# Condensed Data Expansion Using Model Inversion for Knowledge Distillation

**Kuluhan Binici[1], Shivam Aggarwal[2], Cihan Acar[3], Nam Trung Pham[4], Karianto Leman[3], Gim Hee Lee[2], Tulika Mitra[2]**

[1]SAP
[2]National University of Singapore
[3]Institute for Infocomm Research, Agency for Science, Technology and Research (A*STAR)
[4]Intelexvision

## Abstract

Condensed datasets offer a compact representation of larger datasets, but training models directly on them or using them to enhance model performance through knowledge distillation (KD) can result in suboptimal outcomes due to limited information. To address this, we propose a method that expands condensed datasets using model inversion, a technique for generating synthetic data based on the impressions of a pre-trained model on its training data. This approach is particularly well-suited for KD scenarios, as the teacher model is already pre-trained and retains knowledge of the original training data. By creating synthetic data that complements the condensed samples, we enrich the training set and better approximate the underlying data distribution, leading to improvements in student model accuracy during knowledge distillation. Our method demonstrates significant gains in KD accuracy compared to using condensed datasets alone and outperforms standard model inversion-based KD methods by up to 11.4% across various datasets and model architectures. Importantly, it remains effective even when using as few as one condensed sample per class, and can also enhance performance in few-shot scenarios where only limited real data samples are available.

## Introduction

Condensed datasets (Zhao, Mopuri, and Bilen 2021) have emerged as a promising approach for compactly representing large datasets, enabling efficient model training with reduced memory and computational costs. These datasets consist of synthetic samples optimized to capture the information content of much larger datasets. They provide certain privacy benefits, as studied in (Dong, Zhao, and Lyu 2022) and can be produced with modest memory and time resources through recent methods (Zhou, Nezhadarya, and Ba 2022; Zhao and Bilen 2023; Feng, Vedantam, and Kempe 2023). These qualities render condensed samples suitable for scenarios in which the privacy considerations prohibit exposure of individual training samples or the large memory size aggravating their relocation. However, the utility of condensed datasets at small scale can be limited in various learning paradigms (Yu, Liu, and Wang 2023) such as standard supervised learning or knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015). This limited utility stems from the reduced information captured in these compact representations, hindering the ability of models to effectively learn from them. In KD, this limitation is particularly relevant as a student model learns from the guidance of a pre-trained teacher model, whose ability to transfer knowledge is itself constrained by the limited information present in the condensed dataset.



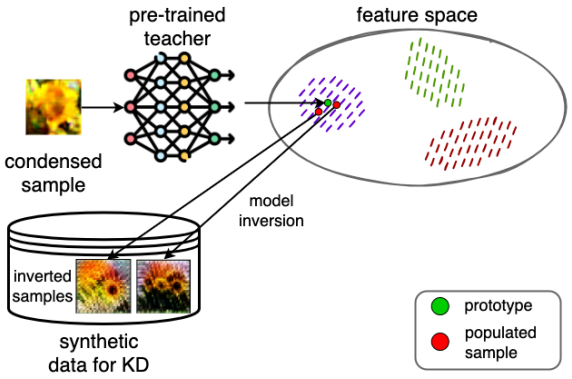



Figure 1: Illustration of our motivation for using condensed samples as prototypes for synthetic data.

In this work, we address the limited utility of condensed datasets, particularly in KD, by expanding them using model inversion (MI) (Lopes, Fenu, and Starner 2017; Yu et al. 2023; Liu et al. 2024). Our goal is to enhance the limited information captured by these compact representations and better approximate the underlying training data. MI is a technique that leverages a pre-trained model as a discriminator to generate synthetic samples resembling real ones. This requirement of a pre-trained model is naturally satisfied in KD, as it inherently involves a pre-trained teacher model. MI operates by training a generative model to produce data points that follow the learned distribution of the teacher's representations (Liu et al. 2021). Since the true data distribution is unknown without access to the original dataset, the generative model is optimized based on certain inductive biases (Zhao et al. 2018; Goyal and Bengio 2022) about the training data. One such bias is the assumption that the teacher classifies real samples with high confidence, resulting in near one-

hot prediction vectors (Chen et al. 2019; Yin et al. 2020).

A trivial approach for expanding the condensed data through MI is simply combining it with the synthetic samples obtained by MI. However, our preliminary experiments suggest that this does not improve accuracy, which is likely caused by the domain gap between the two sample sets (Hennicke et al. 2024; Bai et al. 2024). With this observation, in this paper we propose using the condensed samples as prototypes that represent the real data distribution and changing the MI process to generate samples that are more aligned with the characteristics of the condensed data. By prioritizing the generation of synthetic samples that resemble these prototypes, our method can bridge the domain gap and enhance the performance of the student model trained on the expanded data. Another reason contributing to such performance improvement is that the inductive bias used by MI to model the real data distribution is improved with the knowledge of the prototypes.

Specifically, our method utilizes a small set of condensed samples to query the teacher model and extract more realistic impressions related to the target dataset. These samples are fed to the teacher model for estimating the per-class feature distributions of the target data. Then, the model inversion process is conditioned to produce synthetic samples following a similar feature distribution as the condensed ones. This is achieved by configuring a feature discriminator (Li et al. 2020a) that competes against the sample generator to distinguish condensed samples from the ones generated by model inversion. By doing so, the generator is forced to produce synthetic samples with semantic similarity to the condensed ones, thus reducing the risk of a domain gap and improving accuracy. *One of the major advantages of our approach is its versatility, as it can be applied on top of any model inversion method to improve the accuracy of the student.* This is evident from the experimental evaluation, where we record accuracy improvements of up to $11.44\%$ compared to different state-of-the-art KD baselines across multiple model pairs and datasets. The effectiveness of our method is more pronounced for teacher-student pairs with little structural similarity. In addition, remarkably, even using as few as one condensed sample per class results in a noticeable accuracy improvement.

Moreover, our method is also applicable to scenarios where a limited amount of real samples from the training set are accessible, such as in few-shot learning scenarios (Song et al. 2023; Sauer, Asaadi, and Küch 2022). Experimental results exhibit the advantage of using synthetic data samples generated with guidance from the real samples against pure few-shot KD methods.

# Related Work

## Dataset Condensation

Dataset condensation was introduced by (Zhao, Mopuri, and Bilen 2021) to reduce the training time required for large-scale datasets. It optimizes small batches of synthetic samples to carry almost equal information content as real batches of much larger size. The resulting samples are typically not visually realistic and can better protect data privacy than communicating real samples (Dong, Zhao, and Lyu 2022). To quantitatively assess the privacy benefits, (Zhou, Nezhadarya, and Ba 2022) exercised membership inference attacks (MIA) using condensed samples and showed they yield only around $0.52$ attack AUC, which is almost the same value as random guessing, i.e. $0.5$. Therefore, in cases where samples from a real training set cannot be communicated due to privacy concerns, these condensed samples can be utilized to guide KD's synthetic data generation process. Moreover, the time cost of dataset condensation has significantly decreased due to recent advancements in the area (Zhao and Bilen 2023; Feng, Vedantam, and Kempe 2023). As such, (Zhou, Nezhadarya, and Ba 2022) can generate a condensed dataset of size 10 samples-per-class from ImageNet-200 in less than an hour with 2 GB memory utilization on a single Nvidia Quadro RTX 6000. This allows condensed versions of datasets at various scales (ranging from MNIST to ImageNet) to be conveniently produced and released to public. (He et al. 2023) further reduces the storage requirement of dataset condensation by compressing multiple condensation processes into a single one.

## Knowledge Distillation (KD)

KD (Hinton, Vinyals, and Dean 2015) trains a compact "student" neural network model to approximate the decision space of a more complex one called the "teacher". The inclusion of the soft guidance supplied by the teacher enriches the limited information the student receives from the one-hot encoded class labels. As a result, the student can achieve better performance than supervised training alone can provide. While most commonly, the logit scores or the softmax probabilities of the teacher are considered for regularization (Romero et al. 2014), activation maps or attention scores can also be used (Zagoruyko and Komodakis 2016a).

## Model Inversion

If the training dataset is entirely inaccessible, conventional KD methods can not operate. The textit Model Inversion (MI) technique is developed to infer data samples that the teacher model had observed during training and use them for distillation. Some early works consider the confidence of teachers' predictions as supervision for sample generation (Chen et al. 2019). Some others generate samples that maximize the information gain to the student (Micaelli and Storkey 2019). Following these, DeepInversion (Yin et al. 2020) proposed taking advantage of the batch normalization statistics gathered while training the teacher model. CMI (Fang et al. 2021) improved on this method by diversifying sample synthesis with the help of contrastive learning. Fast-Datafree (Fang et al. 2022) proposed a technique to reduce the significant amount of time that model inversion takes. PRE-DFKD (Binici et al. 2022) introduced a method to eliminate the trade-off between the large memory footprint and the robustness of the process. Recent works have focused on further addressing scalability and efficacy issues (Yu et al. 2023; Liu et al. 2024).

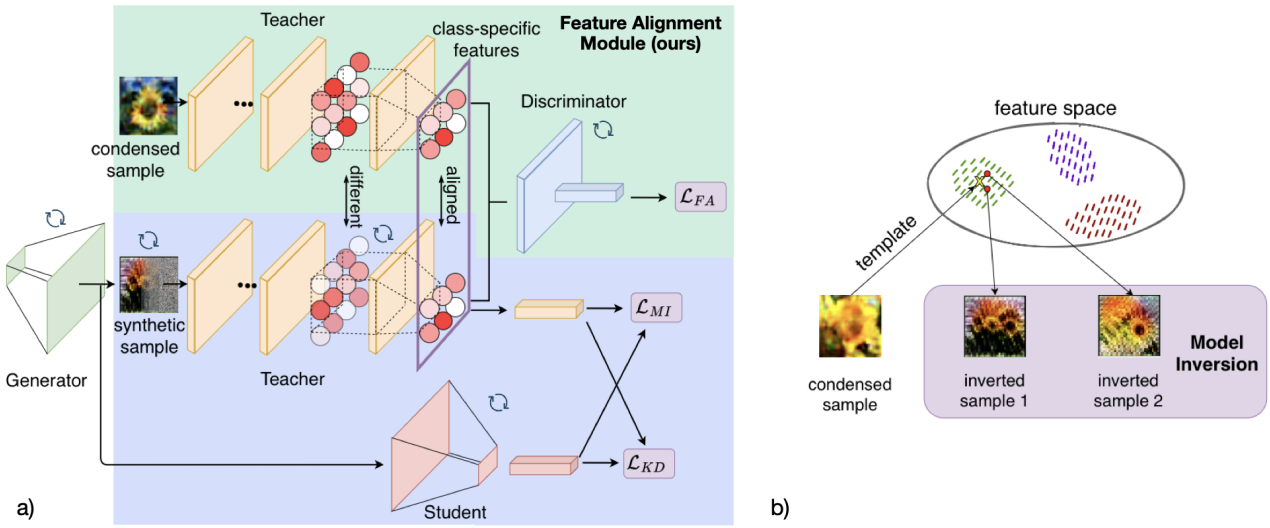



Figure 2: (a) Overview of our condensed-samples guided GMI (generative model inversion) framework. The discriminator is optimized to distinguish real and fake features, while the generator tries to prevent it from doing so by aligning them. (b) Illustration of our motivation for using condensed samples as templates for synthetic data.

### Few-Shot KD

In few-shot KD, only a small subset of real training samples are accessible. To avoid over-fitting, available methods typically reduce the number of parameters required to train the student network. (Bai et al. 2020) feeds activation of teacher and student networks to the layers of one another for cross-correction. FSKD (Li et al. 2020b) obtains the student architecture from the teacher itself via pruning, freezes it, and only learns 1x1 convolutions added after each layer. NetGraft (Shen et al. 2021) performs distillation one layer at a time by grafting the student's layer-tobe-trained to the teacher's architecture.

## Condensed Data Expansion Using MI

Model inversion techniques (Yin et al. 2020; Chen et al. 2019; Fang et al. 2021) typically use variants of the following loss function to guide the synthetic data generation.

$$\mathcal{L}_{MI} = \mathbb{E}_{x,y_{ps}}\left[y_{ps} log(\hat{y}_T)\right] - \mathbb{E}_x\left[D_{KL}(\hat{y_T}||\hat{y_S})\right] \quad (1)$$

The first term targets to maximize the softmax score ($\hat{y}_{(T)}$) (confidence) that the synthetic samples $\hat{x}$ receive from the teacher $T$ for the pseudo-classes ($y_{ps} \sim u(0, c)$) assigned to them. Generally, $\hat{x}$ is obtained using a generative model parameterized by $\theta_g$ ($x \sim p_{\theta_g}(x|z)$). The last term encourages the synthesis of samples that provide high information gain to the student. As this objective is strictly guided by the knowledge that teacher embodies, prototype information cannot be incorporated to provide additional guidance.

### Feature Alignment Mechanism

To incorporate the condensed data prototypes form the training distribution in the model inversion objective, we add the constraint of aligning the feature distribution of generated synthetic samples with that of the condensed ones. This constraint is enforced through the inclusion of a feature discriminator in our model inversion framework. As illustrated in Figure 2, first, the generator outputs synthetic samples. Then, the teacher model encodes both these synthetic batches and condensed samples into feature representations. Later, discriminator classifies these features as condensed or synthetic samples. This classification results in a *feature alignment* loss that quantifies the gap between synthetic and condensed feature distributions. This can be viewed as a minimax game in which the feature discriminator competes against the generator for distinguishing real features from synthetic ones. In equilibrium, the generator will be able to provide samples that can yield similar features as the real ones to trick the discriminator. Essentially, the condensed samples serve as prototypes, guiding the generation of new samples that reflect the training distribution.

The final optimization objective for the generator is constructed by combining the feature alignment loss with the objective of any base model inversion method $\mathcal{L}_{MI}$ as shown in Equation 2.

$$\begin{aligned} &\min \mathcal{L}_G = \mathcal{L}_{MI} + \mathcal{L}_{FA} \\ &\max \mathcal{L}_D = \mathbb{E}_{\hat{x}}\left[\log D(\phi_l(\hat{x}))\right] + \mathbb{E}_{\hat{x}}\left[1 - \log D(\phi_l(\hat{x}))\right] \end{aligned} \quad (2)$$

Where $\mathcal{L}_{FA} = \mathbb{E}_{\hat{x}}\left[1 - \log D(\phi_l(\hat{x}))\right]$ stands for feature alignment loss. As the dimension of feature vectors is high with respect to the limited availability of condensed samples, our method is prone to over-fitting. Therefore we use a simple discriminator architecture introduced by (Li et al. 2020a) that contains very few parameters. To further address the risk of over-fitting, we perform differentiable data augmentation (Zhao et al. 2020) on both the synthetic images output by the generator and the condensed samples before feeding them to the discriminator.

In deciding on the layer index at which the feature alignment will be employed, we considered the type of image features encoded by different parts of the teacher model.

Typically, for image inputs, early layers of neural networks encode structural patterns that are commonly shared across natural images (e.g., edges). In contrast, the image features occurring at the later layers contain semantical information. As our objective is to produce diverse views of objects from the same semantical classes as the condensed samples, we considered features at late layers, specifically the penultimate layer, of the teacher as alignment targets.

Additionally, we posit that simply aligning the cumulative distribution of synthetic features from all classes with the real feature distribution is not ideal. Rather, we provide class-specific alignment using a conditional discriminator (Mirza and Osindero 2014). The contrast between these two alternatives can be seen in the figure given in the appendix. This further changes our discriminator objective to the following.

$$\begin{aligned} \max \mathcal{L}_D = \mathbb{E}_{(x,y)} \left[\log D(\phi_l(x), y)\right] \\ + \mathbb{E}_{(\hat{x},y_{ps})} \left[1 - \log D(\phi_l(\hat{x}), y_{ps})\right] \end{aligned} \tag{3}$$

Here, the discriminator not only predicts if a feature is associated with a condensed or synthetic sample but also determines the class it belongs to. To enforce this, we present three different types of inputs to the discriminator. First, we construct "real" inputs by pairing real features with their labels. Later, we use the teacher to assign labels to the synthetic features and obtain "fake" inputs. Lastly, to prevent the discriminator from neglecting the class information, we construct additional "fake" inputs by pairing the same real features with the wrong class labels. Formally, our real ($\mathcal{R}$) and fake ($\mathcal{F}$) sets can be defined as,

$$\begin{aligned} \mathcal{R} &= \{(x,y) | (x,y) \in \mathcal{X}\} \\ \mathcal{F} &= \{(\hat{x}, y_{ps})\} \cup \{(x,c) | (x,y) \in \mathcal{X}, c \neq y\} \end{aligned} \tag{4}$$

## Combining Condensed and Synthetic Samples

After establishing our feature-alignment strategy to improve model inversion with the available data samples, we discuss how we can join condensed and generated synthetic samples for KD. Some alternatives included pre/post-training the student with condensed samples with respect to model inversion. However, as these methods can cause the student to be biased towards one data type, we avoided them. Instead, we expanded the condensed dataset by adding the iteratively refined synthetic samples (through model inversion) and trained the student with randomly sampled batches from such union. Our distillation objective involves minimizing the distance between the predictions of the teacher and the student models, which can be summarized as,

$$\theta_S^* := \arg\min_{\theta_S} \mathbb{E}_{\hat{x}} \left[D_{KL}(\hat{y}_S || \hat{y}_T)\right] + \mathbb{E}_x \left[D_{KL}(y_S || y_T)\right] \tag{5}$$

In Equation 5, $\hat{x}$ and $x$ denote synthetic samples and condensed samples respectively. The exact procedure we follow in generating synthetic samples and distilling the student is summarised in Algorithm 1. First, we initialize our synthetic dataset $\mathcal{X}$ with the available condensed samples. Later at each epoch, we generate a new synthetic batch via our condensed sample-guided model inversion and add it to $\mathcal{X}$. Later, within the same epoch, we randomly draw a data batch from $\mathcal{X}$ and use it to transfer knowledge from the teacher to the student.

Algorithm 1: Knowledge Distillation

**INPUT:** generator $G$, discriminator$D$, teacher $T$, student $S$ parameterized by $\theta_S$, condensed data $\mathcal{X}$
**OUTPUT:** trained student $\theta_S^*$.

**for** number of epochs **do**
  $\hat{x}_{new} \leftarrow invert\_model(T, S, G, D, \mathcal{X})$
  $\mathcal{X} \leftarrow \mathcal{X} \cup \hat{x}_{new}$
  $(x, y) \sim \mathcal{X}$
  $\mathcal{L}_{KD} \leftarrow \sum_{\mathcal{X}} D_{KL}(\hat{y}_S || \hat{y}_T)$
  $optimizer.step(backward(\mathcal{L}_{KD}), \theta_S)$
**end for**

# Experimental Evaluation

To assess the effectiveness of our method, we incorporate condensed data guidance to three state-of-the-art model-inversion methods and record the improvement in KD performance. These methods are Fast, CMI, and PRE-DFKD. Moreover, we also experiment with applying out method to expand limited real data and observe the advantage against few-shot KD methods. For this comparison, we selected NetGraft and FSKD as baselines. All the results we report on the performance of our baseline methods are either directly taken from the papers or obtained by running the official implementations based on the hyper-parameter configurations shared in the papers or GitHub pages. To standardize the evaluation, we use fixed random seeds borrowed from the official implementations of the baselines.

**Datasets** We use three image classification datasets, which are CIFAR-10/100 (Krizhevsky and Hinton 2009), and ImageNet-200 (Deng et al. 2009). We conducted our experiments using condensed samples generated by three different methods, including those provided by (Zhao and Bilen 2021), (Cazenavette et al. 2022) and (Zhao and Bilen 2023). While the results reported throughout our experiments primarily utilize the condensed samples from (Zhao and Bilen 2021), we also include an ablation study to compare the effectiveness of each condensation method.

**Implementation Details** We used the same generator architectures as shared in the official implementations of the MI methods that we couple our method with (Fast, CMI, and PRE-DKD). Further details on generator and discriminator architectures as well as how we couple our method with individual MI methods can be found in the appendix.

## Impact of Expanded Condensed Data on KD

Tables 1 and 3 show the student accuracies upon coupling our approach with Fast, CMI, and PRE-DFKD. The results achieved by this coupling are indicated by the asterisk symbol with the annotation "*(ours w/ CS)". We also configure naive baselines where we combine these samples with the synthetic datasets generated by model inversion methods. These are denoted with "+ CS" notations. Further, the row "Train w/ full real data" represents the accuracy of full-scale training of students on the target dataset and constitutes the upper bound. "Train w/ cond. samples" reflects the accuracy achieved by only using condensed samples for student train-

| Dataset | CIFAR-10 | | | | CIFAR-100 | | | | ImageNet-200 | |
|---|---|---|---|---|---|---|---|---|---|---|
| Teacher | ResNet-34 | ResNet-34 | WRN-40-2 | WRN-40-2 | ResNet-34 | ResNet-34 | WRN-40-2 | WRN-40-2 | ResNet-34 | ResNet-34 |
| Student | ResNet-18 | MBNet-v2 | WRN-40-2 | MBNet-v2 | ResNet-18 | MBNet-v2 | WRN-40-2 | MBNet-v2 | ResNet-18 | MBNet-v2 |
| Teacher acc. | 95.70 | 95.70 | 94.87 | 94.87 | 78.05 | 78.05 | 75.83 | 75.83 | 71.20 | 71.20 |
| Training student (S) with labeled data | | | | | | | | | | |
| Train w/ full real data | 95.20 | 93.79 | 94.87 | 93.79 | 77.10 | 72.80 | 75.83 | 72.80 | 64.90 | 55.06 |
| Train w/ cond. samples (CS) | 34.66 | 30.13 | 38.92 | 30.13 | 16.54 | 10.69 | 15.55 | 10.69 | 3.50 | 5.60 |
| Distilling student (S) with synthetic data | | | | | | | | | | |
| Fast | 92.62 | 86.12 | 92.82 | 85.06 | 69.76 | 54.62 | 65.05 | 48.21 | 42.99 | 35.31 |
| Fast + CS | 92.72 | 86.37 | 92.84 | 85.69 | 69.96 | 56.57 | 65.51 | 49.42 | 45.43 | 40.68 |
| Fast* (ours w/ CS) | **94.33** | **88.05** | **94.64** | **89.24** | **72.09** | **63.29** | **70.96** | **60.86** | **48.14** | **43.08** |
| CMI | 94.84 | 87.5 | 92.83 | 86.53 | 77.04 | 61.9 | 68.96 | 59.04 | 44.11 | 35.55 |
| CMI + CS | 94.89 | 88.06 | 92.94 | 86.77 | 77.04 | 62.54 | 69.10 | 59.62 | 47.07 | 40.67 |
| CMI* (ours w/ CS) | **94.97** | **89.63** | **94.21** | **90.21** | **77.07** | **70.21** | **72.42** | **68.05** | **48.98** | **45.83** |

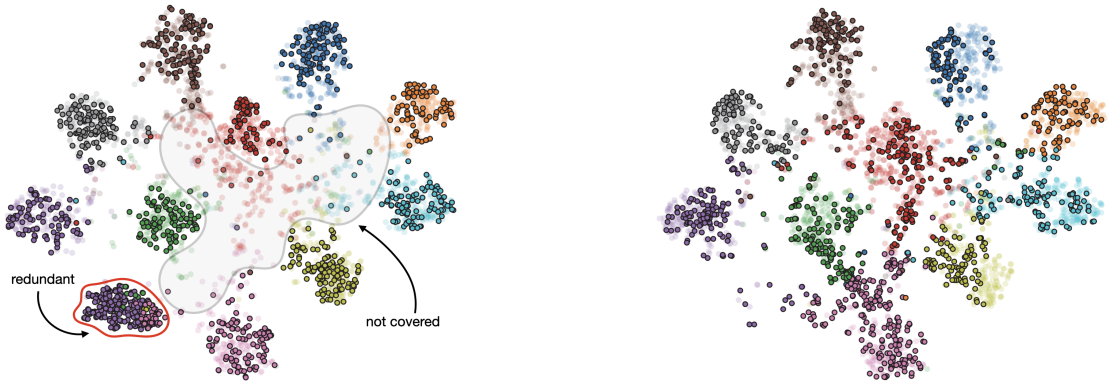
Table 1: Student accuracies (%) obtained by expanding condensed data using Fast and CMI. Condensed datasets with 50 spc from CIFAR10, 10 spc from CIFAR100, and 10 spc from ImageNet-200 were used.

ing, which is the lower bound. In all experiments, we use condensed datasets of 50 samples per class (spc) (total 500 samples) and 10 spc (total 1000 samples) for CIFAR-10 and CIFAR-100, respectively. For ImageNet-200, we use 10 spc (total 2000 samples). These correspond to 1%, 2%, and 2% of the total samples in their respective datasets.

First, we note that the performance of Fast method notably diminishes for pairs with low structural similarity (e.g. ResNet-34 (He et al. 2016) & MobileNet-v2 (Howard et al. 2017)). The improvement was especially significant for WRN-40-2 (Zagoruyko and Komodakis 2016b) & MobileNet-v2 pairs reaching up to $11.44\%$ on CIFAR-100. CMI also has a considerable performance gap with respect to the upper bound for heterogeneous model pairs. Our method again achieves consistent advantage, with substantial accuracy improvements reaching up to $8.43\%$ (WRN-40-2 & MobileNet-v2). Since PRE-DFKD achieves almost the same student accuracy as the upper limit for homogeneous pairs (e.g. ResNet-34 & ResNet-18), we only experiment with heterogeneous ones (e.g. ResNet-34 & MobileNet-v2), where there is still room for improvement. The results in Table 3 shows that our method effectively improves acuracy also for this baseline method.

In all experiments, the simple combination of condensed samples and the synthetic samples from model inversion ("+ CS") neither mitigated this issue nor caused any substantial performance improvement in most cases. On the other hand, our condensed sample-guided model inversion ("*") consistently increased student accuracy across different datasets and teacher-student pairs, which ensures that the benefit of our approach is not simply due to exposing the student to more samples during training.

**Visual Results** We examine the impact of our method on the visual quality of the generated samples. Figure 5 contains synthetic CIFAR100 images obtained by CMI and CMI*. We note that CMI* samples are significantly more realistic and exhibit common class-distinctive patterns across images from the same categories, which is not observed in CMI. This strengthens the claim that feature alignment effectively conditions the synthetic data to contain realistic semantics that is consistent among samples from the same classes. Additionally, this conditioning does not compromise the diversity of the synthetic set, as demonstrated by the varied object views and scales in CMI*. We also note that neither the condensed samples nor the condensed sample-




(a) Synthetic feature distribution by data-free model inversion

(b) Condensed data-guided synthetic feature distribution (ours)

Figure 3: 2D visualisation of feature vectors. Faded and bold markers denote feature space projections of real samples from the CIFAR-10 dataset and synthetic samples, respectively. Condensed data-guided synthetic samples exhibit better alignment with the real data distribution.

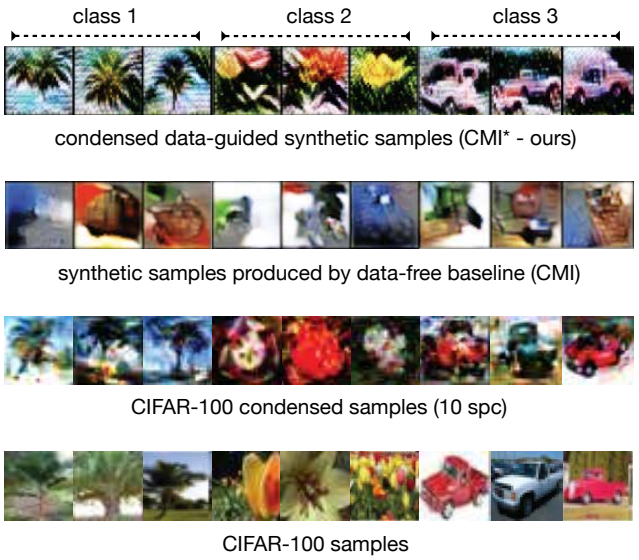



Figure 5: First two rows contain synthetic CIFAR-100 samples obtained with and w/o condensed data-guided model inversion. The last two show condensed and real samples.

guided generated synthetic samples exist in the real dataset. Thus, although the generated synthetic samples are visually realistic, they do not reveal any individual training samples. Moreover, to visually observe the effect of our method on the distribution of generated synthetic samples, we projected the high-dimensional feature space on a 2D plane using the t-SNE algorithm (Van der Maaten and Hinton 2008). On the resulting plane, we compare the feature distributions observed for CMI and CMI*. We use the WRN-40-2 teacher trained on the CIFAR-10 dataset as the feature extractor. The visualizations are displayed in Figure 3 where faded data points mark the features of real samples, and the bold ones represent the synthetic ones. Each sample is marked with the color associated with its class label. Synthetic samples generated by CMI are mostly clustered together and only

| Dataset | CIFAR-10 | | | | | |
|---|---|---|---|---|---|---|
| Teacher | ResNet-34 | | | WRN-40-2 | | |
| Student | MobileNet-v2 | | | MobileNet-v2 | | |
| Ablation Study on Different Dataset Condensation Methods | | | | | | |
| **CS Type / KD Method** | **DSA** | **DM** | **MTT** | **DSA** | **DM** | **MTT** |
| Fast + CS | 86.37 | 86.54 | 86.78 | 85.69 | 80.01 | 80.38 |
| **Fast* (ours w/ CS)** | **88.05** | **87.82** | **87.11** | **89.24** | **90.30** | **90.38** |

Table 2: Impact of different types of condensed samples (CS) on student accuracy (%) for different dataset condensation strategies. Dataset contains 50 spc from CIFAR10.

partially correspond with real samples. Also, some feature clusters have almost no correspondence with real features (circled in red), meaning the associated synthetic samples might not be contributing to the knowledge transfer. In contrast, the synthetic feature distribution formed during CMI* shows better alignment with real data distribution.

**How Does Student Accuracy Scale with the Condensed Data Size?** After establishing that our proposed method can boost the utility of condensed samples in KD, we analyze how the scale of the available data affects the improvement. For this, we consider condensed datasets of 3 different sizes (1 spc, 10 spc, 50 spc) for CIFAR-10 and 2 different sizes (1 spc, 10 spc) for CIFAR-100. These amounts correspond to 0.02%, 0.2% and 1% of the samples contained in CIFAR-10, and 0.2% and 2% for CIFAR-100. The plots in Figure 4 show that Fast and CMI baselines do not benefit from the mere inclusion of condensed samples in their synthetic distillation sets, irrespective of the number of samples available. However, when they are equipped with our feature alignment module, the student accuracies scale up with increasing data availability.

**Does the Condensation Method Affect the Quality of Model Inversion?** We study the impact of the dataset condensation used to produce the condensed samples on the effectiveness of our approach and display the results in Table 2. DSA, DM and MTT refer to (Zhao and Bilen 2021), (Zhao and Bilen 2023), and (Cazenavette et al. 2022) re-

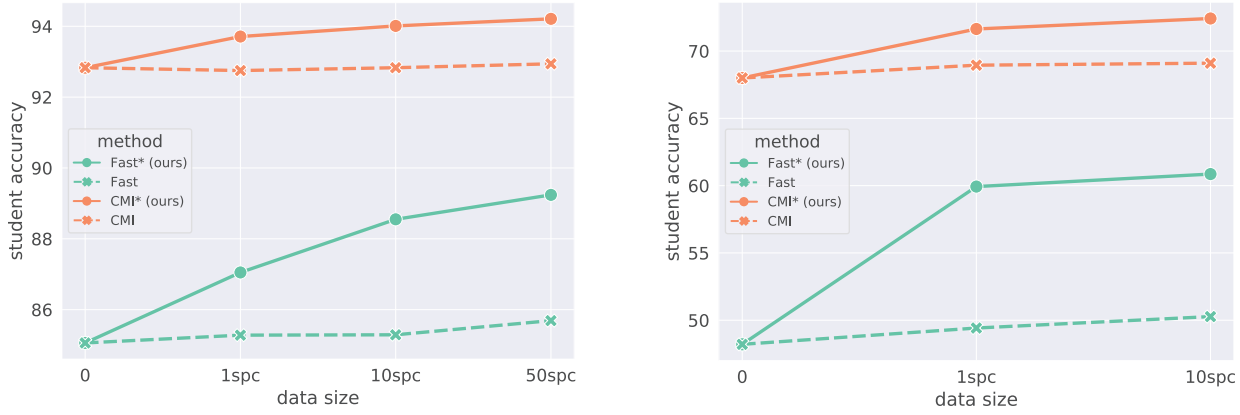



Figure 4: Students distilled by only using samples from model inversion (left), and using expanded data generated by our method (right) for different condensed dataset set sizes. Condensed datasets with 1, 10, and 50 spc from CIFAR10, and 1 and 10 spc from CIFAR100, were used.

| Dataset | CIFAR-10 | | CIFAR-100 | | ImageNet-200 | |
|---|---|---|---|---|---|---|
| Teacher | ResNet-34 | WRN-40-2 | ResNet-34 | WRN-40-2 | ResNet-34 | ResNet-34 |
| Student | MobileNet-v2 | MobileNet-v2 | MobileNet-v2 | MobileNet-v2 | ResNet-18 | MobileNet-v2 |
| Teacher acc. | 95.70 | 94.87 | 78.05 | 75.83 | 71.20 | 71.20 |
| Accuracy of student (S) trained with labeled data | | | | | | |
| Train w/ full real data | 93.79 | 93.79 | 72.80 | 72.80 | 64.90 | 55.06 |
| Train w/ cond. samples (CS) | 27.24 | 27.24 | 10.69 | 10.69 | 3.50 | 5.60 |
| Train w/ few real samples (RS) | 27.55 | 27.55 | 4.66 | 4.66 | 1.42 | 1.30 |
| Accuracy of student (S) trained with synthetic data | | | | | | |
| PRE-DFKD | 83.12 | 83.12 | 66.56 | 61.83 | 54.20 | 47.26 |
| PRE-DFKD* (ours w/ CS) | **86.89** | **88.34** | **67.39** | **63.41** | **55.22** | **49.73** |
| PRE-DFKD* (ours w/ RS) | **86.77** | **87.47** | **71.63** | **64.37** | **54.95** | **49.68** |

Table 3: Student accuracies (%) obtained by utilizing the expanded condensed and real datasets set using PRE-DFKD (CS and RS). Condensed and real sample sets contain 50 spc from CIFAR10, 10 spc from CIFAR100, and 10 spc from ImageNet-200.

| Dataset | CIFAR-10 | | CIFAR100 | |
|---|---|---|---|---|
| Teacher | VGG-11 | VGG-11 | VGG-11 | VGG-11 |
| Student | VGG-11 (50% pruned) | VGG-11 (75% pruned) | VGG-11 (25% pruned) | VGG-11 (50% pruned) |
| Teacher acc. | 92.25 | 92.25 | 71.23 | 71.23 |
| Few-shot distillation accuracy | | | | |
| FSKD | 78.69 | 36.00 | 54.72 | 24.73 |
| PRE-DFKD* | **83.26** | **68.48** | **63.81** | **61.21** |

(a)

| Dataset | CIFAR-10 | CIFAR100 |
|---|---|---|
| Teacher | VGG-16 | VGG-16 |
| Student | ResNet-18 | ResNet-18 |
| Teacher acc. | 94.16 | 74.00 |
| Few-shot distillation accuracy | | |
| NetGraft | 73.69 | 55.51 |
| PRE-DFKD* | **88.38** | **68.54** |

(b)

Table 4: Comparison with different few-shot baselines using few real samples. (a) Student accuracy comparison with FSKD baseline. Real sample sets containing 50 spc from CIFAR10, and 10 spc from CIFAR100 were used. (b) Comparison with few-shot NetGraft baseline. 1 spc from both CIFAR10 / 100 were used.

spectively. The results indicate that while the choice of condensation method can impact the final student accuracy, our method consistently improves performance across all types of condensed samples tested. This suggests that its effectiveness is not dependent on any single condensation approach.

## Impact of Expanded Real Data on KD

Our method is also inherently capable of expanding limited real samples from the target dataset in few-shot scenarios. To show this, we repeated the experiments in Table 3 by replacing the condensed samples with real ones. We assume the availability of the same amount of samples randomly drawn from the target datasets as we used in experiments with condensed data (50 spc for CIFAR10, 10 spc for CIFAR100 and 10 spc for ImageNet-200). The results are displayed in Table 3 as PRE-DFKD* (ours w/ RS). Similar to condensed sample experiments, the improvements observed for heterogeneous pairs are again greater than the homogeneous ones, i.e. ResNet-34 & ResNet-18. The overall improvement yielded by our method upon utilizing few real samples is comparable that using condensed ones.

**Comparison with Few-Shot Methods** We benchmarked the effectiveness of our method in few-shot KD against FSKD and Netgraft baselines using the same experiment setups reported in their papers. To ensure a fair evaluation of our work against these few-shot baselines, we selected identical teacher-student pairs to those in the original papers. We again use the same amount of samples from both datasets as in Table 3. In our comparison with FSKD, we use VGG11 teachers and student models that are channel-pruned versions of the teacher at different rates. As for the comparison with Netgraft, we use VGG16 teachers and ResNet-18 students, with 1 spc from both datasets. From Table 4a, it can be observed FSKD is only effective for low pruning rates while performing poorly for higher rates. Similarly, the accuracies achieved by NetGraft were also much lower than the upperbound as seen in Table 4b. This is expected as both baselines are restricted to the limited information made available by the few-shot sample set. As our method has a generative component, it does not have such restriction and therefore outperforms both few-shot baselines by a large margin.

## Conclusion

We address the challenge of limited information in condensed datasets, which often hinders their effectiveness in KD. We propose a method that leverages MI to expand these condensed datasets, generating synthetic data that enriches the compact representation and improves KD performance. Our approach utilizes condensed samples as prototypes to guide the MI process, ensuring that the generated synthetic data aligns closely with the underlying data distribution represented by the condensed set. Our experiments demonstrate the effectiveness of our method across various datasets, model architectures, and state-of-the-art MI techniques (Fast, CMI, PRE-DFKD). The results show consistent improvements in KD, compared to using condensed datasets alone or standard MI-based KD methods. Our approach also applies to few-shot learning, outperforming existing few-shot KD by effectively leveraging limited real data for synthetic data generation.

## References


Bai, H.; Wu, J.; King, I.; and Lyu, M. 2020. Few shot network compression via cross distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3203–3210.

Bai, X.; Luo, Y.; Jiang, L.; Gupta, A.; Kaveti, P.; Singh, H.; and Ostadabbas, S. 2024. Bridging the Domain Gap between Synthetic and Real-World Data for Autonomous Driving. *Journal on Autonomous Transportation Systems*, 1(2): 1–15.

Binici, K.; Aggarwal, S.; Pham, N. T.; Leman, K.; and Mitra, T. 2022. Robust and Resource-Efficient Data-Free Knowledge Distillation by Generative Pseudo Replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6089–6096.

Cazenavette, G.; Wang, T.; Torralba, A.; Efros, A. A.; and Zhu, J.-Y. 2022. Dataset distillation by matching training trajectories. In *CVPR*, 4750–4759.

Chen, H.; Wang, Y.; Xu, C.; Yang, Z.; Liu, C.; Shi, B.; Xu, C.; Xu, C.; and Tian, Q. 2019. Data-free learning of student networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3514–3522.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Dong, T.; Zhao, B.; and Lyu, L. 2022. Privacy for free: How does dataset condensation help privacy? In *ICML*, 5378–5396. PMLR.

Fang, G.; Mo, K.; Wang, X.; Song, J.; Bei, S.; Zhang, H.; and Song, M. 2022. Up to 100x Faster Data-Free Knowledge Distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 6597–6604.

Fang, G.; Song, J.; Wang, X.; Shen, C.; Wang, X.; and Song, M. 2021. Contrastive model inversion for data-free knowledge distillation. *arXiv preprint arXiv:2105.08584*.

Feng, Y.; Vedantam, S. R.; and Kempe, J. 2023. Embarrassingly simple dataset distillation. In *The Twelfth International Conference on Learning Representations*.

Goyal, A.; and Bengio, Y. 2022. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266): 20210068.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, Y.; Xiao, L.; Zhou, J. T.; and Tsang, I. 2023. Multisize Dataset Condensation. In *The Twelfth International Conference on Learning Representations*.

Hennicke, L.; Adriano, C. M.; Giese, H.; Koehler, J. M.; and Schott, L. 2024. Mind the Gap Between Synthetic and Real: Utilizing Transfer Learning to Probe the Boundaries of Stable Diffusion Generated Data. *arXiv preprint arXiv:2405.03243*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Krizhevsky, A.; and Hinton, G. 2009. Learning Multiple Layers of Features from Tiny Images. *Technical report, University of Toronto*.

Li, K.; Zhang, Y.; Li, K.; and Fu, Y. 2020a. Adversarial feature hallucination networks for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13470–13479.

Li, T.; Li, J.; Liu, Z.; and Zhang, C. 2020b. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14639–14647.

Liu, H.; Wang, Y.; Liu, H.; Sun, F.; and Yao, A. 2024. Small Scale Data-Free Knowledge Distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6008–6016.

Liu, Y.; Zhang, W.; Wang, J.; and Wang, J. 2021. Data-free knowledge transfer: A survey. *arXiv preprint arXiv:2112.15278*.

Lopes, R. G.; Fenu, S.; and Starner, T. 2017. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*.

Micaelli, P.; and Storkey, A. J. 2019. Zero-shot knowledge transfer via adversarial belief matching. *Advances in Neural Information Processing Systems*, 32.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Romero, A.; Ballas, N.; Kahou, S. E.; Chassang, A.; Gatta, C.; and Bengio, Y. 2014. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*.

Sauer, A.; Asaadi, S.; and Küch, F. 2022. Knowledge distillation meets few-shot learning: An approach for few-shot intent classification within and across domains. In *Proceedings of the 4th Workshop on NLP for Conversational AI*, 108–119.

Shen, C.; Wang, X.; Yin, Y.; Song, J.; Luo, S.; and Song, M. 2021. Progressive network grafting for few-shot knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 2541–2549.

Song, Y.; Wang, T.; Cai, P.; Mondal, S. K.; and Sahoo, J. P. 2023. A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities. *ACM Computing Surveys*, 55(13s): 1–40.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Yin, H.; Molchanov, P.; Alvarez, J. M.; Li, Z.; Mallya, A.; Hoiem, D.; Jha, N. K.; and Kautz, J. 2020. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8715–8724.

Yu, R.; Liu, S.; and Wang, X. 2023. Dataset distillation: A comprehensive review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Yu, S.; Chen, J.; Han, H.; and Jiang, S. 2023. Data-free knowledge distillation via feature exchange and activation region constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 24266–24275.

Zagoruyko, S.; and Komodakis, N. 2016a. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*.

Zagoruyko, S.; and Komodakis, N. 2016b. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhao, B.; and Bilen, H. 2021. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, 12674–12685. PMLR.

Zhao, B.; and Bilen, H. 2023. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 6514–6523.

Zhao, B.; Mopuri, K. R.; and Bilen, H. 2021. Dataset Condensation with Gradient Matching. *ICLR*, 1(2): 3.

Zhao, S.; Liu, Z.; Lin, J.; Zhu, J.-Y.; and Han, S. 2020. Differentiable augmentation for data-efficient gan training. *Advances in Neural Information Processing Systems*, 33: 7559–7570.

Zhao, S.; Ren, H.; Yuan, A.; Song, J.; Goodman, N.; and Ermon, S. 2018. Bias and generalization in deep generative models: An empirical study. *Advances in Neural Information Processing Systems*, 31.

Zhou, Y.; Nezhadarya, E.; and Ba, J. 2022. Dataset distillation using neural feature regression. *NeurIPS*.