Hierarchical Features Matter: A Deep Exploration of Progressive Parameterization Method for Dataset Distillation

Supplementary Material

A. Literature Reviews on Dataset Distillation

A.1. Dataset Distillation in Pixel Space

In this section, we review the methodology of optimizing synthetic dataset S with the surrogate objective in pixel space, which provides the basic optimization objective for all parameterization dataset distillation methods.

A.1.1 DC [22].

Dataset Distillation (DD) [19] aims at optimizing the synthetic dataset $\mathcal{S}$ with a bi-level optimization. The main idea of bi-level optimization is that a network with parameter $\theta_{\mathcal{S}}$, which is trained on $\mathcal{S}$, should minimize the risk of the real dataset $\mathcal{T}$. However, due to the need to pass through an unrolled computation graph, DD brings about a significant amount of time overhead. Based on this, DC introduces a surrogate objective, which aims at matching the gradients of a network during the optimization. For a network with parameters θ_S trained on the synthetic data for some number of iterations, the matching loss is

$$\mathcal{L}_{\mathrm{DC}} = 1 - \frac{\nabla_\theta \ell^{\mathcal{S}}(\theta) \cdot \nabla_\theta \ell^{\mathcal{T}}(\theta)}{\|\nabla_\theta \ell^{\mathcal{S}}(\theta)\| \, \|\nabla_\theta \ell^{\mathcal{T}}(\theta)\|}, \tag{1}$$

where $\ell^{\mathcal{T}}(\cdot)$ represents the loss function (e.g., CE loss) calculated on real dataset $\mathcal{T}$, and $\ell^{\mathcal{S}}(\cdot)$ is the same loss function calculated on synthetic dataset $\mathcal{T}$.

A.1.2 DM [21].

Despite DC significantly reducing time consumption through surrogate, bi-level optimization still introduces a substantial amount of time overhead, especially when dealing with high-resolution and large-scale datasets. DM achieves this by using only the features extracted from networks ψ with random initialization as the matching target, the matching loss is

$$\mathcal{L}_{\mathrm{DM}} = \sum_{c} \left\| \frac{1}{|\mathcal{T}_c|} \sum_{\mathbf{x} \in \mathcal{T}_c} \psi(\mathbf{x}) - \frac{1}{|\mathcal{S}_c|} \sum_{\mathbf{s} \in \mathcal{S}_c} \psi(\mathbf{s}) \right\|^2, \tag{2}$$

where $\mathcal{T}_c$ and $\mathcal{S}_c$ represents the real and synthetic images from class c respectively.

A.1.3 MTT [1].

Distinct from the short-range optimization introduced from DC, MTT utilizes many expert trajectories $\{\theta_t^*\}_0^T$ which are obtained by training networks from scratch on the full real dataset and choose the parameter distance the matching objective. During the distillation process, a student network is initialized with parameters θ_t^* by sample expert trajectory at timestamp t and then trained on the synthetic data for some number of iterations N, the matching loss is

$$\mathcal{L}_{\mathrm{MTT}} = \frac{\left\| \hat{\theta}_{t+N} - \theta_{t+M}^* \right\|^2}{\left\| \theta_t^* - \theta_{t+M}^* \right\|^2}, \tag{3}$$

where θ_{t+M}^* represents the expert trajectory at timestamp $t+M$.

A.2. Dataset Distillation in Feature Domain

In this section, we review the methodology of parameterization dataset distillation built upon the aforementioned dataset distillation methods, achieving better performance by employing a differentiable operation $\mathcal{F}(\cdot)$ to shift the optimization space from pixel space to various feature domain, which can be formulated as

$$\mathcal{S} = \{\mathcal{F}(\mathbf{z})\}. \tag{4}$$

where $\mathbf{z}$ represents latent code in the feature domain corresponding to $\mathcal{F}(\cdot)$.

A.2.1 HaBa [13].

HaBa breaks the synthetic dataset into bases and a small neural network called hallucinator which is utilized to produce additional synthetic images. By leveraging this technique, the resulting model could be regarded as a differentiable operation and produce more diverse samples. However, HaBa simultaneously optimizes the bases and the hallucinator, neglecting the relationship between the two feature domains. This leads to unstable optimization during the training process.

A.2.2 IDC [10].

IDC proposes a principle that small-sized synthetic images often carry more effective information under the same spatial budget and utilize an upsampling module as the differentiable operation. Despite employing a differentiable operation, the optimization of IDC is still the pixel space, which resulted in the loss of effective information gain obtained from other feature domains.

A.2.3 FreD [15].

FreD suggests that optimizing for the main subject in the synthetic image is more instructive than optimizing for all the details. Therefore, FreD employs discrete cosine transform (DCT) as the differentiable operation and uses a learnable mask matrix to remove high-frequency information, ensuring that the optimization process only occurs in the low-frequency domain. This allows the synthetic dataset to achieve higher performance and generalization. However, FreD overlooks the effective guiding information within the high-frequency domain and fails to connect the two feature domains produced by DCT, leading to potential incomplete optimization.

A.2.4 GLaD [2].

Different from existing methods [3, 7, 17, 24] utilizing diffusion models [25, 26], GLaD employs a pre-trained generative model (i.e., GAN) and distills the synthetic dataset in the corresponding latent space. By leveraging the capability of a generative model to map latent noise to image patterns, GLaD achieves better generalization to unseen architecture and scale to high-dimensional datasets. However, for StyleGAN, the earlier layers tend to provide the information about the main subject in an image while the later layers often contribute to the details. However, GLaD attempts to balance the low-frequency information with the high-frequency information by selecting an intermediate layer as a fixed optimization space, discarding the guiding information from the earlier layers can lead to incomplete optimization. Another limitation of GLaD is the need for a large number of preliminary experiments. GLaD selects a specific intermediate layer suitable for all datasets for different distillation methods, However, under the same distillation method, the optimal intermediate layer corresponding to different datasets is not the same, especially when the manifold of the datasets varies greatly, which suggests that GLaD cannot spontaneously adapt to different datasets, distillation methods, and GANs.

B. Additional Experimental Results

B.1. More Comparisons with GLaD

To expand the optimization space, the method we proposed utilizes hierarchical feature domains composed of intermediate layers from GAN. To investigate whether optimization across multiple feature domains is superior to optimization within a single fixed feature domain, we evaluate the performance by simply expanding the optimization space based on the baseline. As shown in Table 1, compared to GLaD, which only selects a single yet optimal intermediate layer of the GAN as the optimization space, H-PD has successfully achieved considerable improvement, validating our viewpoint that the optimization result from the previous feature domain can serve as better starting point for subsequent feature domain. Please note the result is obtained by not selecting $\mathcal{S}^*$.

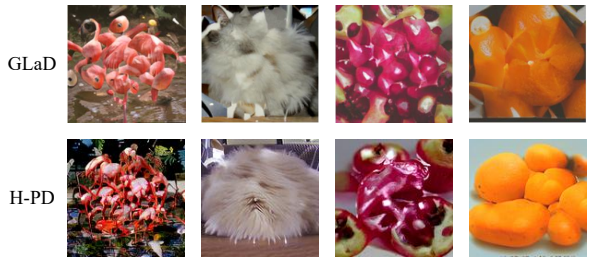


Figure 1. The comparison of visualization.

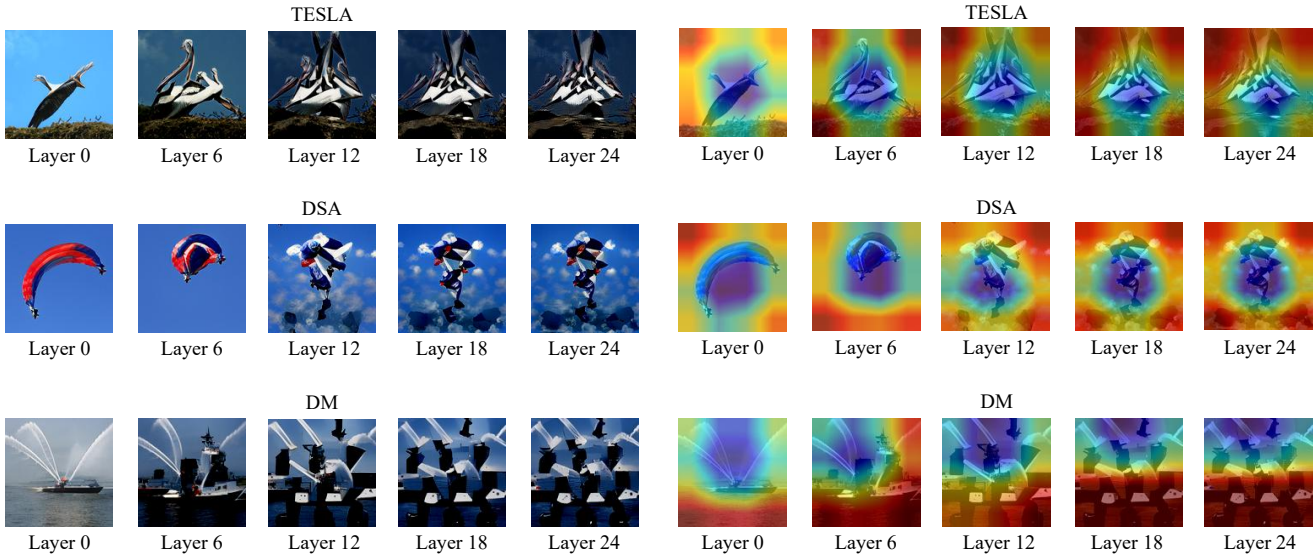
To present a more comprehensive comparison, we evaluate the cross-architecture performance of a high-resolution synthetic dataset under the same setting (i.e., DSA on ImageNet-[A, B, C, D, E] under IPC=1). As shown in Table 2, our proposed H-PD still achieves considerable improvements, demonstrating the stability of our proposed method. Figure 1 illustrates the comparison of synthetic dataset visualization generated by H-PD and GLaD using the same initial image. The images produced by H-PD achieve a good balance between content and style. On one hand, H-PD tends to preserve more main subject information by optimizing in the earlier layers of the GAN. On the other hand, since H-PD also undergoes optimization in the later layers, the synthetic images tend to be sharper and rarely produce the kaleidoscope-like patterns that are common in the GLaD method.

B.2. Visualizing Morphological Transition of Synthetic Images

As shown in Figure 2a, we demonstrate the visualization changes of the synthetic image throughout the optimization process. Layer 0 represents the initial image produced by StyleGAN-XL using averaged noise, and Layer i indicates the image when the optimization space reaches layer i. In the early stage of optimization, since the optimization space is located in the earlier layer of the GAN, the optimization object primarily focus on the main subject of the synthetic image. Meanwhile, GAN still maintains a high degree of integrity which leads to a strong constraint on the slight changes in the latent produced during the optimization process, which can be transformed into patterns resembling real images instead of noises. Thus the tendency in the early stage of optimization is to generate images that better conform to the constraint of distillation loss yet appear more realistic, leading to produce synthetic images that can be regarded as a better starting point for the subsequent

Alg.	Opimization Space	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
TESLA	Fixed (Pixel)	51.7±0.2	53.3±1.0	48.0±0.7	43.0±0.6	39.5±0.9	41.8±0.6	22.6±0.6	37.3±0.8	22.4±1.1	22.6±0.4
	Fixed (GAN)	50.7±0.4	51.9±1.3	44.9±0.4	39.9±1.7	37.6±0.7	38.7±1.6	23.4±1.1	35.8±1.4	23.1±0.4	26.0±1.1
	Unfixed	**53.1**±0.8	**55.4**±0.7	**47.5**±0.9	**44.1**±0.6	**40.8**±0.7	**42.8**±1.0	**27.0**±0.6	**37.6**±0.9	**24.7**±0.7	**28.3**±0.8
DSA	Fixed (Pixel)	43.2±0.6	47.2±0.7	41.3±0.7	34.3±1.5	34.9±1.5	34.2±1.7	22.5±1.0	32.0±1.5	21.0±0.9	22.0±0.6
	Fixed (GAN)	44.1±2.4	49.2±1.1	42.0±0.6	35.6±0.9	35.8±0.9	35.4±1.2	22.3±1.1	33.8±0.9	20.7±1.1	22.6±0.8
	Unfixed	**46.1**±0.7	**50.0**±0.9	**43.8**±1.4	**37.1**±0.9	**36.6**±0.6	**36.2**±0.5	**22.7**±0.3	**34.9**±1.5	**21.2**±0.8	**23.1**±0.3
DM	Fixed (Pixel)	39.4±1.8	40.9±1.7	39.0±1.3	30.8±0.9	27.0±0.8	30.4±2.7	20.7±1.0	26.6±2.6	20.4±1.9	20.1±1.2
	Fixed (GAN)	41.0±1.5	42.9±1.9	39.4±1.7	33.2±1.4	30.3±1.3	32.2±1.7	21.2±1.5	27.6±1.9	21.8±1.8	22.3±1.6
	Unfixed	**42.3**±1.5	**44.1**±1.5	**41.3**±1.7	**33.7**±1.1	**31.5**±1.1	**34.0**±1.2	**23.1**±1.3	**28.9**±1.4	**24.3**±1.3	**22.8**±0.8

Table 1. Abltion study on optimization space comparison. "Fixed (Pixel)" refers to optimize in pixel space and "Fixed (GAN)" refers to GLaD, while Unfixed refers to optimize in multiple feature domains.



(a) Visualization of synthetic images.

(b) Visualization of corresponding CAM.

Figure 2. The visualization change of synthetic images and corresponding CAM during the optimization process using different distillation methods. "Layer" refers to the index of intermediate layers provided by StyleGAN-XL.

Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-D
Pixel	38.3±4.7	32.8±4.1	27.6±3.3	25.5±1.2	23.5±2.4
GLaD	37.4±5.5	41.5±1.2	35.7±4.0	27.9±1.0	29.3±1.2
H-PD	**40.7**±2.1	**42.9**±1.8	**37.2**±2.2	**30.1**±1.7	**29.7**±1.8

Table 2. Higher-resolution (256×256) synthetic dataset (using DSA) cross-architecture performance (%).

optimization process.

In the later stage of optimization, the main subject of the synthetic image no longer undergoes significant changes, and the optimization objective shifts along with the movement of the optimization space, focusing more on the details of the synthetic images. As shown in Figure 2a, due to the weakened generative constraint of the incomplete GAN, the final synthetic image becomes similar to the indistinguishable and distorted image produced by existing distillation methods. Building upon the better synthetic image obtained through the optimization process in the earlier layers, different distillation methods gradually incorporate more guidance-oriented customized patterns into the synthetic image, achieving further performance improvement, which has also been proved by recent work [23].

B.3. Qualitative Interpretation using CAM

We additionally introduce CAM [14] to visualize the heatmap of class-relevant information in the synthetic images as shown in Figure 2b, which also demonstrates our perspective from another aspect. The blue areas represent

Layers	Optimization	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
1	50	53.6±0.2	55.2±1.5	47.3±0.5	44.1±0.7	40.5±1.1	43.8±0.4	26.6±0.7	37.1±0.6	22.9±0.5	27.8±1.0
	100	55.3±0.8	57.1±0.7	49.1±0.9	**46.6**±0.4	42.2±1.5	44.9±1.2	28.6±0.6	39.4±0.8	25.9±0.7	**30.1**±1.2
	200	55.4±0.7	57.5±1.1	48.6±0.8	46.2±0.9	**43.6**±0.6	**45.7**±0.5	**28.7**±0.4	39.4±0.6	25.5±0.5	29.8±0.2
2	50	51.3±0.9	54.2±1.1	46.3±0.8	44.1±1.2	40.3±1.2	41.8±1.4	27.1±0.6	36.5±1.1	23.0±1.2	28.1±1.3
	100	55.1±0.6	57.4±0.3	**49.5**±0.6	46.3±0.9	43.0±0.6	45.4±1.1	28.3±0.2	**39.7**±0.8	25.6±0.7	29.6±1.0
	200	**55.6**±0.9	**57.9**±0.5	49.4±0.3	46.0±0.1	43.5±0.4	45.1±0.7	28.6±0.2	39.3±0.8	**25.9**±1.1	29.9±0.6
4	50	51.8±0.7	52.9±1.2	46.1±1.5	42.3±0.5	39.8±0.5	40.9±1.3	24.7±1.1	35.9±0.5	21.2±1.7	25.3±1.1
	100	53.3±0.8	54.2±1.1	47.3±1.2	41.8±1.7	42.7±0.6	27.7±0.5	27.1±1.0	27.0±0.9	22.5±1.4	26.4±1.2
	200	55.0±1.0	57.0±1.3	48.1±1.6	45.2±0.5	42.1±1.4	45.0±0.5	27.2±0.9	38.8±1.1	24.6±0.5	28.4±0.8

Table 3. Abltion study on layers combination and optimization allocation using TESLA. "Layers" refers to the number of layers per optimization space, "Optimization" refers to the number of SGD steps allocated in each optimization space.

Layers	Optimization	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
1	50	45.2±1.2	48.3±1.3	42.0±0.4	36.2±0.7	35.0±0.8	35.8±1.1	22.7±1.0	33.5±0.5	21.1±1.5	22.7±0.8
	100	46.2±0.7	**51.1**±0.4	43.3±1, 1	37.2±0.5	36.6±0.9	36.7±1.3	22.9±0.8	35.6±1.1	22.1±1.5	23.8±0.7
	200	46.5±0.9	50.7±1.1	43.8±0.2	37.3±0.7	**37.6**±0.7	36.9±1.3	**24.3**±0.5	34.9±0.3	**22.6**±1.3	23.6±0.7
2	50	44.8±0.4	48.9±0.9	42.1±1.1	35.6±1.0	36.6±0.6	34.2±1.1	22.1±0.6	33.3±1.6	20.0±1.3	22.7±0.8
	100	**46.9**±0.8	50.7±0.9	**43.9**±0.7	**37.4**±0.4	37.2±0.3	36.9±0.8	24.0±0.8	35.3±1.0	22.4±1.1	24.1±0.9
	200	46.8±0.5	50.8±0.3	43.4±0.6	37.0±1.3	37.3±0.5	**37.1**±0.7	23.8±1.3	**35.6**±1.1	22.1±1.2	**24.6**±1.3
4	50	43.6±0.7	47.8±0.7	40.4±0.6	34.6±0.5	34.2±0.8	33.4±1.2	21.3±0.9	32.7±1.4	19.9±0.5	21.6±0.6
	100	45.7±0.7	49.4±0.9	43.1±1.1	36.1±1.3	36.4±0.8	35.2±0.6	23.4±1.1	34.7±0.5	21.3±1.1	23.5±1.3
	200	46.3±0.8	50.1±0.9	43.2±0.7	37.0±0.4	36.8±1, 6	36.2±1.0	23.3±1.3	34.4±1.4	21.6±0.8	23.7±0.5

Table 4. Abltion study on layers combination and optimization allocation using DSA.

regions of class-relevant information, which can produce the largest gradient during the training process. Conversely, the red areas indicate regions of class-irrelevant information, with deeper colors signifying higher degrees of corresponding information. In the early stage of optimization, the class-relevant information of the main subject in the synthetic image produced by various distillation methods is compressed.

Interestingly, for the gradient matching methods TESLA and DSA, which rely on long-range and short-range gradient matching respectively, the class-relevant information of the main subject remains unchanged when optimization space changes to later layers, while the gradient that can be produced by the image background (e.g., corners) are further decreased, as indicated by the deeper red color, even though the changes in the background are hardly observable by the naked eye during the optimization process. However, for the feature matching method DM, compared to the visualized kaleidoscope-like pattern, the visualization of corresponding CAM shows an unbalanced distribution and focuses on areas not typically observed by humans. We believe this phenomenon also explains the poorer performance of DM compared with gradient matching methods. Compared to the synthetic images with a centralized concentration of class-relevant information produced by TESLA and DSA, the images generated by DM are too diverse due to fitting all the features of the entire dataset including the class-irrelevant features, which is disadvantageous for training neural networks on tiny distilled datasets.

B.4. Layers Combination and Optimization Allocation

As discussed, we adopt a uniform sampling method that evaluates the synthetic dataset per 100 optimization epochs (even less when using DM) to align with the evaluation method of the baseline (i.e., GLaD). Additionally, we decompose StylGAN-XL into $G_{11} \circ \cdots \circ G_1 \circ G_0(\,\cdot\,)$ to align with the time complexity of the baseline. We present an ablation study on the allocation of optimization epochs per optimization space. Building on this, we further explore the impact of combining different numbers of intermediate layers into a single optimization space and allocating different numbers of optimization epochs to each optimization space on the performance of the synthetic dataset. For all distillation methods, we explore the impact of varying optimization spaces by using combinations of 1, 2, and 4 intermediate layers within each optimization space. Under the same optimization space setting, for TESLA and DSA, we

investigated the effects of different numbers of optimization epochs allocated to each optimization space by using 50, 100, and 200. For DM, due to the overfitting issue caused by feature matching, we used 10, 20, and 50 as the number of optimization epochs per optimization space.

The results for TESLA and DSA are shown in Table 3 and Table 4. Combining 1 or 2 intermediate layers as a single optimization space does not produce a significant impact on the performance, indicating that existing redundant feature spaces provided by GAN contribute little to the distillation tasks and may even lead to a negative effect. Under this setting, allocating 50 optimization epochs per optimization space produces a clear phenomenon of optimization not converging. However, when the number of optimization epochs comes to 100 or 200, the optimization converges without significant performance differences. Achieved by implicitly selecting the optimal synthetic dataset through the proposed class-relevant feature distance metric, allowing us to avoid overfitting issues to some extent through a certain level of optimization path withdrawal. Therefore, we choose 100 epochs as the basic setting to reduce time complexity in the actual training process. When using 4 intermediate layers as an optimization space, the performance is decreased even when setting optimization epochs to 200, indicating that too few feature domains could not provide sufficiently rich guiding information, forcing the optimization process to require more epochs to converge, demonstrating the superiority of our proposed H-PD in utilizing multiple feature domains.

The results for DM are shown in Table 5. Similar to TESLA and DSA, Combining 1 or 2 intermediate layers as a single optimization space results in similar performance, while combining 4 intermediate layers as optimization space leads to a significant performance drop. However, under the same optimization space settings, an excessive number of optimization epochs often leads to a severe decline in performance when using DM as the distillation method. As aforementioned, DM is unable to focus on class-relevant information, which causes an irreversible loss of the main subject information in the synthetic image after deploying a large number of optimization epochs in a specific feature domain, which in turn leads to a situation where the informative guidance provided by subsequent feature domains could not be effectively incorporated into the synthetic image, resulting in performance degradation. In this case, even the proposed class-relevant feature distance could not effectively select a superior synthetic dataset. To align with the approach of decomposing GAN used in TESLA and DSA, we ultimately combine 2 intermediate layers as an optimization space and deploy 20 optimization epochs as the experimental setting for DM.

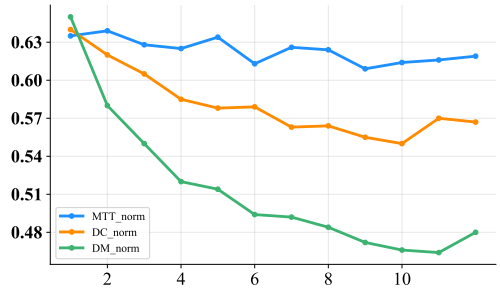


Figure 3. Quantitative results of loss function value using different distillation methods. Note that we normalize all the values for clear comparison.

B.5. Ablation Study on Searching Strategy

To better utilize the informative guidance provided by multiple feature domains, we propose class-relevant feature distance as an evaluation metric for implicitly selecting the optimal synthetic dataset. We demonstrate the ablation study using different implicit evaluation metrics, as shown in Table 6, the metric we proposed outperforms the use of loss function value corresponding to the distillation methods as the metric under all settings. It is worth noting that, although the accuracy of the model trained on the synthetic dataset can be used as an explicit evaluation metric for the data distillation task, the evaluation process incurred much greater time overhead than the distillation task itself, rendering it impractical for actual training processes.

To explore the principle of the superiority of class-relevant feature distance, we first discussed the respective limitations of directly using existing distillation loss function value as the evaluation metric. The tendency of different distillation loss functions is shown in Figure 3. For TESLA, the loss function is obtained by calculating the distance between the student network parameters and the teacher network parameters. However, in order to consider diversity, TESLA selects a random initialization method when initializing the student network parameters, and the expert trajectory also comes from the training process of models with different initialization, leading to a significant fluctuation caused by utilizing different initialization parameters. For DSA, the loss function utilizes neural network gradients as guidance. However, when IPC=1, the proxy neural network used in each optimization process is randomly initialized, causing DSA to face the same issue as TESLA, where the loss function is affected by network parameter initialization. As for DM, the loss function is obtained from the feature distance between the dataset features extracted by randomly initialized networks, resulting in the same impact of network initialization parameters on this loss function. Additionally, DM suffers from severe overfitting in the later stages of optimization due to fitting

Layers	Optimization	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
1	10	42.1±2.2	44.1±1.6	**41.7**±1.7	33.9±1.2	31.3±1.9	34.2±2.1	**24.1**±1.4	**29.7**±0.7	24.1±1.6	22.6±1.3
	20	41.6±1.6	**44.8**±1.8	41.3±1.4	34.1±2.1	31.2±0.5	33.7±0.6	24.0±1.3	29.6±1.7	23.4±0.8	23.7±1.9
	50	40.2±1.6	43.4±1.7	40.2±2.0	33.1±1.3	29.7±1.8	32.6±1.9	23.1±2.1	28.2±1.6	22.1±0.8	21.0±0.5
2	10	41.4±1.7	43.5±1.3	40.4±0.9	34.1±1.3	31.3±1.8	33.6±1.7	22.4±1.6	28.3±2.1	23.1±1.7	22.9±1.5
	20	**42.8**±1.2	44.7±1.3	41.1±1.3	**34.8**±1.5	**31.9**±0.9	**34.8**±1.0	23.9±1.9	29.5±1.5	**24.4**±2.1	**24.2**±1.1
	50	40.1±1.8	42.6±2.0	40.2±1.6	32.6±1.7	29.7±1.3	33.1±0.6	21.6±0.7	27.7±1.6	22.2±1.3	22.4±1.9
4	10	39.9±1.4	42.5±1.0	40.4±1.8	32.4±1.6	30.1±2.4	32.7±2.3	20.9±1.6	27.5±2.2	22.5±1.7	21.8±1.2
	20	40.6±1.3	42.5±1.6	39.6±2.1	32.2±1.5	30.1±1.3	32.9±1.8	21.6±1.5	27.3±1.2	21.7±2.3	22.3±1.6
	50	40.4±1.7	42.7±1.3	39.9±1.2	32.0±1.4	30.3±1.9	32.6±1.6	22.0±1.1	27.8±0.9	21.1±1.7	22.6±1.4

Table 5. Ablation study on layers combination and optimization allocation using DM.

Alg.	Searching Basis	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
TESLA	-	54.7±0.8	56.2±0.7	48.1±0.9	45.4±0.9	41.8±0.6	43.8±0.8	28.1±1.0	38.5±1.2	24.1±0.5	28.7±0.9
	Loss Value	53.6±0.9	56.9±0.7	48.3±0.8	45.0±0.6	41.0±1.2	44.5±0.8	27.5±1.4	37.8±0.7	25.1±0.9	27.6±1.0
	Feature Distance	**55.1**±0.6	**57.4**±0.3	**49.5**±0.6	**46.3**±0.9	**43.0**±0.6	**45.4**±1.1	**28.3**±0.2	**39.7**±0.8	**25.6**±0.7	**29.6**±1.0
DSA	-	45.9±0.7	50.1±1.1	43.1±1.4	36.9±0.8	36.8±0.6	36.0±0.9	23.6±0.8	34.5±0.4	21.9±0.8	23.2±0.9
	Loss Value	46.6±1.3	48.9±1.7	43.6±1.1	36.1±1.2	36.6±0.5	36.2±0.9	23.1±0.6	33.6±0.7	21.3±1.1	22.8±1.0
	Feature Distance	**46.9**±0.8	**50.7**±0.9	**43.9**±0.7	**37.4**±0.4	**37.2**±0.3	**36.9**±0.8	**24.0**±0.8	**35.3**±1.0	**22.4**±1.1	**24.1**±0.9
DM	-	42.4±1.6	44.2±2.1	41.0±1.2	34.0±1.2	31.1±1.0	34.5±2.1	23.1±0.9	29.0±1.5	24.1±1.4	22.6±1.5
	Loss Value	41.6±1.8	44.4±1.4	40.7±2.1	34.6±1.7	30.1±1.3	34.5±1.3	23.6±1.2	28.7±1.3	24.4±1.3	21.2±1.2
	Feature Distance	**42.8**±1.2	**44.7**±1.3	**41.1**±1.3	**34.8**±1.5	**31.9**±0.9	**34.8**±1.0	**23.9**±1.9	**29.5**±1.5	**24.4**±2.1	**24.2**±1.1

Table 6. Quantitative results on searching basis. "-" refers to not employing a searching strategy, "Loss Value" refers to directly using corresponding loss function value as the searching basis, "Feature Distance" refers to the use of proposed class-relevant distance as a searching basis

to the useless features. In summary, the loss functions corresponding to the three distillation methods could not serve as effective evaluation metrics due to the excessive diversity.

Figure 4. The visualization comparison of CAM between pre-trained model and random model using DM.

Distinguished from existing distillation methods, where the loss function is influenced by the need to fit diversity, our proposed class-relevant feature distance effectively addresses this issue by using CAM, which is calculated by utilizing a pre-trained neural network, and we utilize a ResNet-18 trained on ImageNet-1k as a proxy model for computing CAM. As shown in Figure 4, we demonstrate the difference between the visualization obtained using the pre-trained model and those obtained using a randomly initialized model. The observation indicates that there is a significant difference in the regions of interest for the two, by utilizing a pre-trained model with fixed parameters, we can better identify the feature regions that are beneficial for the classification task (i.e., larger gradients). Therefore, our proposed metric successfully leverages this strong supervisory signal to achieve data selection while eliminating the strong correlation between the loss function and the proxy model parameters.

Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E
GLaD-TESLA	50.7±0.4	51.9±1.3	44.9±0.4	39.9±1.7	37.6±0.7
+ Average Initialization	**51.9**±1.0	**53.5**±0.7	**46.1**±0.9	**41.0**±0.7	**39.1**±1.0
GLaD-DSA	44.1±2.4	**49.2**±1.1	**42.0**±0.6	35.6±0.9	**35.8**±0.9
+ Average Initialization	**45.4**±0.6	48.9±0.8	40.6±0.7	**36.4**±0.5	34.8±0.3
GLaD-DM	41.0±1.5	42.9±1.9	39.4±1.7	**33.2**±1.4	30.3±1.3
+ Average Initialization	**41.5**±1.2	**43.2**±1.6	**39.9**±1.7	32.2±0.9	**30.8**±1.3

Table 7. Ablation study of average noise initialization on GLaD.

B.6. Ablation Study on Average Noise Initialization

To investigate the effect of using averaged noise as initialization, we conduct ablation experiments on both GLaD and H-PD respectively. As shown in Table 7, averaged noise often provides a significant gain for GLaD. Indicating that

using averaged noise as input tends to produce images with reduced bias that conform to the statistical characteristics of the real dataset, implying that images generated from averaged noise are usually centered within the real dataset. As aforementioned, since GLaD neglects the informative guidance from the earlier layers, leading to a lack of optimization for the main subject of the synthetic image, averaged noise can to some extent replace this operation.

Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E
H-PD-TESLA	54.1±0.5	56.8±0.4	48.9±1.3	45.0±0.7	42.1±0.6
+ Average Initialization	**55.1**±0.6	**57.4**±0.3	**49.5**±0.6	**46.3**±0.9	**43.0**±0.6
H-PD-DSA	46.5±1.0	50.4±0.4	**44.5**±0.6	**37.7**±1.1	36.9±0.7
+ Average Initialization	**46.9**±0.8	**50.7**±0.9	43.9±0.7	37.4±0.4	**37.2**±0.3
H-PD-DM	42.6±1.6	44.5±0.9	**42.3**±1.4	34.5±1.1	**32.3**±1.3
+ Average Initialization	**42.8**±1.2	**44.7**±1.3	41.1±1.3	**34.8**±1.5	31.9±0.9

Table 8. Ablation study of average noise initialization on H-PD.

As shown in Table 8, average noise initialization provides only a limited improvement for H-PD on TESLA, while using DSA and DM, averaged noise is closer to random initialization. The observation aligns with our perspective that H-PD requires optimization through all layers of the GAN, which has already led to optimization for the main subject information that conforms to the constraints of the loss function during the early stages of training. The role of averaged noise is then reduced to merely providing samples that better conform to statistical characteristics, which is also why we still employ averaged noise for H-PD to obtain a training-free optimization starting point.

Additionally, since DSA tends to optimize towards classification boundary samples or noisy samples, and DM tends to substantially modify synthetic datasets to achieve feature maximum mean discrepancy optimization, neither GLaD nor H-PD with average noise initialization can effectively improve the performance on DSA and DM. Nevertheless, TESLA is most effective in preserving the primary subject information in the synthetic images, which allows for the averaging of noise and the achievement of a relatively stable improvement.

C. Experimental Details

C.1. Dataset

We evaluate H-PD on various datasets, including a low-resolution dataset CIFAR10[11] and a large number of high-resolution datasets ImageNet-Subset.

- CIFAR-10 consists of 32×32 RGB images with 50,000 images for training and 10,000 images for testing. It has 10 classes in total and each class contains 5,000 images for training and 1,000 images for testing.
- ImageNet-Subset is a small dataset that is divided out from the ImageNet[5] based on certain characteristics. By aligning with the previous work, we use the same types of subsets: ImageNette (various objects)[9], ImageWoof (dogs)[9], ImageFruit (fruits) [1], ImageMeow (cats) [1], ImageSquawk (birds) [1], and ImageNet-[A, B, C, D, E] (based on ResNet50 performance) [2]. Each subset has 10 classes. The specific class name in each Imagenet-Subset is shown in Table 9.

C.2. Network Architecture

For the comparison of same-architecture performance, we employ a convolutional neural network ConvNet-3 as the backbone network as well as the test network. For low-resolution datasets, we employ a 3-depth convolutional neural network ConvNet-3 as the backbone network, consisting of three basic blocks and one fully connected layer. Each block includes a 3×3 convolutional layer, instance normalization [18], ReLU non-linear activation, and a 2×2 average pooling layer with a stride of 2. After the convolution blocks, a linear classifier outputs the logits. For high-resolution datasets, we employ a 5-depth convolutional neural network ConvNet-5 as the backbone network for 128×128 resolution, ConvNet-5 has five duplicate blocks, which is as the same as that in ConvNet-3. For 256×256 resolution, we employ ConvNet-6 as the backbone network.

For the comparison of cross-architecture performance, we also follow the previous work: ResNet-18 [8], VGG-11 [16], AlexNet [12], and ViT [6] from the DC-BENCH [4] resource.

C.3. Implementation details

The implementation of our proposed H-PD is based on the open-source code for GLaD [2], which is conducted on NVIDIA GeForce RTX 3090.

To ensure fairness, we utilize identical hyperparameters and optimization settings as GLaD. In our experiments, we also adopt the same suite of differentiable augmentations (originally from the DSA codebase [20]), including color, crop, cutout, flip, scale, and rotate. We use an SGD optimizer with momentum, ℓ_2 decay. The entire distillation process continues for 1200 epochs. We evaluate the performance of the synthetic dataset by training 5 randomly initialized networks on it.

To obtain the expert trajectories used in MTT, we train a backbone model from scratch on the real dataset for 15 epochs of SGD with a learning rate of 10^{-2}, a batch size of 256, and no momentum or regularization. To maintain the integration of different distillation methods, we do not use the ZCA whitening on both high-resolution datasets and low-resolution datasets different from previous work[1], which leads to a same-architecture performance drop, please note that our proposed H-PD still outperforms under the same setting. Different from GLaD which records 1000 expert trajectories for the MTT method,

Dataset	0	1	2	3	4	5	6	7	8	9
ImNet-A	Leonberg	Probiscis Monkey	Rapeseed	Three-Toed Sloth	Cliff Dwelling	Yellow Lady's Slipper	Hamster	Gondola	Orca	Limpkin
ImNet-B	Spoonbill	Website	Lorikeet	Hyena	Earthstar	Trollybus	Echidna	Pomeranian	Odometer	Ruddy Turnstone
ImNet-C	Freight Car	Hummingbird	Fireboat	Disk Brak	Bee Eater	Rock Beauty	Lion	European Gallinule	Cabbage Butterfly	Goldfinch
ImNet-D	Ostrich	Samoyed	Snowbird	Brabancon Griffon	Chickadee	Sorrel	Admiral	Great Gray Owl	Hornbill	Ringlet
ImNet-E	Spindle	Toucan	Black Swan	King Penguin	Potter's Wheel	Photocopier	Screw	Tarantula	Oscilloscope	Lycaenid
ImNette	Tench	English Springer	Cassette Player	Chainsaw	Church	French Horn	Garbage Truck	Gas Pump	Golf Ball	Parachute
ImWoof	Australian Terrier	Border Terrier	Samoyed	Beagle	Shih-Tzu	English Foxhound	Rhodesian Ridgeback	Dingo	Golden Retriever	English Sheepdog
ImNet-Birds	Peacock	Flamingo	Macaw	Pelican	King Penguin	Bald Eagle	Toucan	Ostrich	Black Swan	Cockatoo
ImNet-Fruits	Pineapple	Banana	Strawberry	Orange	Lemon	Pomegranate	Fig	Bell Pepper	Cucumber	Granny Smith Apple
ImNet-Cats	Tabby Cat	Bengal Cat	Persian Cat	Siamese Cat	Egyptian Cat	Lion	Tiger	Jaguar	Snow Leopard	Lynx

Table 9. Corresponding class names in each ImageNet-Subsets. The visualizations follow the same order.

Dataset	IPC	Synthetic steps	Expert epochs	Max expert epoch	Trajectory number	Learning rate (Learning rate)	Learning rate (Teacher)	Learning rate (Latent w)	Learning rate (Latent f)	Steps per space
CIFAR-10	1	20	3	50	100	10^{-6}	10^{-2}	10^{1}	10^{4}	100
	10	20	3	50	100	10^{-6}	10^{-2}	10^{1}	10^{4}	100
ImageNet-Subset	1	20	3	15	200	10^{-6}	10^{-2}	10^{1}	10^{4}	100

Table 10. TESLA hyper-parameters

Dataset	IPC	Learning rate (Latent w)	Learning rate (Latent f)	Steps per space
CIFAR-10	1	10^{-2}	10^{1}	20
	10	10^{-2}	10^{1}	20
ImageNet-Subset	1	10^{-2}	10^{1}	20
	10	10^{-2}	10^{1}	20

Table 11. DM hyper-parameters

Dataset	IPC	inner loop	outer loop	Learning rate (Latent w)	Learning rate (Latent f)	Steps per space
CIFAR-10	1	1	1	10^{-3}	10^{0}	100
	10	50	10	10^{-3}	10^{0}	100
ImageNet-Subset	1	1	1	10^{-3}	10^{0}	100
	10	50	10	10^{-3}	10^{0}	100

Table 12. DSA hyper-parameters

we only record 200 expert trajectories and thus largely reduce the computational costs. Additionally, while GLaD performs 5k optimization epochs on the synthetic dataset using MTT as the distillation method, we only perform 1k optimization epochs and achieve better performance both on same-architecture and cross-architecture settings, further proving the superiority of our H-PD. The detailed hyperparameters are shown in Table 11, Table 12 and Table 10.

D. More Visualizations

We provide additional visualizations of synthetic datasets generated by H-PD using diverse distillation methods, as shown in Figure 5, Figure 6, and Figure 7.

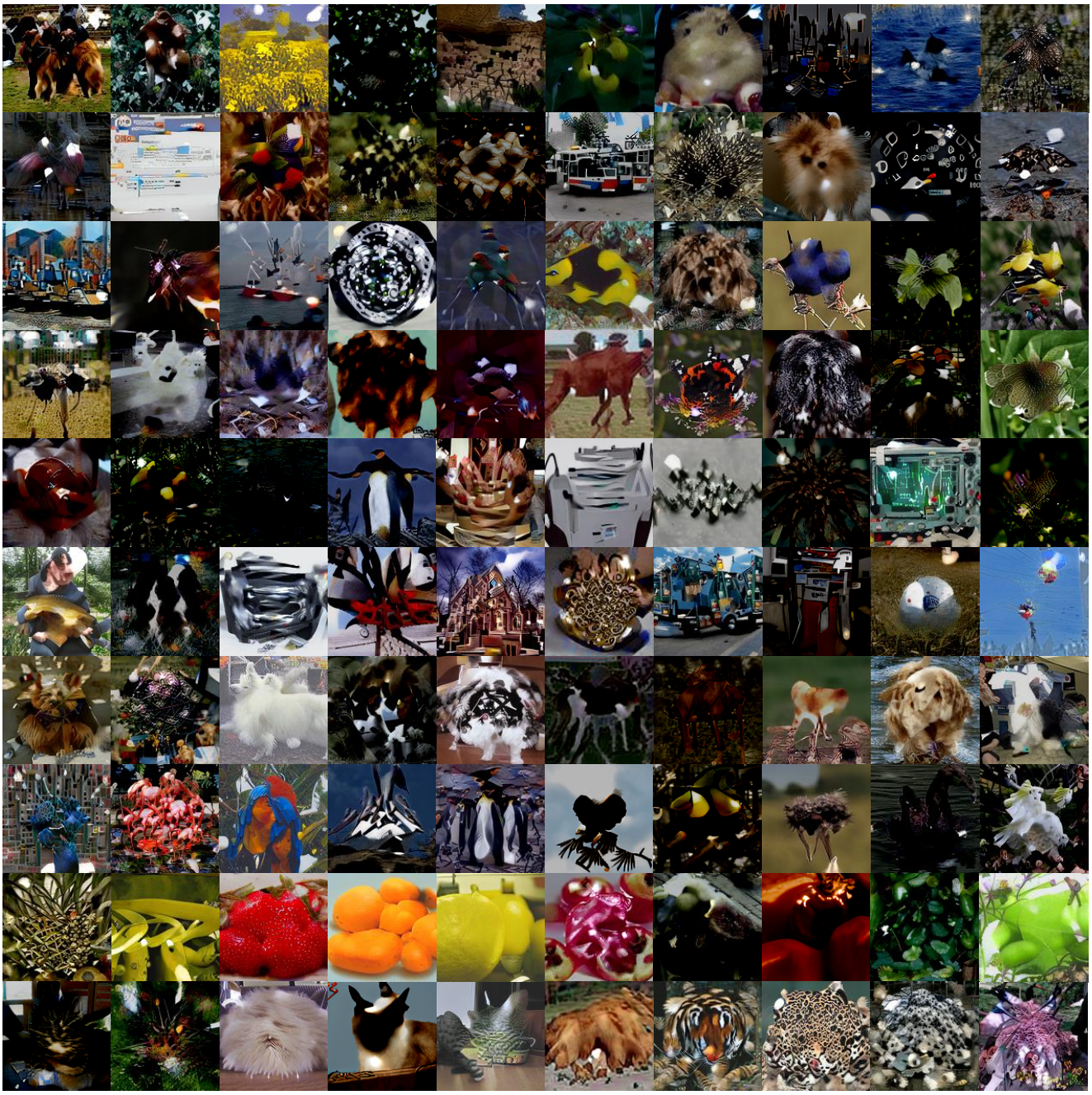


Figure 5. More visualization of the synthetic datasets using TESLA.

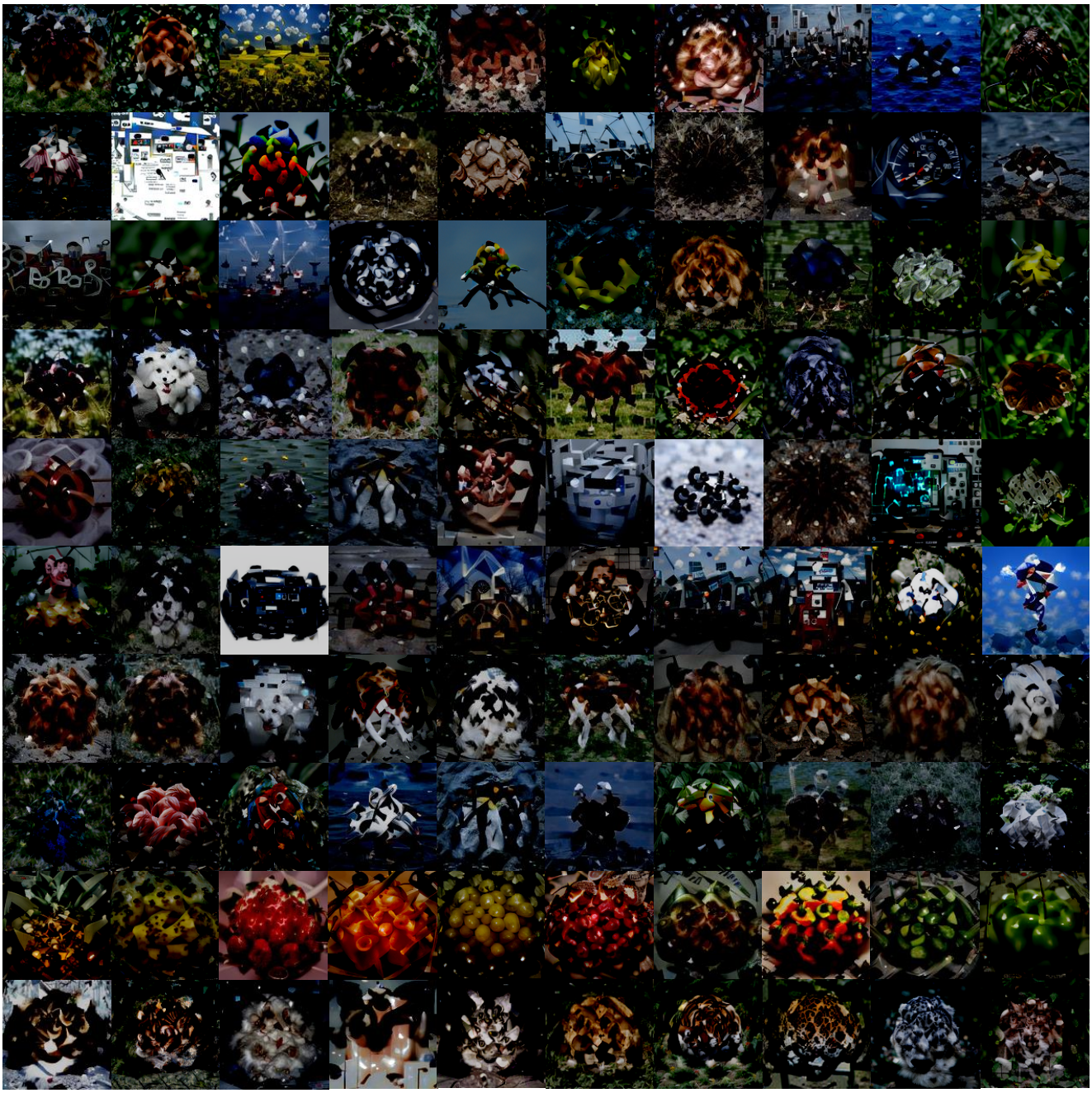


Figure 6. More visualization of the synthetic datasets using DSA.

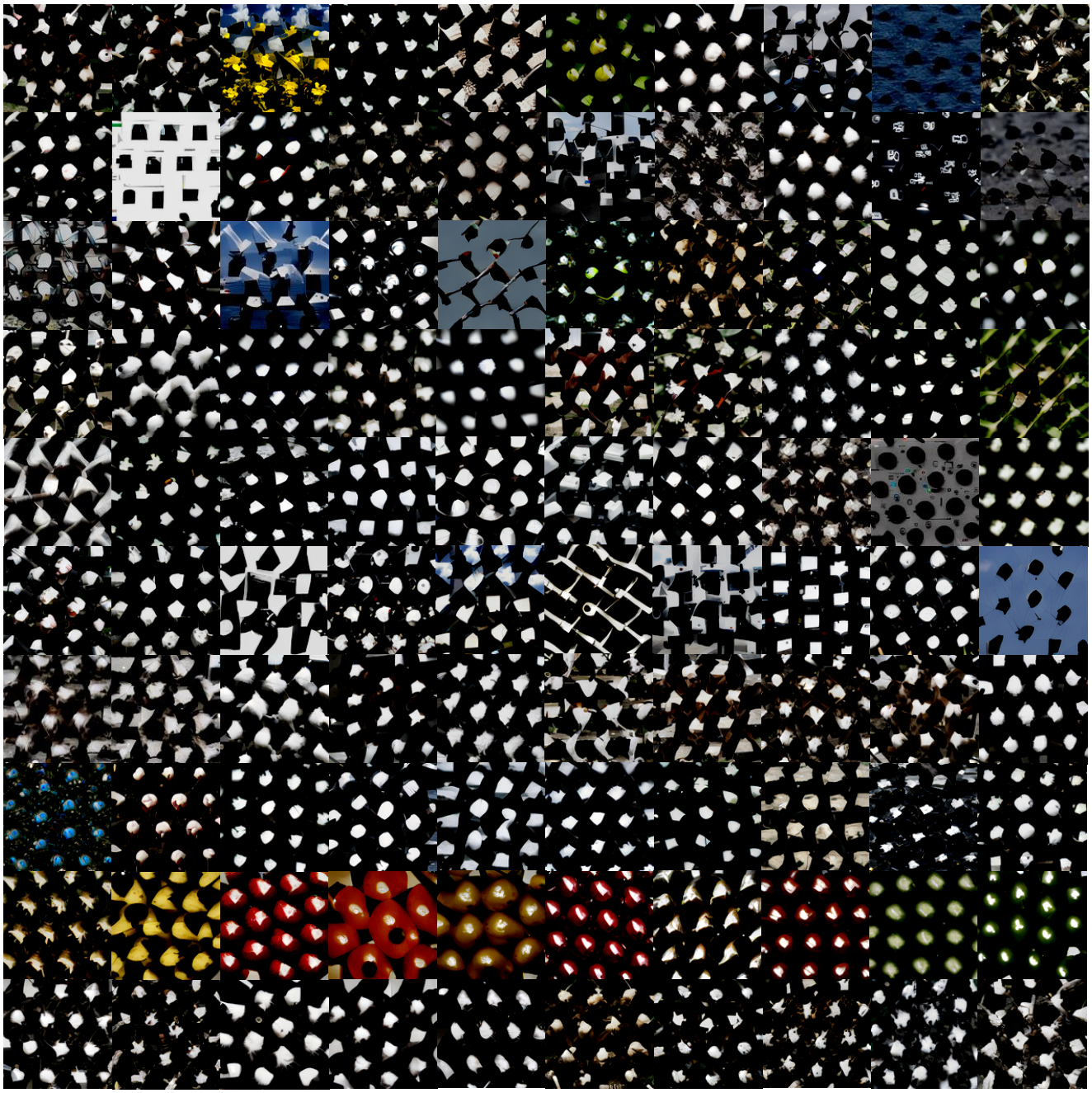


Figure 7. More visualization of the synthetic datasets using DM.

References

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

[2] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Generalizing dataset distillation via deep generative prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3739–3748, 2023. 2, 7

[3] Mingyang Chen, Jiawei Du, Bo Huang, Yi Wang, Xiaobo Zhang, and Wei Wang. Influence-guided diffusion for dataset distillation. In *The Thirteenth International Conference on Learning Representations*, 2025. 2

[4] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Dc-bench: Dataset condensation benchmark. *Advances in Neural Information Processing Systems*, 35:810–822, 2022. 7

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[7] Jianyang Gu, Saeed Vahidian, Vyacheslav Kungurtsev, Haonan Wang, Wei Jiang, Yang You, and Yiran Chen. Efficient dataset distillation via minimax diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15793–15803, 2024. 2

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7

[9] Jeremy Howard. A smaller subset of 10 easily classified classes from imagenet, and a little more french. *URL https://github. com/fastai/imagenette*, 2019. 7

[10] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022. 1

[11] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 7

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 7

[13] Songhua Liu, Kai Wang, Xingyi Yang, Jingwen Ye, and Xinchao Wang. Dataset distillation via factorization. *Advances in Neural Information Processing Systems*, 35:1100–1113, 2022. 1

[14] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020. 3

[15] DongHyeok Shin, Seungjae Shin, and Il-chul Moon. Frequency domain-based dataset distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2

[16] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7

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

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016. 7

[19] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1

[20] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 7

[21] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6514–6523, 2023. 1

[22] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 1

[23] Xinhao Zhong, Bin Chen, Hao Fang, Xulin Gu, Shu-Tao Xia, and En-Hui Yang. Going beyond feature similarity: Effective dataset distillation based on class-aware conditional mutual information. *arXiv preprint arXiv:2412.09945*, 2024. 3

[24] Xinhao Zhong, Shuoyang Sun, Xulin Gu, Zhaoyang Xu, Yaowei Wang, Jianlong Wu, and Bin Chen. Efficient dataset distillation via diffusion-driven patch selection for improved generalization. *arXiv preprint arXiv:2412.09959*, 2024. 2

[25] Chenyang Zhu, Kai Li, Yue Ma, Chunming He, and Xiu Li. Multibooth: Towards generating all your concepts in an image from text. *arXiv preprint arXiv:2404.14239*, 2024. 2

[26] Chenyang Zhu, Kai Li, Yue Ma, Longxiang Tang, Chengyu Fang, Chubin Chen, Qifeng Chen, and Xiu Li. Instantswap: Fast customized concept swapping across sharp shape differences. *arXiv preprint arXiv:2412.01197*, 2024. 2