DiLM: Distilling Dataset into Language Model for Text-level Dataset Distillation

Anonymous ACL submission

Abstract

Dataset distillation aims to compress a training dataset by creating a small number of informative synthetic samples such that neural networks trained on them perform as well as those trained on the original training dataset. Current text dataset distillation methods create each synthetic sample as a sequence of word embeddings instead of a text to apply gradient-based optimization; however, such embedding-level distilled datasets cannot be used for training other models whose word embedding weights are different from the model used for distillation. To address this issue, we propose a novel text dataset distillation approach, called *Distilling dataset into Language Model (DiLM)*, which trains a language model to generate informative synthetic training samples as text data, instead of directly optimizing synthetic samples. We evaluated DiLM on various text classification datasets and showed that distilled synthetic datasets from DiLM outperform those from current coreset selection methods. DiLM achieved remarkable generalization performance in training different types of models and in-context learning of large language models. Our code will be available at `https://github.com/....`

1 Introduction

The successful advancements in machine learning in a wide range of fields are due to the scaling-up of deep neural networks and large training datasets. In the natural language processing (NLP) field, large language models (LLMs), which are pre-trained with a huge amount of text, such as BERT- and GPT-family models (Devlin et al., 2019; Liu et al., 2019; Radford et al., 2019; Brown et al., 2020), have shown remarkable capabilities for various NLP tasks. However, training such large-scale models requires large computational resources and a long time, which makes it difficult to develop new LLMs, and even to fine-tune them.

To address this issue, dataset distillation (Wang et al., 2018b) has attracted much attention in the machine learning community, which aims to reduce training costs by compressing training datasets. In contrast to traditional coreset selection approaches (Wolf, 2011; Sener and Savarese, 2018; Welling, 2009), which heuristically select a small subset of representative training samples from the original dataset, dataset distillation creates more informative synthetic samples by distilling the knowledge from the original dataset. With this approach, synthetic samples are optimized with gradient descent according to objective functions for dataset distillation, including meta-learning (Wang et al., 2018b), gradient matching (Zhao et al., 2021), training trajectory matching (Cazenavette et al., 2022), and feature distribution matching (Wang et al., 2022; Zhao and Bilen, 2023). The recent remarkable performance of dataset distillation, especially in the computer vision (CV) field, has also led to studies of its various applications, including neural architecture search (Such et al., 2020; Medvedev and D'yakonov, 2021), federated learning (Zhang et al., 2022a; Xiong et al., 2023), continual learning (Wiewel and Yang, 2021; Sangermano et al., 2022), and privacy preservation (Dong et al., 2022; Chen et al., 2022).

While most previous studies applied dataset distillation only to image classification datasets, some studies focused on text dataset distillation (Sucholutsky and Schonlau, 2021; Li and Li, 2021; Maekawa et al., 2023; Sahni and Patel, 2023). In contrast to the image, which can be applied gradient-based optimization by considering it as a pixel-wise continuous data, the discrete nature of text makes dataset distillation challenging (Geng et al., 2023; Yu et al., 2023). To address this issue, all existing text dataset distillation methods used the widely used neural NLP technique called embedding, i.e., optimizing a synthetic dataset as continuous input word embeddings instead of dis-

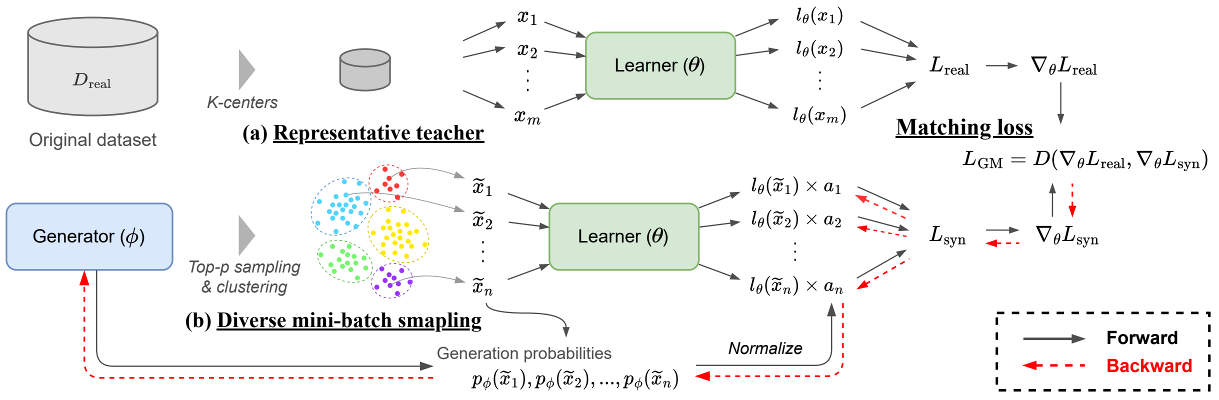

Figure 1: Overview of training with DiLM. Gradient matching loss is computed on the learner model between real samples from the original dataset and generated samples from the generator model. It is then back-propagated to the generator model via generation probabilities, which weight the learner loss for each generated sample. (a) Representative teacher for computing real sample's gradients, which improves the performance and accelerates convergence by using K-center samples, representing the original dataset, rather than randomly sampled ones. (b) Diverse mini-batch sampling, which enables the generator model to explore diverse synthetic samples in each training step.

crete text. However, such embedding-level distilled synthetic datasets cannot be used for training other models that have different word embedding weights, which is a crucial issue in terms of practical applications. Furthermore, distilled word embedding sequences are also completely unreadable to humans, which makes it difficult to interpret and analyze the original training dataset by observing distilled synthetic samples.

Motivated by these shortcomings, this paper explores the text dataset distillation to obtain distilled synthetic datasets at the text-level as the first study. We propose the first text-level dataset distillation approach called "**D**istilling dataset **i**nto **L**anguage **M**odel (**DiLM**)". To overcome the optimization difficulty of discrete text, DiLM uses a language model as a surrogate continuous optimization target instead of directly optimizing a synthetic sample's text. Specifically, DiLM trains a language model to minimize the gradient matching loss (Zhao et al., 2021) of generated synthetic samples as a dataset distillation objective. To enable back-propagating the gradient matching loss to the language model, we design a differentiable backward pass via loss weighting with generation probabilities to bypass the non-differentiable generated text (Figure 1).

In our experiments, we applied DiLM to distill three text classification datasets from the GLUE benchmark (Wang et al., 2018a), SST-2, QQP, and MNLI-m. The results indicate that the synthetic datasets distilled with DiLM outperformed representative real samples selected from the original datasets with current coreset selection methods. Our distilled datasets also achieved remarkable generalization performance not only for training different types of pre-trained models but also for in-context learning of LLMs as few-shot prompts.

Our main contributions are as follows:

- To the best of our knowledge, this is the first study to distill a text dataset into a text-level synthetic dataset that are applicable for training models independent of word embedding weights.
- We present DiLM, which addresses the discreteness of text by using a language model as a surrogate optimization target and back-propagating the distillation loss to the model, bypassing non-differentiable generated text.
- Our experimental results indicate that DiLM outperformed the current coreset selection methods not only for training the same model used for distillation, but also for training different models independent of the word embedding weights, architectures, and training processes.

2 Related Work

2.1 Dataset Distillation

Dataset distillation was first proposed by Wang et al. (2018b), motivated by theoretical interests as well as practical applications for reducing network training costs. Inspired by meta-learning based hyperparameter optimization (Maclaurin et al., 2015), Wang et al. (2018b) optimized a small synthetic dataset by gradient decent such that models trained on it have a lower training loss for the original dataset. Recently, several surrogate objectives have been proposed to improve the performance and efficiency of dataset distillation. DC (Zhao et al., 2021) and DSA (Zhao and Bilen, 2021) focused on gradient matching between real and synthetic samples. DM (Zhao and Bilen, 2023) and CAFE (Wang et al., 2022) proposed feature distribution matching, which requires less GPU memory for optimizing synthetic datasets. MTT (Cazenavette et al., 2022) and TESLA (Cui et al., 2023) optimized synthetic samples to approximate trajectories of model parameters trained with real data. SLDD (Sucholutsky and Schonlau, 2021) and LDD (Bohdal et al., 2020) introduced learnable soft-labels, which are optimized together with input images to make each synthetic sample more informative.

While the most current research on dataset distillation involves only image classification datasets, some studies also focused on text classification datasets. Sucholutsky and Schonlau (2021) and Li and Li (2021) applied the original meta-learning based method by Wang et al. (2018b) to text datasets. To overcome the discrete nature of text, which makes applying gradient-based methods difficult, they optimized synthetic samples in the pre-trained GloVe word embedding space (Pennington et al., 2014) instead of actual words of text as the optimization target. Maekawa et al. (2023) extended the text dataset distillation to the pre-trained BERT model and improved its performance by introducing learnable attention labels, which directly guide the self-attention probabilities of the models. Sahni and Patel (2023) explored dataset distillation in multilingual text classification datasets in the context of fairness, interpretability, and cross-architecture generalization. Although these methods perform well for text classification datasets, distilled synthetic datasets obtained with them cannot be used for training other models that have different word embedding weights. Although Sucholutsky and Schonlau (2021) and Sahni and Patel (2023) transformed their distilled synthetic samples to text by finding a word that has the nearest neighbor embedding, the converted text consists of unrelated words and does not make sense, which makes it difficult to interpret and analyze them. Moreover, the performance of distilled datasets after being converted to text has also not been investigated.

2.2 Generative Models

Recent studies on dataset distillation in the CV field used generative adversarial networks (GANs) (Goodfellow et al., 2014), i.e., training the model parameters and/or their latent input noises instead of synthetic images. These methods generalize distilled synthetic images to different model architectures by restricting them to the generative distribution learned from the original dataset. DiM (Wang et al., 2023) fine-tuned a GAN to generate informative synthetic images from randomly sampled latent noises, where distilled datasets of different sizes can be produced without retraining the model. GTNs (Such et al., 2020) trained a GAN to generate informative images, instead of realistic images, to accelerate neural architecture search. GTNs also learned a latent noise for each synthetic image as a curriculum of training learner networks. IT-GAN (Zhao and Bilen, 2022) and GLaD (Cazenavette et al., 2023) used a pre-trained GAN as a generative prior of synthetic samples and only optimized the latent noises.

Inspired by these studies, we also introduce a generative model with a different motivation for text dataset distillation: to avoid the difficulties of directly optimizing discrete text, we instead optimize the continuous parameters of a generative model to generate distilled synthetic samples. However, since all previous studies that used generative models for image dataset distillation trained them and/or their input latent noises by back-propagating the distillation loss to them via generated images, none of them can be applied to text data, which are non-differentiable due to their discrete nature.

3 Methodology

In this section, we introduce DiLM, which distills text datasets into text data, not word embeddings, for the model-agnostic applicability and interpretability of the distilled synthetic datasets. The main idea of DiLM is to avoid the optimization difficulties of discrete text by instead training continuous parameters of a language model as a surrogate optimization target of dataset distillation.

3.1 Overview

Given a training dataset $\mathcal{D}_{\text{real}} = \{x_i\}_{i=1}^{|\mathcal{D}_{\text{real}}|}$, the goal of DiLM is to obtain a generator model, parameterized by ϕ, that generates a distilled synthetic dataset $\mathcal{D}_{\text{syn}} = \{\tilde{x}_i\}_{i=1}^{|\mathcal{D}_{\text{syn}}|}$ ($|\mathcal{D}_{\text{syn}}| \ll |\mathcal{D}_{\text{real}}|$), such that a learner model, parameterized by θ, trained on $\mathcal{D}_{\text{syn}}$ performs well. To achieve this goal, the overall procedure of DiLM is composed of the following three steps:

1. We fist simply train the generator model to generate synthetic training samples that belong to the same distribution as in the original dataset $\mathcal{D}_{\text{real}}$ (Section 3.2).
2. We then fine-tune the generator model to generate "informative" training samples by minimizing the gradient matching loss between generated and real samples (Section 3.3).
3. We obtain distilled dataset $\mathcal{D}_{\text{syn}}$ by generating synthetic samples with the generator model and selecting representative samples from them by using a clustering-based coreset selection method (Section 3.4).

We describe the details of each step in the following sections.

3.2 Synthetic Training Data Generation with Language Model

Inspired by the remarkable text generation capability of pre-trained transformer language models (Radford et al., 2019), we use them as the generator model to generate synthetic training samples of sufficient quality to be used for training models. Before training the generator model to generate more informative synthetic samples than real samples in the original dataset, we first simply train a language model to generate training samples that belong to the same distribution as in the original training dataset for the initial parameters of the generator model.

When we target at text classification tasks, we need to control the generator model to generate samples for each specific class. Therefore, we introduce class-specific beginning-of-sentence tokens `<bos_i>`, which are added to the head of each training sample to train the generator model to generate samples of the corresponding class following it. For each training sample, an end-of-sentence token `<eos>` is also added, and the sample is fed to the generator model as follows:

`<bos_i>` *sentence of class* i `<eos>`.

To involve text classification tasks that specify the relation between two sentences, such as semantic similarity and natural language inference (NLI), we use a separate token `<sep>` to split two sentences as

`<bos_i>` *sentence 1* `<sep>` *sentence 2* `<eos>`.

The generator model is trained on them with the language modeling loss $\mathcal{L}_\phi(x_i)$ as

$$\mathcal{L}_\phi(x_i) = -\frac{1}{|x_i|}\sum_{w_t \in x_i} \log p_\phi(w_t|w_{<t}), \quad (1)$$

where w_t is a token in x_i and $|x_i|$ is the length of x_i. In this way, we pre-train the generator model parameters ϕ to generate synthetic training data like real data, and use them as the initial parameter for training for gradient matching, described in the following section.

3.3 Training for Gradient Matching

In this section, we explain how to fine-tune the pre-trained generator model, described in Section 3.2, to generate synthetic training samples that are more informative than real samples in the original dataset. Specifically, we describe gradient matching, which is an optimization objective for dataset distillation, and the model updating procedure to deal with the discreteness of text. We also introduce two techniques to improve DiLM: representative teacher and diverse mini-batch sampling.

Gradient Matching. To distill the knowledge of the original dataset $\mathcal{D}_{\text{real}}$ into generated synthetic samples from the generator model, we optimize the gradient matching loss (Zhao et al., 2021) as the objective for dataset distillation. Given a mini-batch of real samples $\{x_i\}_{i=1}^M$ and a mini-batch of synthetic samples $\{\tilde{x}_i\}_{i=1}^N$, which is generated from the generator model, the gradient matching loss $\mathcal{L}_{\text{GM}}$ on the learner model parameters θ is calculated as

$$\mathcal{L}_{\text{GM}} = D\left(\nabla_\theta \mathcal{L}_{\text{real}}, \nabla_\theta \mathcal{L}_{\text{syn}}\right) \quad \text{where}$$
$$\mathcal{L}_{\text{real}} = \frac{1}{M}\sum_{i=1}^{M} l_\theta(x_i),\ \mathcal{L}_{\text{syn}} = \frac{1}{N}\sum_{i=1}^{N} l_\theta(\tilde{x}_i), \quad (2)$$

where $l_\theta(\cdot)$ is the loss function for learning tasks such as cross-entropy loss, and $D(\cdot,\cdot)$ is the cosine similarity-based distance function, expressed as

$$D(A,B) = 1 - \frac{A \cdot B}{\|A\|\|B\|}. \quad (3)$$

Following a previous study (Zhao et al., 2021), we separately calculate the gradient matching loss for

Algorithm 1: Optimization for DiLM

Input : $\mathcal{D}_{\text{real}}$: original dataset; ϕ: generator model; θ: learner model; S: # of outer loop; T: # of inner loop; K: # of learner updating loop in each inner step; M: batch size of real data; N: batch size of synthetic data; η: learning rate of θ; α: learning rate of ϕ.

// Outer loop
for $s = 1, \ldots, S$ **do**
  // Initialize learner
  Initialize $\theta \sim p(\theta_0)$
  // Inner loop
  for $t = 1, \ldots, T$ **do**
    // Compute gradient matching loss for each class
    for $c = 1, \ldots, C$ **do**
      // Compute loss with real samples
      $\{x_i^{(c)}\}_{i=1}^{M} \sim \mathcal{D}_{\text{real}}^{(c)}$
      $\mathcal{L}_{\text{real}}^{(c)} \leftarrow \frac{1}{M}\sum_{i=1}^{M} l_\theta(x_i^{(c)})$
      // Compute loss with synthetic samples
      $\{\tilde{x}_i^{(c)}\}_{i=1}^{N} \sim p_\phi(\tilde{x})$
      for $i = 1, \ldots, N$ **do**
        $a_i \leftarrow p_\phi(\tilde{x}_i^{(c)}) / \sum_{j=1}^{N} p_\phi(\tilde{x}_j^{(c)})$
      $\mathcal{L}_{\text{syn}}^{(c)} \leftarrow \sum_{i=1}^{N} a_i l_\theta(\tilde{x}_i^{(c)})$
      // Gradient matching loss (Eq. (3))
      $\mathcal{L}_{\text{GM}}^{(c)} \leftarrow D(\nabla_\theta \mathcal{L}_{\text{real}}^{(c)}, \nabla_\theta \mathcal{L}_{\text{syn}}^{(c)})$
    // Update generator
    $\phi \leftarrow \phi - \alpha \nabla_\phi \frac{1}{C}\sum_{c=1}^{C} \mathcal{L}_{\text{GM}}^{(c)}$
    // Update learner for K steps
    for $k = 1, \ldots, K$ **do**
      $X_{\text{real}} \sim \mathcal{D}_{\text{real}}$
      $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_\theta(X_{\text{real}})$

Output : ϕ: Parameters of generator model.

each class and combine them to update the generator model parameters ϕ. To consider the gradient on the learner model parameters θ throughout the entire training process, the generator model is trained with the nested loop algorithm, including the outer loop which initializes θ at the beginning, and the inner loop which updates θ for K steps with real samples (see Algorithm 1).

Generator Updating. As we described in Section 2.2, the gradient matching loss $\mathcal{L}_{\text{GM}}$ cannot be directly back-propagated to the generator model parameters ϕ via generated samples $\{\tilde{x}_i\}_{i=1}^{N}$, like the case with image datasets, because they consist of discrete text. To address this issue, we design an alternative backward pass, inspired by a previous study (Hiraoka et al., 2020), which optimizes a tokenization model for the downstream task's loss through a non-differentiable procedure. When computing the generated sample's loss $\mathcal{L}_{\text{syn}}$, instead of simply averaging the losses for each generated sample as in Eq. (2), we weight them with their generation probabilities $p_\phi(\tilde{x}_i)$ as

$$\mathcal{L}_{\text{syn}} = \sum_{i=1}^{N} a_i \, l_\theta(\tilde{x}_i), \tag{4}$$

$$a_i = \frac{p_\phi(\tilde{x}_i)}{\sum_{j=1}^{N} p_\phi(\tilde{x}_j)}. \tag{5}$$

Therefore, $\mathcal{L}_{\text{GM}}$ can be back-propagated to ϕ through the differentiable pass via loss weights a_i, as illustrated in Figure 1. Intuitively, the generator model is updated to increase its generation probabilities of synthetic samples that improve gradient similarity.

Representative Teacher. To improve DiLM, we consider enhancing the gradient teacher of real samples by using representative samples for each mini-batch of real samples instead of randomly selected ones. Inspired by Liu et al. (2023), we select the representative samples with K-centers (Wolf, 2011; Sener and Savarese, 2018), a clustering-based coreset selection method (Figure 1a). Specifically, we divide all the real training samples for each class into M sub-clusters by using the K-means algorithm on the feature space of the learner model, and choose the center sample of each sub-cluster. As shown in (Liu et al., 2023), the representative samples selected by K-centers provide the proper teacher gradient by including diverse samples that cover the overall distribution for each class and eliminating samples near the decision boundaries, which have dominant gradients with large norms. Considering coverage and robustness, we generate 10 representative sample sets by running the K-means algorithm with different random seeds at the beginning of training and use one as a mini-batch of real samples in each training step.[1]

Diverse Mini-batch Sampling. Diversity in a mini-batch of generated samples for each step affects the sample space that the generator model explores in training. If the generator model only generates many samples that are similar to each other, this leads to the biased optimization of the generator model. To address this issue, we introduce diverse mini-batch sampling of generated samples in the training process of DiLM (Figure 1b). Instead of generating N synthetic samples for each step, the generator model generates $N \times I_{\text{int}}$ synthetic samples at the same time, where I_{int} is the

[1] Liu et al. (2023) repeatedly re-generated the K-center representative samples by conducting clustering on the feature space of the different learner model's states throughout the inner loop. However, it is very time consuming with BERT as the learner model, as in our study.

	SST-2 (2 classes, 67.3k)			**QQP** (2 classes, 364k)			**MNLI-m** (3 classes, 393k)		
Data/class	5	10	20	5	10	20	5	10	20
Random	58.1±5.2	64.3±7.4	70.3±6.8	51.5±5.6	56.0±4.8	59.1±3.8	35.6±2.1	37.7±2.6	40.1±3.2
K-centers	70.8±4.1	75.9±4.7	79.8±3.5	**60.7±3.8**	60.9±3.1	62.6±2.7	36.2±2.4	41.8±3.2	45.3±3.0
Herding	70.2±5.7	73.2±5.7	76.9±4.4	56.0±5.6	59.7±4.1	62.3±3.4	36.2±3.8	38.7±3.7	42.8±3.5
TDD (embed.)	89.6±0.4	-	-	81.5±0.2	-	-	75.6±0.2	-	-
TDD (text)	50.2±1.6	-	-	39.6±6.8	-	-	33.4±1.8	-	-
Vanilla LM	65.2±6.8	71.7±6.8	77.6±4.1	56.7±4.4	59.3±3.8	62.5±3.3	36.3±2.7	40.5±2.9	43.6±3.1
DiLM	**72.5±5.9**	**76.3±4.6**	**80.3±2.8**	58.8±5.2	**62.2±3.3**	**64.4±2.6**	**39.7±2.7**	**44.8±3.1**	**48.7±2.6**
Full dataset		92.7			89.6			86.7	

Table 1: Performance comparison of DiLM with coreset selection methods and TDD for training the $\text{BERT}_{\text{BASE}}$ model. Green highlighted results indicate that DiLM outperformed the coreset selection methods. Red highlighted results indicate performance degradation of distilled datasets from TDD after being converted to text. Note that we could not conduct the experiments for TDD with larger DPC settings due to GPU memory requirements.

generation interval. The generated synthetic samples are then divided into N sub-clusters with the K-means algorithm, and a mini-batch of synthetic samples for each step is constructed by randomly choosing one sample from each sub-cluster.

3.4 Generate Synthetic Dataset

We obtain distilled dataset $\mathcal{D}_{\text{syn}}$ by generating synthetic samples with the trained generator model. To include representative samples of the model's generative distribution $p_\phi(\tilde{x})$, we use the coreset selection method again to select generated synthetic samples. Specifically, we generate 100 times as many synthetic samples as the distilled dataset size $|\mathcal{D}_{\text{syn}}|$ by top-p sampling with $p = 0.95$, considering the diversity, and then construct $\mathcal{D}_{\text{syn}}$ with K-center representative samples. This makes $\mathcal{D}_{\text{syn}}$ to include diverse synthetic samples by removing redundant samples caused by the biased generative distribution of the model.

4 Experimental Settings

Datasets. We evaluated DiLM in distilling three major text classification datasets, SST-2, QQP, and MNLI-m, from the GLUE benchmark (Wang et al., 2018a). Following Wang et al. (2018a), we report accuracy for SST-2 and MNLI-m, and the average of accuracy and F1 score for QQP as our results. More details about each dataset are shown in Appendix A.

Baselines. Following previous studies on dataset distillation in the CV field, we compared the performance of DiLM with three coreset selection methods, Random, K-centers (Wolf, 2011; Sener and Savarese, 2018), and Herding (Welling, 2009), as well as TDD (Sucholutsky and Schonlau, 2021), which is a recent embedding-level distillation method. Note that TDD also trains the learnable soft-labels and learning rates for each training step together with the input word embeddings. We also evaluated the vanilla LM, which only learns the synthetic training data generation (Section 3.2), to validate the effectiveness of the training for gradient matching, described in Section 3.3. The details of each baseline are given in Appendix B.

Evaluation. For evaluation, we used $\text{BERT}_{\text{BASE}}$ and other three pre-trained models, $\text{RoBERTa}_{\text{BASE}}$, $\text{BERT}_{\text{LARGE}}$, and $\text{XLNet}_{\text{BASE}}$, as learner models (see more details in Appendix C). We trained a learner model on the distilled datasets for 200 steps by using AdamW (Loshchilov and Hutter, 2019) with a learning rate of 1.0×10^{-4} and a batch size of 64.[2] For Herding and TDD, we trained the learner model on their datasets for 100 times. For other methods, we generated 20 datasets with different random seeds and trained the learner model on each of them for 5 times. We report the average and standard deviation for these 100 models.

Implementation. We used the 128M parameter version of GPT-2[3] (Radford et al., 2019) as the generator model of DiLM, and used $\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019) as the learner model, on which we calculated the gradient matching loss. To reduce the computational costs, we calculated the gradient matching loss only for the randomly initialized last layer parameters, which tend to have dominantly larger gradient than the pre-trained parameters. We set the number of each loop for training DiLM

[2]We did not follow this training protocol for TDD, since TDD optimizes learning rates as well for each step with a specific synthetic sample order.

[3]https://huggingface.co/gpt2

to $S = 2000$, $T = 10$, and $K = 20$, and the generation interval to $I_{\text{int}} = 200$ according to our preliminary experiments. The mini-batch size of real and synthetic samples were respectively set to $M = 200$ and $N = 64$. More details of our implementation are given in Appendix D.

5 Results and Discussion

5.1 Performance for BERT$_{\text{BASE}}$

As shown in Table 1, we first compared DiLM with the other baselines for training BERT$_{\text{BASE}}$, on which DiLM trained gradient matching. We evaluated them for different sizes of distilled synthetic datasets of 5/10/20 data-per-class (DPC) settings.

We first found that the vanilla LM, which was only trained for synthetic training sample generation without gradient matching, clearly underperformed the coreset selection methods. This indicates that, as can be expected, the quality of the generated synthetic samples becomes lower than that of real samples in the original datasets. However, DiLM, which fine-tuned the vanilla LM with gradient matching, improved its performance and even outperformed the coreset selection methods overall. Note that the performance gains from K-centers indicate that DiLM generated synthetic training samples that are more effective for model training than the real samples in the original datasets.

When focusing on the difference between the three datasets, the performance gains of DiLM on QQP and MNLI-m were larger than that on SST-2. We believe this is because QQP and MNLI-m, which are the tasks to specify the relationship between two sentences, are intuitively less likely to have real samples that represent the task than SST-2, which is a relatively simple negative/positive classification task. In addition, it may also be related to the size of the original training dataset of QQP and MNLI-m, which is five times larger than that of SST-2. Since the generator model was trained by gradient matching with self-generated synthetic samples, it can explore broader sample space by pre-training with the original dataset that contains enough diversity samples, which results in the effective performance of DiLM.

For TDD, we also evaluated its distilled datasets as text data by converting them to discrete tokens that have nearest neighbor embeddings. When directly using the distilled datasets as word embeddings, TDD achieved remarkable performance even compared with the full datasets. However, after converting to text, its performance catastrophically degraded even to the lower-bound performances with random prediction. This suggests that the distilled datasets from TDD are strictly overfitted at the word embedding level and cannot be converted to text without acceptable performance degradation, which is necessary for applying them to other models. This point is the clear advantage of DiLM, which distills synthetic datasets at the text-level.

Dataset	Model	Random	K-centers	DiLM
SST-2	BERT$_{\text{BASE}}$ (**S**)	70.3±6.8	79.8±3.5	80.3±2.8
	RoBERTa$_{\text{BASE}}$	74.4±5.3	73.9±5.2	**78.1±3.8**
	BERT$_{\text{LARGE}}$	74.7±8.4	80.4±9.1	**83.1±6.2**
	XLNet$_{\text{BASE}}$	69.9±6.2	71.8±5.8	**77.9±4.7**
QQP	BERT$_{\text{BASE}}$ (**S**)	59.1±3.8	62.6±2.7	64.4±2.6
	RoBERTa$_{\text{BASE}}$	60.1±4.0	63.9±3.2	**66.4±2.3**
	BERT$_{\text{LARGE}}$	58.8±6.9	59.0±8.9	**62.9±8.6**
	XLNet$_{\text{BASE}}$	59.1±3.5	60.9±3.0	**64.4±2.2**
MNLI-m	BERT$_{\text{BASE}}$ (**S**)	40.1±3.2	45.3±3.0	48.7±2.6
	RoBERTa$_{\text{BASE}}$	39.6±2.5	44.5±2.6	**45.0±2.8**
	BERT$_{\text{LARGE}}$	40.9±4.5	48.7±4.2	**49.6±4.4**
	XLNet$_{\text{BASE}}$	39.0±2.0	43.5±2.7	**44.7±2.7**

Table 2: Cross-model generalization performance for settings of DPC=20. (**S**) indicates the source model for gradient matching of DiLM and feature extractor for K-centers.

Models	Random	K-centers	DiLM
GPT-2-XL (1.5B)	64.8±12.0	64.8±13.3	**71.1±13.0**
OPT (2.7B)	89.3±5.9	91.5±3.1	**92.7±1.9**
Llama 2 (7B)	93.6±2.9	94.6±0.7	**95.1±0.7**

Table 3: Performance of distilled datasets as 5-shot prompts for in-context learning of SST-2. Each score is the average and standard deviation for 100 prompts with 20 distilled datasets and 5 random orders.

5.2 Cross-model Generalization

In contrast to the current embedding-level distillation methods, text-level synthetic datasets from DiLM can be leveraged for training different models independent of their word embedding weights. To emphasize this advantage, we evaluated the distilled synthetic datasets for training three models different from BERT$_{\text{BASE}}$, with which the distilled synthetic datasets were obtained, i.e., RoBERTa$_{\text{BASE}}$, BERT$_{\text{LARGE}}$, and XLNet$_{\text{BASE}}$. Table 2 summarizes the performances of Random, K-centers, and DiLM with DPC=20, where DiLM achieved stably good performances.[4] The results

[4]We also show the results with other DPC settings in Appendix E.

	RT	DMS	SST-2	QQP	MNLI-m
DiLM	✓	✓	**72.5 ± 5.9**	**58.8 ± 5.2**	**39.7 ± 2.7**
	✓	-	71.3 ± 5.6	57.5 ± 4.4	38.8 ± 3.0
	-	✓	70.9 ± 5.9	57.6 ± 5.0	39.5 ± 2.8
	-	-	69.2 ± 6.2	57.7 ± 5.2	38.3 ± 2.8

Table 4: Ablation study on the performance improvement techniques of DiLM with the DPC=5 setting. RT and DMS indicates representative teacher and diverse mini-batch sampling, respectively.

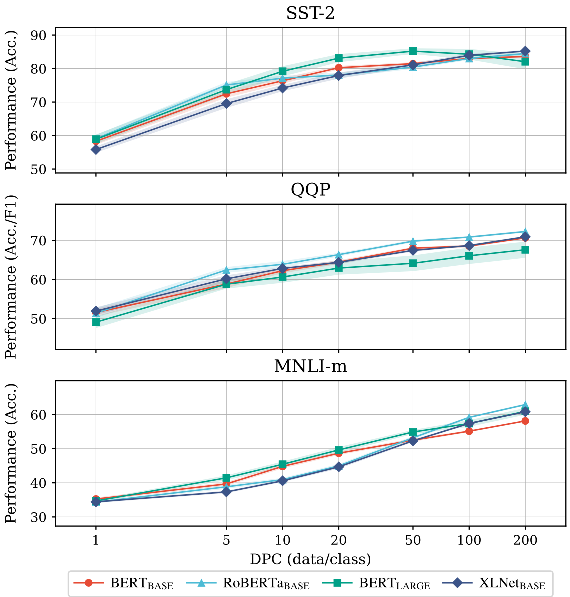

Figure 2: Performance for increasing number of synthetic samples with DPC $\in \{1, 5, 10, 20, 50, 100, 200\}$. We plot the mean and 95% confidence interval for 100 models trained on distilled datasets from DiLM.

indicate that the distilled datasets from DiLM consistently performed well for training the different models, even though DiLM trained gradient matching only for the $\text{BERT}_{\text{BASE}}$ model's parameters. It is worth noting that our distilled datasets show successful generalization performance not only for training $\text{RoBERTa}_{\text{BASE}}$ and $\text{BERT}_{\text{LARGE}}$, which have the same model architecture as $\text{BERT}_{\text{BASE}}$, but also for training $\text{XLNet}_{\text{BASE}}$, which is an autoregressive model using the hidden state of the `<eos>` token for classification, while $\text{BERT}_{\text{BASE}}$ is an autoencoding model using the hidden state of the `[CLS]` token.

We also evaluated the distilled datasets from DiLM as few-shot prompts for in-context learning of LLMs. Table 3 shows the performance of Random, K-centers, and DiLM for in-context learning for SST-2 with three different sizes of LLMs, GPT-2-XL (Radford et al., 2019), OPT (Zhang et al., 2022b), and Llama 2 (Touvron et al., 2023). Surprisingly, the distilled datasets from DiLM consistently performed well for the in-context learning, compared with Random and K-centers.

These remarkable generalization performances across models and training processes strongly support the advantage of DiLM to distill datasets at the text-level.

5.3 Analysis and Discussion

Ablation Study. Table 4 shows the results of the ablation study for the performance improvement techniques of the representative teacher for gradient matching and the diverse mini-batch sampling of synthetic samples during training of DiLM. The results indicate that both two techniques are consistently effective for DiLM.

Scaling of DPC. We investigated the performance of DiLM when increasing the size of synthetic datasets. Note that DiLM does not require retraining the generator model for generating distilled synthetic datasets for different DPCs, which is also the advantage of using generative models for dataset distillation. As shown in Figure 2, the performance of the distilled datasets generally scaled with increasing DPC.

Distilled Data Examples. We gave examples of distilled synthetic samples for each dataset in Appendix F. We found that DiLM successfully generated interpretable synthetic samples that are appropriate for the tasks of the original datasets. Although DiLM consistently generated high quality synthetic samples for SST-2 and QQP, the repetition problem can be observed in some lengthy samples for MNLI-m. This suggests that there is still room for performance improvements of DiLM by using a larger and more sophisticated pre-trained language model for the generator model than the small GPT-2 used in our current experiments.

6 Conclusion

We proposed the first text-level dataset distillation approach, called DiLM, which trains a language model to generate informative synthetic samples as text data for model-agnostic applicability and interpretability of distilled datasets. Experimental results across various text classification datasets indicated that the distilled datasets from DiLM achieve successful performance for training various types of models beyond the source model used for distillation, even for in-context learning of LLMs.

Limitations

The following three points are the limitations of this work. (i) Although DiLM achieved remarkable performance as a text-level distillation method, there is still a performance gap from the full datasets. DiLM has room for the performance improvement by employing larger and more sophisticated pre-trained language models as the generator model or using other dataset distillation objectives as an alternative to the gradient matching. (ii) In our experiments, we applied DiLM to distill only text classification task datasets. DiLM can be applied to text generation tasks as well by just considering the entire original training dataset as the data for a single label. In future work, we should explore the application of DiLM for more difficult settings, such as the text generation tasks and full-scratch training of language models. (iii) While privacy preservation of the original training datasets is one of the applications of dataset distillation, it is difficult to apply DiLM to the privacy preservation because the distilled synthetic datasets from DiLM may include real samples from the original dataset due to the training data memorization of the language model. However, we believe that the advantage of DiLM to generate distilled synthetic datasets at the text-level, enabling the training of models independent of word embedding weights, is more valuable than the application to the privacy preservation in terms of practical applications.

References

Ondrej Bohdal, Yongxin Yang, and Timothy M. Hospedales. 2020. Flexible dataset distillation: Learn labels instead of images. *CoRR*, abs/2006.08572.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2022, New Orleans, LA, USA, June 19-20, 2022*, pages 4749–4758. IEEE.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. 2023. Generalizing dataset distillation via deep generative prior. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 3739–3748. IEEE.

Dingfan Chen, Raouf Kerkouche, and Mario Fritz. 2022. Private set generation with discriminative information. In *Advances in Neural Information Processing Systems*, volume 35, pages 14678–14690. Curran Associates, Inc.

Justin Cui, Ruochen Wang, Si Si, and Cho-Jui Hsieh. 2023. Scaling up dataset distillation to ImageNet-1K with constant memory. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 6565–6590. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Tian Dong, Bo Zhao, and Lingjuan Lyu. 2022. Privacy for free: How does dataset condensation help privacy? In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5378–5396. PMLR.

Jiahui Geng, Zongxiong Chen, Yuandou Wang, Herbert Woisetschlaeger, Sonja Schimmler, Ruben Mayer, Zhiming Zhao, and Chunming Rong. 2023. A survey on dataset distillation: Approaches, applications and future directions. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 6610–6618. International Joint Conferences on Artificial Intelligence Organization. Survey Track.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc.

Tatsuya Hiraoka, Sho Takase, Kei Uchiumi, Atsushi Keyaki, and Naoaki Okazaki. 2020. Optimizing word segmentation for downstream task. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1341–1351, Online. Association for Computational Linguistics.

Yongqi Li and Wenjie Li. 2021. Data distillation for text classification. *CoRR*, abs/2104.08448.

Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. 2023. Dream: Efficient dataset distillation by representative matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17314–17324.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Dougal Maclaurin, David Duvenaud, and Ryan P. Adams. 2015. Gradient-based hyperparameter optimization through reversible learning. In *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 2113–2122. JMLR.org.

Aru Maekawa, Naoki Kobayashi, Kotaro Funakoshi, and Manabu Okumura. 2023. Dataset distillation with attention labels for fine-tuning BERT. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 119–127, Toronto, Canada. Association for Computational Linguistics.

Dmitry Medvedev and Alexander D'yakonov. 2021. Learning to generate synthetic training data using gradient matching and implicit differentiation. In *Recent Trends in Analysis of Images, Social Networks and Texts - 10th International Conference, AIST 2021, Tbilisi, Georgia, December 16-18, 2021, Revised Supplementary Proceedings*, volume 1573 of *Communications in Computer and Information Science*, pages 138–150. Springer.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Shivam Sahni and Harsh M. Patel. 2023. Exploring multilingual text data distillation. *CoRR*, abs/2308.04982.

Mattia Sangermano, Antonio Carta, Andrea Cossu, and Davide Bacciu. 2022. Sample condensation in online continual learning. In *International Joint Conference on Neural Networks, IJCNN 2022, Padua, Italy, July 18-23, 2022*, pages 1–8. IEEE.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net.

Felipe Petroski Such, Aditya Rawal, Joel Lehman, Kenneth Stanley, and Jeffrey Clune. 2020. Generative teaching networks: Accelerating neural architecture search by learning to generate synthetic training data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9206–9216. PMLR.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018a. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Kai Wang, Jianyang Gu, Daquan Zhou, Zheng Zhu, Wei Jiang, and Yang You. 2023. Dim: Distilling dataset into generative model. *CoRR*, abs/2303.04707.

Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. 2022. Cafe: Learning to condense dataset by aligning features. In *2022*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 12186–12195.

Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A. Efros. 2018b. Dataset distillation. *CoRR*, abs/1811.10959.

Max Welling. 2009. Herding dynamical weights to learn. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 1121–1128, New York, NY, USA. Association for Computing Machinery.

Felix Wiewel and Bin Yang. 2021. Condensed composite memory continual learning. In *International Joint Conference on Neural Networks, IJCNN 2021, Shenzhen, China, July 18-22, 2021*, pages 1–8. IEEE.

Gert W. Wolf. 2011. Facility location: concepts, models, algorithms and case studies. series: Contributions to management science. *Int. J. Geogr. Inf. Sci.*, 25(2):331–333.

Yuanhao Xiong, Ruochen Wang, Minhao Cheng, Felix Yu, and Cho-Jui Hsieh. 2023. Feddm: Iterative distribution matching for communication-efficient federated learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 16323–16332. IEEE.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 5754–5764.

Ruonan Yu, Songhua Liu, and Xinchao Wang. 2023. Dataset distillation: A comprehensive review. *CoRR*, abs/2301.07014.

Jie Zhang, Chen Chen, Bo Li, Lingjuan Lyu, Shuang Wu, Shouhong Ding, Chunhua Shen, and Chao Wu. 2022a. Dense: Data-free one-shot federated learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 21414–21428. Curran Associates, Inc.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022b. OPT: open pre-trained transformer language models. *CoRR*, abs/2205.01068.

Bo Zhao and Hakan Bilen. 2021. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12674–12685. PMLR.

Bo Zhao and Hakan Bilen. 2022. Synthesizing informative training samples with GAN. *CoRR*, abs/2204.07513.

Bo Zhao and Hakan Bilen. 2023. Dataset condensation with distribution matching. In *IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2023, Waikoloa, HI, USA, January 2-7, 2023*, pages 6503–6512. IEEE.

Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. 2021. Dataset condensation with gradient matching. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

A Datasets

We used three text classification datasets in the GLUE benchmark (Wang et al., 2018a) from huggingface datasets.[5] SST-2 is a banally sentiment classification (negative/positive) task for movie review sentences. QQP is a task to identify whether a question pair is semantically equivalent or not. MNLI-m is a natural language inference task to predict a premise sentence entails or contradicts a hypothesis sentence or neither (neutral). We reported the evaluation results on the validation set in Section 5, since the test set is not publicly available. For MNLI-m, we used the matched-domain validation set for evaluation. We summarizes the statistics of each dataset in Table 5.

Dataset	Metric	#Train	#Dev	#Class
SST-2	accuracy	67k	872	2
QQP	accuracy/F1	364k	40k	2
MNLI-m	accuracy	393k	9.8k	3

Table 5: Summary of statistics of evaluation datasets

B Baselines Details

In this section, we explain the details of the baseline methods used in our experiments.

B.1 Coreset Selection

Random is the simplest baseline, which randomly selects real samples from the original training dataset.

K-centers (Wolf, 2011; Sener and Savarese, 2018) is a standard coreset selection method that selects the center samples of sub-clusters as a coreset, which eliminates redundant samples and covers the distribution of the original dataset.

Herding (Welling, 2009) is also a standard coreset selection method that greedily selects real samples to match their mean embedding with that of the original dataset.

For K-centers and Herding, we used the last hidden state of the `[CLS]` token in the $\text{BERT}_{\text{BASE}}$ model as a feature of each training sample.

B.2 Embedding-level Dataset Distillation

TDD[6] (Sucholutsky and Schonlau, 2021) is the current embedding level text dataset distillation method. TDD also optimizes learnable soft-labels and learning rates together with input word embeddings by the original meta-learning approach (Wang et al., 2018b). Following the best performing settings in Maekawa et al. (2023), which applied this approach to the BERT model, we used one synthetic sample per class as a mini-batch of a single gradient step and fixed the order of synthetic samples, which means the learner model is trained with 5 gradient steps in the experiments in Section 5 with DPC=5. Similar to DiLM, TDD also used $\text{BERT}_{\text{BASE}}$ as the learner model for distillation.

[5] https://huggingface.co/datasets/glue

[6] We used the implementation by Maekawa et al. (2023), because it also employs BERT as the learner model.

C Learner Models

$\text{BERT}_{\text{BASE}}$[7] (Devlin et al., 2019) was used as the source model for training for dataset distillation and the feature extractor of the coreset selection methods. Following the fine-tuning settings in Devlin et al. (2019), we used a randomly initialized linear layer on the top of the last hidden state of the `[CLS]` token.

$\text{RoBERTa}_{\text{BASE}}$[8] is a BERT derivative model proposed by Liu et al. (2019). This model has the same size and architecture as $\text{BERT}_{\text{BASE}}$, but has different parameters pre-trained with the masked language modeling (MLM) task, without the next sentence prediction (NSP) task, on a larger corpus than the BERT models.

$\text{BERT}_{\text{LARGE}}$[9] is the 24 layer, 340M parameter version of BERT, while $\text{BERT}_{\text{BASE}}$ has 12 layers and 110M parameters.

$\text{XLNet}_{\text{BASE}}$[10] is an autoregressive model in contrast to BERT and RoBERTa. Following (Yang et al., 2019), we used a randomly initialized linear layer on the top of the last hidden state of the `<eos>` token, which involves entire tokens in the sequence.

D Implementation Details

Table 6 shows the details of hyperparameter settings in our experiments. Our implementation was based on PyTorch 2.1.0, and we used pre-trained models from Hugging Face Transformers 4.30.0. All model training and evaluation in our experiments were conducted with the half-precision (BFloat16) on a single RTX 3090 (24GB), RTX

[7] https://huggingface.co/bert-base-uncased

[8] https://huggingface.co/roberta-base

[9] https://huggingface.co/bert-large-uncased

[10] https://huggingface.co/xlnet-base-cased

A6000 (48GB), or A100 PCIe (80GB) according to the required GPU memory size for each experiment.

Pre-training settings of DiLM	
Optimizer	AdamW
Learning rate	1.0×10^{-5}
Learning rate scheduler	Linear warm-up and cosine annealing
Warmup ratio	0.05
Waight decay	0.01
Gradient clipping	1.0
Dropout ratio	0.1
# of training steps	80,000
Batch size	64
Fine-tuning settings of DiLM	
Optimizer	AdamW
Learning rate	3.0×10^{-7}
Learning rate scheduler	Linear warm-up and cosine annealing
Warmup ratio	0.05
Waight decay	0.01
Gradient clipping	1.0
Dropout ratio	0.1
# of outer loop (S)	20,000
# of inner loop (T)	10
# of learner updating steps (K)	20
Batch size of real samples (M)	200
Batch size of synthetic samples (N)	64
Generation interval (I_{int})	200
Learner training settings for evaluation	
Oprimizer	AdamW
Learning rate	1.0×10^{-4}
Learning rate scheduler	Linear warm-up and cosine annealing
Warmup ratio	0.5
Waight decay	0.01
Gradient clipping	1.0
Dropout ratio	0.1
# of training steps	200
Batch size	64

Table 6: Hyperparameter settings in our experiments

E Results for Cross-model Generalization

Tables 7 and 8 show the cross-model generalization performances with DPC=5,10 settings. As in the setting of DPC=20 in Table 2, DiLM also performed well in training different models than the source model.

F Distilled Synthetic Data Examples

We gave examples of distilled synthetic samples from DiLM in Tables 9, 10, and 11. Generated synthetic examples with DiLM were interpretable and seem to represent the tasks of the original training dataset.

Dataset	Model	Random	K-centers	DiLM
SST-2	$\text{BERT}_{\text{BASE}}$ (**S**)	58.1±5.2	70.8±4.1	72.5±5.9
	$\text{RoBERTa}_{\text{BASE}}$	60.6±7.6	74.2±4.9	**75.1±4.6**
	$\text{BERT}_{\text{LARGE}}$	60.4±8.4	70.0±8.2	**73.7±8.4**
	$\text{XLNet}_{\text{BASE}}$	57.0±5.5	66.4±5.0	**69.5±6.6**
QQP	$\text{BERT}_{\text{BASE}}$ (**S**)	51.5±5.6	60.7±3.8	58.8±5.2
	$\text{RoBERTa}_{\text{BASE}}$	52.5±6.0	**63.9±3.3**	62.4±3.7
	$\text{BERT}_{\text{LARGE}}$	53.3±6.7	58.3±5.8	**58.8±5.7**
	$\text{XLNet}_{\text{BASE}}$	52.6±5.2	**62.6±3.1**	60.2±4.6
MNLI-m	$\text{BERT}_{\text{BASE}}$ (**S**)	35.6±2.1	36.2±2.4	39.7±2.7
	$\text{RoBERTa}_{\text{BASE}}$	35.8±2.1	37.4±2.1	**38.8±3.0**
	$\text{BERT}_{\text{LARGE}}$	36.9±2.8	37.4±2.9	**41.5±3.7**
	$\text{XLNet}_{\text{BASE}}$	35.4±1.4	37.0±1.5	**37.3±1.9**

Table 7: Cross-model generalization performance for the setting of DPC=5. (**S**) indicates the source model for gradient matching of DiLM and feature extractor for K-centers.

Dataset	Model	Random	K-centers	DiLM
SST2	$\text{BERT}_{\text{BASE}}$ (**S**)	64.3±7.4	75.9±4.7	76.3±4.6
	$\text{RoBERTa}_{\text{BASE}}$	68.6±7.1	74.6±5.6	**77.1±4.1**
	$\text{BERT}_{\text{LARGE}}$	67.2±8.5	76.6±8.4	**79.2±7.8**
	$\text{XLNet}_{\text{BASE}}$	63.7±7.5	68.0±6.1	**74.2±4.9**
QQP	$\text{BERT}_{\text{BASE}}$ (**S**)	56.0±4.8	60.9±3.1	62.2±3.3
	$\text{RoBERTa}_{\text{BASE}}$	56.4±5.3	**64.0±2.7**	63.9±4.3
	$\text{BERT}_{\text{LARGE}}$	53.7±8.5	59.4±5.6	**60.6±7.5**
	$\text{XLNet}_{\text{BASE}}$	55.0±4.5	61.4±3.2	**62.8±2.2**
MNLI	$\text{BERT}_{\text{BASE}}$ (**S**)	37.7±2.6	41.8±3.2	44.8±3.1
	$\text{RoBERTa}_{\text{BASE}}$	37.1±2.2	**42.1±2.6**	40.9±2.6
	$\text{BERT}_{\text{LARGE}}$	39.7±3.6	43.4±4.4	**45.4±4.1**
	$\text{XLNet}_{\text{BASE}}$	37.0±1.4	**41.5±2.6**	40.6±1.9

Table 8: Cross-model generalization performance for the setting of DPC=10. (**S**) indicates the source model for gradient matching of DiLM and feature extractor for K-centers.

Label	Sentence
negative	is too amateurishly square to work as storytelling, and the ensemble cast lacks depth and resonance.
	is so lousy that you can not enjoy it
	incredibly lifeless, with the lack-of-attention span
	the script's contrived, lame screenplay and listless direction are just the ticket cost.
	a cheap scam that only weak claims to dramatic impact and creepy-crawly humor.
positive	is a wonderous accomplishment of veracity and narrative grace.
	very best
	a fully realized story with keen insights into parapsychological phenomena and the soulful nuances of the grieving process
	it one of the best-sustained ideas i have ever seen on the screen.
	a surprisingly sweet, tender drama that does a superb job contrasting the sleekness of the film's present with the playful paranoia of the film's past.

Table 9: Distilled synthetic samples for SST-2 with DPC=5

Label	Question 1	Question 2
not duplicate	Why should I write a good backmatter for an international conference?	Where can I study internationally on business logic?
	How long does it take you to learn the German language?	How long does it take to learn the English language?
	What are some unexpected things first-time visitors to Colombia notice?	What are some unexpected things first-time visitors to Canada notice?
	Why is red in PFUS something I can't see when I tap PFUS?	Did one have a chance to see one of the real masterpieces being played by Richard Bachardo in MS Dhoni Cricket: Live Streaming, in the Permanent XI Test Center at Mumbai?
	How does digital gatekeeper disable ads on a WiFi band?	How can I enabledisable my WiFi network on my HTC phone?
duplicate	How do I recover my Gmail account after recovery?	How do I recover my Gmail account from recovery?
	How do you prevent hair loss without touching hair?	How do I prevent hair loss without touching hair?
	How do I get successful in C.E.?	How can I get successful in C.E.?
	What is the best word or link you use to explain the meaning of a certain book to a friend?	What is the best word or link you use to explain the meaning of a certain book to a friend?
	How will the ban of Rs 500 and Rs 1000 notes affect Indian economy?	How will the 500 and 1000 rupee notes ban affect the Indian economy?

Table 10: Distilled synthetic samples for QQP with DPC=5

Label	Premise	Hypothesis
entailment	Guess we are all here, friends.	We were all here, friends.
	The costs to the Service, often estimated to be between $100 and $150 million, will be higher because of the reduced volume of post-1991 pleadings by six states and 28 other states requiring service members to produce basic records electronically.	Costs to the Service are higher because of reduced volume of post-1991 pleadings by six states and 28 other states requiring service members to produce basic records electronically.
	uh-huh is that right because like i say a lot of people tell me we could make it cheaper if we wanted but we didn't i mean our family life is just so far so far that	It seems that a lot of people tell me that it could be cheaper if we wanted but we don't really think we could make it cheaper.
	However, the CEF report suggested that some of the following could serve to reduce the burden on small entities with federally or nonfederal support for compliance with the rule and to minimize the number of affected entities receiving small reductions of federal payments.	Some things could be considered part of the CEF report for reducing burdens on small entities.
	If you are a casino business owner looking to expand your profits, opportunities and experiences, or even to retain some intellectual property you acquired during your travels in other countries, it is best to visit Cancio, Parnell's (National Cancia) resort in Montego Bay, where prices and travel policies range from a very reasonable $50.	The casino has plenty of opportunities you can expand your profits with in Cancio, Parnell's resort.
neutral	oh in that case you have to give them uh six months to come and you know and let them go on	They don't have to get their first six months if they return.
	This is highly valued nationally because of its steeply pro-retirement payment culture, which is perceived as a great success rate by the profession and outside of its area of employment, particularly among the field's young professionals.	Out of all the fields in the population, it is highly valued by the professional community because it provides confidence that the community will care more about its growth.
	yeah right now i i still wish they were a little more	The idea of having people tell us what to do is good for their business and prospects.
	In fact, there is one wonder why Republican leaders are afraid to mention his name.	Republican leaders are not afraid of his name because he is in need of attention.
	To me, it's an excellent system.	I think it could be a good system for a number of reasons.
contradiction	yeah well you know i can't i can't i know sometimes i just i'll remember remembering for once the former minister might be sympathetic to some of the Serbian government cases that they might say well there's no way out um no matter what their approach to the possibility of a peace dividend a lot of people i think i think are are willing to compromise and and to stand up and say who's right and who's wrong and i think it's a good idea and	I can't recall the minister's views on different Serbian government cases.
	I suppose you could say, if it were not for the gleam of light in the hour of your death-boom, that the fatal effects were of a furtive rather than a ferocious nature?	I don't think you could confirm it is a furtive either.
	i think something has to change there	They have no plans at all to change.
	The revisions take into account the range of factors that varying units of measure represent when evaluating new disclosure requirements and when determining whether it should be possible to offer various types of similar products for different reasons.	The revisions go against the current practice and do not consider whether it should be possible to offer different types of similar products for different reasons.
	yeah i uh i uh i don't think there's that's a bad place to live in some part of the world and do everything else that it's really not because people have gotten up in arms but it's all it's all a lot of money to run a very very wealthy individual home	I don't think we should be buying a very wealthy home in an undeveloped area in the developed world.

Table 11: Distilled synthetic samples for MNLI-m with DPC=5