errors versus 0 for DiT, demonstrating sustained complementarity across all stages. This 5–10× higher error rate validates that our learnability-guided synthesis conditions each stage on model state to generate non-redundant, complementary samples rather than redundant copies.

A.3. Effect of Initial Data Seed on Our Method

Our method begins with a seed dataset $\mathcal{D}_1$ of size 10 IPC. To understand how this initialization affects the final distilled dataset, we compare three seed sources: IGD [5], DiT [23], and randomly selected real images. We evaluate accuracy as the distilled dataset expands from 10 to 50 IPC on ImageNette using ResNet-AP-10, reporting results under the incremental training setup.

Fig. 9 shows that IGD provides the strongest starting

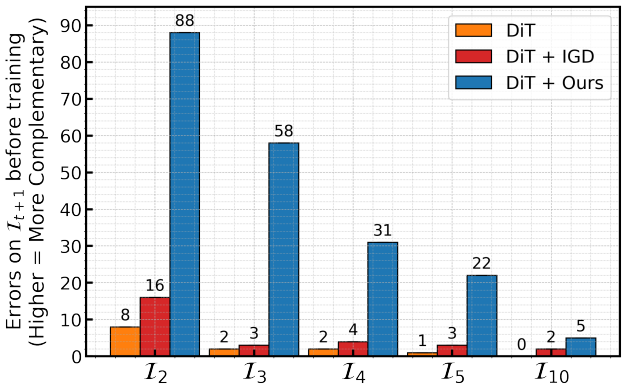


Figure 8. **Cross-increment redundancy analysis.** Errors on increment $\mathcal{I}_{t+1}$ using a model trained only on $\mathcal{I}_t$. DiT and IGD show < 15 errors (high redundancy); our method shows 22-88 errors, confirming reduced redundancy.

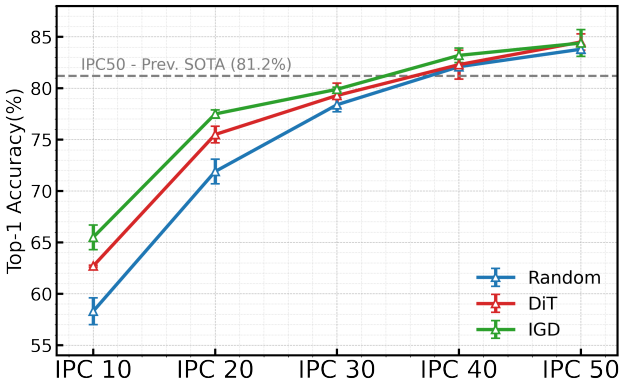


Figure 9. **Incremental accuracy across different seed initializations on our method.** Top-1 accuracy as the distilled dataset grows from 10 to 50 IPC. Three seed sources are compared: IGD [5] (green), DiT [23] (red), and random real images (blue). IGD initialization achieves the best final performance (84.4%), but all seeds substantially outperform previous SOTA (81.2%, dashed line), demonstrating the robustness of our learnability-guided framework to initialization choice.

point, improving final accuracy by 0.6% over random initialization (84.4 vs. 83.8). This advantage due to the fact that IGD's influence-guided generation produces samples aligned with informative low-IPC training gradients (10–20 IPC). DiT achieves a comparable improvement of 0.7% over random (84.5 vs. 83.8). Importantly, our learnability-guided framework amplifies all initializations: even with random seeds, we reach 83.8%. For comparison purposes, MGD^3 achieves 81.2% when distilling 50 IPC under its static evaluation protocol.

While stronger seeds yield higher absolute performance, the key advantage of our method—progressive, complementary synthesis—remains effective regardless of initialization. We default to IGD, though other seeds may be preferable under tighter computational budgets. The consistent relative gains highlight that our incremental, learnability-driven formulation provides benefits largely independent of seed quality.

A.4. Hyperparameter Analysis

Our method uses three key hyperparameters: reference model weight ω (Eq. (7) in main paper), guidance strength λ (Eq. (8) in main paper), and deviation guidance strength γ (Sec. 4.3 in main paper). We systematically evaluate each parameter's impact on distilled dataset quality across multiple IPC budgets.

A.4.1. Reference Model Weight ω

To assess how the reference-model weight ω influences performance, we evaluate $\omega \in 0.0, 0.5, 1.0, 2.0$ across IPC levels 20, 30, 40, and 50. As shown in Fig. 10a, the optimal setting is $\omega = 1.0$, achieving 84.9% at 50 IPC. When $\omega = 0.0$ (i.e., no reference-model guidance), performance drops sharply at 20 IPC, suggesting that the resulting samples are too difficult for low-IPC training. In contrast, $\omega = 2.0$ overemphasizes the reference model and produces overly easy samples, reducing performance at higher IPC levels such as 50. Overall, the method is stable within the range $\omega \in [0.5, 1.0]$, where accuracy remains within 1% of the optimal value.

A.4.2. Guidance Strength λ

We evaluate guidance strength $\lambda \in \{5, 10, 15, 20, 25\}$ on ImageNette with IPC 20, 30, 40, 50, and 100. Fig. 10b shows that $\lambda = 15$ achieves the best performance at 84.4% for 50 IPC. When guidance is too weak ($\lambda = 5$), performance drops to 83.7%. Conversely, when guidance is too strong ($\lambda = 25$), performance declines to 83.1%.

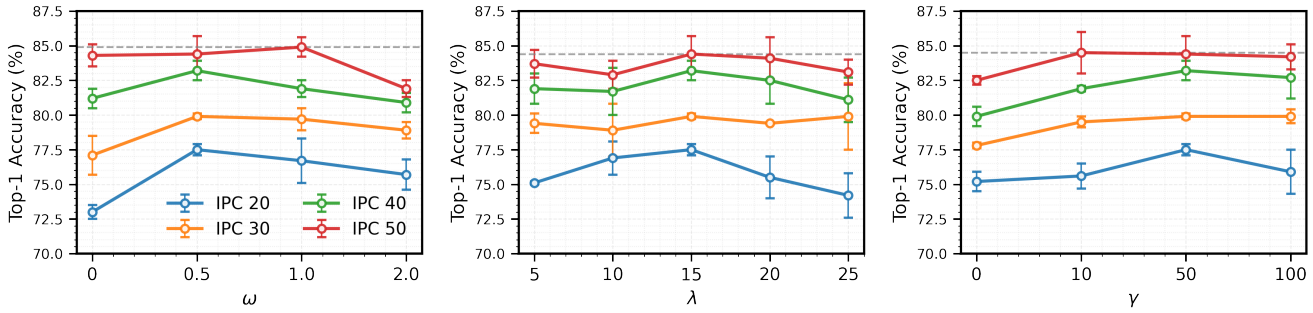
A.4.3. Deviation Guidance Strength γ

To understand how deviation guidance strength affects performance, we evaluate $\gamma \in \{0, 10, 50, 100\}$ across various IPC values. Our results in Fig. 10c show that $\gamma = 50$ provides the best diversity-quality trade-off, achieving 84.7% at 50 IPC. When $\gamma = 0$ (no diversity enforcement), performance drops to 82.5%. Extending to $\gamma = 100$ slightly reduces performance to 84.2%.

We observe consistent trends across IPC budgets, with values at $\omega = 0.5$, $\lambda = 15$, and $\gamma = 50$. The method demonstrates robustness within reasonable ranges: $\omega \in [0.5, 1.0]$, $\lambda \in [10, 20]$, and $\gamma \in [10, 100]$ all achieve competitive performance, enabling practitioners to use default values without extensive per-dataset tuning.

A.5. Effect of Batch Size for Learnability Ranking

Our learnability scoring mechanism (Sec. 4.4 in main paper) evaluates candidate samples in batches of size κ. To assess the impact of batch size, we test $\kappa \in \{1, 3, 10\}$ across



(a) Effect of reference weight ω. (b) Effect of guidance strength λ. (c) Effect of deviation strength γ.

Figure 10. **Hyperparameter sensitivity on ImageNette.** Top-1 accuracy (%) as we vary (a) reference weight ω, (b) guidance strength λ, and (c) deviation strength γ across different IPC settings.

Table 6. **Experiment with Batch Size for Learnability Rank on ImageNette.** Top-1 accuracy (%) for different batch sizes and IPC settings.

κ	IPC 20	IPC 30	IPC 40	IPC 50	IPC 100
1	77.5 ± 0.5	79.9 ± 0.3	83.2 ± 0.8	84.4 ± 1.6	88.5 ± 1.1
3	77.3 ± 1.0	80.9 ± 0.7	83.3 ± 0.6	85.1 ± 0.8	88.4 ± 1.4
10	$\mathbf{77.8 \pm 0.4}$	$\mathbf{81.1 \pm 1.5}$	$\mathbf{83.9 \pm 0.8}$	$\mathbf{85.6 \pm 0.5}$	$\mathbf{89.1 \pm 0.6}$

IPC budgets.

Tab. 6 shows that larger batch sizes consistently improve performance. $\kappa = 10$ achieves the best results with gains of 0.3–1.2% over $\kappa = 1$, particularly notable at higher IPC values (50, 100). For instance, at IPC 50, $\kappa = 10$ reaches 85.6% compared to 84.4% for $\kappa = 1$. We use $\kappa = 3$ as default for the best accuracy-efficiency trade-off.

A.6. Component Ablation Study

We conduct a systematic ablation study to analyze the contribution of each component in our method. Starting from the DiT baseline, we progressively add: (1) learnability-guided generation, (2) deviation guidance for diversity, and (3) learnability-based ranking for sample selection. Tab. 7 shows results across IPC 20, 30, 40, and 50.

Each component provides consistent improvements across all IPC settings. Learnability guidance alone yields 4.7–7.0% gains over the DiT baseline, demonstrating that conditioning generation on model learning state produces more informative samples. Adding deviation guidance further improves performance by 1.9–3.3%, showing that explicitly encouraging diversity helps avoid redundancy. Finally, learnability ranking contributes an additional 0.3–1.2% by selecting the most learnable samples from a larger candidate pool. The full method achieves cumulative improvements of 8.1–10.0% over DiT baseline, with the largest gains at higher IPC values where redundancy is more problematic. Notably, the improvements are additive and consistent, validating that each component addresses a distinct aspect of high-quality dataset distillation.

Table 7. Component ablation study on ImageNette. We progressively add components starting from DiT baseline. Mean±std accuracy (%) across 3 runs with ResNet-AP-10.

Method	IPC 20	IPC 30	IPC 40	IPC 50
DiT (Baseline)	68.2±1.1	73.0±1.7	75.2±1.8	75.6±1.1
+ Learnability Guidance	75.2±0.7	77.8± 0.2	79.9±0.7	82.5±0.3
+ Deviation Guidance	77.5 ± 0.5	79.9± 0.3	83.2±0.8	84.4±1.6
+ Learnability Ranking (Ours)	**77.8± 0.4**	**81.1±1.5**	**83.9± 0.8**	**85.6±0.5**
Improvements over baseline:				
Learnability Guidance	+7.0	+4.8	+4.7	+6.9
+ Deviation Guidance	+9.3	+6.9	+8.0	+8.8
+ Ranking (Ours)	+9.6	+8.1	+8.7	+10.0

B. Incremental Learning Details

Our method employs an incremental training strategy where synthetic data is progressively generated and added to the training set based on the current model's learning state. For a target dataset size of 100 IPC, we divide the process into $K = 10$ increments, each adding 10 IPC.

B.1. Algorithm Overview

At each increment $i \in \{1, 2, ..., K\}$, we: (1) train learner model θ_i on current synthetic dataset $\mathcal{D}_{i-1}$, (2) generate candidate pool using diffusion model with learnability-guided selection, (3) select top samples to form increment $\mathcal{I}_i$, and (4) update dataset $\mathcal{D}_i = \mathcal{D}_{i-1} \cup \mathcal{I}_i$.

The complete procedure is detailed in Algorithm 1, which presents the full incremental distillation framework. The key innovation lies in our learnability-guided diffusion sampling (Algorithm 2), which conditions the generation process on both the current learner state θ_{i-1} and a reference model θ^* to synthesize samples that are informative yet semantically valid.

Incremental Generation and Selection. Unlike prior methods that generate all samples independently, our ap-

proach merges candidate generation with selection to maintain diversity (Algorithm 1, lines 6-16). For each class c and each of the N samples needed, we:

1. Generate κ candidate samples using learnability-guided diffusion
2. Score candidates using the learnability criterion: $S(\mathbf{x}, c) = \mathcal{L}(\theta_{i-1}, \mathbf{x}, c) - \omega \cdot \mathcal{L}(\theta^*, \mathbf{x}, c)$
3. Select the highest-scoring sample and add it to both the increment $\mathcal{I}_i$ and memory $\mathcal{M}_c$

This greedy selection ensures that each newly selected sample benefits from deviation guidance against all previously selected samples within the same increment, maximizing intra-increment diversity.

Reference Model. For learnability scoring, we trained a reference model on the full dataset without MixCut for 300 epochs on the target dataset ImageNette or ImageWoof. For ImageNet-1K we used a pretrained ResNet-18 network provided by PyTorch.

Learner Training. We train the learner from scratch using the initial increment $\mathcal{D}_1$ (always at IPC 10). Training proceeds until the validation loss plateaus, using a patience of 300. We adopt the AdamW optimizer with a learning rate of 0.001, $\beta_0 = 0.9$, $\beta_1 = 0.999$, and a weight decay of 0.01. After convergence, we further fine-tune the model using a cosine-decay schedule with a minimum learning-rate ratio of 0.01 for 300 epochs, which effectively "squeezes" the training signal available from the current distilled data. In the case of ImageNet-1k we train the model 300 epochs using the setup described in Soft-Label Protocol in Sec. C

Learner Model for Synthesis. Because the learner is trained on a small distilled dataset, its instantaneous gradient updates can be noisy and may not reliably reflect the underlying training signal. To obtain a more stable representation of the model's evolving knowledge, we maintain an exponential moving average (EMA) of the learner throughout training. This EMA model is then used to compute learnability guidance, ensuring smoother and more reliable gradient estimates for synthesis.

Memory Management. The memory sets $\mathcal{M}_c$ play a crucial role in deviation guidance. Initialized from the seed dataset $\mathcal{D}_1$ (Algorithm 1, line 3), they are incrementally updated as new samples are selected (line 16). During diffusion sampling, the memory ensures generated samples push away from all previously synthesized examples of the same class, preventing redundancy and promoting diversity.

Algorithm 1: LGD Distillation

Input: Pretrained diffusion model G_ϕ; reference model $\theta^\star$;
seed dataset $\mathcal{D}_1$; number of classes C; images per class N;
number of increments K; over-generation factor κ;
guidance strength λ; reference weight ω; deviation strength γ.
Output: Distilled dataset $\mathcal{D}_S = \bigcup_{i=1}^{K} \mathcal{I}_i$.

// Initialize learner and per-class memory
1 $\theta_1 \leftarrow \textsc{Train}(\mathcal{D}_1)$
2 **for** $c \in \{1, \dots, C\}$ **do**
3 $\mathcal{M}_c \leftarrow \{(\mathbf{x}, y) \in \mathcal{D}_1 \mid y = c\}$

// Incremental distillation
4 **for** $i \leftarrow 2$ **to** K **do**
5 $\mathcal{I}_i \leftarrow \emptyset$
6 **for** $c \leftarrow 1$ **to** C **do**
7 **for** $n \leftarrow 1$ **to** N **do**
8 $\mathcal{C}_c^n \leftarrow \emptyset$
 // Generate κ candidates for position n in class c
9 **for** $k \leftarrow 1$ **to** κ **do**
10 sample $\mathbf{x}_{c,n,k} \sim G_\phi(\cdot \mid \theta_{i-1}, \theta^\star, c, \lambda, \omega, \gamma, \mathcal{M}_c)$
11 $\mathcal{C}_c^n \leftarrow \mathcal{C}_c^n \cup \{(\mathbf{x}_{c,n,k}, c)\}$
 // Score candidates by learnability
12 **foreach** $(\mathbf{x}, c) \in \mathcal{C}_c^n$ **do**
13 $S(\mathbf{x}, c) \leftarrow \mathcal{L}(\theta_{i-1}, \mathbf{x}, c) - \omega\, \mathcal{L}(\theta^\star, \mathbf{x}, c)$
14 $(\mathbf{x}_c^\star, c) \leftarrow \arg\max_{(\mathbf{x},c) \in \mathcal{C}_c^n} S(\mathbf{x}, c)$
 // Add selected sample to increment and memory
15 $\mathcal{I}_i \leftarrow \mathcal{I}_i \cup \{(\mathbf{x}_c^\star, c)\}$
16 $\mathcal{M}_c \leftarrow \mathcal{M}_c \cup \{(\mathbf{x}_c^\star, c)\}$
 // Update cumulative dataset and retrain learner
17 $\mathcal{D}_i \leftarrow \mathcal{D}_{i-1} \cup \mathcal{I}_i$
18 $\theta_i \leftarrow \textsc{Train}(\mathcal{D}_i)$
19 $\mathcal{D}_S \leftarrow \mathcal{D}_K$

B.2. Incremental Evaluation During Synthesis

During the learnability-guided distillation process (Algorithm 1), the learner model θ_i is trained incrementally on the growing distilled dataset $\mathcal{D}_i$ at each stage (line 18). We can evaluate this learner model on the test set and report its top-1 accuracy at the corresponding IPC. This provides insight into the quality of the distilled dataset as it is being constructed, showing how model performance evolves with each added increment. For example, after generating increment $\mathcal{I}_2$ and training θ_2 on $\mathcal{D}_2 = \mathcal{D}_1 \cup \mathcal{I}_2$ (20 IPC total),

Algorithm 2: LGD Sampling

Input: Current learner θ_{i-1}; reference model $\theta^\star$; class c; guidance strength λ; reference weight ω; deviation strength γ; diffusion model G_ϕ; number of timesteps T; per-class memory $\mathcal{M}_c$.
Output: Generated sample $\mathbf{x}_0$.

// Initialize from Gaussian noise
Sample $\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
for $t \leftarrow T$ **to** 1 **do**
  // Standard diffusion denoising prediction
  $\boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c) \leftarrow G_\phi(\text{predict-noise} \mid \mathbf{x}_t, t, c)$
  if $t \in [10, 45]$ **then**
    // Compute learnability score
    $S(\mathbf{x}_t, c) \leftarrow \mathcal{L}(\theta_{i-1}, \mathbf{x}_t, c) - \omega\, \mathcal{L}(\theta^\star, \mathbf{x}_t, c)$
    // Timestep-dependent scaling (normalizes gradient magnitudes)
    $\rho_t \leftarrow \dfrac{\sqrt{1-\bar{\alpha}_t} \cdot \|\boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c)\|}{\|\nabla_{\mathbf{x}_t} S(\mathbf{x}_t, c)\|}$
    // Learnability guidance
    $\tilde{\boldsymbol{\epsilon}}_\phi(\mathbf{x}_t, t, c) \leftarrow \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c) + \lambda\, \rho_t \nabla_{\mathbf{x}_t} S(\mathbf{x}_t, c)$
    // Deviation guidance for diversity
    $\tilde{\mathbf{x}}^\star \leftarrow \arg\min_{\tilde{\mathbf{x}} \in \mathcal{M}_c} \|\mathbf{x}_t - \tilde{\mathbf{x}}\|$
    $\mathcal{G}_D(\mathbf{x}_t) \leftarrow \dfrac{\mathbf{x}_t \cdot \tilde{\mathbf{x}}^\star}{\|\mathbf{x}_t\|\, \|\tilde{\mathbf{x}}^\star\|}$
    $\tilde{\boldsymbol{\epsilon}}_\phi(\mathbf{x}_t, t, c) \leftarrow \tilde{\boldsymbol{\epsilon}}_\phi(\mathbf{x}_t, t, c) - \gamma\, \nabla_{\mathbf{x}_t} \mathcal{G}_D(\mathbf{x}_t)$
  else
    $\tilde{\boldsymbol{\epsilon}}_\phi(\mathbf{x}_t, t, c) \leftarrow \boldsymbol{\epsilon}_\phi(\mathbf{x}_t, t, c)$
  // Reverse diffusion update
  $\boldsymbol{\mu}_\phi(\mathbf{x}_t) \leftarrow \dfrac{1}{\sqrt{1-\beta_t}} \left(\mathbf{x}_t - \dfrac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \tilde{\boldsymbol{\epsilon}}_\phi(\mathbf{x}_t, t, c) \right)$
  Sample $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_{t-1} \leftarrow \boldsymbol{\mu}_\phi(\mathbf{x}_t) + \sigma_t \mathbf{z}$
return $\mathbf{x}_0$

we can evaluate θ_2 and report accuracy at IPC 20 before proceeding to generate increment $\mathcal{I}_3$.

This incremental evaluation is distinct from the standard static evaluation protocol used in Tables 1-3 of the main paper, where a fresh model is trained from scratch on the complete final distilled dataset $\mathcal{D}_K$. The incremental evaluation uses the same model that guides the synthesis process—the learner that has been progressively trained on $\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_i$—whereas static evaluation trains a new model only on $\mathcal{D}_K$. The incremental accuracy curve $\{(i \cdot N, \text{Acc}(\theta_i))\}_{i=1}^K$ shown in Fig. 4c and Tab. 4 directly reflects the learning dynamics during distillation, revealing sustained marginal gains as each curriculum-aligned increment is added.

Table 8. Hyperparameter settings for data synthesis and learnability scoring.

Config	Value	Explanation
(a) Diffusion Synthesis		
Sampler	DDPM	–
Sampling Steps	50	–
CFG Scale	4.0	Classifier-free guidance
(b) Learnability Guidance		
Learner Arch.	ResNet-10-AP	–
Reference Arch.	ResNet-10-AP	Same as learner
Ref. Weight ω	0.5	Eq. 7 in main paper
Guidance λ	15	Eq. 8 in main paper
Deviation γ	50	Diversity control
Learnability Batch K	3	Scoring batch size
(c) Incremental Configuration		
Samples/Increment	10	samples/class
Seed Init.	IGD	Initial 10 IPC

C. Implementation Details

Here, we provide comprehensive implementation details for reproducibility of our experiments, including network architectures, hyperparameter settings. All experiments are conducted in a single NVIDIA H100 (80GB) GPU.

C.1. Diffusion Model Configuration

For diffusion model pre-training, we adopt DiT-XL/2 256×256 following the settings [5, 10]. Generated images are at 256×256 resolution. We use 50 DDPM sampling steps with classifier-free guidance scale 4.0. Generated images are clipped to $[-1, 1]$ and saved as PNG format.

C.2. Evaluation Settings

We outline the evaluation settings used to assess the performance of the distilled datasets.

C.2.1. Network Architectures

We conduct experiments on three commonly adopted network architectures in the area of dataset distillation:

1. **ConvNet-6** is a 6-layer convolutional network. The network contains 128 feature channels in each layer, and instance normalization is adopted.
2. **ResNetAP-10** is a 10-layer ResNet, where the strided convolution is replaced by average pooling for downsampling.
3. **ResNet-18** is a 18-layer ResNet with instance normalization.

Hard-Label Protocol. For validation training, we follow the protocol established in [5]. We use AdamW with a

learning rate of 0.001, momentum parameters $\beta_0 = 0.9$ and $\beta_1 = 0.999$, and a weight decay of 0.01. The number of training epochs for each IPC setting is 2000, 1500, 1500, 1500, 1500, and 1000 for IPC 10, 20, 30, 40, 50, and 100, respectively. We apply learning rate decays at the 2/3 and 5/6 points of training, using a decay factor (gamma) of 0.2.

Soft-Label Protocol. We follow the evaluation protocol used by [10, 29] for ImageNet-1k evaluation. We train ResNet-18 for 300 epochs using AdamW optimizer with learning rate 0.001, weight decay 0.01, and parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We employ smoothing LR schedule with rand augment and RandomHorizontalFlip.

D. Qualitative Analysis

We provide qualitative visualizations comparing samples generated by our learnability-guided approach against baseline methods. Figs. 11 to 14 show representative samples from ImageNette and ImageWoof datasets across different methods.

These visualizations reveal that our method generates samples with greater diversity and semantic richness compared to baseline approaches. While vanilla diffusion models and IGD tend to produce visually similar samples within each increment, our learnability-guided synthesis yields distinct variations that capture complementary visual features, consistent with our quantitative complementarity analysis in Sec. A.2.

Figure 11. **Visual diversity in incrementally distilled datasets.** Samples from increments $\mathcal{I}_1 - \mathcal{I}_5$ (50 IPC total) of the Parachute class.

Figure 12. **Visual diversity in incrementally distilled datasets.** Samples from increments $\mathcal{I}_1 - \mathcal{I}_5$ (50 IPC total) of the Tench class.

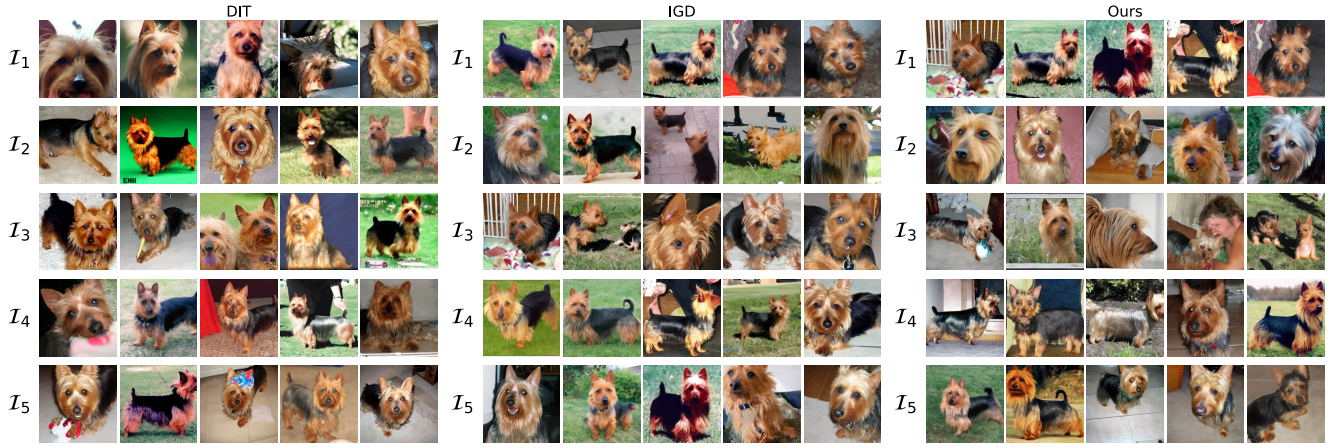


Figure 13. **Visual diversity in incrementally distilled datasets.** Samples from increments $\mathcal{I}_1 - \mathcal{I}_5$ (50 IPC total) of the Australian terrier class.

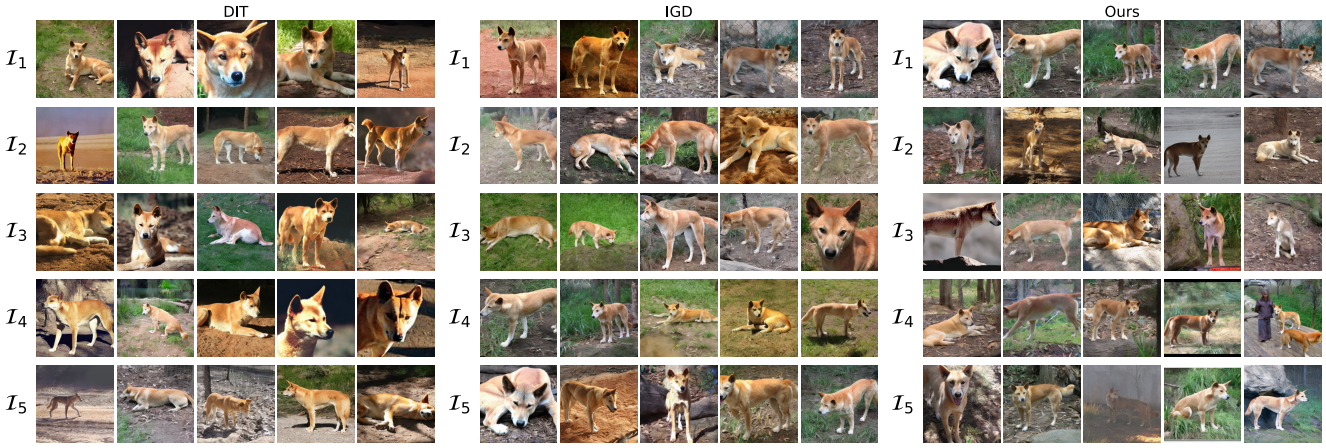


Figure 14. **Visual diversity in incrementally distilled datasets.** Samples from increments $\mathcal{I}_1 - \mathcal{I}_5$ (50 IPC total) of the Dingo class.