# Project Final Report: Dataset Condensation for Text and Diverse Data Modalities

**Vishesh Agrawal, Shivin Dass, Guarav Nuti, Rafael Sanchez**
DeepTrojans

## Abstract

Computational cost of state-of-the art models is rapidly growing as models are being trained on larger datasets. A recent direction to reduce computation and storing costs of large datasets is dataset condensation in which a large dataset is replaced with a significantly smaller learned synthetic set while preserving the original information. This enables faster training times on smaller synthetic sets while achieving a similar level of performance. Dataset condensation has only been applied to image datasets to the best of our knowledge and our objective is extending it to language and explore some of its applications.

## 1 Introduction

Large datasets are becoming the norm to obtain state-of-the-art machine learning models in multiple fields including computer vision, natural language processing and speech recognition. At such scales, not only training the models, but also storing and preprocessing the data becomes expensive and burdensome. An effective way to deal with large data is to represent them using a smaller number of more manageable samples that contains a similar amount of information. There are several approaches that explore reducing the size of the datasets such as coreset selection, dataset distillation and dataset condensation. In this work we extend dataset condensaion using distribution matching (Zhao and Bilen, 2023), one of the best performing techniques in this field to text.

Further, as demonstrated by Zhao and Bilen (2023), a smaller representative dataset can not only reduce costs of storing data and training models, but is efficient in relieving the catastrophic forgetting problem in continual learning. Hence along with applying dataset condensation to continual learning on text, we additionally explore a novel application of dataset condensation in the field of federated learning.

## 2 Related Work

### 2.1 Coreset Selection

Coresets refer to a smaller representative set of a large dataset (Feldman, 2020). The objective of coreset selection algorithms is to sub-sample the large dataset into a smaller set of samples while preserving as much information from the original dataset as possible. Typically, coreset selection methods choose samples that are important for training based on heuristic criteria, for example, minimizing distance between coreset and whole-dataset centers (Belouadah and Popescu, 2020; Castro et al., 2018; Rebuffi et al., 2017; Chen et al., 2010), maximizing the diversity of selected samples (Aljundi et al., 2019), discovering cluster centers (Farahani and Hekmatfar, 2009; Sener and Savarese, 2018) and counting mis-classification frequency (Toneva et al., 2019).

Although coreset selection methods can be very computationally efficient, they have two major limitations. First, most methods incrementally and greedily select samples, which are short-sighted. Second, their efficiency is upper bounded by the information in the selected samples in the original dataset.

### 2.2 Dataset Distillation

Dataset distillation (Wang et al., 2018) is a relatively new field for representing large datasets as a smaller number of synthetically generated samples. Wang et al. (2018) pose the dataset distillation problem modelling the synthetic data as learnable parameters of the model and optimizing it using the loss from the target task. Dataset distillation has been applied to synthesizing labels (Bohdal et al., 2020; Sucholutsky and Schonlau, 2021) and has also been shown to perform well on image as well as language datasets (Sucholutsky and Schonlau, 2021).

In contrast to coreset selection, dataset distilla-

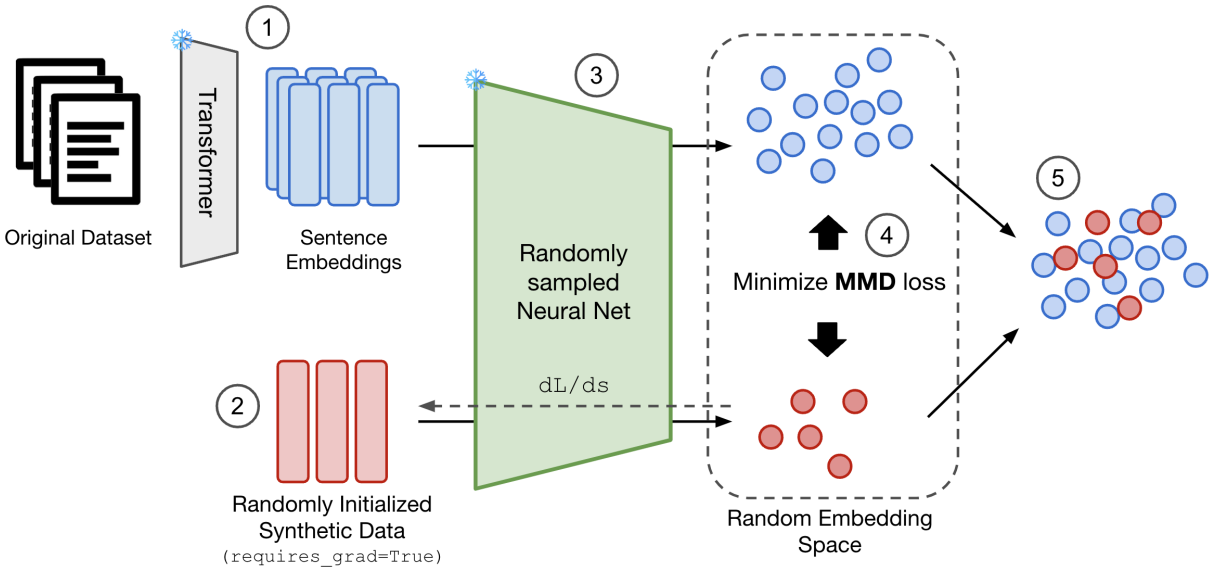



Figure 1: Training loop for text based dataset condensation. 1) Embed the real text dataset into sentence embeddings 2) Initialize the synthetic data 3) Randomly sample the weights of the neural net and process real and synthetic embeddings independently 4) Compute the MMD loss in the embedding space and backpropogate the loss back to the synthetic data 5) After repeating steps 3-4 for N epochs, the embedded distributions of real and synthetic data converge.

tion methods are not bounded by the information in the samples of the original dataset. While datasets synthesized using dataset distillation perform better than coreset construction approaches, they typically require expensive optimization of network weights in the inner loop and hence, struggle with large datasets.

### 2.3 Dataset Condensation

Recent work has shown that some of the limitations of dataset distillation, such as its costly optimization procedure, can be avoided by optimizing a surrogate objective such as matching gradients (Zhao and Bilen, 2021b,a; Jiang et al., 2022), matching training trajectories (Cazenavette et al., 2022; Du et al., 2023), representative matching (Liu et al., 2023), aligning features (Wang et al., 2022) and matching distributions (Zhao and Bilen, 2023). These methods belong to the field of dataset condensation[1].

Although dataset condensation methods have seen an increasing interest in the last few years, their application has primarily been limited to image datasets. Our objective in this project is to extend dataset condensation to language datasets.

[1]There is not a clear distinction between dataset distillation and condensation approaches. For this survey, the term dataset distillation is used when we are solving the original problem formulation by Wang et al. (2018) and use dataset condensation when referring to optimizing a surrogate objective.

### 2.4 Federated Learning

In federated learning, we have a set of locations, also called silos, where we get local data for a task and our objective is to obtain a high performing model (also referred to as the main model or the federation controller) on the task without accessing the local data to ensure privacy. FedAvg (McMahan et al., 2017) is a popular federated learning algorithm however, the difference between the local model losses at each synchronization stage make the algorithm unstable and difficult to train in some scenarios. In (Lu et al., 2020), authors use coresets to train distributed machine learning models.

Federated learning often requires costly communication rounds to obtain good performance and hence, a few papers have made steps towards one-shot federated learning. Guha et al. (2019) try different heuristics for selecting the client models which would form the best ensemble. Zhou et al. (2020) proposes a dataset distillation method for one-shot federated learning. They initialize a model in each silo and distill the dataset at each location. The distilled datasets are shared with the federation controller which is then trained with the aggregation of the local distilled datasets. While previous work has made progress towards one-shot federated learning, our method offers several advantages. Similar to Zhou et al. (2020), our approach shares synthetic samples instead of model weights or gradients, which significantly reduces

communication costs. However, our method improves upon Zhou et al. (2020) by having lower computation costs. Furthermore, our approach is model-independent, allowing each silo to use its own private model, which is particularly useful when silos do not share their proprietary models. This flexibility also extends to the task domain, as our method can be applied to multi-task learning scenarios where the central server wants to learn from multiple tasks simultaneously.

## 3 Method

In dataset condensation, our objective is to condense a large-scale training dataset $\mathcal{T} = \{(x_1, y_1), \ldots, (x_{|\mathcal{T}|}, y_{|\mathcal{T}|})\}$ to a synthetic dataset $\mathcal{S} = \{(x_1, y_1), \ldots, (x_{|\mathcal{S}|}, y_{|\mathcal{S}|})\}$ such that $|\mathcal{S}| << |\mathcal{T}|$. We want the performance of synthetic dataset $\mathcal{S}$ to be similar to $\mathcal{T}$ on an unseen testing set.

We base our method for text based dataset condensation on Distribution Matching (DM) (Zhao and Bilen, 2023). In DM, we aim to align the distributions of the real and synthetic dataset such that the synthetic data distribution can accurately approximate the real data distribution. One way to achieve this objective is by transforming the real and synthetic datasets with a family of parametric functions $\psi_\theta : \mathcal{R}^d \rightarrow \mathcal{R}^{d'}$ where $\theta$ is a parameter, and ensure that the embedding space distributions of the real and synthetic datasets under the transformation of the parameterized function is similar.

Hence we minimize the distance between real and synthetic data distributions in the embedding space using the empirical from of the maximum mean discrepancy (MMD) (Gretton et al., 2012):

$$\mathbb{E}_{\theta\ P_\theta}||\frac{1}{|\mathcal{T}|}\sum_{i=1}^{|\mathcal{T}|}\psi_\theta(\mathbf{x}_i) - \frac{1}{|\mathcal{S}|}\sum_{j=1}^{|\mathcal{S}|}\psi_\theta(\mathbf{s}_j)||^2 \quad (1)$$

where $P_\theta$ is the distribution of network parameters.

In Figure 1 we demonstrate our training algorithm. We first embed the original dataset using transformer (Vaswani et al., 2017) based sentence embedding models such as sentence-transformers (Reimers and Gurevych, 2019). We can initialize the synthetic data randomly but we observe the convergence to be faster when the synthetic data is initialized using randomly sampled embeddings of the original data. We use multi-layered perceptrons as the parameterized family of functions to process the synthetic and real embeddings. As observed by Zhao and Bilen (2023), randomly sampling the weights of the MLP works well and hence we sample a new neural network in each training iteration. Using Eq. (1), we minimize the loss between the randomly transformed samples and propagate the loss back to the synthetic data samples.

## 4 Experiments

We demonstrate the results of dataset condensation on the following text classification datasets (also see Table 1),

1. SST1: The Stanford Sentiment Treebank (SST) (Socher et al., 2013) dataset is a text dataset containing movie reviews for sentiment classification obtained from rottentomatoes.com and originally collected in 2005. Each sample has associated a numerical score ranging from 0 (negative) to 1 (positive). Each review is classified in 5 different classes partitioning the labels in 5 different bins: Very Negative, Negative, Neutral, Positive and Very Positive.

2. SST2: Is a simplified version of SST1 where neutral labels are removed and the bins are merged to form a two-class (positive/negative) sentiment classification problem.

3. Dbpedia-14 : The DBpedia ontology classification dataset (Lehmann et al., 2015) is constructed by picking 14 non-overlapping classes from DBpedia 2014.

4. Yahoo-answers: The Yahoo dataset is a widely used benchmark dataset in natural language processing (NLP) for text classification task (Zhang et al., 2015). The dataset contains a collection of news articles, and each article is labeled with one of 10 categories, such as sports, business, entertainment, and so on.

| Dataset | Classes | Train | Test |
|---|---|---|---|
| SST1 | 5 | 8544 | 2210 |
| SST2 | 2 | 6920 | 1821 |
| Dbpedia-14 | 14 | $5.6 \cdot 10^5$ | $7 \cdot 10^4$ |
| Yahoo-answers | 10 | $1.4 \cdot 10^6$ | $6 \cdot 10^4$ |

Table 1: Dataset statistics showing number of classes, number of training samples and number of test samples in each of the datasets

| Dataset | Samples/class | | | | All Data Accuracy (%) |
|---|---|---|---|---|---|
| | 1 | 10 | 25 | 50 | |
| SST1 | $46.7 \pm 1.00$ | $46.68 \pm 2.82$ | $\mathbf{46.75 \pm 1.29}$ | $46.04 \pm 2.04$ | 52 |
| SST2 | $88.54 \pm 0.28$ | $88.58 \pm 0.12$ | $88.64 \pm 0.16$ | $\mathbf{89.33 \pm 0.1}$ | 91 |
| DBpedia-14 | $82.68 \pm 1.79$ | $79.82 \pm 1.75$ | $87.04 \pm 1.09$ | $\mathbf{88.71 \pm 0.38}$ | 98 |
| Yahoo-answers | $59.33 \pm 1.37$ | $59.81 \pm 0.55$ | $59.40 \pm 0.40$ | $\mathbf{63.2 \pm 0.36}$ | 72 |

Table 2: Test accuracy of a model trained on varying amounts of synthetic data per class from the datasets SST1, SST2, Dbpedia-14 and Yahoo-Answers

We perform dataset condensation on these datasets and compare the results in Table 2. An important hyperparameter in dataset condensation is the number of learnable synthetic samples in each class. In case of the SST datasets, which are orders of magnitude smaller compared to Dbpedia-14 and Yahoo-answers, we notice little performance gain when we increase the number of samples per class since we can get reasonable performance with a single sample. In the case of Dbpedia-14 and Yahoo-answers, we notice significant improvements when we increase the number of learnable synthetic samples but we also note that the performance gains are small when we go beyond 50 samples per class.

To interpret these embeddings, we conduct an experiment where we replace our sentence-transformer model with BART (Lewis et al., 2019) since it has an open source decoder available to convert the embeddings back into sentences. Hence, we use BART embeddings and train some synthetic samples using dataset condensation. We then decode these synthetic embeddings using the BART decoder and show the obtained sentences in Table 3. While the sentences itself don't make much sense, they exhibit the overall expected class sentiment.

We also show that the synthesized samples are able to cover the distribution of embedding space well by plotting the t-SNE (Van der Maaten and Hinton, 2008) of the data embeddings and synthetic embeddings in Figure 2.

Table 3: Decoded synthesized embeddings from each class in the SST2 dataset

| |
|---|
| **Positive Class** |
| 1. extremely well acted by the four primary actors, this is a seriously thought out movie that |
| 2. interesting both as a historical study and an allegory for the relationship between two men |
| 3. intimate and panoramic view of the world at a moment's notice |
| **Negative Class** |
| 1. woefully pretentious. It's all about the same thing: overeating |
| 2. it's so badly made on every level that i'm actually having a hard |
| 3. so mind numbingly awful that you hope britney won't do it one |

## 5 Applications

### 5.1 One-Shot Federated Learning

Federated Learning is a method in which the same model is trained at different physical locations (or silos) with the data available locally. Each silo trains a local model and reports the knowledge acquired to a federation controller (or main model) which aggregates the knowledge and reports it back to the silos so that the silos can continue training with the aggregated model. This allows the local models to obtain knowledge from data at other silos. FedAvg (McMahan et al., 2017) is a popular federation learning algorithm, which reports the local model weights to the federation controller and the federation controller aggregates them by taking their average. The averaged model is sent back to the silos and models continue training for future rounds with the new weights.

Usually, one of the main reasons why federated learning is applied is to obtain an extra layer of privacy for the data. If data does not leave the location where it was acquired, there is less possibility for it to be leaked. This is especially important in medical applications. This is achieved by FedAvg by reporting only the weights of the models. However, federated learning requires many rounds of communication, which can become prohibitively costly especially when using deep neural networks. Therefore we study the setting of one-shot federated learning, where only one round of communication is allowed.

Using dataset condensation, we change the way

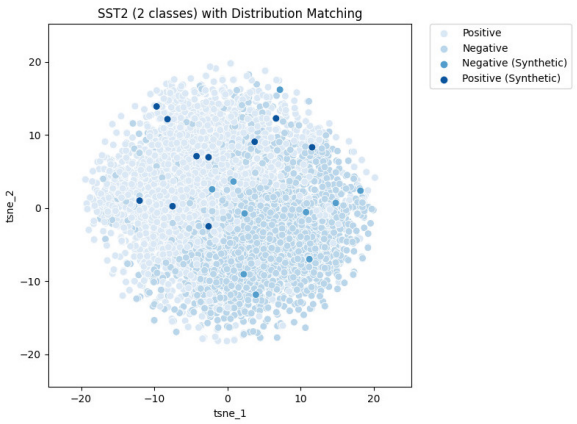
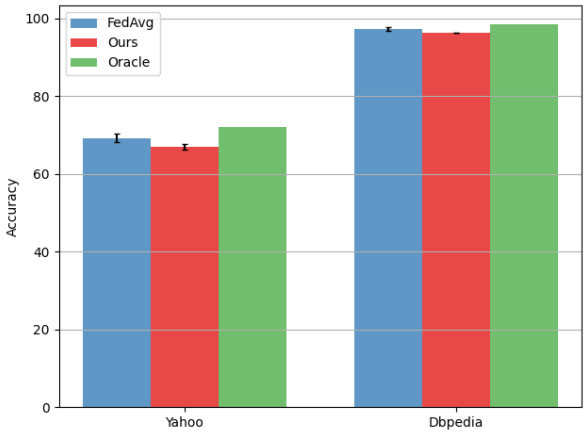
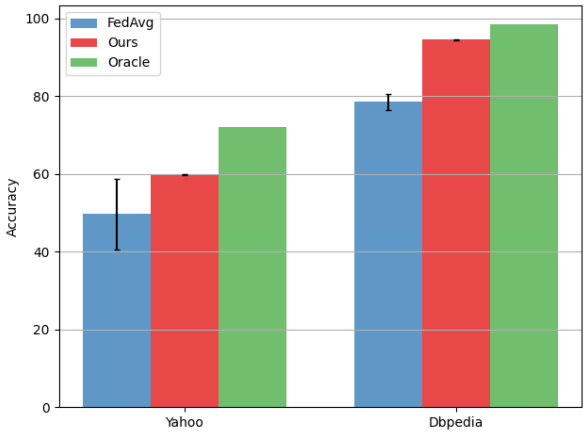



Figure 2: t-SNE of the data embeddings and synthetic embeddings




(a) Comparison of federated learning methods on IID data splits in each silo




(b) Comparison of federated learning methods on non-IID data splits in each silo

Figure 3: Federated learning results on IID and non-IID data splits

in which the knowledge is shared while ensuring privacy. Instead of sharing model weights, we propose that each silo condenses the data it obtains and sends the condensed synthetic data to the federation controller. The federation controller trains a model for the task with the aggregation of all the condensed datasets sent by the silos. We maintain privacy by sharing only encoded synthetic samples. The synthetic samples are also not generally legible, and an attacker would need to have access to the encoder / decoder (which would be kept private) to extract any useful data from them. A possible future direction would be to view our approach from an adversarial perspective and evaluate the condensed synthetic samples for their privacy. We show the efficacy of our method by comparing it to FedAvg as the baseline and an Oracle model trained with all the data available in the silos directly.

We experiment over Dbpedia-14 and Yahoo-answers datasets in a federated setting with 10 silos. We divide the data independently and identically in each of the silos and report the performance in Figure 3a. We observe that the performance of our method is comparable to FedAvg as well as the Oracle. We conduct another experiment in which we use a non-IID data in each of the silos. We achieve this in a similar way to (McMahan et al., 2017), by selecting the number of silos to be equal to number of classes, then dividing each of the classes into two halves and assigning each half to a random silo, such that each silo holds information about at most two different classes. We report the performance of FedAvg, our method and the oracle in Figure 3b. We can see that while FedAvg performs poorly in the non-IID case, our method of using dataset condensation only suffers from a small drop in performance. Furthermore, it is important to note that our method has $50\%$ less communication costs.

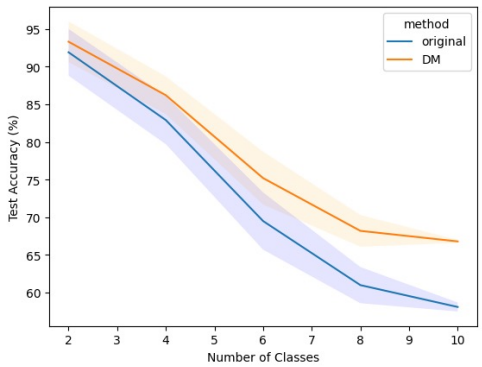
### 5.2 Continual Learning

Continual Learning (CL) is a sub-field of machine learning that deals with the problem of learning from a continuous stream of data, where the distribution of the data can change over time. In traditional machine learning, models are typically trained on a fixed dataset and then evaluated on a separate test set. However, in many real-world scenarios, the data distribution can change over time, making it difficult for models to adapt to new data without forgetting previously learned information. Continual learning aims to address this problem by developing models that can learn from a continuous stream of data and adapt to new tasks without forgetting previously learned information.

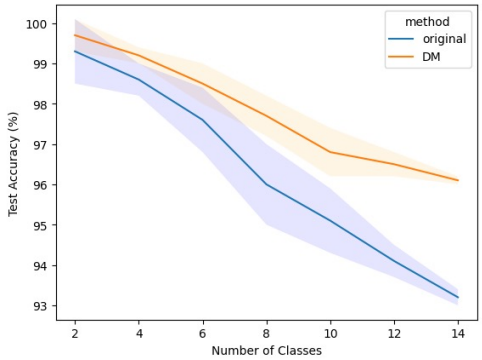
One key challenge in continual learning is avoiding catastrophic forgetting, where the model's performance on previously learned tasks deteriorates as it learns new tasks. There are several approaches to continual learning, including regularization-based methods (Huang et al., 2021; Jung et al., 2020), memory-based methods (Zhou et al., 2022; Lopez-Paz and Ranzato, 2017; Jin et al., 2020; Aljundi et al., 2018), and distillation-based methods (Kang et al., 2022; Hou et al., 2018). A continual learning problem can also be broadly formulated as class-incremental, task-incremental and domain-incremental learning problem.

In this work we follow the experimental methodology given in (Zhao and Bilen, 2023) for continual learning. Using dataset condensation we can store our data in a more efficient manner which helps in tackling catastrophic forgetting. We follow the GDumb (Greedy Sampler and Dumb Learner) framework (Prabhu et al., 2020) for the CL pipeline. GDumb greedily samples k-samples per class from the data seen until a certain point in time and trains a learner model, such as a neural network, over it. The learner learns to classify all the labels seen until that time-step. Since we train the model only on the latest memory, the quality of memory construction plays a vital role in the continual learning performance. Hence, as proposed by Zhao and Bilen (2023), we condense our dataset to the required budget first, and then train our model on it. Since, using condensed data samples allows us to reduce the number of samples to train on, we can minimize the problem of catastrophic forgetting.

For our experiments, we use the Dbpedia-14 and Yahoo-answers datasets in our CL experiments and perform a 5-step and 7-step learning respectively, in which we add 2 new classes to the previously seen data every time-step. We present the results for the two datasets in Figure 4. We can observe that our method is better at maintaining the overall performance as we add more classes to the continual learning setup.




(a) Comparison of DM vs Original GDumb sampling for 5-step CL on Yahoo dataset




(b) Comparison of DM vs Original GDumb sampling for 7-step learning on Dbpedia dataset

Figure 4: Continual Learning Results on Yahoo and Dbpedia datasets

## 6 Conclusion

We presented a way to perform dataset condensation on text. We further explored two applications of dataset condensation - continual learning and federated learning. Our experiments demonstrate that dataset condensation can be a viable solution to the proposed application domains.

The way we apply dataset condensation on the proposed applications is not specific to text datasets and a possible extension of our work could be applying the techniques in multi-modal tasks. We are also interested in comparing federated learning approaches evaluated in this work on harder tasks since we get close to oracle performance with our proposed method as well as the baseline FedAvg when data is independently and identically distributed among the silos. As mentioned previously, an important area of research in federated learning is the level of privacy offered by different algorithms while communicating information. Hence, an important future direction is to measure how robust the condensed synthetic samples are to adversarial attacks.

## References

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, pages 139–154.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua

Bengio. 2019. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825.

Eden Belouadah and Adrian Popescu. 2020. Scail: Classifier weights scaling for class incremental learning. In *The IEEE Winter Conference on Applications of Computer Vision*.

Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. 2020. Flexible dataset distillation: Learn labels instead of images. In *Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Workshop*.

Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. 2018. End-to-end incremental learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 233–248.

George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A. Efros, and Jun-Yan Zhu. 2022. Dataset distillation by matching training trajectories. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4750–4759.

Yutian Chen, Max Welling, and Alex Smola. 2010. Super-samples from kernel herding. *The Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence*.

Jiawei Du, Yidi Jiang, Vincent T. F. Tan, Joey Tianyi Zhou, and Haizhou Li. 2023. Minimizing the accumulated trajectory error to improve dataset distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Reza Zanjirani Farahani and Masoud Hekmatfar. 2009. *Facility location: concepts, models, algorithms and case studies*. Springer Science & Business Media.

Dan Feldman. 2020. Introduction to core-sets: an updated survey.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Neel Guha, Ameet Talwalkar, and Virginia Smith. 2019. One-shot federated learning.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. 2018. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 437–452.

Yufan Huang, Yanzhe Zhang, Jiaao Chen, Xuezhi Wang, and Diyi Yang. 2021. Continual learning for text classification with information disentanglement based regularization. *arXiv preprint arXiv:2104.05489*.

Zixuan Jiang, Jiaqi Gu, Mingjie Liu, and David Z. Pan. 2022. Delving into effective gradient matching for dataset condensation. *arXiv preprint arXiv:2208.00311*.

Xisen Jin, Junyi Du, and Xiang Ren. 2020. Gradient based memory editing for task-free continual learning. In *4th Lifelong Machine Learning Workshop at ICML 2020*.

Sangwon Jung, Hongjoon Ahn, Sungmin Cha, and Taesup Moon. 2020. Continual learning with node-importance based adaptive group sparse regularization. *Advances in Neural Information Processing Systems*, 33:3647–3658.

Minsoo Kang, Jaeyoo Park, and Bohyung Han. 2022. Class-incremental learning by knowledge distillation with adaptive feature consolidation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16071–16080.

Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic web*, 6(2):167–195.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension.

Yanqing Liu, Jianyang Gu, Kai Wang, Zheng Zhu, Wei Jiang, and Yang You. 2023. DREAM: Efficient dataset distillation by representative matching. *arXiv preprint arXiv:2302.14416*.

David Lopez-Paz and Marc'Aurelio Ranzato. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.

Hanlin Lu, Ming-Ju Li, Ting He, Shiqiang Wang, Vijaykrishnan Narayanan, and Kevin S Chan. 2020. Robust coreset construction for distributed machine learning. *IEEE Journal on Selected Areas in Communications*, 38(10):2400–2417.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. 2020. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. 2017. icarl: Incremental classifier and representation learning. In

*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. *ICLR*.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. 2019. An empirical study of example forgetting during deep neural network learning. *ICLR*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12196–12205.

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

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Bo Zhao and Hakan Bilen. 2021a. Dataset condensation with differentiable siamese augmentation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 12674–12685.

Bo Zhao and Hakan Bilen. 2021b. Dataset condensation with gradient matching. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Bo Zhao and Hakan Bilen. 2023. Dataset condensation with distribution matching. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Da-Wei Zhou, Qi-Wei Wang, Han-Jia Ye, and De-Chuan Zhan. 2022. A model or 603 exemplars: Towards memory-efficient class-incremental learning. *arXiv preprint arXiv:2205.13218*.

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

# A Contributions

- **Vishesh**: Setting up the dataset condensation for images baseline pipeline (distribution matching). Running image dataset condensation experiments (distribution matching) to obtain baseline results. Distribution Matching Experiment running for text. Project proposal slides creation. Project survey writing: Literature review for Dataset distillation. Midterm report writing. Project presentation slide creation. Continual learning experiments. Imbalanced datasets experiments. Final report revision.

- **Shivin**: Running image dataset condensation experiments (gradient matching). Text datasets preparation. Pipeline for text Distribution Matching development. Synthetic samples interpretability analysis. Project Proposal slides creation. Project pitch speaker. Literature review. Project Survey writing: Introduction and Literature review for Dataset Condensation. Midterm report writing and revision. Project presentation slide creation. Yahoo and Dbpedia datasets obtention and embedding creation. Final report writing and revision.

- **Gaurav**: Finding NLP baseline datasets (equivalent to CIFAR/MNIST for Image datasets). Running a baseline model on the dataset. Interpretability of synthetic samples with pre-trained transformers pipeline development. Pipeline for text Gradient Matching development. Project Proposal slides creation. Literature review. Project survey writing: Literature review for Coreset Selection. Midterm

report writing. Project presentation slide creation. Imbalanced datasets experiments. Final report revision.

- **Rafael**: Setting up the dataset condensation for images baseline pipeline (gradient matching). Running image dataset condensation experiments (gradient matching). Gradient Matching Experiment running for text. Project Proposal slide creation. Project Survey writing: Introduction and Literature review for Dataset Condensation. Midterm report writing and revision. Project presentation slide creation. Federated learning experiments. Final report writing and revision.