Self-Supervised Representation-Guided Generative Dataset Distillation

Mingzhuo Li[1], Guang Li[1]*, Linfeng Ye[2], Jiafeng Mao[3], Takahiro Ogawa[1], Konstantinos N. Plataniotis[2], and Miki Haseyama[1]

[1]Hokkaido University, Japan
[2]University of Toronto, Canada
[3]University of Tokyo, Japan
{mingzhuo, guang, ogawa, mhaseyama}@lmd.ist.hokudai.ac.jp
linfeng.ye@mail.utoronto.ca, kostas@ece.utoronto.ca, mao@hal.t.u-tokyo.ac.jp

Abstract

Dataset distillation compresses a large training set into a compact synthetic set while retaining its downstream utility. Most existing methods target randomly initialized networks, whereas modern vision systems often adapt frozen pretrained encoders with lightweight modules. Distilled samples should therefore preserve the discriminative geometry of the pretrained representation space, which existing generative objectives do not explicitly consider. We propose self-supervised representation-guided generative dataset distillation (SRG), a framework that translates the SSL geometry into diffusion guidance. Specifically, SRG constructs class-wise prototypes from real-image SSL representations and performs guidance through three SSL-space objectives for prototype alignment, inter-class discrimination, and intra-class assignment. During diffusion sampling, it adopts a stage-wise guidance strategy: early denoising is anchored to the latent of the real image whose SSL representation is nearest to the assigned prototype, whereas later denoising is guided by the SSL-space objectives. This division preserves the visual realism provided by the generative prior while progressively steering samples toward representative and class-discriminative regions of the SSL representation space. SRG consistently outperforms the evaluated generative baselines across multiple datasets and IPC settings. A cross-encoder evaluation further indicates transfer across pretrained representation spaces. These results demonstrate the effectiveness of representation-guided generation for dataset distillation with pretrained SSL models.

Introduction

The rapid progress of computer vision has been supported by increasingly large models and training datasets (Schmidhuber 2015; Talaei Khoei, Ould Slimane, and Kaabouch 2023). However, repeatedly storing, processing, and accessing such datasets incurs substantial computational and storage costs, particularly when models must be trained or evaluated across multiple experimental settings (Strubell, Ganesh, and McCallum 2019). Dataset distillation (DD) (Wang et al. 2018; Li, Zhao, and Wang 2022) provides a potential solution by compressing a large dataset into a much smaller synthetic set. Ideally, models trained on the distilled dataset should retain much of the performance achieved using the original data, thereby reducing the cost of repeated model training.

*Correspondence to Guang Li <guang@lmd.ist.hokudai.ac.jp>

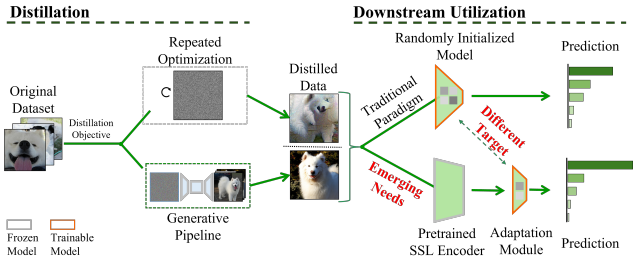


Figure 1: Overview of dataset distillation and downstream utilization. Traditional DD targets training from random initialization, whereas adaptation with pretrained SSL encoders requires distilled samples that preserve discriminative representation structure.

As illustrated on the left side of Fig. 1, existing DD methods can be broadly divided into non-generative and generative approaches. Non-generative methods directly optimize synthetic data as free variables, typically by matching training-related information, such as model gradients (Zhao and Bilen 2021b,a; Li et al. 2023b, 2024a), between the real and distilled datasets. In contrast, generative DD methods construct distilled samples by incorporating distillation objectives into a pretrained generative pipeline (Li et al. 2024b; Su et al. 2024; Wu et al. 2025; Ye et al. 2025; Zou et al. 2025; Cai et al. 2026). By generating samples on demand and avoiding the direct optimization of a fixed set of synthetic data, generative methods provide a flexible way to construct distilled datasets for different images-per-class (IPC) budgets.

As shown on the right side of Fig. 1, DD has traditionally been formulated for training randomly initialized models. Modern vision systems, however, increasingly reuse pretrained self-supervised learning (SSL) models as fixed feature encoders and train only lightweight task-specific modules, such as linear probes, on the extracted representations (Chen et al. 2020; Gui et al. 2024). This shift changes the objective of dataset distillation. Rather than reproducing the complete training dynamics of a model trained from scratch, distilled samples should provide compact supervision that supports effective decision boundaries in a fixed representation space. Therefore, they need to preserve not only visual

characteristics but also the representative and discriminative structure emphasized by the pretrained SSL encoder.

Most existing DD methods are not explicitly designed for this objective. LGM (Cazenavette, Torralba, and Sitzmann 2025) optimizes synthetic samples by matching gradients induced by a downstream linear classifier, while CLP-DD (Peng et al. 2026) improves efficiency using a closed-form solver and a discriminative outer objective. Although these methods demonstrate the feasibility of DD for pretrained encoders, both optimize a fixed synthetic set for each dataset and IPC budget, requiring a new optimization process when either changes. Generative DD offers a more flexible alternative, but existing objectives are mainly designed for training models from scratch. For example, MGD3 (Chan-Santiago et al. 2025) guides diffusion sampling toward modes obtained in the generator's latent space. These modes capture visual variation learned by the generator but do not explicitly retain the discriminative geometry of the downstream SSL encoder.

To address this gap, we propose self-supervised representation-guided generative dataset distillation (SRG), a framework that translates the SSL geometry into distillation signals to guide diffusion sampling. SRG constructs class-wise prototypes from real-image representations and assigns each denoising trajectory to one prototype. It then applies stage-wise guidance to incorporate both visual and representation-level information. During early denoising stages, latent-space guidance anchors the trajectory to a real-image latent associated with the assigned prototype. During later denoising stages, SSL-space guidance refines the generated representation through prototype alignment, inter-class discrimination, and intra-class assignment. These objectives align generated samples with representative regions, separate them from other classes, and encourage distinct assignments within each class. In this way, SRG directly incorporates downstream representation structure into diffusion sampling, obtaining distilled datasets tailored for SSL models.

Our main contributions are summarized as follows:

- We propose SRG, a framework that incorporates the discriminative geometry of pretrained SSL representations into diffusion sampling for dataset distillation.
- We develop a stage-wise guidance strategy that combines early latent-space anchoring with later SSL-space guidance using prototype alignment, inter-class discrimination, and intra-class assignment.
- We conduct extensive experiments across multiple datasets, IPC settings, and SSL encoders, demonstrating the effectiveness, scalability, and cross-encoder transfer of SRG.

Related Work

Pretrained Vision Encoders

Self-supervised learning (SSL) learns transferable visual representations from large-scale data without requiring manually annotated class labels (Gui et al. 2024; Jaiswal et al. 2021). Pretrained encoders such as CLIP (Radford et al. 2021), DINO (Caron et al. 2021), MoCo (He et al. 2020), and EVA (Fang et al. 2023) have become widely used backbones for visual recognition. A common adaptation strategy is to freeze the encoder and train only a lightweight task-specific module on the extracted representations (Chen et al. 2020). Under this setting, the utility of a compact training set depends on whether its samples preserve representative and class-discriminative regions in the pretrained representation space, motivating representation-aware dataset distillation.

Dataset Distillation

Dataset distillation (DD) (Wang et al. 2018) synthesizes a compact dataset that retains the training utility of a much larger original dataset. DD has been applied to various downstream tasks, including fine-grained recognition (Ma et al. 2026), multimodal learning (Li et al. 2025a), and privacy-preserving learning (Li et al. 2020, 2022, 2023a). However, most existing methods are designed for training randomly initialized models (Yu, Liu, and Wang 2023; Liu and Du 2025), although recent studies have begun to consider downstream adaptation with frozen pretrained encoders.

Non-generative methods. Non-generative DD methods typically optimize a fixed set of synthetic images by matching model gradients (Zhao and Bilen 2021b), training trajectories (Cazenavette et al. 2022), feature statistics or distributions (Wang et al. 2022; Zhao and Bilen 2023; Li et al. 2025b), or kernel-based quantities (Nguyen, Chen, and Lee 2021). LGM (Cazenavette, Torralba, and Sitzmann 2025) extends gradient matching to pretrained encoders using gradients from a downstream linear classifier together with Differentiable Siamese Augmentation (Zhao and Bilen 2021a). CLP-DD (Peng et al. 2026) instead employs a closed-form linear-probe solver with a discriminative outer objective. Although both methods achieve strong performance, they optimize a fixed synthetic set for each target dataset and IPC budget, generally requiring re-optimization when either setting changes.

Generative methods. Generative DD methods exploit pretrained generative models to reduce reliance on directly optimizing a fixed synthetic set. IT-GAN (Zhao and Bilen 2022) optimizes informative latent codes of a frozen generative adversarial network, while Minimax (Gu et al. 2024) incorporates representativeness and diversity objectives into the fine-tuning of generative models. MGD3 (Chan-Santiago et al. 2025) and IGD (Chen et al. 2025) modify diffusion sampling using latent-mode and trajectory-influence guidance, respectively, whereas CaO2 (Wang et al. 2025) refines generated samples through post-processing. However, these methods primarily target models trained from scratch and do not explicitly incorporate the representation geometry of a pretrained SSL encoder. SRG addresses this gap by incorporating SSL-space prototypes and objectives as distillation signals to guide diffusion sampling.

Method

Preliminaries

We employ a pretrained diffusion transformer (DiT) (Peebles and Xie 2023) that operates in the latent space of a variational autoencoder (VAE) (Kingma and Welling 2013). Starting from random noise, DiT generates images through iterative denoising.

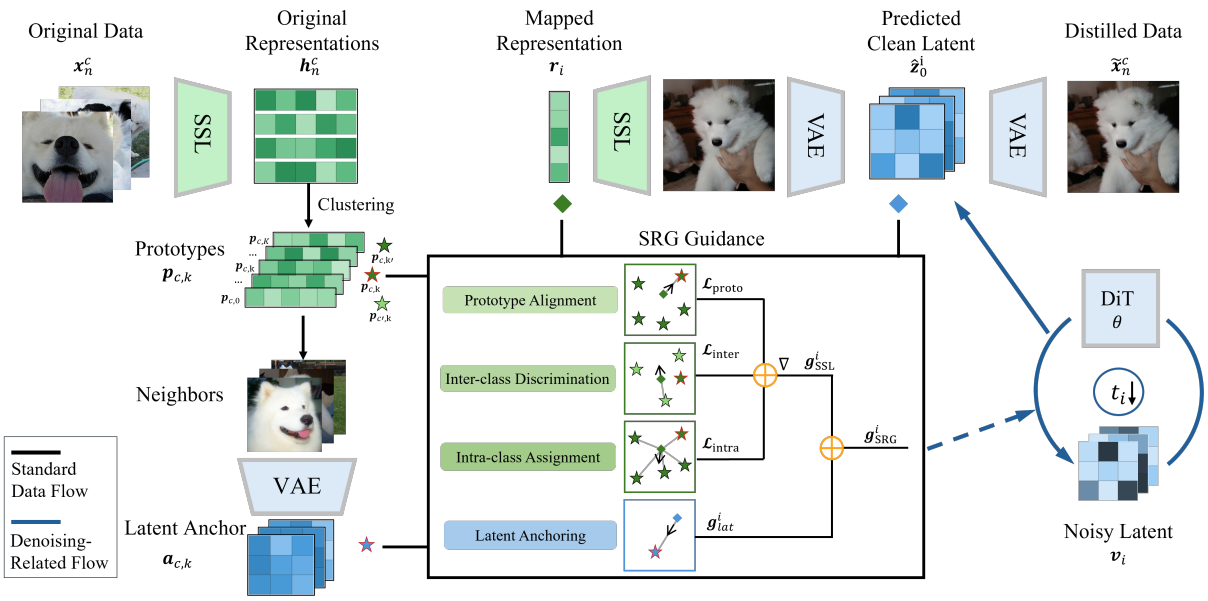


Figure 2: Overview of SRG. Class-wise prototypes are constructed by clustering real-image representations extracted by a frozen SSL encoder, and each denoising trajectory is assigned to one target prototype. Latent-space guidance anchors the trajectory to a prototype-associated real-image latent, while SSL-space guidance improves prototype alignment, inter-class discrimination, and intra-class assignment.

Let $\boldsymbol{v}_i$ denote the noisy latent at the i-th denoising step, corresponding to diffusion timestep t_i. Given the class condition $\boldsymbol{c}$, the pretrained DiT parameterized by θ predicts the noise component in $\boldsymbol{v}_i$. The next latent $\boldsymbol{v}_{i+1}$ is obtained following the Gaussian reverse transition:

$$\boldsymbol{v}_{i+1} = \mu_\theta(\boldsymbol{v}_i, t_i, \boldsymbol{c}) + \sigma_{t_i}\boldsymbol{\epsilon}_i, \tag{1}$$

where $\mu_\theta(\boldsymbol{v}_i, t_i, \boldsymbol{c})$ is the conditional mean of the next latent computed by the DiT sampler. σ_{t_i} is the standard deviation of the reverse transition, and $\boldsymbol{\epsilon}_i$ is the Gaussian noise sampled at the i-th step. At each step, the sampler also estimates the clean latent $\hat{\boldsymbol{z}}_0^i$, which is decoded by VAE to the distilled image $\tilde{\boldsymbol{x}}_n^c$ after denoising.

Although DiT can generate realistic images, the standard sampling process does not explicitly incorporate a distillation objective. MGD3 (Chan-Santiago et al. 2025) addresses this limitation by discovering representative latent modes in the original dataset and aligning each generated sample with a specific mode. Given a target budget of IPC $= K$, MGD3 performs K-means clustering in the VAE latent space separately for each class c to obtain a set of latent modes $\mathcal{M}_c = \{\boldsymbol{m}_{c,k}\}_{k=1}^K$. Each generated sample is assigned to one mode $\boldsymbol{m}_{c,k}$ so that the generated set covers multiple regions of the original data distribution. For clarity, we omit the indices (c, k) from the trajectory variables in the following formulations, and define the MGD3 guidance term as follows:

$$\boldsymbol{g}^i_{\text{MGD3}} = -\lambda_{\text{lat}}\sigma_{t_i}\left(\hat{\boldsymbol{z}}_0^{\,i} - \boldsymbol{m}_{c,k}\right), \tag{2}$$

where λ_{lat} controls the guidance strength, and σ_{t_i} is used for timestep-dependent scaling. The guidance is added to the standard reverse transition in Eq. 1:

$$\boldsymbol{v}_{i+1} = \mu_\theta(\boldsymbol{v}_i, t_i, \boldsymbol{c}) + \sigma_{t_i}\boldsymbol{\epsilon}_i + \boldsymbol{g}^i_{\text{MGD3}}. \tag{3}$$

By assigning different generation trajectories to distinct latent modes, MGD3 is designed to improve the coverage of the distilled dataset. However, using modes constructed in the latent space, its guidance does not explicitly preserve the discriminative structure required for downstream adaptation of pretrained SSL encoders.

Prototype Alignment

As shown in Fig. 2, we address this limitation by complementing latent-space anchoring with guidance objectives derived from the SSL representation space. Specifically, we construct class-wise SSL prototypes and enable guidance through differentiable objectives whose gradients are backpropagated to the predicted clean latent. Unlike latent modes used in MGD3, which characterize the visual distribution encoded by the generator, SSL prototypes summarize the structure emphasized by the SSL encoder, providing guidance that better aligns with the downstream adaptation setting. For each real image $\boldsymbol{x}_n^c$ from class c, we extract the ℓ_2-normalized representation $\boldsymbol{h}_n^c = \text{norm}_2(f_\phi(\boldsymbol{x}_n^c))$ using the frozen SSL encoder f_ϕ, where $\text{norm}_2(\boldsymbol{a}) = \boldsymbol{a}/\|\boldsymbol{a}\|_2$. Normalization removes feature-magnitude effects, avoiding encoder-specific magnitude calibration in the guidance objectives. The class-wise prototypes are obtained by applying spherical K-means within each class as follows:

$$\{\boldsymbol{p}_{c,k}\}_{k=1}^K = \text{SphKMeans}\left(\{\boldsymbol{h}_n^c\}_{n=1}^{N_c}; K\right), \tag{4}$$

where $\boldsymbol{p}_{c,k}$ denotes the normalized k-th prototype of class c and N_c denotes the number of images in class c. Each prototype represents a local region of the class distribution in the SSL representation space. Accordingly, each generation trajectory is assigned to one prototype.

However, the predicted clean latent $\hat{\boldsymbol{z}}_0^i$ and the SSL prototype $\boldsymbol{p}_{c,k}$ lie in different spaces and cannot be compared directly. To solve this problem, we map the predicted clean latent to an SSL representation $\boldsymbol{r}_i = \mathrm{norm}_2(f_\phi(D_\psi(\hat{\boldsymbol{z}}_0^i)))$, where D_ψ denotes the VAE decoder. Since both $\boldsymbol{r}_i$ and $\boldsymbol{p}_{c,k}$ are normalized, their cosine similarity can be measured by:

$$\mathrm{sim}(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^T \boldsymbol{b}. \tag{5}$$

To convert this SSL-space similarity into guidance for diffusion sampling, we formulate differentiable representation objectives and use their gradient as guidance directions. First, we introduce the following prototype alignment objective to drive the mapped representation of the generated sample toward the assigned prototype:

$$\mathcal{L}_{\mathrm{proto}}^i = 1 - \mathrm{sim}\left(\boldsymbol{r}_i, \boldsymbol{p}_{c,k}\right). \tag{6}$$

Minimizing this objective aligns each generated representation with a local, representative region of the corresponding class in the SSL representation space.

Inter-Class Discrimination

The prototype alignment objective pulls each generated representation toward its assigned prototype. However, for semantically similar classes, an assigned prototype may lie near regions occupied by neighboring classes, allowing the generated representation to cross the class boundary. To explicitly preserve separation, we introduce an inter-class discrimination objective. Specifically, we measure the similarity between the generated representation $\boldsymbol{r}_i$ and all prototypes assigned to other classes. Their effects are aggregated using log-sum-exp, which places greater emphasis on highly similar prototypes while retaining contributions from the remaining ones. The objective is defined as follows:

$$\mathcal{L}_{\mathrm{inter}}^i = \log \sum_{\substack{c'=1 \\ c' \neq c}}^{C} \sum_{k'=1}^{K} \exp\left(\frac{\mathrm{sim}\left(\boldsymbol{r}_i,\ \boldsymbol{p}_{c',k'}\right)}{\tau_{\mathrm{inter}}}\right), \tag{7}$$

where τ_{inter} is a temperature parameter controlling the concentration of the objective. A smaller temperature makes the objective more sensitive to the most similar prototype, whereas a larger temperature distributes weight more evenly across prototypes. Minimizing $\mathcal{L}_{\mathrm{inter}}^i$ reduces similarity between the generated representation and prototypes of other classes, thereby encouraging class-level separation in the SSL representation space.

Intra-Class Assignment

Despite these efforts, multiple generation trajectories of the same class may still converge to similar regions, particularly when the pretrained generator strongly favors certain visual patterns. This tendency can weaken correspondence with the assigned prototypes in higher-IPC settings. We therefore introduce an intra-class assignment objective to make the generated representation more similar to the assigned prototype than to other prototypes of the same class. For the generation trajectory assigned to $\boldsymbol{p}_{c,k}$, the objective is defined using a cross-entropy loss as follows:

$$\mathcal{L}_{\mathrm{intra}}^i = -\log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{r}_i, \boldsymbol{p}_{c,k}\right)/\tau_{\mathrm{intra}}\right)}{\sum_{k'=1}^{K} \exp\left(\mathrm{sim}\left(\boldsymbol{r}_i, \boldsymbol{p}_{c,k'}\right)/\tau_{\mathrm{intra}}\right)}, \tag{8}$$

where τ_{intra} is the temperature parameter controlling the sharpness of the prototype assignment. Minimizing $\mathcal{L}_{\mathrm{intra}}^i$ increases similarity to the assigned prototype relative to competing prototypes, thereby strengthening the assignment and discouraging different trajectories from converging to the same region.

Overall Generation

We combine the three representation objectives into the overall SSL-space guidance objective:

$$\mathcal{L}_{\mathrm{SSL}}^i = \mathcal{L}_{\mathrm{proto}}^i + \mathcal{L}_{\mathrm{inter}}^i + \mathcal{L}_{\mathrm{intra}}^i. \tag{9}$$

Since all representations are ℓ_2-normalized, the similarity terms are defined on a consistent range. For simplicity, we use the same weights for all three objectives and rely on the temperature parameters to modulate the effects of the discrimination and assignment terms. The SSL-space guidance is obtained by backpropagating the objective to the predicted clean latent as follows:

$$\boldsymbol{g}_{\mathrm{SSL}}^i = -\lambda_{\mathrm{SSL}} \nabla_{\hat{\boldsymbol{z}}_0^i} \mathcal{L}_{\mathrm{SSL}}^i, \tag{10}$$

where λ_{SSL} controls the guidance strength.

The reliability of the two guidance signals varies over the denoising stages. Early stages form image structure, where latent-space anchoring is effective yet the predicted latent is uncertain for stable semantic comparison. As denoising progresses, representation becomes increasingly discriminative (Kim, Thomas, and Ghadiyaram 2025), making SSL more informative and structural anchoring less critical. We therefore separate the two signals with the following schedule:

$$\begin{aligned} \boldsymbol{g}_{\mathrm{SRG}}^i = \mathbb{I}[t_i \geq t_{\mathrm{lat}}]\, \boldsymbol{g}_{\mathrm{lat}}^i \\ + \mathbb{I}[t_i < t_{\mathrm{SSL}}]\, \boldsymbol{g}_{\mathrm{SSL}}^i, \end{aligned} \tag{11}$$

where $\mathbb{I}[\cdot]$ denotes the indicator function, and t_{lat} and t_{SSL} determine the timesteps at which latent-space guidance stops and SSL-space guidance starts, respectively. The latent-space guidance g_{lat}^i follows Eq. 2 with $\boldsymbol{m}_{c,k}$ replaced by $\boldsymbol{a}_{c,k}$, the latent of the real image whose SSL representation is closest to $\boldsymbol{p}_{c,k}$ in cosine similarity, serving as a trajectory anchor rather than the guidance target in the original definition.

The final guided reverse transition is defined as follows:

$$\boldsymbol{v}_{i+1} = \mu_\theta(\boldsymbol{v}_i, t_i, \boldsymbol{c}) + \sigma_{t_i} \boldsymbol{\epsilon}_i + \boldsymbol{g}_{\mathrm{SRG}}^i. \tag{12}$$

This formulation combines latent-space anchoring with SSL-space prototype alignment, inter-class discrimination, and intra-class assignment, yielding a DD framework tailored to pretrained SSL models. The stage-wise schedule uses the generator-facing signal when the trajectory is structurally uncertain and the encoder-facing signal after semantic content emerges. SRG thereby retains the visual plausibility of diffusion generation while explicitly steering samples toward the discriminative structure used for downstream adaptation.

Table 1: Comparison of downstream validation accuracy between SRG and baseline methods on different ImageNet benchmarks using DINOv2. Random denotes random sampling from the original dataset, while Neighbor denotes selection of the real sample nearest to the assigned SSL prototype. The best result in each setting is highlighted in bold.

Method	ImageNet-IDC			ImageNet-100			ImageNet-1K		
	IPC=1	IPC=3	IPC=5	IPC=1	IPC=3	IPC=5	IPC=1	IPC=3	IPC=5
Random	$85.6_{\pm 2.5}$	$96.2_{\pm 0.8}$	$97.0_{\pm 0.5}$	$77.8_{\pm 0.3}$	$86.2_{\pm 0.1}$	$89.3_{\pm 0.1}$	$53.7_{\pm 0.1}$	$67.5_{\pm 0.1}$	$70.8_{\pm 0.1}$
Neighbor	$93.7_{\pm 0.7}$	$96.9_{\pm 0.6}$	$97.8_{\pm 0.4}$	$86.8_{\pm 0.2}$	$91.2_{\pm 0.1}$	$92.5_{\pm 0.1}$	$71.5_{\pm 0.1}$	$75.6_{\pm 0.0}$	$76.5_{\pm 0.1}$
DiT	$92.2_{\pm 0.9}$	$95.2_{\pm 1.1}$	$95.2_{\pm 0.7}$	$82.8_{\pm 0.1}$	$89.1_{\pm 0.4}$	$89.0_{\pm 0.1}$	$65.5_{\pm 0.2}$	$71.2_{\pm 0.1}$	$72.4_{\pm 0.0}$
MGD3	$88.6_{\pm 1.2}$	$95.6_{\pm 0.6}$	$96.6_{\pm 0.5}$	$81.7_{\pm 0.3}$	$87.8_{\pm 0.1}$	$90.2_{\pm 0.1}$	$65.1_{\pm 0.0}$	$72.0_{\pm 0.0}$	$73.8_{\pm 0.1}$
IGD	$86.6_{\pm 2.5}$	$94.2_{\pm 1.0}$	$95.7_{\pm 0.8}$	$82.7_{\pm 0.3}$	$87.7_{\pm 0.2}$	$89.6_{\pm 0.2}$	$66.0_{\pm 0.1}$	$71.9_{\pm 0.1}$	$73.9_{\pm 0.0}$
LGM	$97.0_{\pm 0.5}$	$98.2_{\pm 0.3}$	$98.3_{\pm 0.2}$	$\mathbf{91.0_{\pm 0.1}}$	$91.7_{\pm 0.1}$	OOM	$75.0_{\pm 0.1}$*	OOM	OOM
CLP-DD	$\mathbf{98.2_{\pm 0.2}}$	$\mathbf{98.4_{\pm 0.2}}$	$98.4_{\pm 0.1}$	$89.2_{\pm 0.1}$	$91.3_{\pm 0.1}$	$91.8_{\pm 0.0}$	$75.6_{\pm 0.1}$*	OOM	OOM
SRG	$95.3_{\pm 0.4}$	$98.1_{\pm 0.5}$	$\mathbf{98.5_{\pm 0.4}}$	$89.0_{\pm 0.1}$	$\mathbf{92.3_{\pm 0.1}}$	$\mathbf{93.2_{\pm 0.2}}$	$\mathbf{73.5_{\pm 0.1}}$	$\mathbf{76.3_{\pm 0.0}}$	$\mathbf{77.6_{\pm 0.1}}$
Full	$99.7_{\pm 0.1}$	$99.7_{\pm 0.1}$	$99.7_{\pm 0.1}$	$95.2_{\pm 0.1}$	$95.2_{\pm 0.1}$	$95.2_{\pm 0.1}$	$83.0_{\pm 0.0}$	$83.0_{\pm 0.0}$	$83.0_{\pm 0.0}$

* We report results from the corresponding original papers because reproduction exceeded the available GPU memory.

Experiments

Experimental Settings

For evaluation, we freeze each pretrained SSL encoder and repeat linear-probe training five times, with $1,000$ epochs per run. All evaluated datasets are derived from ImageNet-1K (Deng et al. 2009; Russakovsky et al. 2015), a large-scale benchmark commonly used in dataset distillation. MGD3 (Chan-Santiago et al. 2025), IGD, and SRG use the same pretrained DiT model (Peebles and Xie 2023). Unless otherwise specified, SRG uses $t_{\text{lat}} = 20$, $t_{\text{SSL}} = 20$, $\lambda_{\text{lat}} = 0.1$, $\lambda_{\text{SSL}} = 15$, $\tau_{\text{inter}} = 1.0$, and $\tau_{\text{intra}} = 5.0$ across all datasets and IPC settings. Each SSL encoder uses a ViT-B (Dosovitskiy et al. 2021) backbone with the patch size specified by the corresponding pretrained model. The prototypes are constructed on the GPU using an encoding batch size of 64. All experiments are conducted on a single NVIDIA RTX A6000 GPU, except for the ImageNet-100 LGM experiment at $\text{IPC} = 3$, which uses four A6000 GPUs due to OOM issues. Further implementation details are provided in the supplementary material.

Benchmark Results

We first compare SRG with representative methods on ImageNet-IDC (Kim et al. 2022), ImageNet-100 (Kim et al. 2022), and ImageNet-1K (Russakovsky et al. 2015) under IPC settings of 1, 3, and 5, using DINOv2 (Oquab, Darcet, and Moutakanni 2024) as the distillation and downstream-evaluation encoder. The compared methods include two selection-based baselines: random sampling from the original dataset (Random) and selection of the real sample nearest to each SSL prototype (Neighbor). For generative approaches, we include direct sampling from DiT (Peebles and Xie 2023), the latent-guidance-based MGD3 (Chan-Santiago et al. 2025), and the influence-guidance-based IGD (Chen et al. 2025). We further compare with SSL-oriented non-generative methods, including the pioneering LGM (Cazenavette, Torralba, and Sitzmann 2025) and the computationally efficient CLP-DD (Peng et al. 2026). We also report performance obtained with the full training set as an upper reference.

Table 2: Comparison of downstream validation accuracy between SRG and baseline methods on fine-grained ImageNet subsets under different IPC settings using DINOv2. The best result in each setting is highlighted in bold.

Subset	Method	IPC=1	IPC=3	IPC=5
Woof	DiT	$83.7_{\pm 1.7}$	$89.1_{\pm 0.4}$	$88.7_{\pm 0.6}$
	MGD3	$78.6_{\pm 1.1}$	$89.0_{\pm 0.3}$	$89.3_{\pm 0.6}$
	LGM	$\mathbf{88.5_{\pm 0.8}}$	$90.1_{\pm 0.3}$	$90.8_{\pm 0.3}$
	SRG	$88.4_{\pm 0.7}$	$\mathbf{91.4_{\pm 0.1}}$	$\mathbf{92.0_{\pm 0.5}}$
Fruits	DiT	$77.7_{\pm 2.1}$	$82.2_{\pm 0.6}$	$84.3_{\pm 0.5}$
	MGD3	$77.4_{\pm 0.6}$	$82.4_{\pm 1.1}$	$85.4_{\pm 1.0}$
	LGM	$\mathbf{86.6_{\pm 0.8}}$	$87.4_{\pm 0.9}$	$87.5_{\pm 0.6}$
	SRG	$85.4_{\pm 0.9}$	$\mathbf{88.1_{\pm 0.4}}$	$\mathbf{89.1_{\pm 0.7}}$
Instruments	DiT	$64.7_{\pm 1.5}$	$65.9_{\pm 0.4}$	$76.3_{\pm 0.9}$
	MGD3	$49.9_{\pm 2.0}$	$67.8_{\pm 0.8}$	$73.5_{\pm 1.0}$
	LGM	$\mathbf{70.6_{\pm 1.5}}$	$\mathbf{81.4_{\pm 0.4}}$	$82.3_{\pm 0.1}$
	SRG	$70.2_{\pm 1.4}$	$81.0_{\pm 0.6}$	$\mathbf{82.6_{\pm 0.2}}$

As shown in Table 1, SRG achieves the best performance among the evaluated generative methods in every reported setting, supporting the effectiveness of incorporating SSL prototypes into generation for downstream adaptation of pretrained SSL encoders. Although optimization-based methods generally outperform at $\text{IPC} = 1$, SRG closes the gap and attains better performance as IPC increases. We hypothesize that direct synthetic-image optimization gives LGM and CLP-DD greater freedom to encode various data patterns into each image, which is especially advantageous under a very small IPC budget. By contrast, SRG constrains samples to the pretrained generator's image manifold and therefore relies more strongly on additional samples to cover diverse patterns.

Table 3: Cross-encoder evaluation of downstream validation accuracy on ImageNet-100 under IPC $= 5$. The Average row reports the mean and standard deviation across the four evaluation encoders.

Evaluation Encoder	Distillation Encoder				Full
	CLIP	DINOv2	EVA-02	MoCov3	
CLIP	$87.8_{\pm 0.1}$	$85.0_{\pm 0.0}$	$85.0_{\pm 0.1}$	$84.6_{\pm 0.1}$	$92.5_{\pm 0.0}$
DINOv2	$91.1_{\pm 0.1}$	$93.2_{\pm 0.1}$	$90.6_{\pm 0.1}$	$90.8_{\pm 0.1}$	$95.2_{\pm 0.1}$
EVA-02	$88.6_{\pm 0.1}$	$88.4_{\pm 0.1}$	$90.4_{\pm 0.1}$	$88.7_{\pm 0.1}$	$94.1_{\pm 0.1}$
MoCov3	$84.0_{\pm 0.1}$	$83.3_{\pm 0.1}$	$82.6_{\pm 0.0}$	$84.8_{\pm 0.1}$	$89.4_{\pm 0.3}$
Average	$87.9_{\pm 2.6}$	$87.5_{\pm 3.8}$	$87.2_{\pm 2.9}$	$87.2_{\pm 2.3}$	$92.8_{\pm 2.2}$

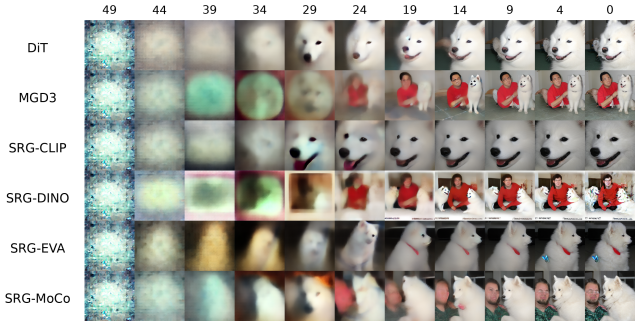


Figure 3: Comparison of intermediate images at different denoising timesteps for DiT, MGD3, and SRG. The denoising timesteps are arranged from 49 to 0.

Fine-Grained Classification

We further evaluate SRG in fine-grained classification settings, where successful distillation requires preserving subtle inter-class differences. Specifically, we compare the performance of SRG with baseline methods on three ImageNet-1K subsets containing visually similar classes, namely *Woof*, *Fruits*, and *Instruments*. As shown in Table 2, SRG outperforms the generative baselines in all the settings and outperforms LGM in most high-IPC settings. These results support the effectiveness of SSL-space guidance for datasets whose classes have similar visual content.

Cross-Encoder Generalization

To examine the generalization of SRG across pretrained representation spaces, we conduct a cross-encoder evaluation on ImageNet-100 at IPC $= 5$. We consider four SSL encoders: CLIP (Radford et al. 2021), DINOv2 (Oquab, Darcet, and Moutakanni 2024), EVA-02 (Fang et al. 2024), and MoCov3 (Chen, Xie, and He 2021). Each encoder is used separately to guide distillation, and every resulting distilled dataset is then evaluated with all four encoders. This protocol measures both SRG's performance with different distillation encoders and its transfer to representation spaces not used during distillation. For each distillation encoder, we also report the mean and standard deviation of accuracy across the four evaluation encoders.

As shown in Table 3, accuracy varies across evaluation encoders even when the full dataset is used, reflecting inherent differences in their representation quality. For each evaluation encoder, the highest accuracy occurs when the same encoder guides distillation. The mean accuracies of datasets distilled with the four encoders lie within a narrow range, indicating stable aggregate performance across distillation encoders. Additional experiments across different IPC settings and comparisons with LGM are provided in the supplementary material.

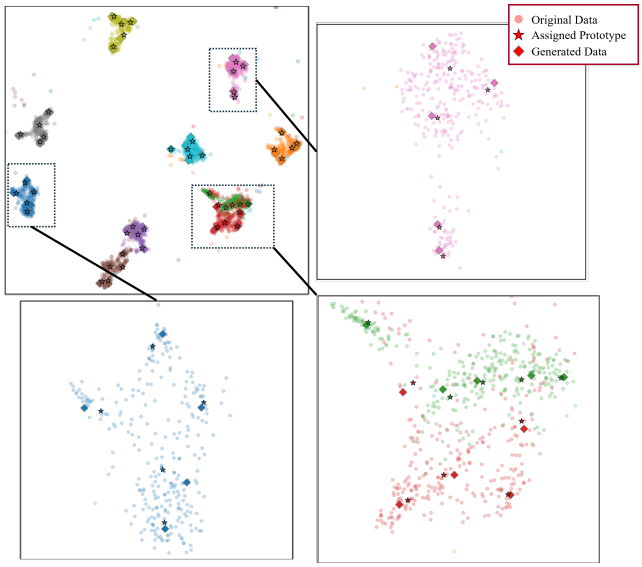
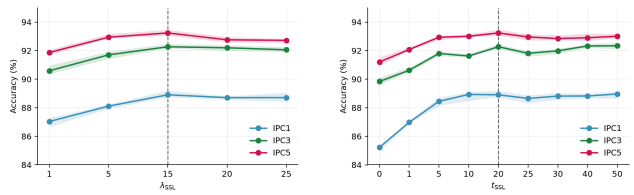


Figure 4: UMAP visualization of real samples, class prototypes, and SRG-generated samples in the SSL representation space on ImageNet-Woof at IPC $= 5$.

Visualization

We provide two qualitative analyses of SRG's behavior. First, Fig. 3 shows intermediate images decoded from the predicted clean latent at different denoising timesteps for DiT, MGD3, and SRG on class n02111889 (Samoyed) using the same initial noise. Under SRG's schedule, latent-space guidance is active at early stages ($t_i \geq 20$), when coarse image structure emerges, whereas SSL-space guidance is active at later stages ($t_i < 20$), when semantic details become visible. The trajectories and final appearances also vary among guidance encoders, indicating different representational preferences across the encoders.

Second, we use UMAP (McInnes et al. 2018) to visualize the distributions of real samples, assigned prototypes, and generated samples in the SSL representation space on ImageNet-Woof. As shown in Fig. 4, SRG-generated samples generally remain near their assigned prototypes. In the displayed well-separated class, generated samples occupy peripheral portions of the class cluster. In comparison, for the overlapping classes, several generated samples lie on the sides of their clusters farther from neighboring classes. These qualitative patterns are consistent with the intended objectives of prototype alignment, intra-class assignment, and inter-class discrimination. Additional visualization analyses



(a) Guidance scale λ_{SSL}. (b) Starting timestep t_{SSL}.

Figure 5: Sensitivity analysis of the SSL-space guidance scale λ_{SSL} and starting timestep t_{SSL} on ImageNet-100 under different IPC settings. Points denote mean accuracy, and shaded regions indicate the minimum-to-maximum range.

Table 4: Ablation study of latent-space guidance $\boldsymbol{g}^i_{lat}$ and SSL-space guidance $\boldsymbol{g}^i_{\text{SSL}}$ on ImageNet-100 under different IPC settings. The best result in each setting is highlighted in bold.

$\boldsymbol{g}^i_{lat}$	$\boldsymbol{g}^i_{\text{SSL}}$	IPC=1	IPC=3	IPC=5
✗	✗	$83.3_{\pm 0.2}$	$87.9_{\pm 0.2}$	$89.1_{\pm 0.1}$
✗	✓	$88.9_{\pm 0.1}$	$91.2_{\pm 0.2}$	$92.3_{\pm 0.2}$
✓	✗	$85.4_{\pm 0.2}$	$89.8_{\pm 0.1}$	$91.3_{\pm 0.1}$
✓	✓	$\mathbf{89.0_{\pm 0.1}}$	$\mathbf{92.3_{\pm 0.1}}$	$\mathbf{93.2_{\pm 0.2}}$

are provided in the supplementary material.

Ablation Study

We conduct an ablation study on ImageNet-100 to evaluate the contributions of latent-space guidance and SSL-space guidance. As shown in Table 4, latent-space guidance alone brings a modest improvement over the unguided baseline, while SSL-space guidance yields more substantial gains. Combining them consistently achieves the best accuracy across all IPC settings, supporting the complementarity of the two signals. Detailed ablations of the individual SSL-space objectives are provided in the supplementary material.

Sensitivity to Hyperparameters

We further analyze the sensitivity of SRG to two hyperparameters associated with SSL-space guidance: the guidance scale λ_{SSL} and the starting timestep t_{SSL}. Analyses of the remaining hyperparameters are provided in the supplementary material. As shown in Fig. 5a, performance is relatively stable across the tested intermediate values of λ_{SSL}. A very small λ_{SSL} leads to lower accuracy, likely because it provides insufficient representation-level supervision. Increasing λ_{SSL} beyond an intermediate range yields no further improvement, possibly because overly strong guidance interferes with DiT's original denoising trajectory. Among the evaluated settings, $\lambda_{\text{SSL}} = 15$ achieves the best overall performance across IPC budgets. Fig. 5b shows that applying SSL-space guidance only during the final several denoising steps already substantially improves performance over using no SSL-space guidance. Extending the guidance to earlier timesteps provides only limited additional gains while increasing the computational cost. We therefore set $t_{\text{SSL}} = 20$ to balance performance and computational efficiency.

Table 5: Comparison of runtime between SRG and baseline methods on different ImageNet benchmarks under various IPC settings, with the prototype-construction cost of SRG reported separately. All runtimes are reported in minutes.

Dataset	Method	Time (Min)		Mem. (GB)	
		1	3	1	3
ImageNet-IDC	LGM	34.8	75.1	10.2	28.8
	CLP-DD	1.4	2.5	1.1	2.3
	DiT	0.2	0.5	3.2	3.2
	SRG	0.3	0.9	5.8	5.8
	Prototype	0.5	0.6	9.5	9.5
ImageNet-100	LGM	81.8	OOM	28.7	> 48
	CLP-DD	8.7	26.8	6.4	18.3
	DiT	1.6	4.7	3.2	3.2
	SRG	2.9	8.5	5.8	5.8
	Prototype	2.7	2.7	9.5	9.5
ImageNet-1K	LGM	OOM*	OOM	> 48	> 48
	CLP-DD	OOM	OOM	> 48	> 48
	DiT	15.5	46.4	3.2	3.2
	SRG	27.8	83.3	5.8	5.8
	Prototype	36.8	39.3	9.5	9.5

* The original paper reports a runtime of approximately 720 minutes using four H200 GPUs.

Computational Cost

Table 5 compares the computational costs of LGM, CLP-DD, DiT, and SRG on a single NVIDIA A6000 GPU, with the prototype-construction cost reported separately. SRG introduces moderate sampling-time and memory overhead over DiT due to SSL-space guidance, while remaining more efficient than the evaluated optimization-based baselines. Prototype construction requires one-time preprocessing, after which the extracted representations are cached for reuse. CLP-DD likewise requires representation extraction as part of its preprocessing. Under this hardware setting, SRG exhibits favorable scalability to large-scale datasets while maintaining moderate overhead over the generative baseline.

Conclusion

In this paper, we proposed SRG, a generative dataset distillation framework for downstream adaptation with pretrained SSL encoders. SRG combines stage-wise latent-space anchoring with SSL-space objectives for prototype alignment, inter-class discrimination, and intra-class assignment. Experiments across multiple datasets and IPC settings demonstrate consistent improvements over generative baselines, competitive performance against optimization-based methods, and favorable computational scalability. Cross-encoder evaluation further indicates transfer across pretrained representation spaces. Nevertheless, its reliance on an ImageNet-pretrained generator currently limits its applicability to broader visual domains.

References

Cai, W.; Zou, Y.; Li, G.; Gu, C.; and Zhang, C. 2026. EVLF: Early Vision-Language Fusion for Generative Dataset Distillation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Caron, M.; Touvron, H.; Misra, I.; Jégou, H.; Mairal, J.; Bojanowski, P.; and Joulin, A. 2021. Emerging Properties in Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9650–9660.

Cazenavette, G.; Torralba, A.; and Sitzmann, V. 2025. Dataset Distillation for Pre-Trained Self-Supervised Vision Models. In *Proceedings of the Advances in Neural Information Processing Systems*.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset Distillation by Matching Training Trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10718–10727.

Chan-Santiago, J. A.; Tirupattur, P.; Nayak, G. K.; Liu, G.; and Shah, M. 2025. MGD3: Mode-Guided Dataset Distillation using Diffusion Models. In *Proceedings of the International Conference on Machine Learning*, 1–16.

Chen, M.; Du, J.; Huang, B.; Wang, Y.; Zhang, X.; and Wang, W. 2025. Influence-Guided Diffusion for Dataset Distillation. In *Proceedings of the International Conference on Learning Representations*.

Chen, T.; Kornblith, S.; Norouzi, M.; and Hinton, G. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, 1597–1607. PMLR.

Chen, X.; Xie, S.; and He, K. 2021. An Empirical Study of Training Self-Supervised Vision Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9640–9649.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Li, F.-F. 2009. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 248–255.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *Proceedings of the International Conference on Learning Representations*, 1–12.

Fang, Y.; Sun, Q.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2024. EVA-02: A Visual Representation for Neon Genesis. *Image and Vision Computing*, 149: 105171.

Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. EVA: Exploring the Limits of Masked Visual Representation Learning at Scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19358–19369.

Fastai. 2019. imagenette. Technical report, GitHub repository. Https://github.com/fastai/imagenette.

Gu, J.; Vahidian, S.; Kungurtsev, V.; Wang, H.; Jiang, W.; You, Y.; and Chen, Y. 2024. Efficient Dataset Distillation via Minimax Diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15793–15803.

Gui, J.; Chen, T.; Zhang, J.; Cao, Q.; Sun, Z.; Luo, H.; and Tao, D. 2024. A Survey on Self-Supervised Learning: Algorithms, Applications, and Future Trends. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 9052–9071.

He, K.; Fan, H.; Wu, Y.; Xie, S.; and Girshick, R. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9729–9738.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; Song, D.; Steinhardt, J.; and Gilmer, J. 2021. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. In *2021 IEEE/CVF International Conference on Computer Vision*, 8320–8329.

Ho, J.; and Salimans, T. 2022. Classifier-Free Diffusion Guidance. *arXiv preprint arXiv:2207.12598*, 1–14.

Jaiswal, A.; Babu, A. R.; Zadeh, M. Z.; Banerjee, D.; and Makedon, F. 2021. A Survey on Contrastive Self-Supervised Learning. *Technologies*, 9(1): 2.

Kim, D.; Thomas, X.; and Ghadiyaram, D. 2025. Revelio: Interpreting and leveraging semantic information in diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4659–4669.

Kim, J.-H.; Kim, J.; Oh, S. J.; Yun, S.; Song, H.; Jeong, J.; Ha, J.-W.; and Song, H. O. 2022. Dataset Condensation via Efficient Synthetic-Data Parameterization. In *Proceedings of the International Conference on Machine Learning*, 11102–11118.

Kingma, D. P.; and Welling, M. 2013. Auto-Encoding Variational Bayes. *arXiv preprint arXiv:1312.6114*, 1–14.

Li, G.; Togo, R.; Ogawa, T.; and Haseyama, M. 2020. Soft-Label Anonymous Gastric X-Ray Image Distillation. In *Proceedings of the IEEE International Conference on Image Processing*, 305–309.

Li, G.; Togo, R.; Ogawa, T.; and Haseyama, M. 2022. Compressed Gastric Image Generation Based on Soft-Label Dataset Distillation for Medical Data Sharing. *Computer Methods and Programs in Biomedicine*.

Li, G.; Togo, R.; Ogawa, T.; and Haseyama, M. 2023a. Dataset Distillation for Medical Dataset Sharing. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), Workshop*, 1–6.

Li, G.; Togo, R.; Ogawa, T.; and Haseyama, M. 2023b. Dataset Distillation using Parameter Pruning. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*.

Li, G.; Togo, R.; Ogawa, T.; and Haseyama, M. 2024a. Importance-Aware Adaptive Dataset Distillation. *Neural Networks*.

Li, G.; Zhao, B.; and Wang, T. 2022. Awesome Dataset Distillation. https://github.com/Guang000/Awesome-Dataset-Distillation.

Li, L.; Li, G.; Togo, R.; Maeda, K.; Ogawa, T.; and Haseyama, M. 2024b. Generative Dataset Distillation: Balancing Global Structure and Local Details. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 7664–7671.

Li, W.; Li, G.; Maeda, K.; Ogawa, T.; and Haseyama, M. 2025a. Decoupled Audio-Visual Dataset Distillation. *arXiv preprint arXiv:2511.17890*.

Li, W.; Li, G.; Maeda, K.; Ogawa, T.; and Haseyama, M. 2025b. Hyperbolic Dataset Distillation. In *Advances in Neural Information Processing Systems*.

Liu, P.; and Du, J. 2025. The Evolution of Dataset Distillation: Toward Scalable and Generalizable Solutions. *arXiv preprint arXiv:2502.05673*, 1–23.

Ma, H.; Li, G.; Wang, S.; Zhou, D.; Sun, B.; Ogawa, T.; Haseyama, M.; and Wang, Z. 2026. FD^2: A Dedicated Framework for Fine-Grained Dataset Distillation. In *European Conference on Computer Vision*.

McInnes, L.; Healy, J.; Saul, N.; and Großberger, L. 2018. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, 3(29): 861.

Nguyen, T.; Chen, Z.; and Lee, J. 2021. Dataset Meta-Learning from Kernel Ridge-Regression. In *Proceedings of the International Conference on Learning Representations*, 1–25.

Oquab, M.; Darcet, T.; and Moutakanni, e. a. 2024. DINOv2: Learning Robust Visual Features without Supervision. *Transactions on Machine Learning Research*.

Peebles, W.; and Xie, S. 2023. Scalable Diffusion Models with Transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4172–4182.

Peng, B.; Li, G.; Liu, P.; Ogawa, T.; and Haseyama, M. 2026. Closed-Form Linear-Probe Dataset Distillation for Pre-trained Vision Models. *arXiv preprint arXiv:2605.07194*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; and Sutskever, I. 2021. Learning Transferable Visual Models from Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 8748–8763. PMLR.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3): 211–252.

Schmidhuber, J. 2015. Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117.

Strubell, E.; Ganesh, A.; and McCallum, A. 2019. Energy and Policy Considerations for Deep Learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1–6.

Su, D.; Hou, J.; Li, G.; Togo, R.; Song, R.; Ogawa, T.; and Haseyama, M. 2024. Generative Dataset Distillation Based on Diffusion Model. In *European Conference on Computer Vision Workshops*.

Talaei Khoei, T.; Ould Slimane, H.; and Kaabouch, N. 2023. Deep learning: systematic review, models, challenges, and research directions. *Neural Computation & Application*, 35: 23103–23124.

Wang, H.; Zhao, Z.; Wu, J.; Shang, Y.; Liu, G.; and Yan, Y. 2025. CaO2: Rectifying Inconsistencies in Diffusion-Based Dataset Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1–14.

Wang, K.; Zhao, B.; Peng, X.; Zhu, Z.; Yang, S.; Wang, S.; Huang, G.; Bilen, H.; Wang, X.; and You, Y. 2022. CAFE: Learning to Condense Dataset by Aligning Features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12196–12205.

Wang, T.; Zhu, J.-Y.; Torralba, A.; and Efros, A. A. 2018. Dataset Distillation. *arXiv preprint arXiv:1811.10959*, 1–14.

Wu, H.; Su, D.; Hou, J.; and Li, G. 2025. Dataset Condensation with Color Compensation. *Transactions on Machine Learning Research*.

Ye, L.; Hamidi, S. M.; Li, G.; Ogawa, T.; Haseyama, M.; and Plataniotis, K. N. 2025. Information-Guided Diffusion Sampling for Dataset Distillation. In *Advances in Neural Information Processing Systems Workshops*.

Yu, R.; Liu, S.; and Wang, X. 2023. Dataset Distillation: A Comprehensive Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(1): 150–170.

Zhao, B.; and Bilen, H. 2021a. Dataset condensation with Differentiable Siamese Augmentation. In *Proceedings of the International Conference on Machine Learning*, 12674–12685.

Zhao, B.; and Bilen, H. 2021b. Dataset Condensation with Gradient Matching. In *Proceedings of the International Conference on Learning Representations*, 1–20.

Zhao, B.; and Bilen, H. 2022. Synthesizing Informative Training Samples with GAN. In *Proceedings of the Advances in Neural Information Processing Systems, Workshop*.

Zhao, B.; and Bilen, H. 2023. Dataset Condensation with Distribution Matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6514–6523.

Zou, Y.; Li, G.; Su, D.; Wang, Z.; Yu, J.; and Zhang, C. 2025. Dataset Distillation via Vision-Language Category Prototype. In *IEEE/CVF International Conference on Computer Vision*.

Algorithm

We provide detailed pseudocode for SRG using the notation and equation numbers from the main paper. As shown in Algorithm 1, SRG first constructs class-specific SSL prototypes and retrieves the nearest real sample to each prototype as the latent-guidance target. It then generates one synthetic sample for each prototype through a stage-wise guided denoising process, in which latent-space guidance is applied during the early stages, and SSL-space guidance is applied during the later stages. Finally, the generated samples from all classes are collected to form the distilled dataset.

Algorithm 1: Self-Supervised Representation-Guided Generative Dataset Distillation (SRG).

Equation numbers refer to the main paper.

Require: $\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$: original dataset; f_θ: pretrained DiT; f_ϕ: pretrained SSL encoder; E_ψ, D_ψ: VAE encoder and decoder (Kingma and Welling 2013); K: IPC; $\mathcal{T} = (t_0, \ldots, t_{T-1})$: decreasing denoising schedule; $t_{\mathrm{lat}}, t_{\mathrm{SSL}}, \lambda_{\mathrm{lat}}, \lambda_{\mathrm{SSL}}, \tau_{\mathrm{inter}}, \tau_{\mathrm{intra}}$: guidance hyperparameters.

Ensure: $\mathcal{S}$: distilled dataset.

Initialize $\mathcal{S} \leftarrow \varnothing$
Stage 1: Prototype and latent-anchor construction
for each class $c = 1, \ldots, C$ **do**
  Collect $\mathcal{D}_c = \{\boldsymbol{x}_n^c\}_{n=1}^{N_c}$ and compute $\boldsymbol{h}_n^c \leftarrow \mathrm{norm}_2(f_\phi(\boldsymbol{x}_n^c))$ for all n
  Obtain $\{\boldsymbol{p}_{c,k}\}_{k=1}^{K}$ by spherical K-means with Eq. (4)
  for $k = 1, \ldots, K$ **do**
    $n_{c,k}^\star \leftarrow \arg\max_{1 \le n \le N_c} \mathrm{sim}(\boldsymbol{h}_n^c, \boldsymbol{p}_{c,k})$
    $\boldsymbol{a}_{c,k} \leftarrow E_\psi(\boldsymbol{x}_{n_{c,k}^\star}^c)$
  end for
end for
Stage 2: Stage-wise guided generation
for each class $c = 1, \ldots, C$ **do**
  for each prototype $k = 1, \ldots, K$ **do**
    Sample $\boldsymbol{v}_0 \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$
    for $i = 0, \ldots, T-1$ **do**
      Set $t_i \leftarrow \mathcal{T}[i]$ and obtain $\mu_\theta(\boldsymbol{v}_i, t_i, c)$ and $\hat{\boldsymbol{z}}_0^i$ from DiT
      Initialize $\boldsymbol{g}_{\mathrm{lat}}^i, \boldsymbol{g}_{\mathrm{SSL}}^i \leftarrow \boldsymbol{0}$
      if $t_i > t_{\mathrm{lat}}$ **then**
        Compute $\boldsymbol{g}_{\mathrm{lat}}^i$ with target $\boldsymbol{m}_{c,k}$ with Eq. (2)
      end if
      if $t_i < t_{\mathrm{SSL}}$ **then**
        $\boldsymbol{r}_i \leftarrow \mathrm{norm}_2\big(f_\phi\big(D_\psi(\hat{\boldsymbol{z}}_0^i)\big)\big)$
        Compute $\mathcal{L}_{\mathrm{proto}}^i$, $\mathcal{L}_{\mathrm{inter}}^i$, and $\mathcal{L}_{\mathrm{intra}}^i$ with Eqs. (6)–(8)
        Compute $\mathcal{L}_{\mathrm{SSL}}^i$ and $\boldsymbol{g}_{\mathrm{SSL}}^i$ with Eqs. (9) and (10)
      end if
      Compute $\boldsymbol{g}_{\mathrm{SRG}}^i$ with Eq. (11)
      Sample $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ and update $\boldsymbol{v}_{i+1}$ with Eq. (12)
    end for
    $\tilde{\boldsymbol{x}}_{c,k} \leftarrow D_\psi(\boldsymbol{v}_T)$ and $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\tilde{\boldsymbol{x}}_{c,k}, c)\}$
  end for
end for
return $\mathcal{S}$

Implementation Details

We use four pretrained SSL encoders with ViT-B backbones (Dosovitskiy et al. 2021): CLIP ViT-B/16 (Radford et al. 2021), DINOv2 ViT-B/14 (Oquab, Darcet, and Moutakanni 2024), EVA-02 ViT-B/14 (Fang et al. 2024), and MoCo-v3 ViT-B (Chen, Xie, and He 2021). The feature dimension is 512 for CLIP ViT-B/16 and 768 for DINOv2, EVA-02, and MoCo-v3. SSL features are extracted at a resolution of 224×224.

For SRG, we use a DiT-XL/2 backbone (Peebles and Xie 2023) at an image resolution of 256×256, with 50 denoising steps and a classifier-free guidance (Ho and Salimans 2022) scale of 4.0. Prototype centers are computed using spherical K-means in the normalized SSL feature space, with a maximum of 50 iterations. The default parameters are the latent-guidance stopping timestep $t_{\mathrm{lat}} = 20$, SSL-guidance starting timestep $t_{\mathrm{SSL}} = 20$, SSL-guidance scale $\lambda_{\mathrm{SSL}} = 15$, latent-guidance scale $\lambda_{\mathrm{lat}} = 0.1$, inter-class discrimination temperature $\tau_{\mathrm{inter}} = 1.0$, and intra-class assignment temperature $\tau_{\mathrm{intra}} = 5.0$.

Unless otherwise specified, the methods reproduced in our experiments use the default parameters provided by their original authors. For LGM (Cazenavette, Torralba, and Sitzmann 2025), the number of augmentation views is set to 3 on ImageNet-100 (Kim et al. 2022) because of GPU memory limitations. For CLP-DD (Peng et al. 2026), we use the unpublished version code provided by the authors.

For evaluation, we train a linear probe on the frozen SSL features for $1{,}000$ epochs and repeat each experiment five times. When augmenting the distilled training data for evaluation, we use resize-only preprocessing for CLP-DD and spatial augmentation for LGM, following the respective default settings provided by the authors. For the generative methods, we use weak photometric augmentation to introduce mild natural variation.

All experiments are conducted on a GPU server equipped with an AMD EPYC 7413 24-core processor and four NVIDIA A6000 GPUs with 48 GB of memory each, running Ubuntu 22.04.4 LTS. Except for LGM on ImageNet-100 at IPC $= 3$, which uses all four GPUs, each experiment uses a single A6000 GPU. The software environment consists of Python 3.14, PyTorch 2.12.0, torchvision 0.27.0, CUDA 12.6, and cuDNN 9.10. We use a default fixed random seed 0. However, we do not enforce exact bitwise reproducibility because some optimized CUDA kernels used for efficient sampling are nondeterministic. Enabling fully deterministic algorithms would substantially slow sampling and may disable some operations.

Dataset Details

For ImageNet-IDC and ImageNet-100, we follow the dataset configurations introduced by Kim et al. (2022). ImageNet-100 is a subset of ImageNet-1K (Deng et al. 2009; Russakovsky et al. 2015) containing 100 selected classes, while ImageNet-IDC contains 10 classes from ImageNet-100. The class composition of ImageNet-IDC is provided below. Each class is reported in the format "ImageNet class index: class name (WordNet ID)", where the class name corresponds to

Table 6: Cross-encoder comparison of downstream validation accuracy between SRG and LGM on ImageNet-100 at $\mathrm{IPC} = 1$ and $\mathrm{IPC} = 3$. Each average is computed from the mean accuracies obtained with the four evaluation encoders. The best average values are highlighted in bold for each distillation setting.

	Evaluation Encoder	CLIP		DINOv2		EVA-02		MoCo-v3	
		LGM	SRG	LGM	SRG	LGM	SRG	LGM	SRG
IPC = 1	CLIP	$82.4_{\pm 0.1}$	$78.5_{\pm 0.2}$	$76.3_{\pm 0.2}$	$71.6_{\pm 0.2}$	$74.3_{\pm 0.2}$	$74.5_{\pm 0.2}$	$64.5_{\pm 0.2}$	$72.7_{\pm 0.2}$
	DINOv2	$78.2_{\pm 0.3}$	$85.2_{\pm 0.3}$	$90.7_{\pm 0.1}$	$88.7_{\pm 0.2}$	$85.8_{\pm 0.3}$	$85.2_{\pm 0.2}$	$85.6_{\pm 0.1}$	$84.0_{\pm 0.2}$
	EVA-02	$80.8_{\pm 0.4}$	$80.9_{\pm 0.4}$	$86.0_{\pm 0.3}$	$80.8_{\pm 0.2}$	$88.5_{\pm 0.1}$	$85.8_{\pm 0.2}$	$81.2_{\pm 0.1}$	$80.9_{\pm 0.1}$
	MoCo-v3	$55.7_{\pm 0.9}$	$75.1_{\pm 0.0}$	$76.3_{\pm 0.4}$	$74.9_{\pm 0.1}$	$64.8_{\pm 0.7}$	$74.6_{\pm 0.1}$	$83.1_{\pm 0.1}$	$77.8_{\pm 0.1}$
	Average	$74.3_{\pm 10.8}$	$\mathbf{79.9_{\pm 3.7}}$	$\mathbf{82.3_{\pm 6.3}}$	$79.0_{\pm 6.5}$	$78.3_{\pm 9.4}$	$\mathbf{80.0_{\pm 5.5}}$	$78.6_{\pm 8.3}$	$\mathbf{78.8_{\pm 4.2}}$
IPC = 3	CLIP	$86.5_{\pm 0.1}$	$86.2_{\pm 0.1}$	$80.4_{\pm 0.2}$	$82.5_{\pm 0.1}$	$79.7_{\pm 0.2}$	$83.7_{\pm 0.1}$	$71.3_{\pm 0.2}$	$84.5_{\pm 0.1}$
	DINOv2	$83.0_{\pm 0.2}$	$90.4_{\pm 0.2}$	$91.7_{\pm 0.2}$	$92.1_{\pm 0.0}$	$88.0_{\pm 0.2}$	$90.1_{\pm 0.2}$	$86.4_{\pm 0.3}$	$90.3_{\pm 0.2}$
	EVA-02	$83.9_{\pm 0.2}$	$87.0_{\pm 0.2}$	$87.0_{\pm 0.3}$	$88.0_{\pm 0.1}$	$88.7_{\pm 0.1}$	$89.6_{\pm 0.1}$	$81.7_{\pm 0.4}$	$88.3_{\pm 0.2}$
	MoCo-v3	$66.2_{\pm 0.6}$	$82.1_{\pm 0.0}$	$79.4_{\pm 0.5}$	$81.8_{\pm 0.1}$	$69.6_{\pm 0.3}$	$80.9_{\pm 0.1}$	$83.7_{\pm 0.0}$	$84.5_{\pm 0.1}$
	Average	$79.9_{\pm 8.0}$	$\mathbf{86.4_{\pm 3.0}}$	$84.6_{\pm 5.0}$	$\mathbf{86.1_{\pm 4.2}}$	$81.5_{\pm 7.7}$	$\mathbf{86.1_{\pm 3.9}}$	$80.8_{\pm 5.7}$	$\mathbf{86.9_{\pm 2.5}}$

the first entry in the ImageNet label list.

ImageNet-IDC. 452: Bonnet (n02869837); 64: Green Mamba (n01749939); 374: Langur (n02488291); 236: Doberman (n02107142); 993: Gyromitra (n13037406); 176: Saluki (n02091831); 882: Vacuum (n04517823); 904: Window Screen (n04589890); 503: Cocktail Shaker (n03062245); 74: Garden Spider (n01773797).

For fine-grained evaluation, the *Woof* subset follows the ImageWoof configuration of Fastai (2019). We additionally construct the *Fruits* and *Instruments* subsets by selecting semantically related and visually similar classes from ImageNet-1K. Their class compositions are listed below.

***Woof*.** 155: Shih-Tzu (n02086240); 159: Rhodesian Ridgeback (n02087394); 162: Beagle (n02088364); 167: English Foxhound (n02089973); 182: Border Terrier (n02093754); 193: Australian Terrier (n02096294); 207: Golden Retriever (n02099601); 229: Old English Sheepdog (n02105641); 258: Samoyed (n02111889); 273: Dingo (n02115641).

***Fruits*.** 953: Pineapple (n07753275); 954: Banana (n07753592); 949: Strawberry (n07745940); 950: Orange (n07747607); 951: Lemon (n07749582); 957: Pomegranate (n07768694); 952: Fig (n07753113); 945: Bell Pepper (n07720875); 943: Cucumber (n07718472); 948: Granny Smith (n07742313).

***Instruments*.** 432: Bassoon (n02804610); 513: Cornet (n03110669); 558: Flute (n03372029); 566: French Horn (n03394916); 593: Harmonica (n03494278); 683: Oboe (n03838899); 684: Ocarina (n03840681); 699: Panpipe (n03884397); 776: Saxophone (n04141076); 875: Trombone (n04487394).

Cross-Encoder Generalization

We further validate SRG in the cross-encoder setting on more IPC settings, and conduct comparison with LGM. As shown in Table 6, at $\mathrm{IPC} = 1$, LGM generally retains an advantage when the same encoder is used for both distillation and evaluation. However, its performance transfers less consistently across encoders, with several combinations exhibiting substantial degradation and less stable average performance than SRG. At $\mathrm{IPC} = 3$, SRG outperforms LGM for nearly all encoder combinations and achieves substantially higher average performance, indicating better scalability and transfer across encoders. One possible explanation is that LGM directly optimizes synthetic data for a specific encoder, which can yield stronger encoder-specific adaptation but weaker transfer to other representation spaces. In contrast, SRG performs distillation by guiding a pretrained generative model. In addition to encoder-specific SSL guidance, the model's visual prior preserves general object structure and appearance, thereby reducing overfitting to the distillation encoder.

Detailed Hyperparameter Analyses

Guidance Timesteps

Table 7 examines the joint effect of the latent-guidance stopping timestep t_{lat} and the SSL-guidance starting timestep t_{SSL} on ImageNet-100 at $\mathrm{IPC} = 5$. As the differences in the experimental results were not clear under some settings, all values are reported to two decimal places for clarity. At the boundary setting $t_{\mathrm{SSL}} = 0$, where SSL-space guidance is disabled, extending latent-space guidance beyond the earliest denoising stages initially provides clear improvements, but the gains diminish at later timesteps. Similarly, at $t_{\mathrm{lat}} = 51$, where latent-space guidance is disabled, introducing SSL-space guidance during the later denoising stages substantially improves performance, whereas extending it to earlier timesteps produces only limited additional gains, consistent with the discussion in the main paper. These observations agree with the intended roles of the two guidance signals: latent-space guidance mainly helps determine coarse structure during early denoising, whereas SSL-space guidance mainly helps refine semantic details during later stages. A similar trend is observed when the two guidance intervals overlap, suggesting that the two signals can operate together effectively. Because both guidance operations introduce additional computation, particularly the representation map-

Table 7: Joint sensitivity analysis of the latent-guidance stopping timestep t_{lat} and SSL-guidance starting timestep t_{SSL} on ImageNet-100 at IPC $= 5$. The default setting is underlined, and the best result is highlighted in bold.

$t_{\text{lat}} \backslash t_{\text{SSL}}$	0	10	20	25	30	40	50
51	$89.12_{\pm 0.06}$	$91.79_{\pm 0.11}$	$92.29_{\pm 0.15}$	$92.02_{\pm 0.15}$	$92.28_{\pm 0.17}$	$92.16_{\pm 0.07}$	$92.24_{\pm 0.09}$
40	$89.28_{\pm 0.12}$	$91.58_{\pm 0.14}$	$91.79_{\pm 0.15}$	$92.24_{\pm 0.15}$	$92.53_{\pm 0.15}$	$92.01_{\pm 0.15}$	$92.31_{\pm 0.15}$
30	$89.88_{\pm 0.15}$	$92.06_{\pm 0.16}$	$92.77_{\pm 0.09}$	$92.71_{\pm 0.08}$	$92.70_{\pm 0.08}$	$92.64_{\pm 0.15}$	$92.52_{\pm 0.15}$
25	$90.51_{\pm 0.15}$	$92.54_{\pm 0.10}$	$92.93_{\pm 0.18}$	$92.80_{\pm 0.08}$	$93.03_{\pm 0.15}$	$92.89_{\pm 0.12}$	$93.06_{\pm 0.09}$
20	$91.21_{\pm 0.19}$	$93.00_{\pm 0.08}$	$\mathbf{93.24_{\pm 0.15}}$	$92.96_{\pm 0.15}$	$92.85_{\pm 0.14}$	$92.90_{\pm 0.18}$	$93.01_{\pm 0.12}$
10	$91.28_{\pm 0.10}$	$92.68_{\pm 0.14}$	$93.03_{\pm 0.10}$	$92.80_{\pm 0.07}$	$93.02_{\pm 0.15}$	$92.83_{\pm 0.05}$	$93.11_{\pm 0.13}$
0	$91.33_{\pm 0.09}$	$92.49_{\pm 0.10}$	$92.74_{\pm 0.10}$	$92.91_{\pm 0.14}$	$92.96_{\pm 0.06}$	$92.88_{\pm 0.14}$	$93.03_{\pm 0.11}$

Table 8: Joint sensitivity analysis of the temperature parameters τ_{inter} and τ_{intra} on ImageNet-100 at IPC $= 5$. The default setting is underlined, and the best result is highlighted in bold.

τ_{inter}	τ_{intra}				
	0.1	0.5	1.0	5.0	10.0
0.1	$91.2_{\pm 0.2}$	$92.0_{\pm 0.1}$	$92.2_{\pm 0.1}$	$91.6_{\pm 0.2}$	$91.7_{\pm 0.1}$
0.5	$92.2_{\pm 0.1}$	$92.5_{\pm 0.2}$	$93.0_{\pm 0.1}$	$92.7_{\pm 0.2}$	$92.9_{\pm 0.2}$
1.0	$91.7_{\pm 0.1}$	$92.8_{\pm 0.1}$	$93.0_{\pm 0.1}$	$\mathbf{93.2_{\pm 0.2}}$	$93.1_{\pm 0.1}$
5.0	$92.3_{\pm 0.1}$	$92.5_{\pm 0.1}$	$92.3_{\pm 0.1}$	$92.8_{\pm 0.1}$	$92.7_{\pm 0.1}$
10.0	$92.2_{\pm 0.1}$	$92.7_{\pm 0.1}$	$93.0_{\pm 0.1}$	$92.7_{\pm 0.1}$	$92.9_{\pm 0.2}$

Table 9: Joint sensitivity analysis of the guidance scales λ_{lat} and λ_{SSL} on ImageNet-100 at IPC $= 5$. The default setting is underlined, and the best result is highlighted in bold.

λ_{SSL}	λ_{lat}				
	0.01	0.05	0.1	0.2	0.3
1	$89.9_{\pm 0.2}$	$90.8_{\pm 0.0}$	$92.0_{\pm 0.0}$	$92.3_{\pm 0.1}$	$88.1_{\pm 0.3}$
10	$92.0_{\pm 0.2}$	$92.7_{\pm 0.2}$	$93.1_{\pm 0.1}$	$92.8_{\pm 0.1}$	$90.2_{\pm 0.1}$
15	$92.4_{\pm 0.1}$	$93.0_{\pm 0.0}$	$\mathbf{93.2_{\pm 0.2}}$	$93.1_{\pm 0.1}$	$90.4_{\pm 0.1}$
20	$92.6_{\pm 0.1}$	$92.6_{\pm 0.1}$	$92.8_{\pm 0.1}$	$93.0_{\pm 0.1}$	$90.5_{\pm 0.1}$
30	$92.4_{\pm 0.1}$	$92.6_{\pm 0.1}$	$92.3_{\pm 0.1}$	$92.6_{\pm 0.1}$	$90.6_{\pm 0.2}$

ping and gradient backpropagation required by SSL-space guidance, we select $t_{\text{lat}} = t_{\text{SSL}} = 20$ as the default setting to balance performance and computational cost.

Temperature parameters

The temperature parameters τ_{inter} and τ_{intra} control the sharpness of their corresponding objectives. A smaller temperature makes an objective more sensitive to the most similar prototype, whereas a larger temperature distributes weight more evenly across prototypes. As shown in Table 8, accuracy generally increases and then decreases as either temperature grows. We adopt $\tau_{\text{inter}} = 1.0$ and $\tau_{\text{intra}} = 5.0$ as the default values because this combination achieves the highest accuracy. The difference between the selected temperatures is consistent with the structures modeled by the two objectives. Prototypes from other classes are typically more separated from the predicted representation, making the nearest ones particularly informative for discrimination. In contrast, intra-class prototypes are distributed more closely around the predicted representation, and repelling the representation from only one competing prototype may move it closer to another.

Table 10: Joint sensitivity analysis of the objective weights w_{inter} and w_{intra} on ImageNet-100 at IPC $= 5$. The default setting is underlined, and the best result is highlighted in bold.

w_{inter}	w_{intra}				
	0.1	1.0	5.0	10.0	30.0
0.1	$92.9_{\pm 0.1}$	$92.9_{\pm 0.2}$	$92.8_{\pm 0.1}$	$92.1_{\pm 0.1}$	$89.6_{\pm 0.2}$
1.0	$93.1_{\pm 0.1}$	$\mathbf{93.2_{\pm 0.2}}$	$92.8_{\pm 0.1}$	$92.4_{\pm 0.0}$	$90.0_{\pm 0.2}$
5.0	$93.1_{\pm 0.1}$	$92.9_{\pm 0.1}$	$92.9_{\pm 0.1}$	$92.3_{\pm 0.1}$	$90.0_{\pm 0.2}$
10.0	$92.4_{\pm 0.1}$	$92.7_{\pm 0.1}$	$92.6_{\pm 0.3}$	$92.4_{\pm 0.0}$	$90.6_{\pm 0.2}$
30.0	$89.3_{\pm 0.1}$	$89.9_{\pm 0.1}$	$88.6_{\pm 0.1}$	$90.0_{\pm 0.1}$	$87.4_{\pm 0.3}$

Guidance scales

The guidance scales λ_{lat} and λ_{SSL} determine the strengths of latent-space and SSL-space guidance, respectively. When a scale is too small, the corresponding guidance has only a limited effect on the sampling trajectory, resulting in performance close to that of unguided DiT. Increasing either guidance scale produces clear improvements, whereas excessively strong guidance can dominate the diffusion and steer samples toward regions not supported by the pretrained generative prior, resulting in performance degradation. We set $\lambda_{\text{lat}} = 0.1$ and $\lambda_{\text{SSL}} = 15$ as the default values because this combination achieves the best performance.

Objective weights

Since the overall strength of SSL guidance is already controlled by λ_{SSL}, we only examine the relative weights of $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$, using the following modified SSL-space guidance objective:

$$\mathcal{L}'^{i}_{\text{SSL}} = \mathcal{L}^{i}_{\text{proto}} + w_{\text{inter}}\mathcal{L}^{i}_{\text{inter}} + w_{\text{intra}}\mathcal{L}^{i}_{\text{intra}}, \qquad (13)$$

where w_{inter} and w_{intra} are relative weights. Because both the predicted representations and the prototypes are ℓ_2-normalized, weights of 1.0 place the three SSL-space objectives on comparable numerical scales. As shown in Table 10, small weights yield behavior similar to that obtained using $\mathcal{L}_{\text{proto}}$ alone, whereas excessively large weights cause

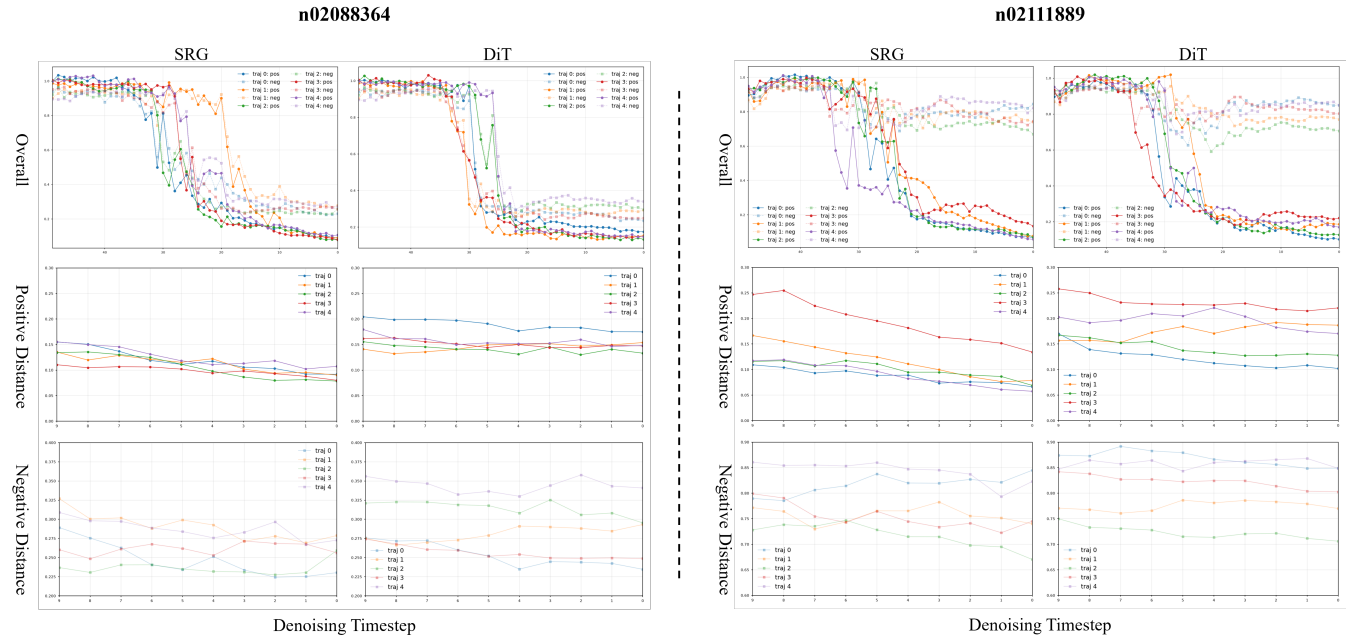


Figure 6: Evolution of prototype distances during denoising. The first row shows the complete trajectories. The second and third rows show the positive and negative distances, respectively, during the final 10 denoising steps. The horizontal axis shows the denoising timestep, and the vertical axis shows cosine distance.

Table 11: Comparison of DiT, MGD3 (Chan-Santiago et al. 2025), and SRG on ImageNet-R. The best result in each IPC setting is highlighted in bold.

Method	IPC=1	IPC=3	IPC=5
DiT	$58.6_{\pm 0.3}$	$62.3_{\pm 0.3}$	$62.6_{\pm 0.2}$
MGD3	$56.0_{\pm 0.5}$	$61.1_{\pm 0.2}$	$63.5_{\pm 0.2}$
SRG	$\mathbf{63.9_{\pm 0.3}}$	$\mathbf{64.4_{\pm 0.2}}$	$\mathbf{64.9_{\pm 0.3}}$

the corresponding objective to dominate the guidance, interfere with prototype assignment, and degrade performance. This trend is consistent with the objective ablation: $\mathcal{L}_{\text{proto}}$ provides the primary improvement, while $\mathcal{L}_{\text{inter}}$ and $\mathcal{L}_{\text{intra}}$ serve as auxiliary signals. We set $w_{\text{inter}} = w_{\text{intra}} = 1.0$ to simplify the formulation while maintaining balanced performance.

Out-of-Distribution Generalization

To evaluate whether the distilled datasets preserve semantic information beyond the visual style of the source domain, we conduct a cross-style generalization experiment on ImageNet-R (Hendrycks et al. 2021). ImageNet-R contains 200 ImageNet classes rendered in diverse artistic and non-photorealistic styles, including cartoons, paintings, and sculptures, and therefore provides a challenging benchmark for robustness to style shifts. In this experiment, all distilled datasets are generated from the original ImageNet domain, while evaluation is performed on ImageNet-R.

SRG consistently outperforms both DiT and MGD3 across all IPC settings. This result suggests that representation-guided sampling helps preserve semantic structure that remains useful under style shifts rather than merely improving visual realism in the source domain.

Table 12: Performance of SRG on ImageNet-100 across different generation seeds. The Mean row reports the mean and standard deviation across the five seeds. The best result in each IPC setting is highlighted in bold.

Seed	IPC 1	IPC 3	IPC 5
0	$89.0_{\pm 0.1}$	$92.3_{\pm 0.2}$	$\mathbf{93.2_{\pm 0.2}}$
1	$88.9_{\pm 0.2}$	$92.3_{\pm 0.1}$	$93.1_{\pm 0.2}$
2	$88.0_{\pm 0.2}$	$92.1_{\pm 0.1}$	$93.1_{\pm 0.1}$
3	$88.6_{\pm 0.2}$	$\mathbf{92.6_{\pm 0.1}}$	$92.7_{\pm 0.1}$
4	$\mathbf{89.1_{\pm 0.1}}$	$92.3_{\pm 0.1}$	$93.0_{\pm 0.1}$
Mean	$88.7_{\pm 0.4}$	$92.3_{\pm 0.2}$	$93.0_{\pm 0.2}$

Robustness across Generation Seeds

We further evaluate the robustness of SRG across different generation seeds. As shown in Table 12, performance exhibits moderate variation across seeds, especially under the most challenging IPC $= 1$ setting. Nevertheless, SRG consistently maintains strong performance across all tested seeds. Compared with the generative baselines reported in the main results, SRG remains clearly superior even when seed-level variation is considered, indicating that its performance is not overly sensitive to the generation seed.

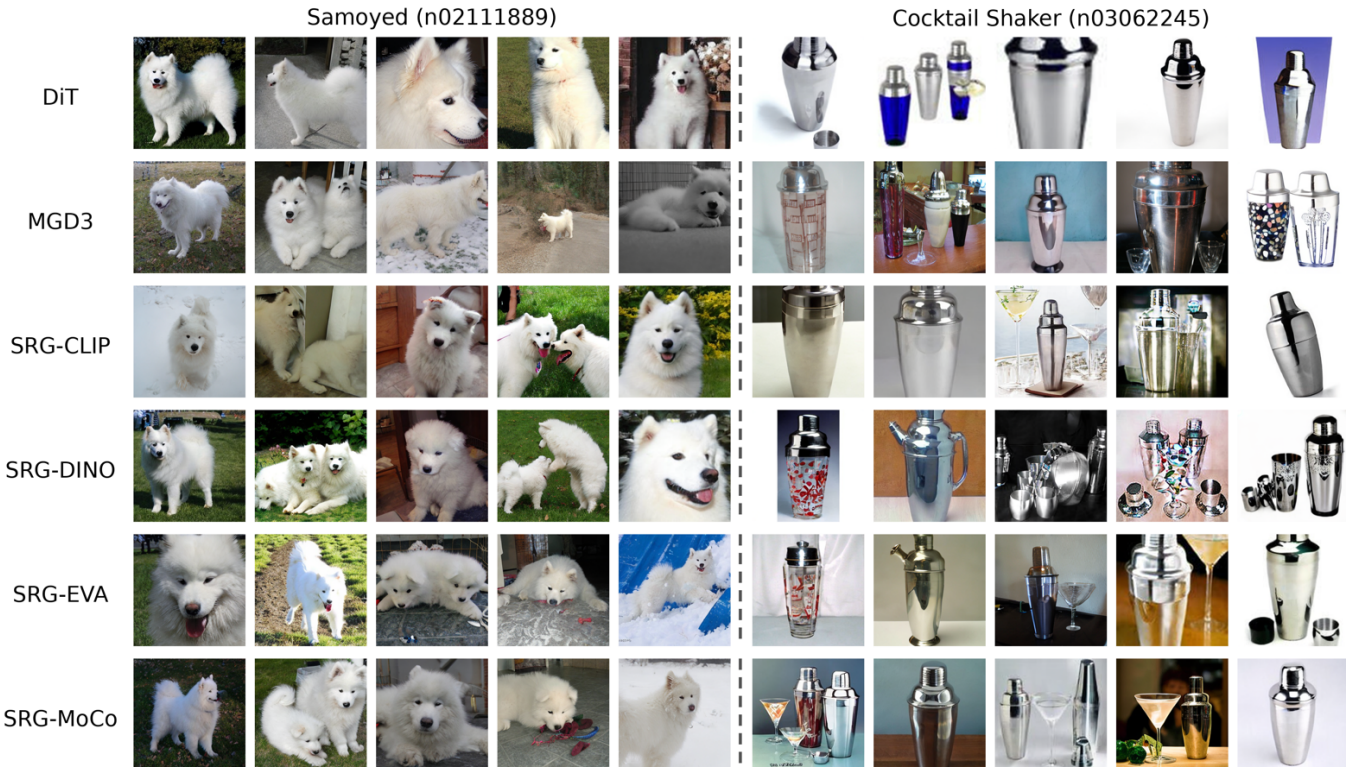


Figure 7: Samples generated by DiT, MGD3, and SRG with different SSL encoders for two representative classes at IPC $= 5$. The generated samples exhibit different visual characteristics across methods and guidance encoders.

Qualitative Analyses

Evolution of Prototype Distances

We analyze the evolution of SSL-space representations throughout the denoising process for two representative ImageWoof classes: Beagle (n02088364) and Samoyed (n02111889). At each denoising step, we measure the cosine distance, defined as one minus cosine similarity, between the generated sample and its nearest prototype from the target class (positive distance), as well as its nearest prototype from a non-target class (negative distance). As shown in Fig. 6, the first row presents the complete trajectories, while the second and third rows provide detailed views of the positive and negative distances, respectively, during the final 10 steps. Compared with unguided DiT, SRG consistently moves the generated samples closer to target-class prototypes, as indicated by the lower positive distances, while retaining separation from the nearest non-target prototypes. These results confirm that SRG modifies prototype-based distances as intended and suggest that improved coverage of SSL prototypes can benefit downstream adaptation of the SSL model.

Distilled Samples

Fig. 7 presents distilled samples produced by DiT, MGD3, and SRG with different SSL encoders for Samoyed (n02111889) from ImageWoof and Cocktail Shaker (n03062245) from ImageNet-IDC at IPC $= 5$. The unguided DiT samples exhibit relatively similar visual characteristics. In contrast, MGD3 and the SRG variants produce more diverse appearances, including variations in object shape, pose, texture, and background. The samples generated by SRG also differ across SSL encoders, suggesting that each encoder emphasizes different representational properties and consequently induces a different generation trajectory. Overall, the results indicate that the choice of guidance space affects not only downstream utility but also the visual modes captured by the distilled dataset.

Comparison of Samples and Neighbors

We compare each distilled sample with the real sample nearest to its assigned prototype (Prototype Neighbor) and the real sample nearest to the generated sample (Sample Neighbor). As shown in Fig. 8, SRG preserves the dominant visual attributes of the prototype neighbor in many cases without exactly reproducing it. This observation indicates that SRG balances SSL-space guidance with the visual prior of the pretrained diffusion model. An illustrative example appears in the fourth Samoyed column: the prototype neighbor contains a person wearing a dog-like costume, whereas SRG generates an actual dog with a similar low-level appearance but different semantic content. Overall, these results show that SRG integrates representation-space targets with the generative prior, producing samples that preserve structure around the assigned prototypes while remaining on a plausible visual manifold.

Figure 8: Comparison of prototype neighbors, SRG-generated samples, and sample neighbors in the DINOv2 representation space at IPC $= 5$. Each column corresponds to one prototype.

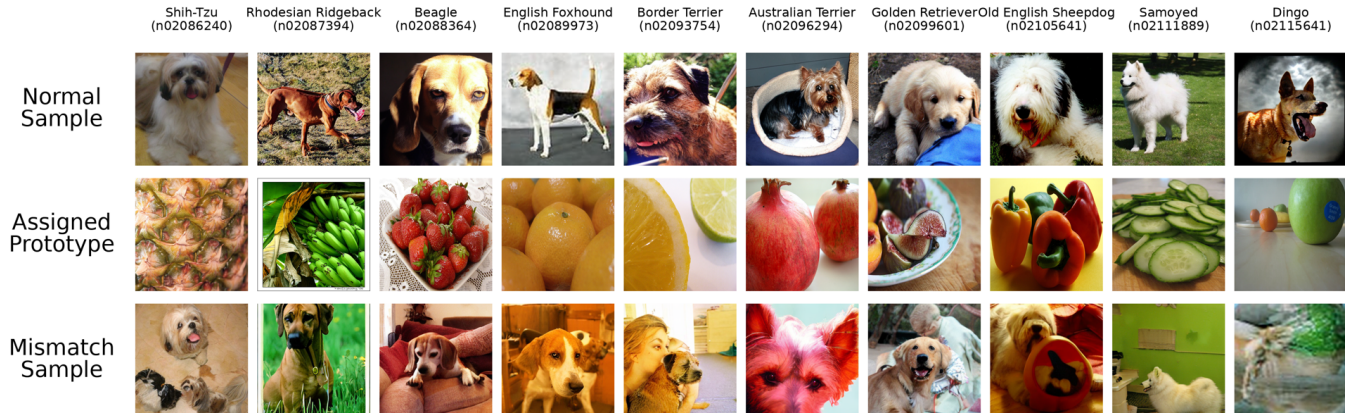


Figure 9: Comparison of samples generated with standard SRG guidance, mismatched prototype neighbors, and mismatched generated samples. The first samples generated using IPC $= 5$ are illustrated.

Table 13: Comparison of downstream validation accuracy between standard SRG guidance and mismatched prototype guidance using IPC $= 5$.

Set	Woof	Fruits
Woof	$92.0_{\pm 0.5}$	$10.8_{\pm 2.5}$
Fruits	$8.1_{\pm 1.8}$	$89.1_{\pm 0.7}$
Mismatch	$78.5_{\pm 2.1}$	$72.2_{\pm 2.0}$

Cross-Class Prototype Guidance

To further examine the interaction between the generative prior and SSL-space guidance, we conduct a controlled intervention in which samples use class conditions from Woof but are guided by prototypes constructed from Fruits, and compare the generated samples (Mismatch Sample) with samples obtained using standard SRG guidance (Standard Sample), as well as the nearest neighbor to the assigned mismatch prototype (Assigned Prototype). As shown in Fig. 9, the mismatch samples retain the overall shapes and object structures of dogs but exhibit atypical illumination and local details. We compare their downstream behavior with that of standard SRG samples in Table 13. When evaluated on Woof, the mismatched samples achieve lower accuracy than standard Woof SRG samples, yet attain substantial accuracy on Fruits. Such cross-dataset performance far exceeds that when standard samples are evaluated with another dataset. These results indicate SRG samples contain information associated with both the generator conditions that keep global visual identity, and the SSL space prototypes that remain discriminative semantic structure.