

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

Xinhao Zhong^{1,*} Hao Fang^{3,*} Bin Chen^{1,2†} Xulin Gu¹ Meikang Qiu⁴ Shuhan Qi¹ Shu-Tao Xia^{2,3}

¹Harbin Institute of Technology, Shenzhen ²Peng Cheng Laboratory

³Tsinghua Shenzhen International Graduate School, Tsinghua University ⁴Augusta University

xh021213@gmail.com, fang-h23@mails.tsinghua.edu.cn, chenbin2021@hit.edu.cn, 210110720@stu.hit.edu.cn,
 shuhanqi@cs.hitsz.edu.cn, qiumeikang@yahoo.com, xiast@sz.tsinghua.edu.cn;

Abstract

Dataset distillation is an emerging dataset reduction method, which condenses large-scale datasets while maintaining task accuracy. Current parameterization methods achieve enhanced performance under extremely high compression ratio by optimizing determined synthetic dataset in informative feature domain. However, they limit themselves to a fixed optimization space for distillation, neglecting the diverse guidance across different informative latent spaces. To overcome this limitation, we propose a novel parameterization method dubbed Hierarchical Parameterization Distillation (H-PD), to systematically explore hierarchical feature within provided feature space (e.g., layers within pre-trained generative adversarial networks). We verify the correctness of our insights by applying the hierarchical optimization strategy on GAN-based parameterization method. In addition, we introduce a novel class-relevant feature distance metric to alleviate the computational burden associated with synthetic dataset evaluation, bridging the gap between synthetic and original datasets. Experimental results demonstrate that the proposed H-PD achieves a significant performance improvement under various settings with equivalent time consumption, and even surpasses current generative distillation using diffusion models under extreme compression ratios $IPC=1$ and $IPC=10$. Our code is available at <https://github.com/ndhg1213/H-PD>

1. Introduction

In recent years, deep learning has made significant strides in various research fields, encompassing computer vision [12, 18] and natural language processing [3, 11]. These advancements have been facilitated by utilizing larger and more intricate deep neural networks (DNNs) in conjunction with

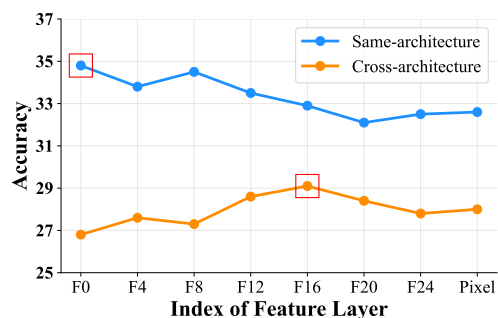


Figure 1. Performance of synthetic datasets condensed from various feature domains provided by GAN under the same settings (DSA on ImageNet-Birds).

numerous datasets tailored for diverse application fields. However, as the complexity of various learning tasks increases, neural networks have grown both deeper and wider, resulting in an exponential surge in the size of datasets required for training these models. This has presented a substantial challenge to data storage and processing efficiency [24], further exacerbating the bottleneck in deep learning due to the mismatch between the enormous data volume and limited computing resources.

Dataset distillation (DD) [41] has emerged as a promising solution to the aforementioned issues. It allows for the generation of a more compact synthetic dataset, where each data point encapsulates a higher concentration of task-specific information than its real counterparts. When trained on the synthetic dataset, the network can achieve performance comparable to its counterpart using the original dataset. By significantly reducing the size of the training data, dataset distillation offers a substantial reduction in training costs and memory consumption. Various methods have been proposed to enhance the performance of the condensed dataset.

Subsequently, synthetic dataset parameterization methods [5, 25, 36, 46] employ differentiable operations to pro-

*Equal Contribution.

†Corresponding Author.

cess synthetic images, shifting the optimization space from pixels to feature domains. These methods benefit from the efficient guidance of hidden features, thus achieving better performance. However, existing parameterization methods focus on one fixed optimization space, overlooking the informative guidance across multiple corresponding feature domains. For example, FreD [36] optimizes the synthetic dataset in the low-frequency space using discrete cosine transform (DCT), while ignoring informative guidance in the high-frequency domain. In recent years, several recent studies have exploited the rich semantic information encoded in the generators to enhance dataset distillation. With the rapid advancement achieved by generative models, one category of distillation methods utilizes diffusion models [32] to generative informative samples [17, 38]. However, under high compression rates (i.e., IPC=1/10), these methods can degrade into coreset selection methods. Another category of distillation methods employs GANs in an optimization-based manner [5, 46] to parameterize the synthetic datasets and achieve reliable results.

In contrast to the aforementioned methods, GAN-based parameterization distillation methods possess an optimization space with richer semantic information. ITGAN[46] directly optimizes the initial latent space of GAN and achieves significant performance improvements on low-resolution datasets. To fully utilize the GAN prior, GLaD [5] decomposes the GAN structure and manually selects the intermediate layer, greatly enhancing the cross-architecture performance of the synthetic dataset. However, existing method exhibit a performance decrease in the same-architecture settings when coupled with certain dataset distillation methods as suggested in Figure 1. In this manner, even though synthetic datasets are condensed from the optimally selected intermediate layer through preliminary experiments by manual picking, the diverse model architectures still lead to changes in the optimal performance. As aforementioned parameterization methods, current GAN-based approaches limit the optimization space to a specific feature domain and necessitate extensive computing time and resources to manually select the optimal feature domain for different settings. This naturally raises a question: *Does a fixed optimization space meet the demands of dynamic data distribution and model architectures during parameterization dataset distillation?*

To address this question, we propose a straightforward and efficient approach based on the parameterization prior, Hierarchical Parameterization Distillation (H-PD), which explores the significance of hierarchical features. To verify our intuition, we design a well-design framework on GAN-based parameterization method. The proposed H-PD embraces adaptive exploration across all hierarchical feature domains within GAN models. Specifically, we decompose the GAN structure, undertaking a greedy search that spans

different hierarchical feature domains. During the distillation process, we optimize these hierarchical latents within the GAN model, guided by the loss from the dataset distillation task. Throughout this optimization, we track the best hierarchical latents at the current layer, feeding them into the next layer. This iterative process continues until the optimizer traverses the hierarchical layers and reaches the pixel domain. To mitigate the time-consuming nature of performance evaluation, we introduce a class-relevant feature distance metric between the synthetic and real datasets to search for the optimal latent feature. This metric serves as a performance estimation for the synthetic datasets, encapsulating the significance of hierarchical features. Crucially, our method explores hierarchical features more comprehensively than previous approaches, which only rely on a single fixed feature domain as image priors.

Our main contributions can be summarized as follows:

- We revisited the shortcomings of existing parameterization methods, and provided a novel parametrization framework to enhance their efficiency by leveraging information across various feature domains.
- Through a well-designed framework, we effectively enhanced the performance of GAN-based parameterization methods, demonstrating the validity of our insights.
- To mitigate the computational demands associated with searching feature domains, we introduce a novel class-relevant feature distance metric, saving valuable computational time by approximating the real performance of the synthetic dataset.

2. Related Work

2.1. Dataset Distillation

Dataset distillation was initially regarded as a meta-learning problem [41]. It involves minimizing the loss function of the synthetic dataset using a model trained on the synthetic dataset. Since then, several approaches have been proposed to enhance the performance of dataset distillation. One category of methods utilizes ridge regression models to approximate the inner loop optimization [1, 29, 30, 49]. Another category of methods selects an informative space as a proxy target to address the unrolled optimization problem. DC [48], DSA [45] and DCC [23] match the weight gradients of models trained on the real and synthetic dataset, while DM [47], CAFE [40] and DataDAM [33] use feature distance between the real and synthetic dataset as matching targets. MTT [4] and TESLA [9] match the model parameter trajectories obtained from training on the real dataset and the synthetic dataset. In recent years, some studies have argued that the bi-level optimization structure required by traditional dataset distillation is redundant and computationally expensive.

Other studies suggest that the pixel domain, where im-

ages reside, is considered a high-dimensional space. Therefore, performance improvement can be achieved by parameterizing the synthetic dataset and transferring the optimization space. IDC [19] and HaBa [25] perform optimization in a low-dimensional space using differentiable operations, while GLaD [5] and ITGAN [46] utilize the feature domain provided by GANs as the optimization space, both of them employ pre-trained GANs as priors. FreD [36] employs traditional compression methods (e.g., DCT) to provide a low-frequency space as the optimization space. However, existing parameterization methods fix the optimization space thus neglecting the guidance from multi-feature domains.

The proposed H-PD introduces an innovative approach to parameterizing synthetic datasets and be verified on GAN-based parameterization methods, representing a broader and more encompassing enhancement compared to previous approaches utilizing fixed optimization space.

2.2. GAN for parameterization distillation

GAN [7] is a deep generative model trained adversarially to learn the distribution of real images. Recent studies have shown that GANs can tackle inverse problems by mapping images into their latent space [6, 13–15, 31, 42], enabling tasks like image editing [2, 39]. Incorporating image distribution information into GAN enhances the performance of dataset distillation by utilizing GAN to parameterize the synthetic dataset. GLaD employs GAN (e.g., StyleGAN-XL [34]) as a prior and significantly improves the cross-architecture performance of the synthetic dataset by selecting the feature domain of GAN’s intermediate layers as the optimization space. However, it overlooks the fact that the optimal optimization space may vary when dealing with different datasets, even with the same dataset distillation method. Additionally, it ignores the guidance offered by GAN’s earlier layers.

Compared with the current dataset distillation methods [17, 38] based on diffusion models, which could be more suitable for generative methods. GLaD can achieve significantly better performance under extreme compression ratios of IPC=1 and IPC=10 by enhancing the optimization-based methods as an efficient plugin. With the observation, the proposed H-PD further explores hierarchical feature domains of pre-trained GANs to address the limitation of GLaD, resulting in a novel parameterization method that successfully leveraging informative guidance within unfixed optimization space.

3. Method

In this section, we first present the problem definition of dataset distillation and discuss existing methods that parameterize synthetic datasets using GANs. Subsequently, we delve into the specifics of our method, aiming to improve upon previous works by exploring the feature do-

main provided by GANs. Finally, we propose an alternative evaluation scheme that assesses the synthetic dataset’s performance by measuring the layer-wise feature distance between it and the real one.

3.1. Preliminaries

Dataset distillation necessitates a real large-scale dataset $\mathcal{T} = \{(\mathbf{x}_t^i, y_t^i)\}_{i=1}^T$ and aims to create a smaller synthetic dataset $\mathcal{S} = \{(\mathbf{x}_s^i, y_s^i)\}_{i=1}^S$ ($|\mathcal{S}| \ll |\mathcal{T}|$), minimizing the performance gap between models trained on the two datasets. To achieve this, a well-designed matching objective $\phi(\cdot)$ is employed to extract feature distances in a specific informative space, representing the performance gap between the real and synthetic datasets. The optimization process involves initializing the synthetic dataset from the real dataset and iteratively updating it by minimizing the feature distance, which can be formulated below:

$$S^* = \arg \min_{\mathcal{S}} \mathcal{M}(\phi(\mathcal{S}), \phi(\mathcal{T})), \quad (1)$$

where $\mathcal{M}(\cdot, \cdot)$ denotes some matching metric, e.g., neural network gradients [48], exacted features[47], and training trajectories[4].

Building upon these findings, methods that parameterize synthetic datasets shift the optimization space from the pixel domain to the feature domain by employing differentiable operations. For instance, GAN priors-based methods [5, 46] can be formulated uniformly below:

$$\mathbf{z}^* = \arg \min_{\mathbf{z} \in \mathcal{Z}} \mathcal{M}(\phi(G_w(\mathbf{z})), \phi(\mathcal{T})), \quad (2)$$

where $\mathbf{z} \in \mathcal{Z}$ represents the latent in a specific feature domain of a pre-trained generative model $G_w(\cdot)$. Guided by GAN priors, these methods demonstrate substantial performance improvements.

3.2. Theoretical Insights of Unfixed Optimization Space and Fixed Optimization Space

Previous parameterization methods distill knowledge within a fixed optimization space, where the starting point in a given feature domain is typically randomly initialized. In contrast, our proposed H-PD framework performs progressive optimization on the latents. Figure 2 provides a detailed comparison of the optimization processes in fixed versus flexible feature spaces. Beyond the limited guidance across various feature domains, the fixed optimization space encounters a critical bottleneck: it restricts further enhancement by limiting the selection of superior synthetic datasets based on explicit or implicit criteria. In particular, the optimization within a fixed space can be viewed as a continuous process. If a temporarily optimal result, denoted as S^i , is chosen before convergence, and the corresponding optimization epoch does not mark the end of the optimization trajectory, a fundamental issue arises: *should S^i serve*

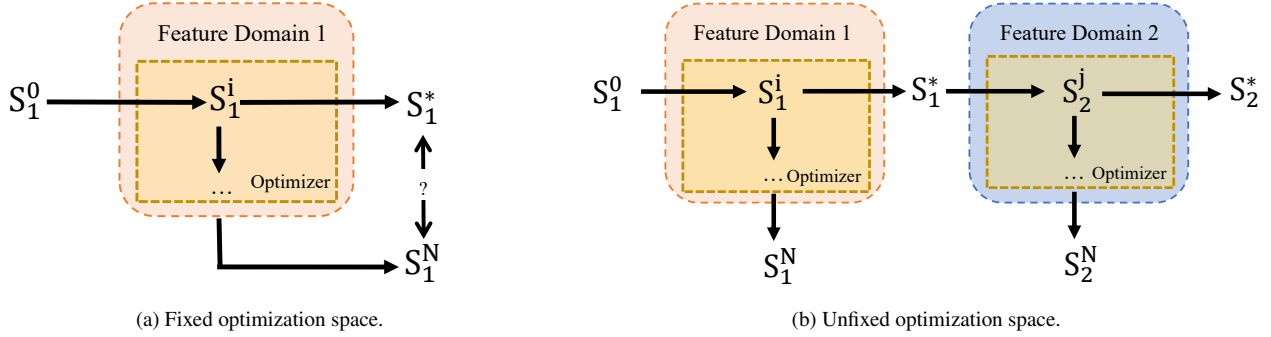


Figure 2. The comparison between fixed optimization space and unfixed optimization space. S^i is the synthetic dataset at optimization steps i , S^* is the optimal synthetic dataset selected during the optimization path, S_j is the synthetic dataset optimized in feature domain j .

as the starting point for the next phase of the optimization, or should the suboptimal result S^N be used instead? Opting for the latter can trap the optimization in a local minimum due to the absence of robust criterion-driven guidance, while choosing the former risks creating an optimization loop, effectively discarding the progress between S^i and S^N and forcing a reset.

Next, we provide the theoretical insights of its effectiveness. Specifically, we denote the random latent and optimized latent as $\mathbf{z}_{\text{rand}}$ and $\mathbf{z}_{\text{opt}}$, respectively. For an effective dataset distillation method, we assume that the distilled data distribution $P(Z_{\text{lic}})$ fits the original dataset distribution in a lossless manner. Let the coupling latent $(\mathbf{z}_{\text{rand}}, \mathbf{z}_{\text{opt}})$ be the observed values of random variables Z_{rand} and Z_{opt} respectively. Let $c(\cdot)$ be a cost function defined in optimal transport theory [26], which satisfies $\mathbb{E}[c(X - Y)] \propto 1/I(X; Y)$. Then we can obtain:

$$\begin{aligned}
 & \mathbb{E}[c(Z_{\text{lic}} - Z_{\text{opt}})] \\
 &= k/D_{\text{KL}}(P(Z_{\text{lic}}, Z_{\text{opt}}) || P(Z_{\text{lic}}) \cdot P(Z_{\text{opt}})) \\
 &= k/[H(Z_{\text{lic}}) - H(Z_{\text{lic}}|Z_{\text{opt}})] \\
 &\leq k/[H(Z_{\text{lic}}) - H(Z_{\text{lic}}|Z_{\text{rand}})] \quad (3) \\
 &= k/D_{\text{KL}}(P(Z_{\text{lic}}, Z_{\text{rand}}) || P(Z_{\text{lic}}) \cdot P(Z_{\text{rand}})) \\
 &= \mathbb{E}[c(Z_{\text{lic}} - Z_{\text{rand}})],
 \end{aligned}$$

where k denotes a constant, $D_{\text{KL}}(\cdot||\cdot)$ and $H(\cdot)$ stand for Kullback-Leibler divergence and entropy, respectively. The Inequality (3) follows since the conditional entropy $H(Z_{\text{lic}}|Z_{\text{opt}})$ is smaller than $H(Z_{\text{lic}}|Z_{\text{rand}})$.

The theoretical analysis indicates that the proposed H-PD method of selecting partially optimized latents, rather than random initialization, reduces optimization cost across different spaces. By leveraging multiple feature domains, it accelerates convergence and, with its implicit evaluation criterion, effectively avoids local optima—an issue common in fixed optimization spaces.

3.3. Progressive Optimization with Hierarchical Feature Domains

To verify our insights about the utilization of unfixed parameterization optimization space, we apply it with GAN-based parameterization method (i.e., GLaD [5]). As depicted in Algorithm 1, our approach diverges from restricting the optimization space to a specific feature domain of the GANs. Instead, we aim to explore the hierarchical layers of the GAN, striving to enhance the effective utilization of the prior information.

To sufficiently utilize the informative guidance from the hierarchical feature domains, we decompose a pre-trained GAN $G_w(\cdot)$ for hierarchical layer searching, i.e.,

$$G_w(\cdot) = G_{K-1} \circ G_{K-2} \circ \cdots \circ G_1 \circ G_0(\cdot). \quad (4)$$

For each hierarchical layer G_i provided by GAN, we repeat the following steps. Firstly, we generate images from $\mathbf{z}_i$ only using the remaining synthesizing network $G_{k-1}(\cdot) \circ G_{k-2}(\cdot) \circ \cdots \circ G_i(\cdot)$. Then, we employ the distillation method (e.g., MTT [4]) to calculate $\mathcal{L}$ based on the synthetic dataset composed of generated images and the real dataset to optimize $\mathbf{z}_i$ with an SGD optimizer, the optimization process lasts for a pre-determined and fixed N steps. After completing the optimization process for a specific layer, we implicitly evaluate the latents synthesized during the SGD optimization process and record $\mathbf{z}_i^*$ as the optimal latents for the current layer. Finally, we pass $\mathbf{z}_i^*$ into the next layer to obtain $\mathbf{z}_{i+1}^0$ as the initial latent for the next layer.

When the optimization space reaches the ultimate pixel domain, we choose the optimal latent $\mathbf{z}^*$ from the recorded latents $\mathbf{z}_i^*$ based on the real performance of the synthetic dataset $\mathcal{S}$ generated by the corresponding remaining synthesizing network $G_{k-1}(\cdot) \circ G_{k-2}(\cdot) \circ \cdots \circ G_i(\cdot)$. In this way, we fully explore the feature domain of the GAN, leveraging its rich information.

Algorithm 1 Pseudocode of H-PD with GAN-based parameterization method

Input: $G_w(\cdot)$: a pre-trained generative model; K : the number of hierarchical layers; N : distillation steps; $\mathcal{T}$: real training dataset; P_z : distribution of latent initializations; $\mathcal{L}$: distillation loss; $Acc(\cdot)$: evaluate real performance of synthetic dataset;

```

1: Initial average latent  $\mathbf{z} \sim P_z$ 
2: Dissemble  $G_w(\cdot)$  into  $G_{k-1} \circ G_{k-2} \circ \dots \circ G_0(\cdot)$ 
3:  $acc_{max} = 0$ 
4: for  $i \leftarrow 0$  to  $K - 1$  do
5:    $d_{min} = \mathcal{D}(G_{k-1} \circ \dots \circ G_i(\mathbf{z}_i^0), \mathcal{T})$ 
6:   for  $j \leftarrow 0$  to  $N - 1$  do
7:      $\mathcal{S}_i^j = G_{k-1} \circ \dots \circ G_i(\mathbf{z}_i^j)$ 
8:      $\mathcal{L} = \mathcal{M}(\phi(\mathcal{S}_i^j), \phi(\mathcal{T}))$ 
9:      $\mathbf{z}_i^{j+1} \leftarrow SGD(\mathbf{z}_i^j; \mathcal{L})$ 
10:    if  $\mathcal{D}(\mathcal{S}_i^j, \mathcal{T}) \leq d_{min}$  then
11:       $\mathbf{z}_i^* = \mathbf{z}_i^j, \mathcal{S}_i^* = \mathcal{S}_i^j$ 
12:    end if
13:  end for
14:  if  $Acc(\mathcal{S}_i^*) > acc_{max}$  then
15:     $acc_{max} = Acc(\mathcal{S}_i^*)$ 
16:     $\mathcal{S} = \mathcal{S}_i^*$ 
17:  end if
18:   $\mathbf{z}_{i+1}^0 = G_i(\mathbf{z}_i^*)$ 
19: end for

```

Output: Synthetic dataset $\mathcal{S}$

3.4. Enhancing Performance with Efficient Searching Strategy

Ensemble-Averaging Latent Initialization To mitigate the undesirable time overhead brought by existing methods [27] using clustering or GAN inversion [42], we propose an inactive searching initialization by calculating the average value of multiple noises, and passing it through the GAN’s mapping network to obtain the initial latent $\mathbf{z}_0$ with reduced bias. our method showcases simplicity without compromising effectiveness, as confirmed by sufficient experimental results.

Class-relevant Feature Distance To search for optimal latent as the optimization starting point of the subsequent feature domain, an efficient implicit evaluation metric is needed to replace the time-consuming evaluation of the synthetic dataset’s real performance. We first attempt to use the loss value as a substitute evaluation metric. However, it fails to yield desired results and, in some cases, performs even worse than not searching at all.

To utilize gradient information while maintaining diversity, we adopt the class activation map (CAM) [28] by utilizing the gradients of the corresponding class with respect

to the feature maps to localize the class-specific features. With the output logits $q = f^d(w^d; \mathbf{z})$ from the classifier w^d , the CAM is defined as the gradients of output logits q^y of class y with respect to features $\mathbf{z}$ as follows:

$$g_{\mathbf{z}} = \frac{\partial q^y}{\partial \mathbf{z}}. \quad (5)$$

To focus attention on the class-relevant region, we propose a novel class-relevant feature distance $\mathcal{D}(\mathcal{S}, \mathcal{T})$ between the real dataset $\mathcal{S}$ and the synthetic dataset $\mathcal{T}$. i.e.,

$$\mathcal{D}(\mathcal{S}, \mathcal{T}) = \left\| \frac{1}{|\mathcal{T}|} \sum_{i=1}^{|\mathcal{T}|} w^e(\mathbf{x}_t^i) \cdot \text{ReLU}(g_{\mathbf{z}}^t) - \frac{1}{|\mathcal{S}|} \sum_{j=1}^{|\mathcal{S}|} w^e(\mathbf{x}_s^j) \cdot \text{ReLU}(g_{\mathbf{z}}^s) \right\|^2, \quad (6)$$

where $w^e(\cdot)$ represents the feature extractor of a pre-trained network, and $\text{ReLU}(\cdot)$ is the rectified linear unit function.

4. Experiments

To verify the efficiency of our proposed method, we conduct experiments using code derived from the open-source GLaD¹. We utilize ImageNet-1K [10] subsets and CIFAR-10 [20] to generate high-resolution and low-resolution distilled datasets respectively, with StyleGAN-XL as the deep generative network. To ensure a fair comparison, we maintain consistency by adopting the same network architecture and employing identical hyperparameters. Our code is available at <https://github.com/ndhg1213/H-PD>.

4.1. Settings and Implementation Details

Datasets and Network Architectures In this study, we build upon previous research by utilizing CIFAR10 and Tiny-ImageNet [22] as low-resolution dataset, For high-resolution experiments, we choose ImageNet-1K and ten subsets from it. These subsets, each consisting of ten categories, are divided into the training and validation sets. The detailed categories combination can be found in Appendix.

For the surrogate model for dataset distillation, we choose ConvNet-5 [16] as the backbone network for DM, DSA and TESLA. For ImageNet-1K and Tiny-Imagenet, we conduct experiments on SR²L [43] and adopt ResNet-18 [18] as backbone. To evaluate the performance of the synthetic dataset, we employ various models, including ConveNet, AlexNet [21], VGG-11 [37], ResNet-18, and a Vision Transformer model [12] from the DC-BENCH [8] resource. It is important to note that all of these evaluation models are versions specifically tailored for corresponding resolution datasets.

¹<https://georgecazenavette.github.io/glad>

Alg.	Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
TESLA	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
	GLaD	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
	H-PD	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	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
	GLaD	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
	H-PD	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	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
	GLaD	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
	H-PD	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 1. Synthetic dataset same-architecture performance (%) on ImageNet-Subset (128×128) under IPC=1. "Pixel" refers to not deploying GAN.

Alg.	Method	Tiny-ImageNet		ImageNet-1K	
		IPC-1	IPC-10	IPC-1	IPC-10
SRe ² L	Pixel	2.6±0.1	16.1±0.2	0.1±0.1	21.3±0.6
	GLaD	3.1±0.3	15.7±0.2	1.2±0.1	21.9±0.8
	H-PD	4.5±1.0	18.3±0.5	2.6±0.2	23.5±0.4

Table 2. Synthetic dataset performance comparison with SRe²L on Tiny-ImageNet (64x64) and ImageNet-1K (224x224) evaluated by ResNet-18.

4.2. Performance Improvements

The performance comparison of our method with previous works GLaD is shown in Table 1 and Table 2. We report the same-architecture performance of the synthetic dataset. Since GLaD did not conduct experiments on SRe²L, we chose to adopt the same layers settings (i.e., 12) as TESLA to ensure fairness. compared to optimizing only in a fixed feature space, it can be observed that our method achieves consistent and significant improvements with all the optimization-based methods. This indicates that our method successfully leverages the guidance information provided by all feature domains. The visualized images of synthetic datasets are depicted in Figure 3.

Alg.	Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E
DSA	Pixel	52.3±0.7	45.1±8.3	40.1±7.6	36.1±0.4	38.1±0.4
	GLaD	53.1±1.4	50.1±0.6	48.9±1.1	38.9±1.0	38.4±0.7
	H-PD	54.1±1.2	52.0±1.1	49.5±0.8	39.8±0.7	40.1±0.7
DM	Pixel	52.6±0.4	50.6±0.5	47.5±0.7	35.4±0.4	36.0±0.5
	GLaD	52.8±1.0	51.3±0.6	49.7±0.4	36.4±0.4	38.6±0.7
	H-PD	55.1±0.5	54.2±0.5	50.8±0.4	37.6±0.6	39.9±0.7

Table 3. Synthetic dataset cross-architecture performance (%) on ImageNet-Subset under IPC=10.

4.3. More Comparisons with GLaD

To align the proposed H-PD with GLaD on higher IPC, i.e. only IPC = 10 under DSA and DM is reported from original paper, our current confirmatory trials achieves a performance improvement of 1% to 3% compared to GLaD with

IPC = 10 under DSA and DM as shown in Table 3, respectively, which demonstrates the effectiveness of H-PD.

Alg.	Method	ConvNet	AlexNet	ResNet-18	VGG-11	ViT
TESLA	Pixel	46.3±0.8	26.8±0.6	23.4±1.3	24.9±0.8	21.2±0.4
	GLaD	35.5±0.6	27.9±0.6	30.2±0.6	31.3±0.7	22.7±0.4
	H-PD	37.2±0.4	28.5±0.3	31.4±0.4	32.2±0.2	24.1±0.4
DSA	Pixel	28.3±0.3	25.9±0.2	27.3±0.5	28.0±0.5	22.9±0.3
	GLaD	29.2±0.8	26.0±0.7	27.6±0.6	28.2±0.6	23.4±0.2
	H-PD	30.2±0.5	26.6±0.4	28.2±0.4	28.0±0.6	24.4±0.5
DM	Pixel	26.0±0.6	22.9±0.2	22.2±0.7	23.8±0.5	21.3±0.5
	GLaD	27.1±0.7	25.1±0.5	22.5±0.7	24.8±0.8	23.0±0.1
	H-PD	27.6±0.7	27.5±0.6	25.6±0.6	25.4±0.8	23.6±0.5

Table 4. Performance(%) across different models on CIFAR-10.

The cross-architecture performance on CIFAR-10 [20] is shown in Table 4. The results demonstrate that using a shallower StyleGAN-XL structure on the lower-resolution dataset CIFAR-10, H-PD still improves the performance of synthetic datasets distilled by different distillation methods. Please note that the released code of GLaD does not include the data augmentation and hyperparameter settings used by TESLA on CIFAR10, which leads to a poor performance on ConvNet.

To present a more comprehensive comparison, we assess the performance of the synthetic dataset across different architectures as shown in Table 5. The cross-architecture accuracy is calculated by averaging the performance of the remaining four models, excluding the backbone model. The results of previous studies are acquired directly from the original papers.

Alg.	Method	ImNet-A	ImNet-B	ImNet-C	ImNet-D	ImNet-E
TESLA	Pixel	33.4±1.5	34.0±3.4	31.4±3.4	27.7±2.7	24.9±1.8
	GLaD	39.9±1.2	39.4±1.3	34.9±1.1	30.4±1.5	29.0±1.1
	H-PD	40.2±0.3	39.8±0.8	35.8±0.7	31.2±1.0	29.5±0.7
DSA	Pixel	38.7±4.2	38.7±1.0	33.3±1.9	26.4±1.1	27.4±0.9
	GLaD	41.8±1.7	42.1±1.2	35.8±1.4	28.0±0.8	29.3±1.3
	H-PD	42.4±1.2	42.6±1.1	36.1±1.1	28.7±1.1	29.6±0.9
DM	Pixel	27.2±1.2	24.4±1.1	23.0±1.4	18.4±0.7	17.7±0.9
	GLaD	31.6±1.4	31.3±3.9	26.9±1.2	21.5±1.0	20.4±0.8
	H-PD	34.9±2.1	33.8±2.0	27.8±1.7	23.6±1.5	22.5±1.3

Table 5. Synthetic dataset cross-architecture performance (%) on ImageNet-Subset.

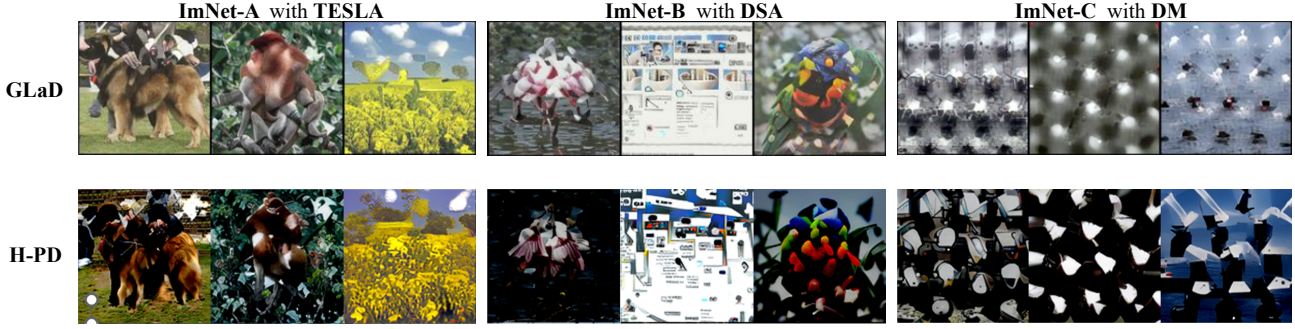


Figure 3. Visualization comparison of the synthetic datasets with different distillation methods.

4.4. Comparison with Diffusion Model Based Methods

Other generative distillation methods consider diffusion as an image generator rather than parameterization methods. Minimax [17] and D⁴M [38] fine-tune the diffusion model and corresponding latent respectively to generate entirely new samples. However, these methods only achieve random sampling outcomes under a high compression ratio (e.g., IPC=1/10). Contrary to these diffusion-based methods, our method in conjunction with GLaD belongs to an optimization-based method, which can effectively accomplish distillation in a more efficient way.

For a fair comparison, we adhere to GLaD’s experimental setting (e.g., a 128x128 resolution image corresponds to a one-hot label), disregarding training time matching [35] and mixup [44] strategy. As shown in Table 6, our method demonstrates significant advantages at a mechanism compression ratio of IPC=1 without requiring fine-tuning of the pre-trained model.

Method	ImNette	ImWoof	ImNet-Birds	ImNet-Fruits	ImNet-Cats
Minimax	22.8±0.5	17.8±0.3	23.2±0.7	17.5±0.5	19.8±0.3
D ⁴ M	15.2±0.5	17.4±0.6	18.2±0.4	17.6±0.6	23.4±0.7
H-PD	45.4±1.1	28.3±0.2	39.7±0.8	25.6±0.7	29.6±1.0

Table 6. Comparison with generative dataset distillation methods on ImageNet-Subsets under IPC=1.

4.5. Evaluation Protocol

The approach to assessing the performance of a synthetic dataset is as follows: firstly, a set of models is trained using the synthetic dataset. Once the training is complete, the trained models are validated using the corresponding validation set from the real dataset. For a specific model architecture, this process is repeated five times, and the average performance is calculated based on these repetitions. In previous studies, the evaluation method involved continuously optimizing the entire distillation process for 1000 epochs, with sampling the synthetic dataset every 100 epochs. The

Metric	Method	TESLA	DSA	DM
Time	GLaD	75	69	64
	H-PD	70	73	15
performance	GLaD	45.0±0.9	41.3±1.2	37.4±1.6
	H-PD	50.3±0.6	43.2±0.6	39.1±1.2

Table 7. Time complexity (min) and performance (%) averaged on ImageNet-[A, B, C, D, E].

best performance among all sampled examples would then be selected. To ensure a fair comparison, we decompose StyleGAN-XL into $G_{11} \circ \dots \circ G_1 \circ G_0(\cdot)$ and apply the same optimizer and learning rate for each layer, optimizing for 100 steps. This ensures that the total number of optimization epochs remains consistent, thereby preventing performance improvements solely due to a higher number of optimization epochs. The comparison of time complexities and corresponding performance is shown in Table 7. Additionally, we adopt the same setup in our evaluation and sample the synthetic dataset after optimizing for 100 epochs in all of these different feature domains (i.e., $\mathbf{z}_i^{K-1}$). This approach prevents performance improvements obtained from implicitly selecting a dataset with a higher quality.

Component	ImNet-B	ImNet-C	ImWoof	ImNet-Fruits
GLaD-TESLA	51.9±1.3	44.9±0.4	23.4±1.1	23.1±0.4
+ Average Initialization	53.5±0.7	46.1±0.9	24.8±1.1	22.7±1.2↓
+ Hierarchical Layers	56.2±0.7	48.1±0.9	28.1±1.0	24.1±0.5
+ Distance Metric	57.4±0.3	49.5±0.6	28.3±0.2	25.6±0.7
GLaD-DSA	49.2±1.1	42.0±0.6	22.3±1.1	20.7±1.1
+ Average Initialization	48.9±0.8↓	40.6±0.7↓	22.8±1.4	21.3±1.0
+ Hierarchical Layers	50.1±1.1	43.1±1.4	23.6±0.8	21.9±0.8
+ Distance Metric	50.7±0.9	43.9±0.7	24.0±0.8	22.4±1.1
GLaD-DM	42.9±1.9	39.4±1.7	21.2±1.5	21.8±1.8
+ Average Initialization	43.2±1.6	39.9±1.7	21.1±1.9↓	22.3±1.3
+ Hierarchical Layers	44.2±2.1	41.0±1.2	23.1±0.9	24.1±1.4
+ Distance Metric	44.7±1.3	41.1±1.3	23.9±1.9	24.4±2.1

Table 8. Ablation study of each component with different distillation method across various ImageNet-Subset.

4.6. Ablation Studies

Effectiveness of Each Component As Table 8 shows, the two major components of our method, i.e., hierarchical feature domains and class-relevant feature distance both improve the performance across various ImageNet-Subsets with all distillation methods, especially on TESLA. Optimizing in an unfixed feature space can bring significant gains, and on this basis, using class-relevant feature distance for implicit evaluation can yield a slight additional improvement. Please note that using class-relevant feature distance is infeasible without unfixed optimization spaces. Despite Initialization with averaged noise improves the performance of TESLA and DM to some degree, it cannot achieve stable improvement in the performance of the DSA method. We attribute this discrepancy to the inherent inclination towards noise and edge samples in DSA.

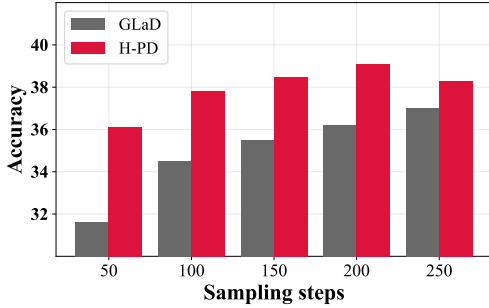


Figure 4. The comparison of performance(%) at the same optimization epoch.

Steps	TESLA	DSA	DM
20	46.9±1.2	39.8±1.1	39.1±1.2
50	47.2±0.8	41.6±0.8	37.0±1.7
100	50.3±0.6	43.2±0.6	36.5±1.4
200	50.5±0.4	43.0±0.6	35.8±1.1

Table 9. Ablation results of optimization steps per optimization space averaged on ImageNet-[A, B, C, D, E].

Optimization Steps We perform 100 optimization steps in each layer to align with the sampling method in previous works. We conduct more experiments across different optimization steps to explore the correct optimization steps. By observing the results shown in Table 9, we find that optimization steps beyond 100 do not yield significant performance improvements for TESLA and DSA methods. Meanwhile, optimization steps below 100 result in performance degradation. Considering the trade-off between effects and costs, we set the steps at 100 for both TESLA and DSA. For DM, however, optimal performance is attained at the least number of steps per layer, i.e., 20 steps. We compare the performance of GLaD and H-PD at the same epoch as

shown in Figure 4, our method outperforms and converges faster, demonstrating the superiority of the approach in utilizing the hierarchical features.

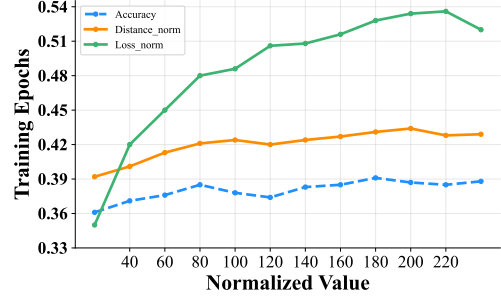


Figure 5. The relationship between searching basis and performance. Note that higher loss-norm values indicate lower loss values and the same applies to feature distances.

Searching Basis To avoid the time-consuming task of directly evaluating the synthetic dataset, we opt for class-relevant feature distance for implicit searching. Specifically, we evaluate the synthetic dataset at specific epochs during the optimization process and subsequently record its ground-truth performance, loss value, and feature distance. Figure 5 demonstrates the recorded values during the optimization process. We normalize the loss values and feature distance to range $[0, 1]$ for clear clarity and comparison. Our observation indicates that compared with the loss value, the feature distance consistently exhibits a stronger negative correlation with the performance.

5. Conclusion

In this paper, we present a novel approach to dataset distillation by exploring hierarchical parameterization space and successfully enhance the GAN-based parameterization method. Our method transforms the optimization space from a specific GAN feature domain to a broader feature space, addressing challenges seen in previous GAN-based parameterization methods. An advantage is that our approach provides a new insight for parameterization methods. Additionally, we anticipate that further improvements can be achieved through detailed optimization steps and optimization space combinations. The proposed H-PD re-explores and showcases the potential of hierarchical features in parameterization distillation for enhancing the performance under extreme compression ratios, contributing to an advanced dataset distillation approach.

Acknowledgement. This work is supported in part by the National Natural Science Foundation of China under grant 62171248, 62301189, Peng Cheng Laboratory (PCL2023A08), and Shenzhen Science and Technology Program under Grant KJZD20240903103702004, JCYJ20220818101012025, RCBS20221008093124061, GXWD20220811172936001.

References

- [1] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020. 2
- [2] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093*, 2016. 3
- [3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 1
- [4] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4750–4759, 2022. 2, 3, 4
- [5] 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. 1, 2, 3, 4
- [6] Lucy Chai, Jonas Wulff, and Phillip Isola. Using latent space regression to analyze and leverage compositionality in gans. *arXiv preprint arXiv:2103.10426*, 2021. 3
- [7] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018. 3
- [8] 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. 5
- [9] Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. Scaling up dataset distillation to imagenet-1k with constant memory. In *International Conference on Machine Learning*, pages 6565–6590. PMLR, 2023. 2
- [10] 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. 5
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1
- [12] 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. 1, 5
- [13] Hao Fang, Bin Chen, Xuan Wang, Zhi Wang, and Shu-Tao Xia. Gifd: A generative gradient inversion method with feature domain optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4967–4976, 2023. 3
- [14] Hao Fang, Jiawei Kong, Wenbo Yu, Bin Chen, Jiawei Li, Shutao Xia, and Ke Xu. One perturbation is enough: On generating universal adversarial perturbations against vision-language pre-training models. *arXiv preprint arXiv:2406.05491*, 2024.
- [15] Hao Fang, Yixiang Qiu, Hongyao Yu, Wenbo Yu, Jiawei Kong, Baoli Chong, Bin Chen, Xuan Wang, and Shu-Tao Xia. Privacy leakage on dnns: A survey of model inversion attacks and defenses. *arXiv preprint arXiv:2402.04013*, 2024. 3
- [16] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4367–4375, 2018. 5
- [17] 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, 3, 7
- [18] 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. 1, 5
- [19] 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. 3
- [20] A Krizhevsky. Learning multiple layers of features from tiny images. *Master's thesis, University of Tront*, 2009. 5, 6
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012. 5
- [22] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 5
- [23] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoo Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR, 2022. 2
- [24] Shiye Lei and Dacheng Tao. A comprehensive survey to dataset distillation. *arXiv preprint arXiv:2301.05603*, 2023. 1
- [25] 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, 3
- [26] Yanbin Liu, Makoto Yamada, Yao-Hung Hubert Tsai, Tam Le, Ruslan Salakhutdinov, and Yi Yang. Lsmi-sinkhorn: Semi-supervised mutual information estimation with optimal transport. In *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part I 21*, pages 655–670. Springer, 2021. 4
- [27] Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. Dream: Efficient dataset distillation by representative matching. *arXiv preprint arXiv:2302.14416*, 2023. 5

- [28] 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. 5
- [29] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. 2
- [30] Timothy Nguyen, Roman Novak, Lechao Xiao, and Jaehoon Lee. Dataset distillation with infinitely wide convolutional networks. *Advances in Neural Information Processing Systems*, 34:5186–5198, 2021. 2
- [31] Yixiang Qiu, Hao Fang, Hongyao Yu, Bin Chen, MeiKang Qiu, and Shu-Tao Xia. A closer look at gan priors: Exploiting intermediate features for enhanced model inversion attacks. In *European Conference on Computer Vision*, pages 109–126. Springer, 2024. 3
- [32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2
- [33] Ahmad Sajedi, Samir Khaki, Ehsan Amjadian, Lucy Z Liu, Yuri A Lawryshyn, and Konstantinos N Plataniotis. Datadam: Efficient dataset distillation with attention matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17097–17107, 2023. 2
- [34] Axel Sauer, Katja Schwarz, and Andreas Geiger. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH 2022 conference proceedings*, pages 1–10, 2022. 3
- [35] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European conference on computer vision*, pages 673–690. Springer, 2022. 7
- [36] DongHyeok Shin, Seungjae Shin, and Il-chul Moon. Frequency domain-based dataset distillation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 1, 2, 3
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5
- [38] Duo Su, Junjie Hou, Weizhi Gao, Yingjie Tian, and Bowen Tang. D⁴: Dataset distillation via disentangled diffusion model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5809–5818, 2024. 2, 3, 7
- [39] Ayush Tewari, Mohamed Elgharib, Florian Bernard, Hans-Peter Seidel, Patrick Pérez, Michael Zollhöfer, and Christian Theobalt. Pie: Portrait image embedding for semantic control. *ACM Transactions on Graphics (TOG)*, 39(6):1–14, 2020. 3
- [40] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12196–12205, 2022. 2
- [41] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2
- [42] Weihao Xia, Yulun Zhang, Yujiu Yang, Jing-Hao Xue, Bolei Zhou, and Ming-Hsuan Yang. Gan inversion: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3121–3138, 2022. 3, 5
- [43] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *Advances in Neural Information Processing Systems*, 36, 2024. 5
- [44] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 7
- [45] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 2
- [46] Bo Zhao and Hakan Bilen. Synthesizing informative training samples with gan. *arXiv preprint arXiv:2204.07513*, 2022. 1, 2, 3
- [47] 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. 2, 3
- [48] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 2, 3
- [49] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 2