Rethinking Data Distillation: Do Not Overlook Calibration

Dongyao Zhu
University of California, San Diego
doz022@ucsd.edu

Bowen Lei
Texas A&M University
bowenlei@stat.tamu.edu

Jie Zhang
Zhejiang University
zj_zhangjie@zju.edu.cn

Yanbo Fang
Certik
yanbo.fang@certik.com

Yiqun Xie
University of Maryland, College Park
xie@umd.edu

Ruqi Zhang
Purdue University
ruqiz@purdue.edu

Dongkuan Xu*
North Carolina State University
dxu27@ncsu.edu

Abstract

Neural networks trained on distilled data often produce over-confident output and require correction by calibration methods. Existing calibration methods such as temperature scaling and mixup work well for networks trained on original large-scale data. However, we find that these methods fail to calibrate networks trained on data distilled from large source datasets. In this paper, we show that distilled data lead to networks that are not calibratable due to (i) a more concentrated distribution of the maximum logits and (ii) the loss of information that is semantically meaningful but unrelated to classification tasks. To address this problem, we propose[1] Masked Temperature Scaling (MTS) and Masked Distillation Training (MDT) which mitigate the limitations of distilled data and achieve better calibration results while maintaining the efficiency of dataset distillation.

1. Introduction

Dataset distillation (DD) has recently gained growing attention because of its ability to reduce the need for large amounts of data during deep neural network (DNN) training, thereby reducing training time and storage burden [40]. Despite the efficiency of training, studies have pointed out that DD still has multiple limitations. On the one hand, the distillation process is found to be time-consuming, computationally expensive, and storage intensive [40, 53, 52, 7, 27, 28, 14, 49]. On the other hand, DNNs trained on DD data are said to be poorly generalizable to different models or downstream tasks [40, 53, 52]. Efforts have been conducted to address these issues [3, 50, 24]. However, the calibration of DD has been overlooked, which is important for deploying DD safely in real-world applications.

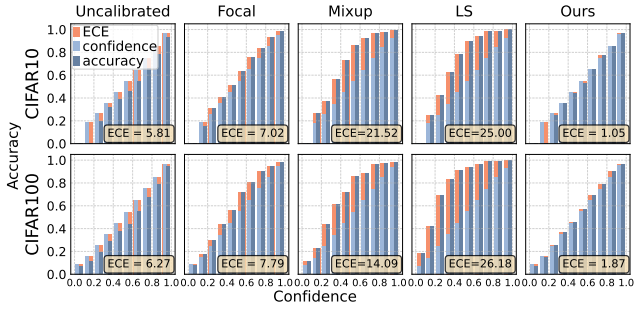


Figure 1. ECE (red area, smaller is better) of different calibrations on an over-confident ConvNet trained on MTT [3] distilled CIFAR10 and CIFAR100. Our proposed techniques achieve the best calibration results compared to the over-calibration of other methods. Focal: Focal loss. LS: Label Smoothing.

An increasing number of studies are investigating calibration as an important property of DNNs, which means that a DNN should know when it is likely to be wrong [10, 26, 1]. In other words, the confidence (probability related to the predicted category label) of a model should reflect its ground truth correctness likelihood (accuracy). Previous work has found that DNNs are often too confident to realize when they are making mistakes [10, 30], which leads to safety issues, especially in safety-critical tasks, e.g., automated healthcare and self-driving cars [6, 32].

We for the *first* time identify and study the calibration problem of DNNs trained on distilled data (DDNNs).

Problem 1. *We find that DDNNs still suffer from over-confidence problem.*

We evaluate the calibration quality of DDNNs by Ex-

[1]Code available at https://github.com/DongyaoZhu/calibrate-networks-trained-on-distilled-datasets

pected Calibration Error (ECE) [10], which is a common metric to quantitatively measure the difference between confidence and accuracy. Specifically, to calculate the ECE, we categorize the output probability and accuracy into different levels and calculate the average absolute difference. The lower the ECE, the better the calibration. As shown in Figure 1, the ECE (red area) of DDNNs is quite visible in the figures of the first column, which means that the probability of DDNNs' output is usually higher than the actual accuracy of its prediction. Thus, it is desirable to calibrate DDNNs for reliable prediction and decision-making.

Problem 2. *We find that DDNNs are not calibratable when using existing calibration methods.*

There are calibration methods designed to align the confidence and accuracy of DNNs trained on full datasets (FDNNs). They either modify loss term during network training [21], use soft labels [47, 36], or scale down the logits after training [10]. However, when training on distilled data, we find that most of the existing methods tend to over-calibrate DDNNs. As shown in Figure 1, a DDNN trained on distilled CIFAR10 (the first column) has an initial ECE of 6.17% (red area). After calibrating with focal loss (the second column), mixup (the third column), or label smoothing (the fourth column), the DDNN becomes under-confident with increased ECE of 7.79%, 14.09%, and 26.18% respectively, as shown by the inverted and enlarged red bars. This over-calibration problem also occurs for various distillation methods on common datasets (Table 1).

In order to address the issues mentioned above, we raise the following questions:

Question 1. *Why are DDNNs not calibratable when using existing calibration methods?*

We first dive deep into the differences between the source full data and the distilled data. We find that the distilled data tend to retain information relevant to the classification task while discarding other distributional information in the full data, which may result in limiting DDNNs to pursuing higher accuracy in the classification task while losing more abilities in latent representation learning of FDNNs [37, 29]. By decomposing distilled and full data into smaller components and studying their corresponding significance to model training accuracy, we show that distilled data contains very condensed information, implying a loss of information and leading to harder during-training calibration. Then, we also investigate the differences between DDNNs and FDNNs. We observe that DDNNs have a more concentrated distribution of logit values, leading to less room for after-training calibration methods such as temperature scaling.

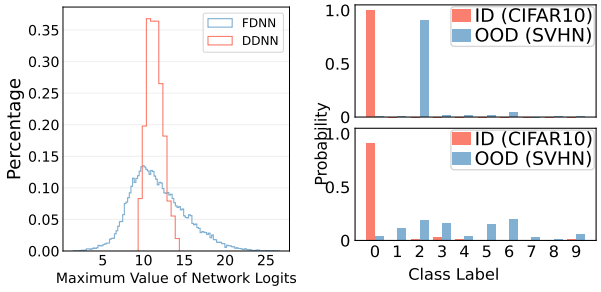
Question 2. *How to calibrate DDNNs efficiently?*

To enable DDNNs to be calibratable, we propose (i) Mask Temperature Scaling and (ii) Masked Distillation Training that can be applied both during and after the training of DDNNs. We design a binary masking method for synthetic input when training for distillation objection, which effectively forces the distillation model to extract richer information from the source dataset into distilled datasets, leading to better encoding abilities and thus better calibration of DDNNs. We also show that our proposed masked temperature scaling better improves after-training calibration results on DDNNs by introducing more dynamics to network outputs. Our proposed techniques thus allow for more powerful and more calibratable DDNNs. We summarize contributions as follows:

- We for the *first* time study the calibration of DDNNs and find that DDNNs are not calibratable.
- We find that DD discards semantically meaningful information and that DDNNs produce a concentrated logit distribution, which explains the difficulty of calibrating DDNNs.
- We propose two masking techniques that can improve the calibration of DDNNs better than existing calibration methods, i.e., masked distillation training and masked temperature scaling. In addition, our proposed techniques can be readily deployed in existing dataset distillation methods with minimal extra cost.
- We perform extensive experiments on multiple benchmark datasets, model architectures, and data distillation methods. Our techniques reduce ECE values by up to **91.05%** with comparable accuracy.

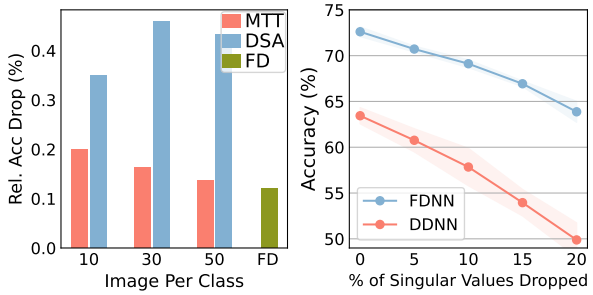
2. Related Work

Dataset Distillation. First introduced by [40], dataset distillation is the task of synthesizing a smaller dataset from a large-scale dataset such as CIFAR100 [17], so that the network trained on the distilled data has a performance comparable to that of the network trained on the source large-scale data. Recent work has significantly improved the performance of networks trained on distilled data and reduced the computational and time overhead of the distillation process while compressing the dataset size to one image per class [3, 7, 24, 27, 28, 53, 52, 39, 48]. Dataset distillation problem is treated as a gradient-based hyperparameter optimization [40]. DC performs distillation by matching the gradients generated from distilled data and full data [53]. DSA further improves the results by differentiable Siamese augmentations [52]. Other SOTA methods include matching trajectories of each parameter between the training on distilled data and full data [3], optimizing soft labels [35], minimizing reconstruction errors [45], and using neural networks to regress features from synthetic samples to real ones [54]. The current focus of DD is on computational expense and training performance, and to



(a) Maximum logits produced by DDNNs and FDNNs. (b) Prob. of DDNN (top) vs. Ours (bottom) on ID / OOD samples.

Figure 2. Left: The more calibratable FDNN outputs more evenly distributed logits, while the less calibratable DDNN outputs a more concentrated logit distribution. Top-Right: The less calibratable DDNN struggles to distinguish between an in-distribution (ID) and an out-of-distribution (OOD) sample using its max logits.



(a) DDNNs lose more accuracy than FDNNs across DD settings. (b) Trend of accuracy loss vs. dropping singular values (MTT).

Figure 3. Effects on model accuracy of discarding major singular values during SVD reconstruction of distilled and full CIFAR10. DDNNs suffer from more accuracy drop as we discard more singular values during reconstruction, indicating that distilled data contains more condensed information that can be easily grouped by a simple SVD decomposition. Right: IPC = 10.

the best of our knowledge, the difficulties in calibrating over-confident DDNNs remain untouched.

Neural Network Calibration. The importance of neural network calibration has been emphasized and received increasing attention [10], with the aim of matching the output probability of a neural network (also known as the network output confidence) with the actual accuracy. [10] also introduces the concept of Expected Calibration Error (ECE), which has now become a standard metric for quantitatively measuring calibration quality. A higher ECE implies a poorer calibration of the neural network, while a 0 implies a perfect calibration. Recent calibration methods that have been proposed for networks trained on large-scale datasets include Label Smoothing (LS) [46], which smooths a one-hot class label with uniform noise during training, forcing the model to learn loose predictions. Mixup is similar to label smoothing, where different data-label pairs are mixed to form new data points [36, 47]. Focal loss (FL), originally designed to address the class imbalance, modifies the traditional cross-entropy loss in classification problems by adding a moderation term, thus allowing the model to focus more on difficult examples that are easily misclassified but difficult to learn [21, 25]. Temperature scaling (TS) is an after-training calibration method applied to fully trained and fixed-weight networks [10]. As an extension of Platt scaling [31], the temperature scaling method scales the output, denoted by z, of the last layer of the network with a scaler T before converting it into a probability:

$$\hat{q}_i = \max_k \sigma_{softmax}\left(z_i/T\right)^{(k)} \; z_i \in \mathbb{R}^D. \tag{1}$$

Other work has discussed the necessity [42] and hardness of network calibration [5, 9, 51], as well as the degradation of calibration with distribution shift or model size [16, 20].

3. Limitation Analysis of DDNNs' Calibration

We focus on the difficulties of calibrating over-confident DDNNs. As shown in the first column of Figure 1 and the raw ECE reported in Table 1, DDNNs show the common over-confidence problem of neural networks, giving higher probabilities than actual accuracy; however, when applied with existing calibration methods, DDNNs are often over-calibrated and become under-confident. In this section, we analyze the reasons that may account for the DDNNs that are not calibratable from 2 aspects: (i) the after-training prediction behaviors and (ii) the during-training network capacity in terms of feature encoding ability. We also discuss the decomposed significance of full data and distilled data on the training accuracy of the network.

3.1. DDNNs are Less Calibratable

We find that the logit distribution of the DDNNs' output is more concentrated, making it difficult to calibrate. In general, a neural network can be considered as a mapping function from the source data domain to the target label distribution, and in the classification task, we use the softmax function to convert logits into label probabilities. The higher the maximum logit value compared to other values, the higher the argmax probability will be and thus the more likely such prediction is over-confident. Therefore we study the distribution of maximum logit values for fully trained DDNNs and FDNNs. As shown in Figure 2(a), the more calibratable FDNNs (blue) output a more dispersed logit distribution, while the less calibratable DDNNs (red) output a concentrated logit distribution with a larger mean.

This mismatched behavior causes problems for after-training calibration methods such as TS and Mixup that operate on scaling output logits, because DDNNs with tight distributions of max logits struggle to distinguish between hard (e.g., out-of-distribution, OOD) and easy (e.g., in-distribution, ID) samples (top of Figure 2(b)) using the cor-

responding max logits [41]. Similar theoretical assumptions appear in recent work [38], where they show that the small range of logits due to regularization during training, and a large mean of logits due to the network trying to fit on hard examples may lose information about the different hardness of data points, causing the networks of after-training calibration methods to fail to calibrate.

Therefore, we infer that DDNNs are less calibratable using distilled data due to their more concentrated output distribution and larger mean values. Thus, in order to make DDNNs more calibratable in after-training calibration without modifying network weights, we aim to utilize data that force DDNNs to produce more diverse and smaller outputs.

3.2. DD Contains Limited Semantic Information

By reconstructing distilled and full data with SVD, we find that distilled data contains only condensed information about the classification task, resulting in the limited ability of DDNNs in latent representation learning. Intuitively, distilled images should be more informative, or more representative than source full images, in order to keep the number of images small. But do distilled images discard too much source information that is not so much useful for the classification tasks they are optimized for? We hypothesize that distilled data is "simpler" than source full data, such that dropping the same amount of information from distilled datasets should hurt the training performance worse than it does on full data. We start by breaking down full datasets into smaller components of different significance. Singular value decomposition (SVD) [15] is a powerful algorithm in linear algebra for matrix approximation:

$$U, \Sigma, V = \mathrm{SVD}(X), \tag{2}$$

where higher singular values in Σ correspond to more significant components of X. Source data can then be approximately reconstructed by

$$X' \approx U \cdot \Sigma' \cdot V^T \tag{3}$$

SVD has been widely used in DNN research for model reconstruction [43, 44], knowledge distillation [19], and analyzing data [11]. For our purposes of analyzing data information diversity and significance of data components, we gradually throw away the highest singular values during SVD reconstruction and check for accuracy drop when trained on the approximately reconstructed data.

Our assumption is that distilled data contains dense information that can be easily grouped, such that SVD decomposes distilled data into several important components and other very small components, compared to full data components whose importance can be more evenly distributed, such that dropping the same number of important components from distilled data would lose more accuracy than in full data. We drop from 0% to 20% of the highest singular values from full CIFAR10 and CIFAR10 distilled by [3] with IPC = 10, 30 and 50, then train a ConvNet for 300 epochs on the resulting data. As shown in Figure 3, DDNNs suffer much more severely from the loss of principle components than FDNNs.

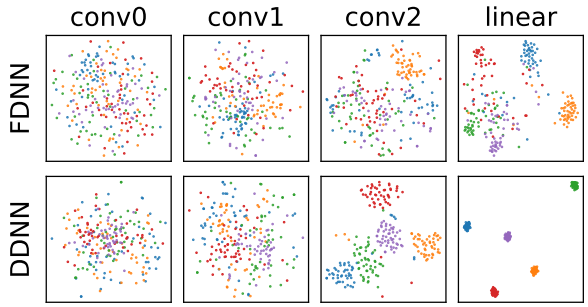


Figure 4. T-SNE projections of feature vectors from each layer of a 4-block ConvNet trained with Mixup on distilled and full data. FDNNs better encode source information as visualized by the rich features not separated until the last layer. DDNN poorly encodes source information, as shown by the feature projections already separated in layer conv2.

We thus conclude that the distilled data discards too much other semantically meaningful information from the original full data due to over-optimization of the classification task, resulting in condensed information that can be easily decomposed by SVD.

3.3. Limited Semantic Information Weakens Encoding Capacity

We further infer that DDNNs may be less capable of tasks other than classification due to the likely loss of non-classification information. Usually, outputs of intermediate layers of DNNs could be used as feature vectors for other interesting non-classification tasks such as style transfer [8, 13] due to their unique encodings of source information. To see this, we visualize outputs of layers at different depths in the ConvNet using t-SNE that projects feature vectors down to 2 dimensions. We can see that in Figure 4, features from FDNNs cluster slowly, and become visually separable only in the last layer, thus retaining most of the original information in its latent vectors; features from DDNNS, however, form visible cluster already in layer conv2, making more compact final clusters that are more valuable for classifications than other tasks such as feature extraction. Moreover, clusters of DDNNs from each class are closer to each other than those from FDNNs. Clearly, outputs of middle layers from DDNNs are already alike label distributions, discarding too much non-classification information. Similar observations on other distillation backbones are reported in [39], in which they also account for long-tailed gradients as possible reasons.

Table 1. ECE (%) of different calibration methods on DDNNs with different DD backbones. Our proposed method yields the best or comparable ECE results in all distillation settings, reducing ECE of DSA by 91.05%. More importantly, our method does not over-calibrate DDNNs as other calibration methods do, as shown in the last row. Although MX and LS outperform our method in distillation backbones of inferior accuracy (RTP, DC), we surpass them by fine-tuning a more aggressive masking ratio, as described in Section 6.

DD Backbone		Raw	TS	MX	LS	FL	Ours
MTT	CIFAR10	4.93 ± 0.2	*7.45 ± 2.1*	*21.06 ± 0.8*	*25.01 ± 0.2*	*6.62 ± 0.2*	**1.20 ± 0.3**
	CIFAR100	5.95 ± 0.4	*7.76 ± 0.4*	*14.19 ± 0.4*	*26.36 ± 0.4*	*8.30 ± 0.5*	**2.18 ± 0.2**
	Tiny ImageNet	15.78 ± 0.3	2.44 ± 0.3	2.42 ± 0.3	12.14 ± 0.3	3.61 ± 0.3	**2.26 ± 0.3**
	ImageNette	8.68 ± 1.9	4.85 ± 0.6	5.19 ± 0.6	*23.45 ± 1.4*	6.87 ± 1.3	**4.78 ± 0.5**
RTP	CIFAR10	2.96 ± 0.5	*3.28 ± 0.7*	*13.35 ± 1.5*	*9.58 ± 0.5*	*8.35 ± 1.4*	**2.22 ± 0.5**
	CIFAR100	29.71 ± 0.6	23.72 ± 0.6	**3.55 ± 0.6**	7.94 ± 0.2	18.51 ± 0.5	10.14 ± 0.4
DC	CIFAR10	23.60 ± 0.7	5.00 ± 0.7	1.83 ± 0.3	**1.28 ± 0.1**	13.31 ± 0.9	10.39 ± 0.8
DSA	CIFAR10	19.91 ± 0.3	1.95 ± 0.4	6.44 ± 0.8	2.32 ± 0.5	7.95 ± 0.7	**1.70 ± 0.4**
# over-calibration		-	3	3	4	3	**0**

Therefore, DDNNs do not exhibit good encoding capability due to being trained on distilled data optimized specifically for the classification task and may be susceptible to being over-calibrated by calibration methods. We provide more details in the supplementary material.

4. Our Proposed Techniques

We respond to the analyses in Sections 3.1-3.3 so that our method can be applied during and after training, providing calibration options at different times and computational budget levels.

4.1. Masked Temperature Scaling

As discussed in Section 3.1, compared to FDNNs, DDNNs produce a more concentrated distribution of logit values with larger values, and these large and condensed logit values lead to networks that are not calibratable. Since after-training calibration methods such as temperature scaling [10] make use of these large and concentrated logit values from a forward pass of validation data, we seek to overcome this source of difficulty in calibration by perturbing the validation data such that the model could output more various and smaller logit values. Inspired by dropout [34], we apply a simple zero-masking on the validation data of temperature scaling. Our proposed method, which we refer to as Masked Temperature Scaling (MTS), thus modifies Eq (1) as follows:

$$\hat{q}_i = \max_k \sigma_{softmax} \left(z_i * mask / T \right)^{(k)}, \tag{4}$$

where q, z, mask $\in R^D$ and the number of zeros in the mask is controlled by a hyperparameter masking ratio r. Note that masking is only applied when updating T, such that MTS does not change model accuracy. We use a sampled portion of the training data we have to update the temperature parameter T, instead of using separate validation data as in traditional temperature scaling.

Algorithm 1: Masked Distillation Training

Input: Source training data $\mathcal{T}$, number of classes N_c, deep neural network ψ_θ parameterized with θ, criterion C, loss function l, total number of training steps T, masking ratio r
Output: Distilled dataset $\mathcal{S}$

for $t \leftarrow 0$ **to** T **do**
  custom pre-processing
  for $c \leftarrow 0$ **to** N_c **do**
    Sample $T_c \sim \mathcal{T}$, $S_c \sim \mathcal{S}$
    Update synthetic data $\mathcal{S}_c$:
    $S_c \leftarrow S_c - \lambda \nabla_{S_c} C\left(S_c, Mask\left(T_c, r\right), l, \theta\right)$
  end
  custom post-processing
  Update θ of network ψ_θ using $T \sim \mathcal{T}$
end

This is particularly necessary in the dataset refinement setting, as we may simply not have any extra data, for example, when each class of images is set to 1 (see Section 6 for more details).

4.2. Masked Distillation Training

In response to the analyses in Sections 3.1-3.3, we avoid over-concentration of distillation data on easily identifiable information in the source complete data by perturbing the binary mask during distillation, so that the distillation data also contain more semantically complete information.

A typical DD training paradigm tries to minimize the differences of certain characteristics, as measured by some criterion C, between data batch B' from synthetic data and data batch B from source full data. The loss function $l\left(\theta; X\right)$ used in C is usually the cross entropy loss for training θ on x in classification tasks. We thus put the binary mask on synthetic data before feeding it into C. We

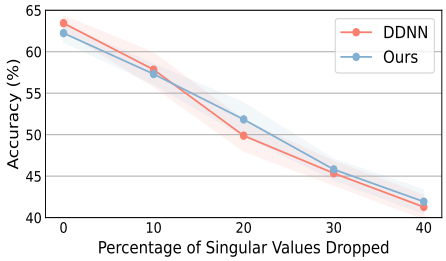


Figure 5. Model trained on MDT (Ours) distilled data suffers from less accuracy drop than the model (DDNN) trained on MTT distilled data when dropping major singular values during SVD reconstruction of DD, showing that ours alleviates the issue of condensed information of distilled data.

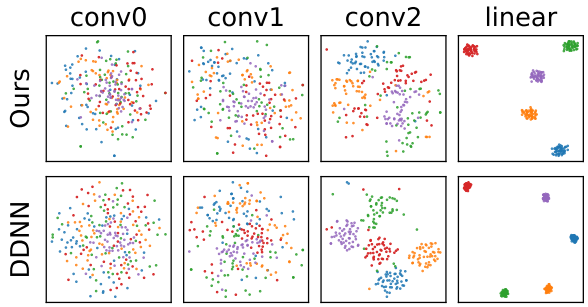


Figure 6. T-SNE projections of feature vectors from each layer of a 4-block ConvNet trained on original distilled data and Ours. The model trained on Ours better encodes source information than the original DDNN, as visualized by the features that are hardly separated in layer conv2.

give the details of our method, Masked Distillation Training (MDT), in Algorithm 1. MDT is applicable to various distillation backbones. For instance, in Efficient Dataset Distillation [50], they set the criterion C as the differences between gradients back-propagated from l given source data B and distilled data B':

$$C\left(B, B'; l, \theta\right) = \left\|\nabla_\theta \ell(\theta; B) - \nabla_\theta \ell\left(\theta; B'\right)\right\| \quad (5)$$

When applied with MDT, this now becomes:

$$C\left(B, B'; l, \theta\right) = \left\|\nabla_\theta \ell(\theta; B) - \nabla_\theta \ell\left(\theta; Mask\left(B', r\right)\right)\right\| \quad (6)$$

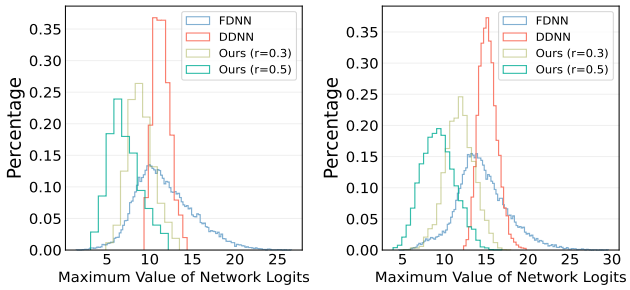
In another distillation backbone MTT [3], the criterion C measures parameter trajectory differences between DDNNs and FDNNs:

$$\begin{aligned} \hat{\theta}_{t+N} &\leftarrow \hat{\theta}_{t+N-1} - \lambda \nabla l\left(B', \theta_{t+N-1}\right) \\ C\left(B', \hat{\theta}; l, \theta\right) &= \left\|\hat{\theta}_{t+N} - \theta^*_{t+M}\right\|_2^2 / \left\|\theta^*_t - \theta^*_{t+M}\right\|_2^2 \end{aligned} \quad (7)$$

and when applied with MDT, this becomes:

$$\hat{\theta}_{t+N} \leftarrow \hat{\theta}_{t+N-1} - \lambda \nabla l\left(Mask\left(B', r\right), \theta_{t+N-1}\right). \quad (8)$$

We find that a masking ratio of 10% works well for our purposes while losing minimal test accuracy, and we provide more details in Sections 5.3 and 5.5.



(a) Maximum Logits on CIFAR10 (b) Maximum Logits on CIFAR100

Figure 7. Histogram of maximum logits of DDNNs (Ours) on CIFAR10 and CIFAR100. As we increase the ratio of masking in our method, DDNNs produce logits that cover more values, thus becoming more calibratable by after-training calibration methods.

4.3. Connection to Dropouts

Dropout [34] is a common practice to prevent overfitting of neural networks. Two popular types are unit dropout (U-DP) and weight dropout (W-DP), which randomly discard units (neurons) and individual weights at each training step, respectively. The formulas are shown in Eq (9).

$$\text{U-DP: } Y = (X \odot M)W; \quad \text{W-DP: } Y = X(W \odot M), \quad (9)$$

where M denotes dropout mask and W refers to weights.

Our proposed Masked Distillation Training can be viewed as a new version of dropout on the input, i.e., $X = S_c \odot M$. There are practices using dropout on inputs as data augmentation [2]. In contrast to existing efforts, we apply masking in distillation backbones on synthetic data during their forward passes. Masking some of the synthetic data makes it harder to collect easily reachable information from the source dataset, and thus forces the distillation to focus on other structurally and semantically meaningful information that has not received sufficient attention in previous data distillation.

5. Experiments

5.1. Experiment Setup

We thoroughly evaluate our proposed MTS and MDT on different dataset distillation setups and compare them with existing calibration methods. **Dataset Distillation Backbones:** We follow the exact settings in MTT [3], RTP [7], DC [53] and DSA [52]. Our experiments are based on 4 benchmark datasets: CIFAR10 & CIFAR100 [17], Tiny ImageNet [18], and ImageNette (a subset of ImageNet) [12]. We mainly set image-per-class to larger values, e.g. 50 in MTT, and results on different IPCs are provided in supplement materials. **Calibration Methods:** We compare our method with existing calibration methods including Temperature Scaling (TS) [10], mixup (MX) [47], Label Smoothing (LS) [46], and Focal Loss (FL) [21]. **Implementations:** For TS, we use an initial temperature of 1.5

Table 2. ECE (%) of different calibration methods on distilled datasets trained with our methods. Tiny: Tiny ImageNet. Nette: Nette subset of ImageNet. Our results are in shadow.

Dataset	Best of Others	MDT	MTS	MDT + MTS
CIFAR10	3.64 ± 0.2 (TS)	3.66 ± 0.3	$\mathbf{1.20 \pm 0.3}$	2.50 ± 0.5
CIFAR100	5.95 ± 0.4 (Raw)	4.65 ± 0.3	2.18 ± 0.2	$\mathbf{2.00 \pm 0.5}$
Tiny	2.42 ± 0.3 (MX)	7.44 ± 1.4	$\mathbf{2.26 \pm 0.3}$	5.91 ± 1.4
Nette	4.85 ± 0.6 (TS)	7.32 ± 1.7	$\mathbf{4.78 \pm 0.5}$	5.14 ± 1.2

Table 3. ECE (%) of MDT with dynamically sampled r and MTS on CIFAR10, MTT with different IPCs.

IPC	MDT^{ds}	MTS	MDT^{ds} + MTS
10	1.79 ± 0.9	1.36 ± 0.4	$\mathbf{1.13 \pm 0.2}$
50	5.10 ± 0.4	1.20 ± 0.3	$\mathbf{1.26 \pm 0.2}$

and LBFGS [22] optimizer with a learning rate of 0.02. For MX, we use a β distribution with $\alpha = 1.0$ for the mixup ratio. For LS, we set $\epsilon = 0.1$. For FL, we set $\gamma = 1.$, which calibrates better on DDNNs than the best value 2 reported in the paper. For our proposed MTS, on distillation backbone MTT, we use a fixed masking ratio of 0.3, 0.3, 0.5, and 0.1 for each of the 4 datasets respectively. On backbones RTP, DSA, and DC, due to their inferior performance in accuracy, we use a more aggressive masking ratio of 0.8.

Since the number of examples of distillation data is usually limited, we draw 10% of all distillation data as the validation set for the after-training method, as in other existing work. The experiments are repeated five times, and the mean and standard deviation are reported. More experimental setups are available in supplementary materials.

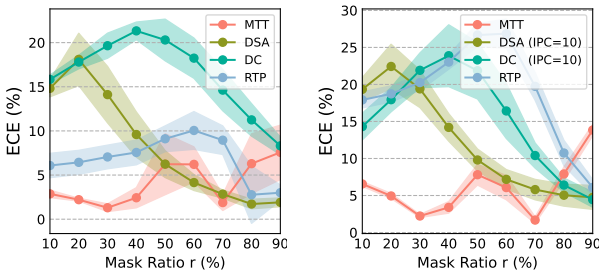
5.2. Empirical Analysis of MTS

We show in Figure 1 that our proposed method is able to reduce the ECE (red bars) to almost zero for each confidence bin when using MTT as the distillation backbone on CIFAR10 and CIFAR100. Although traditional calibration methods such as mixup *can* perform well, they could also over-calibrate and result in under-confident networks. We visualize the under-confidence in Figure 1, in which the red bars are enlarged and switched from left to right in each bin. Additional calibration results are reported in Table 1, and our proposed MTS gives the best numerical ECE results in almost all settings. In real-world settings where no mistakes are allowed, traditional methods are regarded as unsafe due to their potential over-calibration.

As a contrast, we propose masked temperature scaling, which not only has better performance but does not show any lack of confidence in the results at all and is therefore considered a safer choice.

5.3. Empirical Analysis of MDT

We show that MDT improves the calibration results. As reported in Tables 2-3, applying MDT alone or combining



(a) CIFAR10 (b) CIFAR100

Figure 8. Effects on ECE (%) of different masking ratios r in our method. For DD methods with better performance reported (MTT, RTP on CIFAR10), our method is robust to r and saves efforts in fine-tuning. For DD methods with inferior performance (DC, DSA, RTP on CIFAR100), a more aggressive masking ratio ($r > 0.7$) could still calibrate reasonably well.

it with MTS yields comparable or better calibration performance. WE note that naively combining MDT + MTS may increase ECE due to DDNNs overfitting to the fixed masking ratio in MDT, then being over-calibrated by MTS. Thus we further improve (Table 3) MDT + MTS by dynamically sampling the r in MDT from 0 to 0.1 (denoted MDT^{ds}) so the resulting DDNNs are more calibratable. We fix r in MTS due to the limited amount of validation data in DD. This indicates that our proposed MDT produces more robust and calibratable DDNNs than the original backbone when sufficient computational resources are available to train the distillation process from the beginning. We use MTT as the distillation backbone.

We find that MDT gives comparable model accuracy albeit altering the distillation process. With a 10% zero masking during the distillation process, MDT only leads to a loss of as large as 1.26% in DDNNs' accuracy on CIFAR100 and as low as 0.14% on Tiny ImageNet. As reported in Table 4, this is even better than traditional during-training calibration methods such as mixup, label smoothing, and focal loss that lead to different model results.

This suggests that MDT yields better calibration potential at a negligible performance cost, which is desirable in an environment where security is a major concern [6, 32].

5.4. Enabling Calibratable DDNNs

In response to the discussion of the after-training behavior of DDNNs in Section 3.1, we examined improvements in DDNN calibrability. On the validation data for Temperature Scaling, we apply zero-masking with ratio r = 10%, 20%, and 30% to see its effects on resulting logit distributions of DDNNs. We show in the bottom-right of Figure 2(b) that our MDT produces lower probabilities on OOD samples, leading to more distinguishable logits and more calibratable DDNNs than before. We also show in Figure 7 that DDNNs given these mask-perturbed data will produce

Table 4. Accuracy (%) of different during-training calibration methods on MTT distilled datasets. While all during-training calibration methods lead to a loss in accuracy, ours loses only as small as 0.14% at a masking ratio of 10%. Our results are in shadow.

Dataset	Raw	MX	LS	FL	Ours
CIFAR10	70.48 ± 0.2	65.50 ± 0.5	67.42 ± 0.5	68.79 ± 0.5	**69.98** ± 0.4
CIFAR100	47.47 ± 0.2	39.65 ± 0.3	47.02 ± 0.2	46.79 ± 0.4	46.21 ± 0.4
Tiny ImageNet	27.76 ± 0.2	21.48 ± 0.4	25.76 ± 0.3	27.42 ± 0.3	**27.62** ± 0.4
ImageNette	63.04 ± 1.3	55.60 ± 1.0	63.40 ± 0.9	61.32 ± 0.9	62.80 ± 1.2

similarly diverse logits as if they are processing normal full data, allowing masked temperature scaling to better calibrate DDNNs with similar good performance on FDNNs.

5.5. Enhancing Semantic Information of DDNNs

We investigate whether the semantic information of DD is enhanced according to the discussion in Section 3.2. As shown in Figure 5, when trained with MDT, our DDNNs start with a little lower accuracy than the normal MTT model. However, as we gradually drop more singular values following Eqs (2)-(3), the accuracy of the MDT model drops slower and even stays higher than the accuracy of the MTT models. This indicates that MDT distillation effectively retains more semantically meaningful information than normal distillation does, making MDT distilled data more difficult to be decomposed by SVD.

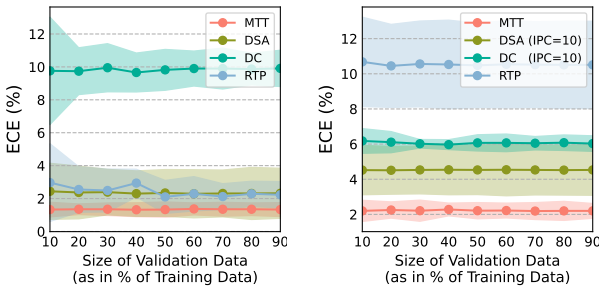
5.6. Improving Encoding Capacity of DDNNs

We study the improvement in the feature encoding capability of the DDNNs, responding to the discussion of DDNN behavior during training in Section 3.3. We experiment on the distillation backbone MTT, in which they collect network parameter trajectories from training on synthetic data in each iteration. We apply masked distillation training with masking ratio = 10% on the synthetic data before the forward pass in each iteration.

We show in Figure 6 that the hidden layers in the MDT model form larger clusters than the original MTT model, and that the clusters in each category are more intertwined with each other, retaining more information from the complete dataset and forming better feature vectors as desired.

6. Ablation Studies

Analysis of Mask Ratio r in MTS. We analyze the effects of mask ratio r on the calibration results of MTS. We set r from 0.1 to 0.9, increasing by 0.1. As shown in Figure 8, on CIFAR10 and CIFAR100, MTS works well for most of the possible r ranging from 0.1 to 0.5, indicating that MTS can be tuned with minimal effort. On variants of the ImageNet dataset, however, we find that 0.3 works best for Tiny ImageNet, and 0.5 for ImageNet Subset. This is probably due to the large number of classes in these more complex datasets, as well as the relatively low accuracy of their corresponding distilled datasets.



(a) CIFAR10 (b) CIFAR100

Figure 9. Effects on ECE (%) of different sizes of the validation data (N, as in % of total training data) in our method. On different distillation backbones, a small N gives identical calibration performance to a larger N, indicating that our method is also applicable in scenarios with extremely scarce validation data.

Table 5. ECE (%) of different calibration methods with IPC=1. Under this extreme compression rate, our method still outperforms other calibration methods. Our results are in shadow.

Dataset	Raw	MX	LS	FL	Ours
CIFAR10	10.15 ± 1.2	8.40 ± 1.1	12.79 ± 0.6	2.05 ± 0.9	**1.81 ± 0.7**
CIFAR100	2.46 ± 0.6	4.45 ± 0.5	8.89 ± 0.6	3.24 ± 0.9	**2.19 ± 0.5**

Analysis of Validation Set Size for MTS. We study the impact of how much data is drawn from all the distilled data as validation data for MTS. We denote N as the proportion we sample, and we set N from 10% to 50%, increased by 10%. We can see in Figure 9 that the number of samples has little effect on calibration results. We hypothesize that this is due to we only update the temperature parameter T for only one step, thus not being affected by the number of examples in this step. This also indicates that MTS can be applied when we have only a small amount of data available, such as distillation with IPC=1 or medical image analysis scenarios [23, 4, 33].

Calibration in Lower Accuracy Settings. Our method outperforms other calibration methods at DD settings with extreme compression ratio, i.e. only 1 synthetic image for each label. This means traditional temperature scaling no longer applies because it requires additional validation data. As reported in Table 5, while other calibration methods over-calibrate or don't work at all, ours still produces better results, indicating its generality to various DD settings.

7. Conclusion

In this paper, we find for the first time that networks trained on distillation data are not calibratable and have poor encoding ability because the distillation process focuses on the classification task while discarding other semantically meaningful information. Our proposed methods, namely Masked Distillation Training during training and Masked Temperature Scaling after training, effectively alleviate these limitations and make the DDNNs recalibrated.

In future work, we will look for better distillation methods that retain most of the source information and lead directly to calibratable networks. In addition, beyond calibrating DDNNs on in-distribution data, we will rethink DDNNs in terms of more general reliability, i.e., out-of-distribution detection, robust generalization, and adaptation, which are important properties for the safety of DDNN applications.

References

[1] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U Rajendra Acharya, et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. *Information Fusion*, 76:243–297, 2021. 1

[2] Xavier Bouthillier, Kishore Konda, Pascal Vincent, and Roland Memisevic. Dropout as data augmentation. *arXiv preprint arXiv:1506.08700*, 2015. 6

[3] 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, 4, 6

[4] Xuxin Chen, Ximin Wang, Ke Zhang, Kar-Ming Fung, Theresa C Thai, Kathleen Moore, Robert S Mannel, Hong Liu, Bin Zheng, and Yuchen Qiu. Recent advances and clinical applications of deep learning in medical image analysis. *Medical Image Analysis*, page 102444, 2022. 8

[5] Jiacheng Cheng and Nuno Vasconcelos. Calibrating deep neural networks by pairwise constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13709–13718, 2022. 3

[6] Lavindra De Silva and Alan Mycroft. Toward trustworthy programming for autonomous concurrent systems. *AI & SOCIETY*, pages 1–3, 2022. 1, 7

[7] Zhiwei Deng and Olga Russakovsky. Remember the past: Distilling datasets into addressable memories for neural networks. In *Neural Information Processing Systems (NeurIPS)*, 2022. 1, 2, 6

[8] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016. 4

[9] Biraja Ghoshal and Allan Tucker. On calibrated model uncertainty in deep learning. *arXiv preprint arXiv:2206.07795*, 2022. 3

[10] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR, 2017. 1, 2, 3, 5, 6

[11] ER Henry and J Hofrichter. [8] singular value decomposition: Application to analysis of experimental data. In *Methods in enzymology*, volume 210, pages 129–192. Elsevier, 1992. 4

[12] Jeremy Howard. Imagenette: A smaller subset of 10 easily classified classes from imagenet, March 2019. 6

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 4

[14] 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

[15] Virginia Klema and Alan Laub. The singular value decomposition: Its computation and some applications. *IEEE Transactions on automatic control*, 25(2):164–176, 1980. 4

[16] Ranganath Krishnan and Omesh Tickoo. Improving model calibration with accuracy versus uncertainty optimization. *Advances in Neural Information Processing Systems*, 33:18237–18248, 2020. 3

[17] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 2, 6

[18] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 6

[19] Seung Hyun Lee, Dae Ha Kim, and Byung Cheol Song. Self-supervised knowledge distillation using singular value decomposition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–350, 2018. 4

[20] Bowen Lei, Ruqi Zhang, Dongkuan Xu, and Bani Mallick. Calibrating the rigged lottery: Making all tickets reliable. *arXiv preprint arXiv:2302.09369*, 2023. 3

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 3, 6

[22] Dong C Liu and Jorge Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989. 7

[23] Tianming Liu, Eliot Siegel, and Dinggang Shen. Deep learning and medical image analysis for covid-19 diagnosis and prediction. *Annual Review of Biomedical Engineering*, 24:179–201, 2022. 8

[24] Noel Loo, Ramin Hasani, Alexander Amini, and Daniela Rus. Efficient dataset distillation using random feature approximation. *arXiv preprint arXiv:2210.12067*, 2022. 1, 2

[25] Jishnu Mukhoti, Viveka Kulharia, Amartya Sanyal, Stuart Golodetz, Philip Torr, and Puneet Dokania. Calibrating deep neural networks using focal loss. *Advances in Neural Information Processing Systems*, 33:15288–15299, 2020. 3

[26] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. 1

[27] Timothy Nguyen, Zhourong Chen, and Jaehoon Lee. Dataset meta-learning from kernel ridge-regression. *arXiv preprint arXiv:2011.00050*, 2020. 1, 2

[28] 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. 1, 2

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018. 2

[30] Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. *Advances in neural information processing systems*, 32, 2019. 1

[31] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in large margin classifiers*, 10(3):61–74, 1999. 3

[32] Khansa Rasheed, Adnan Qayyum, Mohammed Ghaly, Ala Al-Fuqaha, Adeel Razi, and Junaid Qadir. Explainable, trustworthy, and ethical machine learning for healthcare: A survey. *Computers in Biology and Medicine*, page 106043, 2022. 1, 7

[33] Zohaib Salahuddin, Henry C Woodruff, Avishek Chatterjee, and Philippe Lambin. Transparency of deep neural networks for medical image analysis: A review of interpretability methods. *Computers in biology and medicine*, 140:105111, 2022. 8

[34] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 5, 6

[35] Ilia Sucholutsky and Matthias Schonlau. Soft-label dataset distillation and text dataset distillation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2021. 2

[36] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019. 2, 3

[37] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017. 2

[38] Deng-Bao Wang, Lei Feng, and Min-Ling Zhang. Rethinking calibration of deep neural networks: Do not be afraid of overconfidence. *Advances in Neural Information Processing Systems*, 34:11809–11820, 2021. 4

[39] 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 (CVPR)*, pages 12196–12205, June 2022. 2, 4

[40] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018. 1, 2

[41] Hongxin Wei, Renchunzi Xie, Hao Cheng, Lei Feng, Bo An, and Yixuan Li. Mitigating neural network overconfidence with logit normalization. In *International Conference on Machine Learning*, pages 23631–23644. PMLR, 2022. 4

[42] Longfeng Wu, Bowen Lei, Dongkuan Xu, and Dawei Zhou. Towards reliable rare category analysis on graphs via individual calibration. *arXiv preprint arXiv:2307.09858*, 2023. 3

[43] Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369, 2013. 4

[44] Jian Xue, Jinyu Li, Dong Yu, Mike Seltzer, and Yifan Gong. Singular value decomposition based low-footprint speaker adaptation and personalization for deep neural network. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6359–6363. IEEE, 2014. 4

[45] Ruonan Yu, Songhua Liu, and Xinchao Wang. Dataset distillation: A comprehensive review. *arXiv preprint arXiv:2301.07014*, 2023. 2

[46] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020. 3, 6

[47] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 3, 6

[48] Jie Zhang, Chen Chen, and Lingjuan Lyu. Ideal: Query-efficient data-free learning from black-box models. In *The Eleventh International Conference on Learning Representations*, 2022. 2

[49] Jie Zhang, Chen Chen, Weiming Zhuang, and Lingjuan Lv. Addressing catastrophic forgetting in federated class-continual learning. *arXiv preprint arXiv:2303.06937*, 2023. 1

[50] 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. *arXiv preprint arXiv:2212.06152*, 2022. 1, 6

[51] 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

[52] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021. 1, 2, 6

[53] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020. 1, 2, 6

[54] Yongchao Zhou, Ehsan Nezhadarya, and Jimmy Ba. Dataset distillation using neural feature regression. *arXiv preprint arXiv:2206.00719*, 2022. 2