Label-Augmented Dataset Distillation

Seoungyoon Kang[1,2]*, Youngsun Lim[2]*, Hyunjung Shim[2]
[1]Yonsei University, [2]KAIST AI
sy.kang@yonsei.ac.kr, {youngsun_ai, kateshim}@kaist.ac.kr

Abstract

Traditional dataset distillation primarily focuses on image representation while often overlooking the important role of labels. In this study, we introduce Label-Augmented Dataset Distillation (LADD), a new dataset distillation framework enhancing dataset distillation with label augmentations. LADD sub-samples each synthetic image, generating additional dense labels to capture rich semantics. These dense labels require only a 2.5% increase in storage (ImageNet subsets) with significant performance benefits, providing strong learning signals. Our label-generation strategy can complement existing dataset distillation methods and significantly enhance their training efficiency and performance. Experimental results demonstrate that LADD outperforms existing methods in terms of computational overhead and accuracy. With three high-performance dataset distillation algorithms, LADD achieves remarkable gains by an average of 14.9% in accuracy. Furthermore, the effectiveness of our method is proven across various datasets, distillation hyperparameters, and algorithms. Finally, our method improves the cross-architecture robustness of the distilled dataset, which is important in the application scenario.

1. Introduction

Dataset distillation, also called dataset condensation, creates a small synthetic training set to reduce training costs. The synthesized dataset enables faster training while maintaining a performance comparable to that achieved with the source dataset. For example, FrePo [45] attained 93% of full dataset training performance using merely one image per class in MNIST [6]. Dataset distillation can be applied in various fields. These include privacy-free training data generation (e.g., federated learning [12,31,46], medical image computing [20,31]), fast training (e.g., network architecture search [41–43]), or compact training data generation (e.g., continual learning [41–43]).

*Equal contribution

The efficacy of distilled datasets is typically evaluated based on the test accuracy achieved by models trained by these datasets. The distilled dataset must maximally encapsulate essential information of the source dataset within a limited number of synthetic samples. Prior research [2, 21, 22, 33, 36, 42, 43] has refined the optimization objective within the bi-loop nested meta-learning framework for dataset synthesis. Some methods have further explored optimization spaces beyond image [3,9] and efficient ways to utilize pixel-space [17]. Additionally, several approaches [4, 34, 45] develop algorithms to reduce the computational cost induced by the bi-loop optimization. However, these efforts mostly focus on data representation in images, overlooking the important roles of labels.

Labels, pivotal in supervised learning, pair with images to provide strong learning signals. In contrast to images, labels provide highly compressed representations because they are defined in a semantic space. For instance, in the ImageNette-128 [16], representing a "cassette player" requires 49,000 scalars ($128 \times 128 \times 3$) for the image, but only ten scalars for its one-hot vector label. This substantial difference between image and label suggests a new perspective to dataset distillation, emphasizing the potential of harnessing more information from labels rather than images.

Addressing the overlooked potential of labels in dataset distillation, we introduce Label-Augmented Dataset Distillation (LADD). LADD effectively exploits labels in a distilled dataset. Our approach comprises two main stages: distillation and deployment, as depicted in Fig. 1. In the distillation stage, we first generate synthetic images using existing distillation algorithms. Subsequently, we apply an image sub-sampling algorithm to each synthetic image. For each sub-image (termed a local view), we generate a dense label, sub-image's soft label, which encapsulates high-quality information. During the deployment stage, LADD uniquely merges global view images with their original labels and local view images with the corresponding dense labels, delivering diverse learning signals.

LADD presents three key benefits over prior methods: (1) enhanced storage efficiency with smaller increments in dataset sizes, (2) reduced computational demands, and (3)

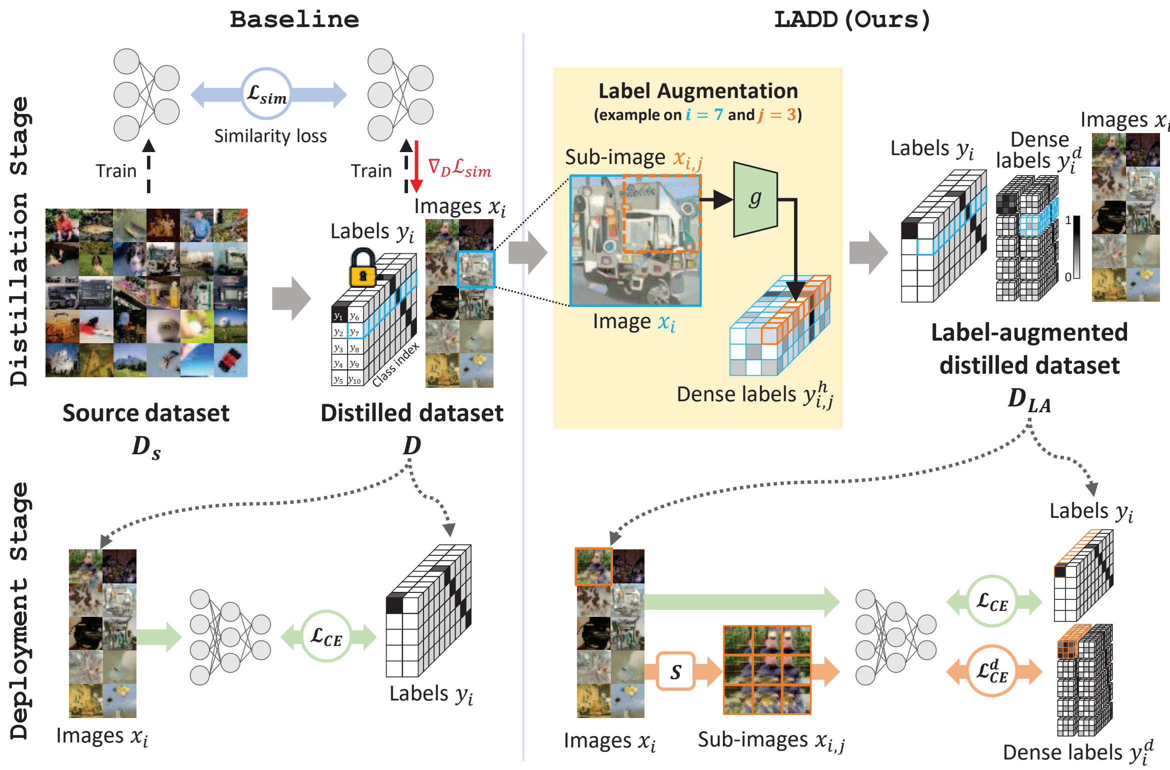


Figure 1. **Overview of LADD.** Once the distilled dataset D is synthesized by baseline, LADD initiates label augmentation. It divides each image in D into $N \times N$ sub-images, as illustrated in Fig. 1 ($N = 3$). Then, N^2 soft labels are computed using the labeler g to produce the dense label. Label augmented distilled dataset D_{LA} consists of images, labels, and dense labels; it is utilized in the deployment stage to train the evaluation model.

improved performance and robustness across different testing architectures. First, LADD employs a fixed-parameter sampling rule for sub-image generation, ensuring minimal memory overhead (e.g., only 2.5% regardless of IPC (images per class)). Second, the computational demands are significantly lowered as the label augmentation process only involves dense label predictions. Lastly, rich information encoded in labels serves as effective and robust training signals at the deployment stage. In this way, LADD leverages the diverse local information obtained from dense labels.

Experimental results validate these key advantages of our LADD. At 5 IPC, LADD consistently surpasses the 6 IPC baseline while consuming 87% less memory. This underscores the memory efficiency of our method. Additionally, in this setup, LADD only requires an extra 0.002 PFLOPs for label augmentation compared to the 5 IPC baseline. This is notably lower than the additional 211 PFLOPs required by the 6 IPC setup. Furthermore, LADD improves the performances of three baselines by an average of 14.9%, validated across five test model architectures and five distinct datasets. Finally, GradCAM [28] visualizations show that LADD-trained models capture objects within images more accurately. This demonstrates the robustness of our label-augmented distilled dataset approach.

Our contributions can be summarized as follows:

- We recognize the crucial role of labels in dataset distillation, an aspect neglected in existing research.
- We introduce a novel framework, label-augmented dataset distillation, which utilizes dense labels for local views of each synthetic image. We offer an effective training method for the deployment stage to maximize the use of the distilled dataset.
- Extensive experiments reveal that our method significantly improves computation efficiency, storage efficiency, and cross-architecture robustness. Moreover, our approach can be effectively integrated with existing image-focused distillation methods.

2. Related work

Preliminary: dataset distillation. Dataset distillation is the process of synthesizing a dataset, denoted as D, which comprises a small, representative subset of samples extracted from a larger source dataset D_s. With the number of total classes C and the number of images per class (IPC), the distilled dataset D contains $C \times$ IPC image-label pairs (i.e., $D = \{(x_i, y_i)_{i=1}^{C\times\text{IPC}}\}$).

To achieve dataset distillation, algorithms employ a bi-loop optimization strategy consisting of two phases: the inner-loop and the outer-loop. The inner loop simulates the training of two models with the source dataset D_s and the synthetic dataset D, respectively. In detail, two models $f(x_s, \theta_s)$ and $f(x, \theta)$ with the same structure are trained on D_s and D for one or several iterations from the identical initial weights θ_O. Subsequently, with pre-trained models, the outer loop updates the distilled dataset such that the model trained on D approximates the model trained on D_s. The optimization objective for the outer loop is to minimize $\mathcal{L}_{sim}$ loss that measures the difference between two trained models at the inner loop:

$$\mathcal{L}_{sim}(D_s, D) = \text{dist}(f(\cdot;\theta_s), f(\cdot;\theta)). \tag{1}$$

Then, the distilled dataset D is updated to reduce the dissimilarity:

$$D := D - \beta \nabla_D \mathcal{L}_{sim}(D_s, D), \tag{2}$$

where β is the learning rate for the dataset.

We refer to the aforementioned process as the distillation stage. Subsequently, during the deployment stage, we utilize the distilled dataset to train a model, represented as $y = h(x;\phi)$. This model undergoes evaluation on the real validation dataset D_s^{val}.

Trends in dataset distillation algorithm. Various distillation methods have been proposed to define the similarity loss, denoted as L_{sim}. Performance matching [36] and distribution matching [26, 35, 39, 42, 44] utilize a distance function to measure similarity in predictions or features, respectively. Gradient matching [43] aligns gradients of the network parameter θ_s and θ for increased efficiency by reducing multiple inner-loop iterations. Trajectory matching [2, 13] focuses on minimizing the parameter distance between θ_s and θ after several inner-loop updates. This approach captures the long-range relationship between parameters, an aspect that gradient matching does not address. In contrast, DiM [34] and SRe^{2}L [38] bypass bi-loop optimization by using conditional GANs and reversing fully-trained models for distilled data synthesis, respectively.

Other methods enhance the robustness or image representation of the distilled dataset. DSA [41] utilizes an augmentation-aware synthesis for diverse image augmentations. ModelAug [40] increases the synthesis robustness of D by diversifying the θ configuration during distillation. AST [29] uses a smooth teacher in trajectory matching [2] to emphasize essential trajectory for D and employs additive noise to augment the teacher while distillation. To improve image representation, GLaD [3] and LatentDD [9] regularize the manifold of D based on GAN [27] and Diffusion Model [24]. IDC [17] enriches representation by embedding multiple small images within a single image of D.

Our focus is on enriching label space information to enhance distilled dataset quality. We emphasize that our method is both compatible with and capable of synergizing with other distillation methods in image synthesis.

A few methods draw focus to utilizing labels. FDD [1] optimizes only labels while images are randomly selected from the source dataset. FrePo [45] optimizes both images and labels at once. TESLA [4] uses a soft label for each image. These methods are limited to using a single label per image. On the other hand, we augment a single label into multiple informative labels, achieving enhancements in both memory efficiency and performance.

3. Method

We propose Label-Augmented Dataset Distillation (LADD), a specialized label augmentation method for dataset distillation. During the dataset distillation stage, LADD conducts a label augmentation process to images distilled by conventional image-level dataset distillation algorithms. For each image x, we produce additional groups of soft labels, denoted dense labels, and create a label-augmented dataset D_{LA}. Specifically, to obtain D_{LA}, the label augmentation step goes through two processes: (1) an image sub-division and (2) a dense label generation. In the deployment stage, LADD uses both global (i.e., full images with hard labels) and local data (i.e., sub-sampled images with dense labels) to train the network effectively. Fig. 1 depicts the overview of our method.

In the following section, we describe details of the label augmentation process (Sec. 3.1) and the labeler acquisition (Sec. 3.2). Finally, we demonstrate the training procedure of the deployment stage (Sec 3.3).

3.1. Label Augmentation

We denote the image-level distilled dataset $D = \{(x_i, y_i) | i \in [1, C \times \text{IPC}]\}$, where C is the number of classes in the source dataset D_s and IPC is the number of images per class. In our framework, D is generated using an existing image-level distillation algorithm. By preserving the effectiveness of the image-level distilled dataset, our method synergizes with state-of-the-art dataset distillation algorithms, leveraging their strengths.

Image Sub-Sampling. We define a function S that samples synthetic image $x_i \in D$ into several sub-images. Considering the memory-constrained environment, dynamic sub-image sampling is not an optimal choice because it requires saving additional sampling parameters. Therefore, we restrict S to be a static strategy sampler. We sample N^2 sub-images from x_i. Each sub-image covers $R\%$ of each axis. To achieve a uniform sampling across x_i, we maintain a consistent stride $(100\% - R\%)/(N-1)$ for cropping. For example, for x_i of 128×128 pixels, using $R = 62.5\%$ and $N = 5$, we obtain 25 sub-images of 80×80 pixels each, applying a 12-stride. After the sub-sampling, we resize each sub-image to match the dimension of x_i. For clarity, we

Algorithm 1 Label Augmentation

Input: Distilled dataset $D = \{(x_i, y_i)\}$, Labeler g, Sub-sampling function S
Output: Label augmented dataset D_{LA}
for each image x_i in D **do**
  for $j = 1$ to N^2 **do**
    $x_{i,j} \leftarrow S_j(x_i)$ ▷ Generate j-th sub-image
    $y^d_{i,j} \leftarrow g(x_{i,j})$ ▷ Generate sub-image soft label
  end for
  Add (x_i, y_i, y^d_i) to D_{LA}
end for
return D_{LA}

denote the sub-sampling function S as below:

$$x_{i,j} = S_j(x_i), \tag{3}$$

where $j \in [1, N^2]$ is the index of sub-sampled image.

Dense Label Generation. Sub-images, derived from the same original image, vary in visual content. In detail, each sub-image exhibits distinct patterns, conveying different levels of class information. We generate labels for each sub-image $x_{i,j}$, resulting in N^2 labels for each synthetic image x_i. To capture rich information in these labels, we opt for soft labeling. We develop the labeler $y^s = g(x)$, where x denotes the image and y^s is the corresponding soft label. We train the labeler on the source dataset D_s from scratch. Then, we obtain a dense label y^d from each sub-image:

$$y^d_{i,j} = g(S_j(x_i)). \tag{4}$$

We will discuss how to train g in Sec 3.2.

After the dense label generation, we obtain the original hard label y_i and a dense label y^d_i containing N^2 soft labels for a synthetic image x_i. We denote the label augmented dataset as $D_{LA} = \{(x_i, y_i, y^d_i) | i \in [1, C \times \text{IPC}]\}$. The synthesis process of D_{LA} is illustrated in Algorithm 1.

One straightforward approach might involve optimizing labels as part of the distillation process. However, it adds complexity to an already complicated optimization process, potentially leading to instability. Furthermore, it reduces computational efficiency due to slower convergence and increased operations per iteration. Instead, our LADD first applies existing distillation methods for image-level distillation. Subsequently, we perform a label-augmentation step on the distilled data, producing final datasets with our generated labels. In this way, LADD enjoys significant performance gains with minimal computational overhead.

Both LADD and knowledge distillation [15] use a teacher model but differ in the medium of knowledge transfer. Knowledge distillation transfers knowledge through an online teacher during the evaluation stage. However, LADD produces a dataset of images and augmented labels which are fixed after the distillation. In other words, LADD do not require any online model, such as a teacher, during the deployment stage.

3.2. Acquiring Labeler g.

LADD employs a labeler g to generate dense labels, employing the same labeler across all evaluations for fairness. To minimize overhead, we design g as a small network mirroring the distillation architecture (ConvNetD5). We train it for 50 epochs with a learning rate of 0.015, saving parameters at epochs 10, 20, 30, 40, and 50. We use the model trained up to 10 epochs as our early-stage labeler g, as it provides general and essential information for sub-images. This is well-aligned with existing dataset distillation methods [2, 13]. Although g is trained on a source dataset, it appropriately predicts labels for distilled images because the distilled dataset retains local structures of the source data.

Apart from our chosen method, classifiers trained on different data, including zero-shot models like CLIP [23], can be used as g. However, they do not produce more effective dense labels than our method. This is because these pre-trained models are not trained on the distilled dataset and have different architectures from those used in distillation.

3.3. Training in Deployment Stage

We closely follow the deployment stage from existing approaches. Given the dataset D_{LA} and an optimized learning rate η, we conduct standard classification training on the target network $h(x, \phi)$. Additionally, we modify the data input and training loss to effectively utilize informative dense labels in D_{LA}:

$$L_{cls} = CE(h(x_i, \phi), y_i) + \sum_{j=1}^{N^2} CE(h(S_j(x_i), \phi), y^d_{i,j}), \tag{5}$$

where $CE(\cdot, \cdot)$ is a cross-entropy loss. The dimensions of y_i (one-hot) and $y^d_{i,j}$ (soft) are the same as $\mathbb{R}^C$, and the dimension of y^d_i is $\mathbb{R}^{N^2 \times C}$. Through this process, we provide diverse training feedback through augmented dense labels beyond the signal provided by D.

4. Experiment

4.1. Implementation details

Image Sub-Sampling. The sub-sampling function is selected as a uniform sampler S with R = 62.5% and N = 5; R and N are determined experimentally (experiments are in Sup.A). Throughout the experiments, 25 sub-images are generated per synthetic image, and each sub-image is 80×80 in size when using 128×128 source dataset.

IPC	Method	MTT	AST	GLaD (MTT)	Overhead
1	Baseline	38.3±0.9	39.0±1.2	34.3±1.0	-
	Baseline++	**42.6±1.0**	41.8±1.2	**41.8±1.4**	100.1%
	LADD (ours)	40.9±1.3	**41.9±1.6**	40.7±1.2	2.5%
5	Baseline	49.5±1.4	51.4±1.2	48.0±1.1	-
	Baseline++	50.5±1.0	52.1±1.3	48.6±1.2	20.7%
	LADD (ours)	**52.6±0.8**	**60.1±0.9**	**58.4±0.9**	2.5%
10	Baseline	54.6±1.3	53.2±0.9	52.3±1.1	-
	Baseline++	55.4±1.2	54.2±1.3	52.4±1.2	10.0%
	LADD (ours)	**55.6±1.2**	**62.0±0.5**	**62.8±0.9**	2.5%
20	Baseline	58.2±1.2	55.5±1.5	53.3±1.2	-
	Baseline++	59.2±1.3	56.9±1.3	54.9±1.0	5.0%
	LADD (ours)	**59.6±0.5**	**59.4±1.0**	**66.5±0.8**	2.5%

Table 1. **ImageNette (128×128) Performance on Various IPC (images-per-class).** Each result reports an average of validation set accuracy of training ConvNetD5, AlexNet, VGG11, and ResNet18 on synthetic datasets which are distilled using a ConvNetD5 (4-CAE, four cross-architecture evaluation). The numbers after the '±' symbol are the average standard deviation of five trials per evaluation. The best performance is bolded, and the second-best performance is underlined.

Dataset. Various high-resolution image datasets are used as the source and evaluation datasets. They include ImageNet [5] and its subsets, such as ImageNette, ImageWoof [16], ImageFruit, ImageMeow, and ImageSquawk [2]. Each subset contains 10 classes and around 1,300 images per class. All images are center-cropped and resized into 128 × 128.

Baselines. We benchmark our method against a range of notable dataset distillation methods. These include MTT [2], AST [29], GLaD [3], DC [43], DM [42], and TESLA [4]. We re-implement DC and DM within the GLaD framework. For all distillation processes, we employ the ConvNetD5, a 5-layer convolutional network [11], as the standard distillation model architecture. For ImageNet-1K, we compare TESLA [4], SRe^{2}L [38], and RDED [32].

Labeler g. To ensure fairness, we use the same labeler g for all experiments. We train g on each source dataset for ten epochs using stochastic gradient descent (SGD) with a learning rate of 0.01 and a batch size of 256, following [2].

Cross-Architecture Evaluation. To evaluate the robustness of distilled data across various architectures, we use five different models [3] including four unseen models (ConvNetD5 [2], AlexNet [19], VGG11 [30], ResNet18 [14], and ViT [7]) except in Tab. 1. We refer to this protocol as 5-CAE. The scores represent the average of five independent trainings for each model. Each test model is trained for 1,000 epochs using the synthetic dataset. We adhere to the learning rate and decay strategy for each model as in [3]. Both baseline and LADD use the same data augmentations [41].

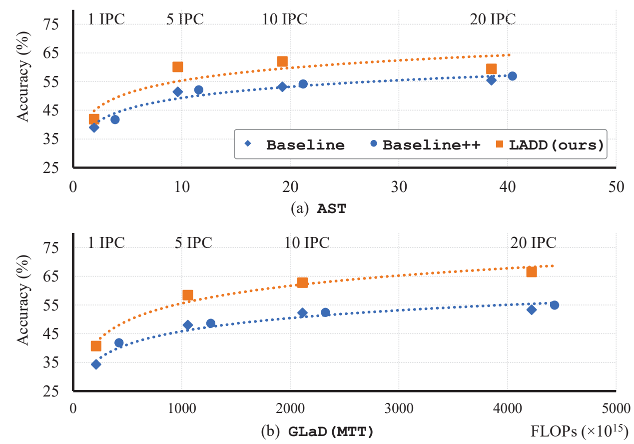


Figure 2. **FLOPs-Accuracy Plot for Distillation.** x-axis indicates the total computational cost to obtain D in FLOPs. For LADD, we compute FLOPs for both synthesizing D and creating dense labels. Each result uses ImageNette.

4.2. Quantitative evaluation

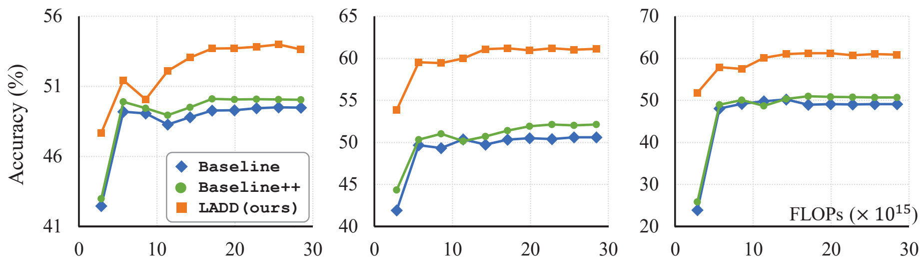
We quantitatively evaluate LADD by benchmarking it against representative distillation methods (MTT [2], AST [29], and GLaD [3]) in various IPC settings. LADD incurs additional memory usage compared to the baseline because of labeler training and label augmentation. For fair comparison, we evaluate the baselines with incremented IPC (i.e., IPC+1), labeled as baseline++. We focus on 4-CAE results in Tab. 1 since MTT and AST are not fully compatible with heterogeneous architectures (e.g., several experiments failed to converge on ViT architecture). The additional memory overhead for both images (uint8) and labels (float32) is calculated utilizing the Python *zipfile* library [10], the standard compression method.

Tab. 1 presents the results for varying IPC on the ImageNette. The quantitative analysis reveals that LADD surpasses the baseline, showing an average improvement of 15% at 5 IPC. Notably, our method outperforms baseline++ in all cases except at 1 IPC. At 1 IPC, baseline++ entails a 100.1% increase in memory usage. In contrast, LADD achieves comparable performance with only a 2.5% storage overhead, resulting in 40 times greater memory efficiency. For 5 IPC, baseline++ requires 20.7% more memory to accommodate an extra image per class. Conversely, LADD requires only an additional 2.5% memory while achieving, on average, a 13.2% better performance than baseline++ across three models. Consequently, we conclude that our approach shows impressive performances in terms of accuracy and efficiency, creating synergies with existing dataset distillation algorithms.

We evaluate the cross-architecture robustness of our method. Tab. 2 shows results for five architectures during the deployment stage. Notably, the baseline's ViT exhibits the weakest performance due to the architectural divergence

	MTT		AST		GLaD (MTT)	
	Baseline	LADD	Baseline	LADD	Baseline	LADD
ConvNetD5	61.2±1.5	**62.1±0.8**	63.8±0.5	**66.8±0.4**	61.2±0.4	**69.0±0.8**
VGG11	49.6±1.8	**50.6±1.5**	48.3±1.4	**58.1±0.7**	49.0±1.0	**60.0±1.3**
ResNet18	57.3±1.9	**59.0±1.6**	54.9±0.7	**63.6±0.6**	55.6±1.9	**65.5±0.7**
AlexNet	46.4±0.6	**51.0±0.6**	45.6±1.1	**59.4±0.3**	43.3±0.9	**56.7±0.8**
ViT	35.9±0.8	**37.8±0.5**	31.0±1.3	**32.6±2.2**	32.6±0.2	**42.5±1.2**
Avg.	50.1±1.3	**51.8±1.3**	48.7±1.0	**56.1±0.8**	48.3±0.9	**58.7±1.0**

Table 2. **Detail Results in Cross-Architecture Evaluation.** All results are measured on ImageNette dataset at 10 IPC.



(a) MTT (b) AST (c) GLaD (MTT)

Figure 3. **FLOPs-Accuracy Plot at the Deployment Stage.** x-axis indicates the total computational cost at the deployment stage in FLOPs. Among the three algorithms, LADD shows the best performance. Each result uses ImageNette at 5 IPC.

Method	Accuracy (%)	Assumption compliance
TESLA [38] (ICML'23)	7.7±0.1	✔
SRe^{2}L [38] (NeurIPS'23)	21.3±0.6	✔
RDED-I (H)	12.4±0.3	✔
RDED-I (S)	23.6±0.3	✔
LADD-RDED-I (ours)	**28.8±0.5**	✔
RDED [32] (CVPR'24)	42.0±0.3	✘

Table 3. **Performance on ImageNet-1K Dataset.** Each model uses ResNet-18 [14] as a test model. IPC is set to 10.

between the models in the distillation and deployment stages. Therefore, ViT's performance is a key indicator of the architecture robustness of the distilled dataset. LADD enhances performance across various architectures, particularly boosting ViT performance by 31% in GLaD (MTT). The dense label in LADD improves the representation quality and generalization within the distilled dataset.

Additionally, we show that LADD surpasses other dataset distillation methods on the ImageNet-1K [5], as shown in Tab. 3. ImageNet-1K presents significant challenges in dataset distillation due to high GPU consumption and complex optimization. For RDED, we remove the labeling process that uses the teacher model at the deployment stage. Using the teacher model at deployment stage violates the assumption of dataset distillation because it aligns more with knowledge distillation (Sec. 3.1). We denote the modified model as RDED-I (H or S), which consists of the distilled image and either hard or soft labels. Without online knowledge transfer of the RDED, we observe that RDED-I (H) only achieves 12.4% accuracy. RDED-I (S) shows better accuracy at 23.6%, which is better than SRe^{2}L. Our method demonstrates the best performance. We conclude that our approach improves the performance on a large dataset. More details are described in the Sup.B.

We compute the FLOPs requirement to assess the computational overhead for creating distilled data D and D_{LA}. Fig. 2 presents the total FLOPs necessary to distill D (⧫, ●) and D_{LA} (■). It also shows their corresponding deployment stage accuracies for baseline, baseline++, and LADD. Our observations indicate that LADD is more resource-efficient and achieves higher accuracy than both baseline and baseline++. There's a noticeable offset between the trend lines of LADD and baseline. This difference highlights our greater computational efficiency compared to previous studies. According to Fig. 2, the computational cost of LADD is slightly higher than that of the baseline, but significantly lower than that of baseline++. This is because LADD's computation includes labeler training and label augmentation in addition to the baseline distillation. However, these additional costs are much smaller than those for baseline distillation. Thus, it is a fair comparison of computational efficiency.

Furthermore, for an equitable comparison of the training cost, we conduct the experiments using the same batch size and number of iterations during the deployment stage. Fig. 3 depicts the accuracy of each model relative to the training cost. LADD outperforms both the baseline and baseline++ under the same training cost.

In Tab. 4, we report performances across various

Method	ImageNette	ImageFruit	ImageWoof	ImageMeow	ImageSquawk
MTT	45.3±1.1	31.7±1.8	28.3±1.2	33.0±1.1	41.5±1.0
LADD-MTT (ours)	**49.2±0.9**	**35.5±1.2**	**31.0±0.8**	**36.4±0.7**	**48.2±0.8**
AST	47.3±1.2	32.9±1.9	29.3±1.1	32.0±1.5	35.1±2.1
LADD-AST (ours)	**53.4±1.1**	**40.3±1.4**	**33.0±1.1**	**36.0±1.0**	**43.2±1.0**
GLaD(MTT)	44.2±1.0	27.5±1.0	24.5±0.9	30.0±0.8	34.0±1.3
LADD-GLaD(MTT) (ours)	**53.9±0.9**	**32.5±1.2**	**26.1±0.6**	**33.7±1.1**	**42.1±0.8**

Table 4. **Performance Improvement on Various Datasets.** All methods are trained on each dataset at 5 IPC. All values are 5-CAE results.

	Baseline	LADD
MTT	45.3±1.1	**49.2±0.9**
AST	47.3±1.2	**53.4±1.1**
GLaD(MTT)	44.2±1.0	**53.9±0.9**
GLaD(GM)	39.8±0.7	**52.1±1.0**
GLaD(DM)	37.2±1.2	**49.9±1.0**
TESLA	19.2±0.7	**27.3±0.7**

Table 5. **Performance on Various Algorithms.** All 5-CAE results are measured in ImageNette dataset at 5 IPC.

datasets. These results consistently demonstrate that LADD significantly enhances the performance of baselines across different source datasets. For each baseline model, we calculated the percentage improvement of LADD over the original models for all five datasets and then averaged them. We further averaged the improvements across the three baselines. This comprehensive calculation shows that LADD achieves an average performance improvement of 14.9% across the five datasets. This consistent improvement is a strong indication of our method's generalizability, regardless of the dataset. Tab. 5 presents the results from using various distillation algorithms. Analogous to the previous results, LADD significantly outperforms the various baselines. TESLA depicts low accuracy in both Tab. 3 and Tab. 5 because it reduces computations by ignoring training feedback. Detailed information is described in the Sup.C. Based on the experiments, we conclude that LADD demonstrates robustness and efficiency across a range of IPC settings, datasets, and architectures.

In conclusion, our extensive experiments establish that our method is effective in several key aspects. First, it demonstrates resource efficiency, as illustrated in Fig. 2. Second, it provides high compactness relative to its performance, evidenced in Tab. 1. Third, it consistently delivers superior training performance in diverse environments, as shown in Tab. 4 and 5. These findings collectively confirm that LADD significantly improves the quality of distilled datasets via efficient label augmentation.

4.3. Impact of Dense Labels in LADD

In this section, we investigate the most efficient ways to utilize a distilled dataset. We designate GLaD(MTT) as our baseline model. Tab. 6 presents the deployment stage performance using different combinations of datasets and labels. We note that the performance differences are negligible when training each image in D with hard labels, soft labels, or a mix of both. Additionally, using only sub-images with hard labels yields results comparable to the baseline. However, employing sub-images with corresponding dense labels results in a significant performance improvement of 7%p. This underscores that the combined strategy of image sub-sampling and dense label generation in LADD is highly effective for label utilization. Furthermore, integrating training with full images and their hard labels into previous experiments leads to an extra 2.8%p boost. This demonstrates that LADD, which leverages both local views with dense labels and global views of distilled images, is the most effective approach for label augmentation.

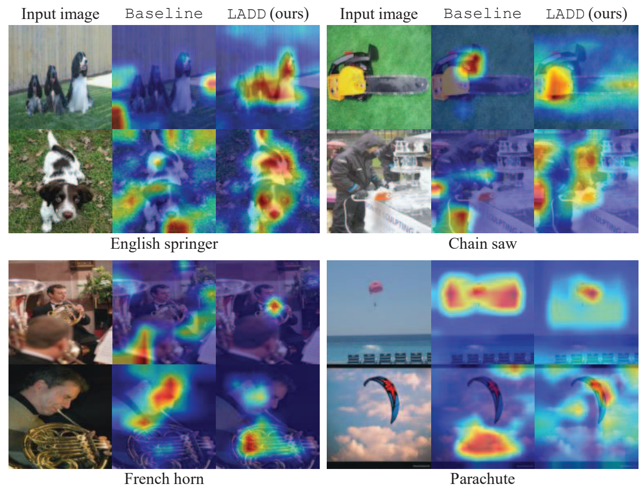


Figure 4. **Analysis on the Dataset Quality.** The second and third columns depict GradCAM [28] visualization of each prediction from GLaD(MTT) (baseline) and LADD-GLaD(MTT) (LADD), respectively.

4.4. Dataset Quality Analysis

We employ GradCAM [28] to visually investigate the reasons behind performance improvements from label augmentation. Fig. 4 displays the GradCAM results for GLaD(MTT) and LADD, both trained on ImageNette at 5 IPC. Our observations reveal that LADD more accurately identifies objects than the baseline, which often focuses

Images: Full	Images: Sub-sampled	Labels: Hard	Labels: Soft	ConvNetD5	VGG11	ResNet18	AlexNet	ViT	Avg.
✓		✓		58.7	45.5	50.6	37.0	29.4	44.2
✓			✓	60.1	44.5	51.2	37.7	28.8	44.5
✓		✓	✓	60.8	44.1	51.9	36.3	29.2	44.5
	✓	✓		54.3	49.7	49.5	37.3	29.6	44.1
	✓		✓	62.5	53.8	57.0	49.4	32.6	51.1
	✓	✓	✓	59.8	54.7	55.4	48.9	34.6	50.7
✓	✓	✓	✓	**66.5**	**55.7**	**61.2**	**50.2**	**35.9**	**53.9**

Table 6. **Performance Analysis on Image and Label Combinations.** `GLaD(MTT)` is set to the baseline model. All results are 5-CAE values measured on ImageNette at 5 IPC.

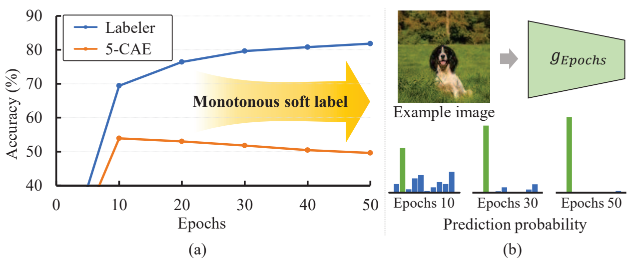


Figure 5. **Analysis on the Labeler g.** (a) The Blue line indicates the labeler performance. The orange line depicts the accuracy of the test model in the deployment stage where dense labels in the distilled dataset are obtained from the labeler of each epoch. (b) Each bar graph depicts the prediction probability of the example image using the labeler for each epoch.

on surroundings rather than primary objects. For example, `LADD` effectively concentrates on the main object, identifying all three English springers. Another shortcoming of the `baseline` is its tendency to detect only parts of an object, while `LADD` captures entire objects for accurate classification. Additionally, `LADD` excels at detecting small objects like a miniature French horn and a Parachute, outperforming the `baseline`. Overall, models trained with `LADD` classify objects with diverse features better, regardless of size, quantity, and structure. This demonstrates `LADD`'s ability to learn multiple representations of a single object using diverse dense labels with sub-images, significantly enhancing classification accuracy. Challenging categories like Chain saw, French horn, Gas pump, and Golf ball are difficult to classify (accuracies $\leq 40\%$) due to variations in size and quantity. `LADD` improves classification accuracies from 32%, 36%, 32%, and 40% to 56%, 60%, 40%, and 56%, respectively, marking up to a 24% improvement.

4.5. Ablation Study

The ablation study on `LADD-GLaD(MTT)` using the ImageNette at 5 IPC concentrates on identifying the ideal training steps for the labeler. The labeler creates soft labels that encapsulate meaningful information for specific sub-images. We evaluate the contribution of training labeler on the source dataset to the distilled dataset. Fig. 5 (a) displays the performance of labeler and `LADD` across various training epochs. Fig. 5 (b) shows that soft labels from less extensively trained labelers exhibit greater diversity (indicating less overconfidence) compared to those trained for longer periods. This occurs as, during initial training stages, the model primarily absorbs general information about the source dataset. Subsequently, the model begins to memorize the training data, leading to overconfident results. Consequently, we employ a labeler trained only for ten epochs, capitalizing on this early-stage learning.

5. Conclusion and Limitation

In this work, we highlight the overlooked role of labels in distilled datasets. Addressing this limitation, we introduce Label-Augmented Dataset Distillation (LADD), a method that effectively utilizes labels. Our approach enriches labels with useful information, orthogonal to the images. This yields three major advantages: (1) enhanced efficiency in distillation computation, (2) improved memory capacity efficiency, and (3) increased dataset robustness.

Extensive experiments demonstrate that LADD enhances various distillation methods with minimal extra computational and memory resources. On five ImageNet subsets and three baseline methods, LADD achieves an average performance improvement of 14.9% with only a 2.5% memory increase. Remarkably, LADD surpasses baselines with more images per class while using fewer computational resources and memory capacity. LADD with 5 IPC delivers 12.9% more accuracy than a 6 IPC baseline while using eight times less memory. We confirmed that datasets distilled using LADD enable more robust training across diverse architectures. Additionally, results from Grad-CAM [28] visualizations show that models trained with our dataset accurately and robustly capture object locations.

Acknowledgments

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the MSIP (NRF-2022R1A2C3011154, RS-2023-00219019), the IITP grant funded by the Korea government(MSIT) (RS-2019-II190075, Artificial Intelligence Graduate School Program(KAIST), No. RS-2024-00457882, AI Research Hub Project), and KEIT grant funded by the Korea government(MOTIE) (No. 2022-0-01045, No. 2022-0-00680). We thank Yunji Jung for her valuable contributions to the discussions and experiments.

References

[1] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020. 3

[2] 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, 3, 4, 5, 11, 12

[3] 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, 3, 5

[4] 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, 3, 5

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 5, 6, 11

[6] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012. 1

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 5

[8] 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. 12

[9] Yuxuan Duan, Jianfu Zhang, and Liqing Zhang. Dataset distillation in latent space. *arXiv preprint arXiv:2311.15547*, 2023. 1, 3

[10] Python Software Foundation. Python documentation: zipfile module, 2023. Accessed: 2023-07-14. 5

[11] 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

[12] Jack Goetz and Ambuj Tewari. Federated learning via synthetic data. *arXiv preprint arXiv:2008.04489*, 2020. 1

[13] Ziyao Guo, Kai Wang, George Cazenavette, HUI LI, Kaipeng Zhang, and Yang You. Towards lossless dataset distillation via difficulty-aligned trajectory matching. In *The Twelfth International Conference on Learning Representations*, 2023. 3, 4, 11, 12

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '16, pages 770–778. IEEE, June 2016. 5, 6, 11

[15] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 4

[16] Jeremy Howard. Imagewang. 1, 5

[17] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. In *International Conference on Machine Learning*, pages 11102–11118. PMLR, 2022. 1, 3

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 11

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 5

[20] Guang Li, Ren Togo, Takahiro Ogawa, and Miki Haseyama. Soft-label anonymous gastric x-ray image distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 305–309. IEEE, 2020. 1

[21] Noel Loo, Ramin Hasani, Mathias Lechner, and Daniela Rus. Dataset distillation with convexified implicit gradients. *arXiv preprint arXiv:2302.06755*, 2023. 1

[22] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. In *International Conference on Learning Representations*, 2021. 1

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 4

[24] 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

[25] Noveen Sachdeva and Julian McAuley. Data distillation: A survey. *Transactions on Machine Learning Research*, 2023. Survey Certification. 11

[26] 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. 3

[27] 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

[28] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 2, 7, 8

[29] Jiyuan Shen, Wenzhuo Yang, and Kwok-Yan Lam. Efficient dataset distillation through alignment with smooth and high-quality expert trajectories. *arXiv preprint arXiv:2310.10541*, 2023. 3, 5

[30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 5

[31] Ilia Sucholutsky and Matthias Schonlau. Secdd: Efficient and secure method for remotely training neural networks. *arXiv preprint arXiv:2009.09155*, 2020. 1

[32] Peng Sun, Bei Shi, Daiwei Yu, and Tao Lin. On the diversity and realism of distilled dataset: An efficient dataset distillation paradigm. *arXiv preprint arXiv:2312.03526*, 2023. 5, 6, 11

[33] Paul Vicol, Jonathan P Lorraine, Fabian Pedregosa, David Duvenaud, and Roger B Grosse. On implicit bias in overparameterized bilevel optimization. In *International Conference on Machine Learning*, pages 22234–22259. PMLR, 2022. 1

[34] 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. 1, 3

[35] 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. 3, 12

[36] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 3

[37] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *International Conference on Learning Representations*, 2020. 11

[38] 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. 3, 5, 6

[39] Hansong Zhang, Shikun Li, Pengju Wang, Dan Zeng, and Shiming Ge. Echo: Efficient dataset condensation by higher-order distribution alignment. *arXiv preprint arXiv:2312.15927*, 2023. 3

[40] Lei Zhang, Jie Zhang, Bowen Lei, Subhabrata Mukherjee, Xiang Pan, Bo Zhao, Caiwen Ding, Yao Li, and Dongkuan Xu. Accelerating dataset distillation via model augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11950–11959, 2023. 3

[41] 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, 12

[42] 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, 5, 12

[43] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. In *International Conference on Learning Representations*, 2021. 1, 3, 5, 12

[44] Ganlong Zhao, Guanbin Li, Yipeng Qin, and Yizhou Yu. Improved distribution matching for dataset condensation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7856–7865, 2023. 3

[45] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *Advances in Neural Information Processing Systems*, 35:9813–9827, 2022. 1, 3, 12

[46] Yanlin Zhou, George Pu, Xiyao Ma, Xiaolin Li, and Dapeng Wu. Distilled one-shot federated learning. *arXiv preprint arXiv:2009.07999*, 2020. 1