

D⁴M: Dataset Distillation via Disentangled Diffusion Model

Duo Su^{1,5,6,†} Junjie Hou^{2,5,6,†} Weizhi Gao³ Yingjie Tian^{4,5,6,7,*} Bowen Tang⁸

¹School of Computer Science and Technology, UCAS ²Sino-Danish College, UCAS

³Department of Computer Science, NCSU ⁴School of Economics and Management, UCAS

⁵Research Center on Fictitious Economy and Data Science, CAS

⁶Key Laboratory of Big Data Mining and Knowledge Management, CAS

⁷MOE Social Science Laboratory of Digital Economic Forecasts and Policy Simulation, UCAS

⁸Institute of Computing Technology, CAS

<https://junjie31.github.io/D4M/>

Abstract

Dataset distillation offers a lightweight synthetic dataset for fast network training with promising test accuracy. To imitate the performance of the original dataset, most approaches employ bi-level optimization and the distillation space relies on the matching architecture. Nevertheless, these approaches either suffer significant computational costs on large-scale datasets or experience performance decline on cross-architectures. We advocate for designing an economical dataset distillation framework that is independent of the matching architectures. With empirical observations, we argue that constraining the consistency of the real and synthetic image spaces will enhance the cross-architecture generalization. Motivated by this, we introduce Dataset Distillation via Disentangled Diffusion Model (D⁴M), an efficient framework for dataset distillation. Compared to architecture-dependent methods, D⁴M employs latent diffusion model to guarantee consistency and incorporates label information into category prototypes. The distilled datasets are versatile, eliminating the need for repeated generation of distinct datasets for various architectures. Through comprehensive experiments, D⁴M demonstrates superior performance and robust generalization, surpassing the SOTA methods across most aspects.

1. Introduction

The rapid growth in machine learning, resulting in large models and vast datasets, poses a challenge to researchers due to the escalating computational and storage demands. Can the 'Divide-and-Conquer' algorithm [1] mitigate this

† Equal contribution. * Corresponding author.

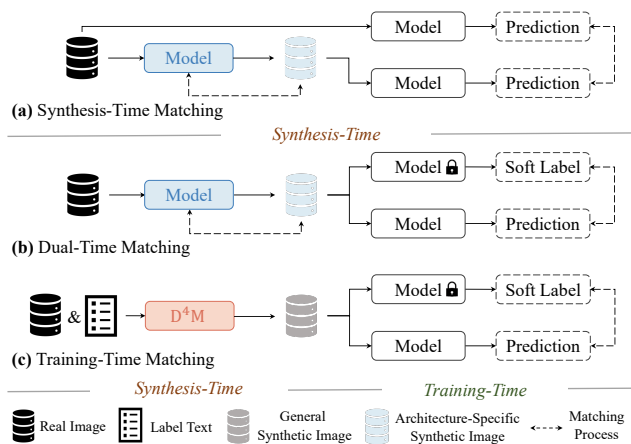


Figure 1. Comparison of various matching strategies in dataset distillation. (a) The bi-level optimization implements data matching at synthesis time. (b) Dual-Time Matching strategy decouples the bi-level optimization process into synthesis time and training time to save computational overhead. (c) D⁴M utilizes multi-modal features (image and texts) to synthesize high-quality images. D⁴M does not require matching process at Synthesis-Time.

challenge? From the perspective of dataset, recent research extends the coreset selection [3, 7, 39] to distillation techniques aimed at reducing dataset scales. Dataset Distillation (DD) aims to synthesize a small dataset \mathcal{S} from the original large-scale dataset \mathcal{T} , where $|\mathcal{S}| \ll |\mathcal{T}|$. The information in \mathcal{T} is condensed into a small dataset through DD. Initially, the DD framework uses the bi-level optimization to generate datasets where the inner loop updates the network used for testing the classification performance and the outer loop synthesizes images according to matching strategies, such as gradient [24, 51, 53], distribution [40, 52] or trajectory [4, 8].

Unfortunately, the existing solutions of DD mainly focus on small and simple datasets, such as CIFAR [21] and MNIST [23, 44]. When it comes to large-scale and high-resolution datasets such as ImageNet [9], there exists unaffordable computational requirements and reduced performance. Another challenge in DD is the cross-architecture generalization. Previous methods conduct data matching within a fixed discriminative architecture, which makes the output space biased from the original image space. As demonstrated in Fig. 2, this kind of dataset may be insightful for the networks but suffers from the lack of semantic information for humankind. Furthermore, the dataset has to be distilled from scratch again and again to adapt to the emerging network architectures. Obviously, these limitations constrain the scientific value and practical utility of the current solutions. In this paper, we argue that an ideal DD method should meet the following properties.



Guess What These Are?

Figure 2. Visualizations of previous DD methods. Synthesis-Time Matching sacrifices part of the visual semantic expression in order to imitate the performance of the original dataset.

1) *The synthesis process should not depend on a specific network architecture.* Typically, a fixed architecture is required for data matching, which leads to low cross-architecture generalization performance because the output space is constrained by the architecture. This problem arises once the matching process occurs in the synthesis time as shown in Fig. 1(a) and (b). Some work leverages a model pool instead of an individual matching model to alleviate this issue but makes the network hard to optimize [41, 54]. When the distillation process is architecture-free, there is no need to distill datasets for different architectures repeatedly. In addition, constraining the consistency of input and output spaces will make the distilled images more realistic. GlAD [5] seems to be a solution where the images are synthesized via Generative Adversarial Networks. However, the synthetic images are still matched by the inner loop.

2) *The method is capable of distilling datasets of various sizes and resolutions with limited computational resources.* As illustrated in Fig. 1(a), most DD solutions use bi-level optimization during synthesis time. While the large-scale datasets are unable to perform a number of unrolled iterations on such a nested loop system. Some works attempt to distill the ImageNet-1K but yield low testing accuracy [4, 8]. A more effective method is depicted in Fig. 1(b): the bi-level optimization is decoupled into synthesis time and training time [48]. However, the Dual-Time Matching

(DTM) strategy leads to information loss at each stage, posing challenges for distillation on small datasets instead.

Inspired by these insights, we propose the **Dataset Distillation via Disentangled Diffusion Model (D⁴M)**, an efficient approach designed for DD across varying sizes and resolutions as depicted in Fig. 1(c). In D⁴M, the Synthesis-Time Matching (STM) is superseded by Training-Time Matching (TTM) which facilitates the fast distillation of large-scale datasets with constrained computational resources. Furthermore, D⁴M alleviates the architectural dependency and improves the cross-architecture generalization performance of the distilled dataset. As the generative model, Diffusion Models ensure the consistency between input and output spaces, and its synthesis process does not rely on any specific matching architecture. To mitigate the information loss due to insufficient data matching, the conditioning mechanism in Latent Diffusion Model (LDM) consistently infuses the semantic information of labels into the synthetic data during the denoising process. The synthesis process of D⁴M solely depends on the prototypes extracted from the original data, with synthesis speed scaling linearly with the size of datasets. Moreover, the synthetic images exhibit realism at a high resolution of 512×512 . Our pivotal contributions are summarized as follows:

- To the best of our knowledge, this is the first work that overcomes the pronounced dependency on specific architectures inherent in traditional DD frameworks. We introduce the TTM strategy, which paves the way for the generation of a curated and versatile distilled dataset.
- We propose D⁴M that integrates the diffusion model into DD task for the first time. By leveraging label texts and the learned prototypes, we construct a multi-modal DD model that simultaneously enhances distillation efficiency and model performance.
- The method realizes the attainment of resolutions up to 512×512 that exhibit high-fidelity and robust adaptability in the realm of DD. This improvement is evidenced across a spectrum of datasets, extending from the ImageNet-1K to CIFAR-10/100.
- We conduct extensive experiments and ablation studies. The results outperform the SOTA in most cases, substantiating the superior performance, computational efficiency, and robustness of our method.

2. Related Work

2.1. Dataset Distillation

The existing DD approaches are taxonomized into meta-learning matching and data matching frameworks [13, 25, 34, 49]. The meta-learning matching aims to optimize the meta-test loss on real dataset for the model meta-trained by the distilled dataset. The gradients are back-propagated to supervise the DD directly [10, 27, 31, 32, 42, 54].

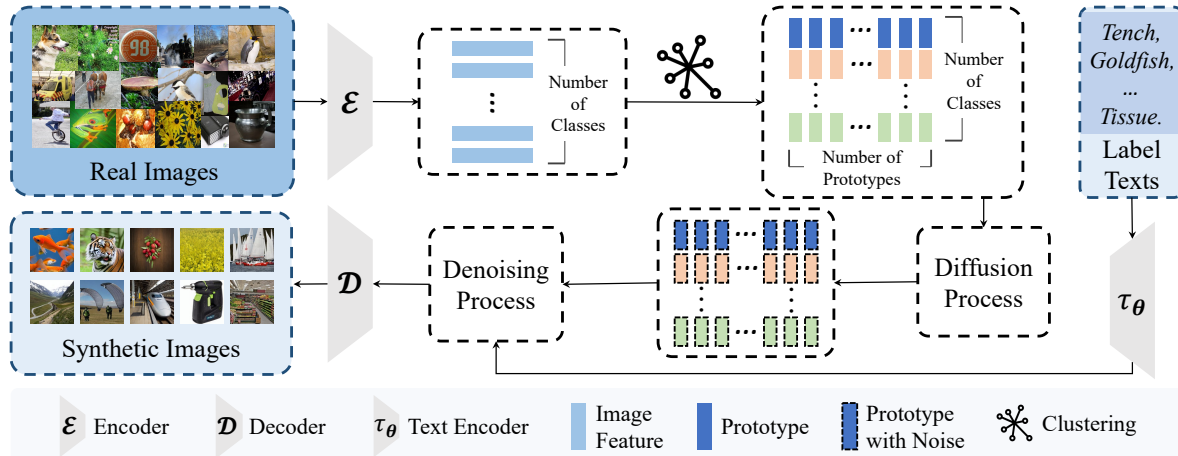


Figure 3. **Pipeline of Dataset Distillation via Disentangled Diffusion Model (D⁴M)**. Rather than using the embedded features directly, D⁴M disentangles feature extraction from image generation in diffusion models through prototype learning.

Unlike optimizing the performance on the DD explicitly, data matching encourages the consistency between the same network architecture trained by distilled and real dataset. Matching the gradients generated by the networks is a reliable surrogate task [18, 24, 51, 53]. Matching Training Trajectory (MTT) [4, 11] is then proposed to solve the issue that errors are accumulated during validation in gradient matching. TESLA [8] reduced the complexity of gradients calculating with constant memory, allowing DD to be achieved in ImageNet for the first time. Besides, distribution matching optimizes the distance between the two distributions, such as MMD [52] and CAFE [40].

The aforementioned methods only implement various matching strategies at synthesis time. SRe²L [48] argues that decoupling the bi-level optimization into Squeeze, Recover, and Relabel leads to a good performance on large-scale datasets. Inspired by this, we summarize previous works into STM and DTM. D⁴M implements the TTM with the help of soft labels, which is considered a feature distribution matching approach.

2.2. Diffusion Models

The Diffusion Model has demonstrated remarkable capabilities within the generative models. Given samples x observed from a target distribution, the goal of generative models is approximating the true distribution $P(x)$, enabling the generation of novel samples from it. Denoising Diffusion Probabilistic Models (DDPM) [16] aims to learn a reverse process of a fixed Markov Chain for generating images. However, DDPM is expensive to optimize and evaluate in the original pixel space.

Latent Diffusion Model (LDM) [33], a recent state-of-the-art diffusion model, addresses this by abstracting high-frequency, imperceptible details into a compact latent space,

thereby streamlining both training and inference. LDM has been applied in image editing [38, 43], video processing [2, 12], audio generation [17, 36] and 3D model reconstruction [6, 19, 20, 29]. Notably, the proficiency of LDM in abstracting and generating images within the latent space exactly resonates with the foundational tenets of DD.

3. Method

3.1. Preliminaries on Diffusion Models

A pivotal step in DD is the generation of the distilled images. Distinct from the data-matching approaches, our method harnesses the prior knowledge embedded in the pre-trained generative models, offering a high-quality initialization for TTM. Recently, diffusion models have emerged as SOTA in generative models [28, 46]. As aforementioned, the synthesis process of the diffusion model does not rely on any specific matching architecture, ensuring the consistency between input and output spaces. For a sequence of denoising autoencoders ϵ_θ , the training objective of Denoising Diffusion Probabilistic Model (DDPM) [16] is defined as

$$L_{DM} = \mathbb{E}_{x, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(x_t, t)\|_2^2 \right], \quad (1)$$

with the timestamp t uniformly sampled from $\{1, \dots, T\}$. Although the DDPM does not cater to our goal of synthesizing images within the condensed features, we turn our attention to the LDM [33].

LDM effectively compresses the working space from the original pixel space x to a more compact latent space z . Such a transition is close to our intent of encapsulating images into condensed features. LDM constructs an optimized low-dimensional latent space by training a perceptual compression model composed of the encoder (\mathcal{E}) and decoder (\mathcal{D}). This latent space effectively abstracts high-frequency

imperceptible details than pixel space [33]. In this case, the objective function with text encoder τ_θ is redefined as

$$L_{LDM} = \mathbb{E}_{\mathcal{E}(x), y, \epsilon \sim \mathcal{N}(0,1), t} \left[\|\epsilon - \epsilon_\theta(z_t, t, \tau_\theta(y))\|_2^2 \right]. \quad (2)$$

3.2. Disentangled Diffusion Model

The existing diffusion methods are capable of generating high-quality images directly from the given images and prompts. However, it is imperative for the DD model to aggregate the given images into a few condensed features before synthesis. The images in the original dataset encapsulate a spectrum of information from low-level texture patterns to high-level semantic information, along with potential redundancies. Since the diffusion models do not have the capability of aggregating this information among images, it is necessary to extract the salient feature representative of each category before employing the generative model. Consequently, it is essential to disentangle the diffusion models.

Employing prototypes in standard classification tasks offers the benefit of addressing the open-world recognition challenge, thereby enhancing the robustness of models [26, 45, 50]. Therefore, initializing the input of the diffusion model with prototypes not only reduces data redundancy but also elevates the quality of the distilled dataset. As illustrated in Fig. 3, we leverage the pre-trained autoencoder \mathcal{E} inherent in the LDM to extract feature representations from original images. Subsequently, we perform a clustering algorithm to calculate the cluster centers as prototypes for each category. Given the considerable size of the original dataset, we adopt the Mini-Batch k -Means [35] to mitigate the memory overhead of large-scale clustering. This approach iteratively optimizes a mini-batch of samples in each step, accelerating the clustering process with a minimal compromise in accuracy.

Specifically, the clustering algorithm consists of two primary steps: assignment z

$$z^c \leftarrow z \quad (3)$$

$$\text{s.t. } \arg \min_c \|z - z^c\|^2, c = 1, \dots, C \quad (4)$$

and update z^c

$$z^c \leftarrow (1 - \eta)z^c + \eta z. \quad (5)$$

Here z is the latent variable generated by \mathcal{E} , and z^c represents the cluster centers (prototypes), C is the number of cluster centers. The learning rate η is often calculated by $\frac{1}{|z^c|}$. Ultimately, we employ the prototypes $\tilde{Z} = \{z_l^c | c = 1, \dots, C, l = 1, \dots, L\}$ from all categories as input to the diffusion process for image synthesis.

Algorithm 1 Dataset Distillation via Disentangled Diffusion Model ($\mathbf{D}^4\mathbf{M}$)

Input: $(\mathcal{T}, \mathcal{L})$: Real images and their label texts.

Input: \mathcal{E} : Pre-trained encoder.

Input: \mathcal{D} : Pre-trained decoder.

Input: τ_θ : Pre-trained text encoder.

Input: \mathcal{U}_t : Pre-trained time-conditional U-Net.

Input: C : Number of prototypes.

```

1:  $Z = \mathcal{E}(\mathcal{T}) \sim P_z$  ▷ Compressed latent space
2: for each  $L \in \mathcal{L}$  do
3:   for mini-batch  $z \in L$  do
4:      $z^c \sim P_z, c = 1, \dots, C$  ▷ Initialize cluster centers
5:      $z^c \leftarrow z, \text{s.t. } \arg \min_c \|z - z^c\|^2$  ▷ Assignment
6:      $\eta = \frac{1}{|z^c|}$  ▷ Update learning rate
7:      $z^c \leftarrow (1 - \eta)z^c + \eta z$  ▷ Update
8:   end for
9:    $y = \tau_\theta(L)$  ▷ Label text embedding
10:  for each  $z^c$  do
11:     $z_t^c \sim q(z_t^c | z^c)$  ▷ Diffusion process
12:     $\tilde{z}^c = \mathcal{U}_t(\text{Concat}(z_t^c, y))$  ▷ Denoising process
13:  end for
14: end for
15:  $\mathcal{S} = \mathcal{D}(\tilde{Z}^c)$  ▷ Generate image

```

Output: \mathcal{S} : Distilled images.

Moreover, LDM is capable of modeling the conditional distribution, enabling DD tasks to incorporate the label information into synthetic images. In Eq. (2), LDM introduces a domain-specific encoder τ_θ to map the textual labels (prompts) into the feature space. This mapping is seamlessly integrated into the U-Net architecture (\mathcal{U}_t) through a cross-attention layer, facilitating the fusion of multi-modal features. For each prototype z^c and its corresponding label L , the synthesis process is formulated as

$$\text{output} = \mathcal{D}(\mathcal{U}_t(\text{Concat}(z_t^c, \tau_\theta(L)))) \quad (6)$$

where z_t^c represents the c -th prototype with noise. The distillation process is summarized in Algorithm 1.

3.3. Training-Time Matching

Since eliminates the necessity of matching with a specific architecture, separating data matching from the synthesis process reduces the computational overhead on large-scale datasets and addresses the cross-architecture issue inherent in the STM strategy. However, based on previous research [4, 8, 48] and preliminary experiments, we find that training large-scale distilled datasets with hard labels is prone to low testing accuracy. To address this, we introduce the TTM strategy, which is considered a distribution matching approach.

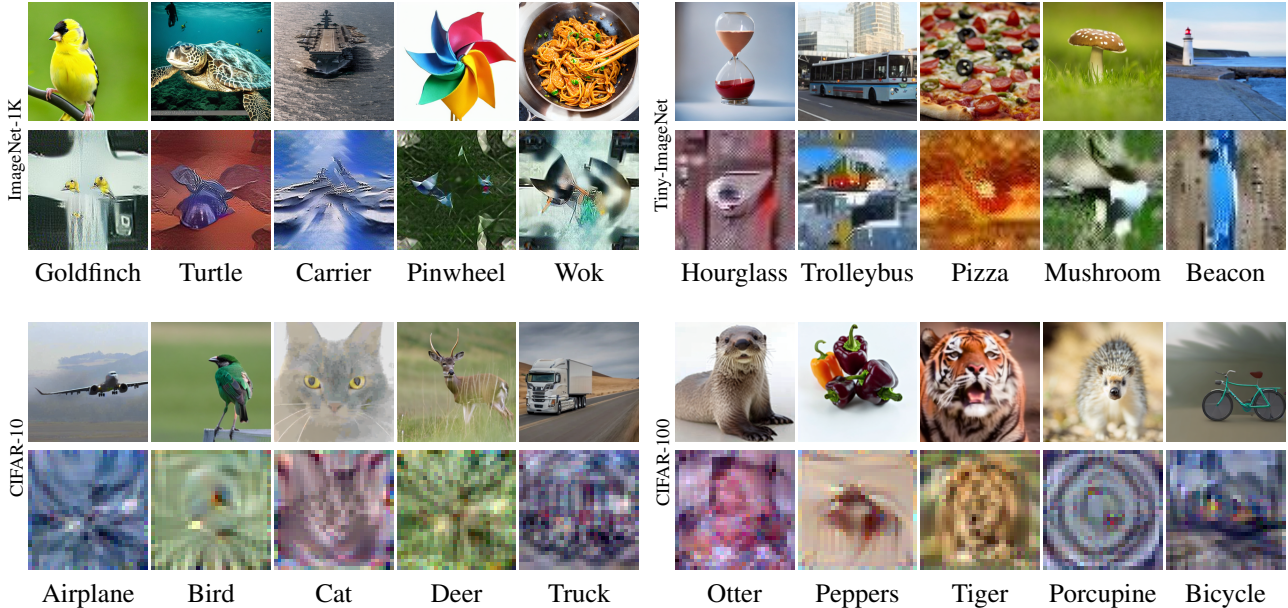


Figure 4. Visualization results. The top row of each dataset comes from D⁴M and the bottom comes from SRe²L [48] (ImageNet-1K and Tiny-ImageNet) and MTT [4] (CIFAR-10/100). The images generated by D⁴M have better resolution and are more lifelike.

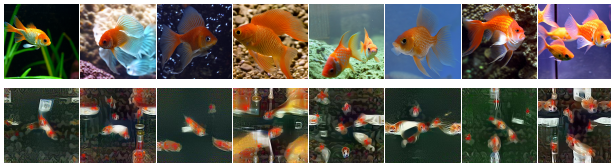


Figure 5. Visualization results within one category. D⁴M (top) provides richer semantic information than SRe²L.

TTM refers to training on distilled datasets with soft labels. Label softening is widely adapted in distillation tasks [15, 30, 47]. Since D⁴M infuses the label features into the synthetic data, it is natural to use the soft label during TTM. We employ soft label to align the distribution of student prediction $S_{\theta}(x)$ with teacher network T :

$$\theta_{\text{student}} = \arg \min_{\theta \in \Theta} L_{KL}(T(x), S_{\theta}(x)) \quad (7)$$

where $T(x)/S_{\theta}(x)$ is the teacher/student prediction for the distilled image x and L_{KL} represents the KL divergence. The output of the teacher network, also known as soft prediction or soft label, encapsulates richer semantic information compared to hard labels. Matching with the soft labels during training will enhance the robustness and generalization capability of the trained model [15]. For a fair comparison, we use the soft label storage method similar to the FKD [37] method, which generates soft labels and conducts matching at each training epoch:

$$\theta_{\text{student}}^{t+1} = \arg \min_{\theta \in \Theta} L_{KL}(T^t(x), S_{\theta}^t(x)). \quad (8)$$

4. Experiments

4.1. Setting and Evaluation

We evaluate the performance of D⁴M across various datasets and networks. All models employed for ImageNet-1K and Tiny-ImageNet are sourced from the PyTorch official model repository, while the ConvNet utilized for CIFAR-10/100 is based on the architecture proposed by Gidaris *et al.* [14]. Performance validation was carried out using PyTorch on NVIDIA V100 GPUs. Detailed training and validation hyperparameters are available in the supplementary material.

4.2. Dataset Distillation Results

In our comparative analysis, we evaluate the D⁴M against a range of techniques, encompassing both meta-learning and data-matching strategies. For small datasets, our comparison included two meta-learning methods: KIP [32] and FRePO [54], alongside four data-matching techniques: DSA [51], CAFE [40], TESLA [8], and SRe²L [48]. In the context of large-scale datasets, our focus shifted to a detailed comparison between TESLA and SRe²L.

CIFAR-10 and CIFAR-100 For small dataset distillation, the STM strategy outperforms when the number of categories and IPC (Image Per Class) are limited. However, as the category increases, the TTM strategy becomes more effective. This shift is attributed to the fact that the optimal solution derived from STM fails to ensure the convergence of the network training with large category numbers, thereby capping the testing performance. As evidenced in

Dataset	IPC	Meta-Learning		Data-Matching			SRe ² L [†]	D ⁴ M	Full Dataset
		KIP	FRePO	DSA	CAFE	TESLA			
CIFAR-10	10	62.7±0.3	65.5±0.6	52.1±0.5	50.9±0.5	66.4±0.8	(60.2)	56.2±0.4	84.8±0.1
	50	68.6±0.2	71.7±0.2	60.6±0.5	62.3±0.4	72.6±0.7		72.8±0.5	
CIFAR-100	10	28.3±0.1	42.5±0.2	32.3±0.3	31.5±0.2	41.7±0.3	-	45.0±0.1	56.2±0.3
	50	-	44.3±0.2	42.8±0.4	42.9±0.2	47.9±0.3	-	48.8±0.3	

Table 1. **Top-1 Accuracy \uparrow on small datasets.** We train the ConvNet-W128 [14] from scratch 5 times on the distilled dataset and evaluate them on the original test dataset to get the $\bar{x} \pm std$. [†]: SRe²L [48] achieves 60.2% Top-1 Accuracy on CIFAR-10 with IPC-1K.

Dataset	IPC	Method	R18	R50	R101	
ImageNet-1K	Full Dataset [†]		69.8	80.9	81.9	
	10	TESLA	7.7	-	-	
		SRe ² L	21.3	28.4	30.9	
		D ⁴ M	27.9	33.5	34.2	
	50	SRe ² L	46.8	55.6	60.8	
		D ⁴ M	55.2	62.4	63.4	
	100	SRe ² L	52.8	61.0	65.8	
		D ⁴ M	59.3	65.4	66.5	
	200	SRe ² L	57.0	64.6	65.9	
		D ⁴ M	62.6	67.8	68.1	
	Tiny-ImageNet	Full Dataset [‡]		61.9	62.0	62.3
		50	SRe ² L	44.0	47.7	49.1
D ⁴ M			46.2	51.8	51.0	
D ⁴ M-G			46.8	51.9	53.2	
100		SRe ² L	50.8	53.5	54.2	
		D ⁴ M	51.4	54.8	55.3	
		D ⁴ M-G	53.3	54.9	54.5	

Table 2. **Top-1 Accuracy \uparrow on large-scale datasets.** SRe²L [48] and our D⁴M employ ResNet18 as the teacher model to generate the soft label while TESLA [8] uses the ConvNetD4. All standard deviations in this table are < 1 . [†]: The results of ImageNet-1K come from the official PyTorch [websites](#). [‡]: The results of Tiny-ImageNet come from the model trained from scratch with the official PyTorch [code](#).

Tab. 1, when applied to CIFAR-100, D⁴M attains a Top-1 accuracy of 45.0% with merely IPC-10. This performance surpasses that of FRePO and TESLA by 2.5% and 3.3%.

ImageNet-1K and Tiny-ImageNet The TTM strategy demonstrates remarkable efficacy in large-scale DD tasks as presented in Tab. 2. The effectiveness stems from its ability to improve the quality of the synthetic data rather than imitate the performance of the original data. Consequently, it facilitates the processing of large-scale datasets with reduced computational complexity and memory demands. In terms of accuracy, the proposed D⁴M sets new benchmarks, achieving 66.5% and 51.0% with IPC-100 on ImageNet-1K and Tiny-ImageNet. Notably, it replicates the full dataset

Ablation	R18	R50	R101
Teacher: R18			
w/ STM	23.6	29.7	32.3
w/o STM	27.9(+4.3)	33.5(+3.8)	34.2(+1.9)
Teacher: R50			
w/ STM	15.8	20.6	22.3
w/o STM	20.7(+4.9)	24.7(+4.1)	26.7(+4.4)
Teacher: R101			
w/ STM	12.5	16.0	17.6
w/o STM	19.4(+6.9)	23.0(+7.0)	24.2(+6.6)

Table 3. **Comparison of Top-1 Accuracy \uparrow on different matching strategy.** We use the R18 as the distribution matching architecture. All methods are evaluated with IPC-10.

performance with 81.2% and 81.9%, respectively. Moreover, our approach significantly surpasses the leading data-matching method, SRe²L, across both datasets. This superiority is attributed to the integration of multi-modal fusion embedding in D⁴M.

Benefit to the architecture-free synthesis process, the datasets distilled by D⁴M exhibit versatility. To substantiate this characteristic, we extract 200 categories from the distilled ImageNet as the distilled Tiny-ImageNet in accordance with the predefined mapping [22]. The experimental outcomes of D⁴M-G in Tab. 2 demonstrate that our method not only manifests a pronounced distillation effect but also retains the applicability inherent to the original dataset.

4.3. Matching Strategy Analysis

As mentioned in Sec. 2, the DD task often uses the STM strategy to generate images. In order to validate the superiority of TTM strategy, we conduct the comparative experiments listed in Tab. 3. We execute the synthesis process through BN distribution matching on images distilled via D⁴M, resulting in distribution-matched synthetic images.

It is evident that the test performance with STM failed regardless of the chosen teacher network. The images dis-

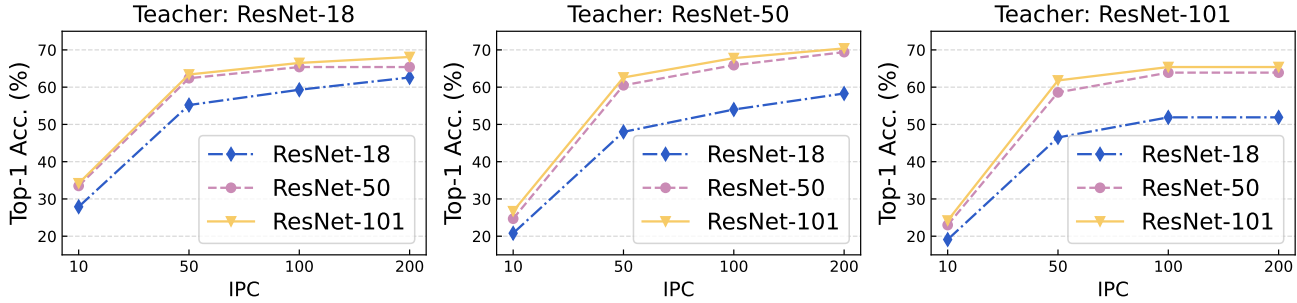


Figure 6. **Top-1 Accuracy \uparrow of ImageNet-1K on various teacher-student pairs.** The result of each pair increases consistently with larger IPC

Ablation	R18	R50	R101
Dataset: ImageNet-1K			
w/o PT	15.6	20.7	20.6
w/ PT	27.9(+12.3)	33.5(+12.8)	34.2(+13.6)
Dataset: Tiny-ImageNet			
w/o PT	30.5	35.6	37.3
w/ PT	46.2(+15.7)	51.8(+16.2)	51.0(+13.7)

Table 4. **Comparison of Top-1 Accuracy \uparrow on different initialization of diffusion process.** PT is the abbreviation of Prototype. All methods are evaluated with IPC-10.

titled via D^4M encapsulate not only the salient features of the original prototypes but also the text information of category labels. Therefore, the network solely trained with the original images proves inadequate for effectively managing such fused multi-modal features. Should the fused features be aligned with these networks, it would result in the disruption of the fused information, thereby diminishing the overall accuracy. It is worth noting that D^4M potentially offers high-quality initialization for STM, as it synthesizes images with higher testing accuracy compared to those derived from random white noise initialization.

4.4. Prototype Analysis

To ascertain the critical role of prototypes in D^4M , we conduct an ablation study on the diffusion process with random initialization and prototype initialization. The results listed in Tab. 4 demonstrate that the incorporation of a learned prototype markedly enhances the effectiveness of D^4M .

To showcase the merits of the prototype intuitively, we employ ResNet-18 for feature extraction from the distilled dataset, followed by t-SNE for dimensionality reduction. The visualization results (Fig. 7) reveal that the data synthesized via D^4M demonstrates enhanced inter-class discrimination and intra-class consistency.

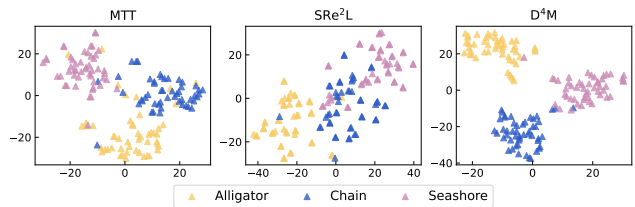


Figure 7. **T-SNE visualizations on Tiny-ImageNet.** The feature embedding distribution of D^4M displays more compact within classes and discriminative among classes.

4.5. Teacher-Student Network Analysis

We studied the performance of different teacher-student models with D^4M and the experimental results are shown in Fig. 6. Under the same teacher network, the accuracy of ResNet-18, ResNet-50, and ResNet-101 increases gradually. When IPC is small (such as 10 and 50), the student network trained with an enhanced teacher is prone to overfitting, resulting in reduced testing accuracy. As IPC increases, the large network shows stronger learning ability and the Top-1 accuracy improves. We further compare the performance of the distilled ImageNet on different teacher-student pairs, including CNNs and ViTs (Tab. 5). As a student network, the ViT-based networks assimilate the inductive bias inherent in CNN-based teachers, leveraging its global attention mechanism to attain the best Top-1 accuracy. Conversely, as a teacher network, ViT does not have such an inductive bias characteristic, yielding suboptimal results on their student networks. Nevertheless, ViT-based students consistently achieve superior Top-1 accuracy.

4.6. Qualitative Analysis

A pivotal advantage of D^4M lies in its utilization of the outputs from the image decoder \mathcal{D} as the distilled dataset, avoiding the need for STM. This implies that the pixel space of the generated image remains unaltered by any matching optimization, thereby preserving the reality of the distilled

Teacher Network	Student Network				
	ResNet-18	MobileNet-V2	EfficientNet-B0	Swin-T	ViT-B
ResNet-18	55.2	47.9	<u>55.4</u>	58.1	45.5
MobileNet-V2	47.6	42.9	49.8	58.9	<u>50.4</u>
Swin-T	27.5	21.9	26.4	38.1	<u>34.2</u>

Table 5. **Top-1 Accuracy \uparrow on ImageNet-1K with various teacher-student architectures.** ViT-based students show powerful learning ability with IPC-50.

image. Figures 4 and 5 exemplify the superior image quality achieved by D⁴M in comparison to its counterparts. It is evident that the D⁴M method not only guarantees the high resolution of the distilled image and preserves the integrity of semantic information but also ensures the richness of features within the same category. More visualizations and analysis can be found in supplementary material.

Method	Resolution	Time(s) \downarrow	GPU(GB) \downarrow
Dataset: ImageNet-1K			
MTT [†]	128 \times 128	45.0	79.9
TESLA [†]	64 \times 64	46.0	13.9
SRe ² L	224 \times 224	5.2	34.8
D ⁴ M	512 \times 512	2.7	6.1
Dataset: Tiny-ImageNet			
MTT	64 \times 64	5.4	48.9
SRe ² L	64 \times 64	11.0	33.8
D ⁴ M	512 \times 512	2.7	6.1

Table 6. **Synthesis time \downarrow and GPU memory \downarrow cost on large-scale datasets.** †: The runtime of MTT [4] and TESLA [8] on ImageNet-1K are measured for 10 iterations (500 matching steps).

4.7. Distillation Cost Analysis

We conduct the analysis of GPU memory consumption across various DD methods, with the corresponding results presented in Tab. 6. Notably, the architecture-free nature of D⁴M during synthesis ensures the fixed time and GPU memory costs. When considering STM and DTM, we observe an increase in both time and GPU memory usage with the enlargement of the matching architecture. For instance, the peak GPU memory utilization for SRe²L in the recovery of a 64 \times 64 image on ConvNet is 4.2 GB, whereas on ResNet-50, it reaches a substantial 33.8 GB. Similarly, when synthesizing a 64 \times 64 image on ConvNet, MTT demands a peak GPU memory of 48.9 GB. Furthermore, the number of iteration steps impacts the generation time for a single image in data matching. With the increased iteration steps, the time cost for SRe²L to recover a 224 \times 224 image on ResNet-50 gradually rises from 1.31s to 10.48s. Notably,

D⁴M demonstrates a remarkable reduction in time cost by a factor of 3.82 when compared to SRe²L. Figure 8 reveals that D⁴M attains best accuracy at a constant time cost.

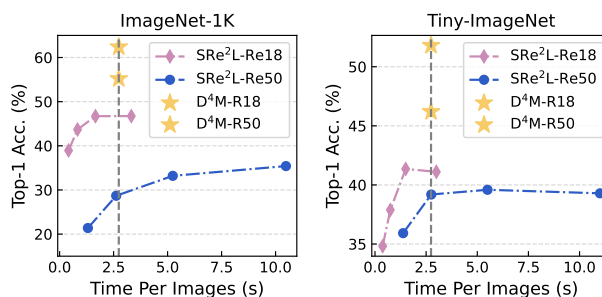


Figure 8. **Top-1 Accuracy \uparrow and synthesis time \downarrow on large-scale datasets.** D⁴M is architecture-free at synthesis time, thereby a constant runtime cost. Re is the abbreviation of Recover.

5. Conclusion

We introduce D⁴M, a novel and efficient dataset distillation framework leveraging the TTM strategy. For the first time, D⁴M addresses the cross-architecture generalization issue by integrating the principles of diffusion models with prototype learning. The distilled dataset not only boasts realistic and high-resolution images with limited resources but also exhibits a versatility comparable to that of the full dataset. D⁴M demonstrates outstanding performance compared to other dataset distillation methods, particularly when applied to large-scale datasets such as ImageNet-1K. Last but not least, rethinking the relationship between generative models and dataset distillation offers fresh perspectives, paving the way for the community to develop more efficient dataset distillation methods in future endeavors.

Limitation and future works. In the situation of extreme distillation (IPC-1/10), we observe a significant performance degradation. Our future work will concentrate on refining the distillation process for this challenging scenario and try to distill more real-world multi-modal datasets.

Acknowledgement. This work is supported by the National Natural Science Foundation of China (No. 12071458).

References

- [1] Richard E Blahut. *Fast algorithms for signal processing*. Cambridge University Press, 2010. 1
- [2] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22563–22575, 2023. 3
- [3] Zalán Borsos, Mojmír Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33:14879–14890, 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. 1, 2, 3, 4, 5, 8
- [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. 2
- [6] Xin Chen, Biao Jiang, Wen Liu, Zilong Huang, Bin Fu, Tao Chen, and Gang Yu. Executing your commands via motion diffusion in latent space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18000–18010, 2023. 3
- [7] Yutian Chen, Max Welling, and Alex Smola. Super-samples from kernel herding. *arXiv preprint arXiv:1203.3472*, 2012. 1
- [8] 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. 1, 2, 3, 4, 5, 6, 8
- [9] 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. 2
- [10] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. *Advances in Neural Information Processing Systems*, 35:34391–34404, 2022. 2
- [11] Jiawei Du, Yidi Jiang, Vincent YF Tan, Joey Tianyi Zhou, and Haizhou Li. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3749–3758, 2023. 3
- [12] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7346–7356, 2023. 3
- [13] Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woitschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. A survey on dataset distillation: Approaches, applications and future directions. *arXiv preprint arXiv:2305.01975*, 2023. 2
- [14] 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, 6
- [15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 5
- [16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3
- [17] Yujin Jeong, Wonjeong Ryoo, Seunghyun Lee, Dabin Seo, Wonmin Byeon, Sangpil Kim, and Jinkyu Kim. The power of sound (tpos): Audio reactive video generation with stable diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7822–7832, 2023. 3
- [18] Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z Pan. Delving into effective gradient matching for dataset condensation. In *2023 IEEE International Conference on Omni-layer Intelligent Systems (COINS)*, pages 1–6. IEEE, 2023. 3
- [19] Seung Wook Kim, Bradley Brown, Kangxue Yin, Karsten Kreis, Katja Schwarz, Daiqing Li, Robin Rombach, Antonio Torralba, and Sanja Fidler. Neuralfield-ldm: Scene generation with hierarchical latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8496–8506, 2023. 3
- [20] Juil Koo, Seungwoo Yoo, Minh Hieu Nguyen, and Minhyuk Sung. Salad: Part-level latent diffusion for 3d shape generation and manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14441–14451, 2023. 3
- [21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario, 2009. 2
- [22] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6
- [23] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist>, 2, 2010. 2
- [24] Saehyung Lee, Sanghyuk Chun, Sangwon Jung, Sangdoon Yun, and Sungroh Yoon. Dataset condensation with contrastive signals. In *International Conference on Machine Learning*, pages 12352–12364. PMLR, 2022. 1, 3
- [25] Shiye Lei and Dacheng Tao. A comprehensive survey to dataset distillation. *arXiv preprint arXiv:2301.05603*, 2023. 2
- [26] Gen Li, Varun Jampani, Laura Sevilla-Lara, Deqing Sun, Jonghyun Kim, and Joongkyu Kim. Adaptive prototype learning and allocation for few-shot segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8334–8343, 2021. 4
- [27] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *Advances in Neural Information Processing Systems*, 35:13877–13891, 2022. 2
- [28] Calvin Luo. Understanding diffusion models: A unified perspective. *arXiv preprint arXiv:2208.11970*, 2022. 3

- [29] Zhaoyang Lyu, Jinyi Wang, Yuwei An, Ya Zhang, Dahua Lin, and Bo Dai. Controllable mesh generation through sparse latent point diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 271–280, 2023. 3
- [30] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 5
- [31] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. 2
- [32] 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, 5
- [33] 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. 3, 4
- [34] Noveen Sachdeva and Julian McAuley. Data distillation: A survey. *arXiv preprint arXiv:2301.04272*, 2023. 2
- [35] David Sculley. Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178, 2010. 4
- [36] Shuai Shen, Wenliang Zhao, Zibin Meng, Wanhua Li, Zheng Zhu, Jie Zhou, and Jiwen Lu. DiffTalk: Crafting diffusion models for generalized audio-driven portraits animation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1982–1991, 2023. 3
- [37] Zhiqiang Shen and Eric Xing. A fast knowledge distillation framework for visual recognition. In *European Conference on Computer Vision*, pages 673–690. Springer, 2022. 5
- [38] Yu Takagi and Shinji Nishimoto. High-resolution image reconstruction with latent diffusion models from human brain activity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14453–14463, 2023. 3
- [39] Mariya Toneva, Alessandro Sordani, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 1
- [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. 1, 3, 5
- [41] Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. Dim: Distilling dataset into generative model. *arXiv preprint arXiv:2303.04707*, 2023. 2
- [42] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 2
- [43] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7378–7387, 2023. 3
- [44] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 2
- [45] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3474–3482, 2018. 4
- [46] Ling Yang, Zhilong Zhang, Yang Song, Shenda Hong, Runsheng Xu, Yue Zhao, Wentao Zhang, Bin Cui, and Ming-Hsuan Yang. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 2022. 3
- [47] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4133–4141, 2017. 5
- [48] Zeyuan Yin, Eric Xing, and Zhiqiang Shen. Squeeze, recover and relabel: Dataset condensation at imagenet scale from a new perspective. *arXiv preprint arXiv:2306.13092*, 2023. 2, 3, 4, 5, 6
- [49] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*, 2023. 2
- [50] Baoquan Zhang, Xutao Li, Yunming Ye, Zhichao Huang, and Lisai Zhang. Prototype completion with primitive knowledge for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3754–3762, 2021. 4
- [51] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 1, 3, 5
- [52] 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, 3
- [53] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 1, 3
- [54] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 2, 5