Two-stage Dataset Compression for 3D Point Cloud Data

Shengyao Chen[1], Yuxuan Zhang[1], Yonglin Liu[1], and Qihao Luo[1]

[1]Shanghai Jiao Tong University, Shanghai, China

ABSTRACT

Currently, there is a multitude of work based on 3D point cloud datasets. However, the current 3D datasets lack effective distillation methods. This paper presents a two-stage dataset compression approach for 3D point cloud data, addressing the challenges of high VRAM consumption.We propose a method that decouples feature extraction from dataset compression, utilizing an encoder-decoder framework to transform and compress data. Our approach involves two distinct strategies: data synthesis through distillation and data selection via k-medoids clustering. The latter achieved good results, constructing a feasible two-stage process for data compression of 3D point cloud data. We also analyzed the poor performance of the former and believe that the dataset distillation process is severely affected by information loss. This work provides insights into the limitations of traditional data distillation methods and contributes to the understanding of dataset compression for complex data types like 3D point clouds. Code and data are accessible at Github.[*]

Keywords: Dataset Distillation, Dataset Compression, Coreset Selection

1. INTRODUCTION

Dataset Compression[1] is a crucial step in enhancing learning efficiency. It involves selecting or synthesizing a very small amount of high-quality data from a large dataset, such that models trained on this reduced dataset can achieve performance comparable to those trained on the entire original dataset. The challenge lies in the fact that the process of synthesizing or selecting the dataset requires loading a significant amount of raw data, which consumes a large amount of VRAM, especially for 3D point cloud data.

Our work focuses on decoupling feature extraction from dataset compression: first, we use an encoder to transform the original data into lower-dimensional features, then perform data compression on these features, and finally, use a decoder to convert the synthesized or selected data back into the original data format. This approach can greatly reduce the overhead.

We attempted two distinct strategies: synthesizing data through dataset distillation and selecting more representative data. The reason for trying the former is that dataset distillation methods have shown good performance in image classification, and we wanted to explore whether they would also be superior in synthesizing features for 3D point cloud data. The latter is a strategy that has similar idea to the former, retaining more of the original information but lacking comprehensiveness.

We conducted experiments with these two methods under different IPC (Images Per Class) conditions. The results indicated that the approach of synthesizing features through dataset distillation and then decoding yielded poor performance, even worse than randomly selecting the same number of original data points when IPC was high. However, another method we designed, which involved selecting representative features from the extracted features and then finding their corresponding original data to construct a dataset, resulted in better quality. We analyzed the results obtained from this approach and proposed possible reasons for the outcomes.

To summarize, our contributions are as follows: **1)** We We propose a method for 3D point cloud dataset distillation. **2)** We have proposed a two-stage method that may be able to extend to other fields. **3)** We compared the performance of distilled dataset and raw dataset under different scale of dataset

[*]The code repository for the paper can be found at https://github.com/csyer/PointCloud_Distillation_Demo.

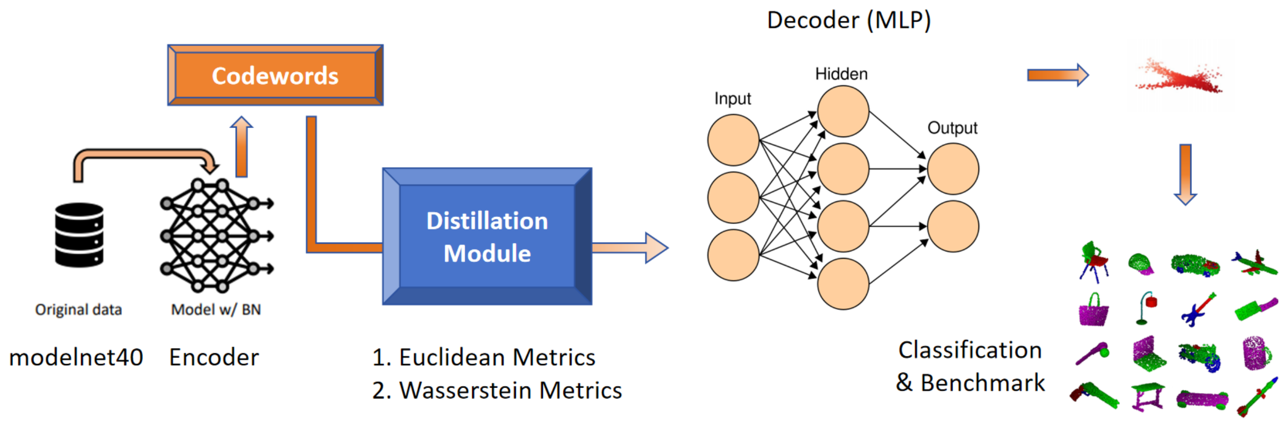

Figure 1: Data Distillation Pipeline

2. RELATED WORK

2.1 Coreset

Coreset selection is a technique used in the field of machine learning for data compression. Its goal is to select the most representative subset of samples from the target dataset, so as to reduce the scale of the data while maintaining the performance of models trained on the full dataset as much as possible. We have studied the concept of coreset selection and believe that when the distribution of the small-scale dataset we select or synthesize is closer to the distribution of the original dataset, the performance of the models trained on it will be better.

2.2 Dataset Distillation

Dataset distillation is a process that extracts key features from a large dataset and synthesizes them into a very small dataset. A well-designed dataset distillation method can ensure that the synthesized dataset retains as much of the critical information and characteristics of the original data as possible, enabling models trained on this dataset to approach or even achieve the performance of models trained on the original data.

In our work, to achieve decoupling of the feature extraction and distillation processes, we employ a distillation approach based on the data itself rather than the training process. More specifically, our primary method of distillation relies on synthesizing a small-scale dataset that has a similar distribution to the original data.The loss we use is based on the distance between the synthesized features and the original features.

2.3 FoldingNet

FoldingNet[2] is a deep learning network for point cloud processing that achieves point cloud auto-encoding through deep grid deformation. This network structure falls within the category of self-supervised learning, capable of projecting the input point cloud into a high-dimensional space rich in semantic information to form high-dimensional feature vectors, which is the encoding process. Subsequently, these high-dimensional feature vectors are restored to high-dimensional input point clouds through the decoding network.

3. METHOD

3.1 Data Synthesizing

We process the original point cloud data with the encoder of FoldingNet[2] to transform it into feature vectors (Figure 2). Afterwards, we input all the feature vectors into an MLP and output a small number of synthesized feature vectors, with the loss function using the Wasserstein Metric[3] between the synthesized vectors and the original feature vectors. Then, we restore these synthesized vectors back into point cloud data through the decoder. In this way, we obtain a very small point cloud dataset whose feature distribution is close to that of the original dataset's features.

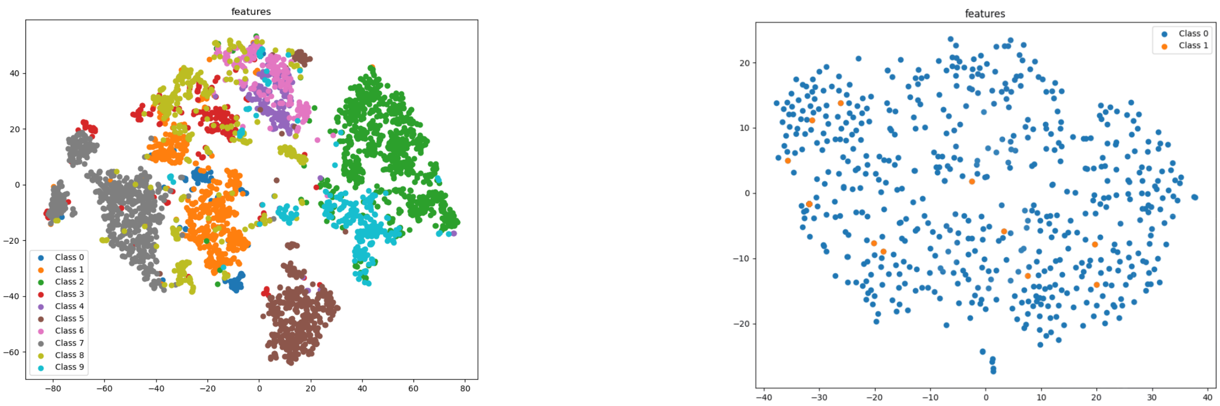

Figure 2: Features(Left) & Distilled features(Right)

3.2 Data Selecting

We found that the quality of the point cloud data reconstructed from the distilled synthetic vectors and decoded was not good. To improve the method and analyze the reasons, we designed another dataset reduction strategy: using the k-medoids algorithm to select some representative features from the feature vectors output by the encoder, and then finding the original data corresponding to these feature vectors. This method does not require a decoder, and the resulting compressed dataset is a subset of the original dataset, rather than newly synthesized data.

4. EXPERIMENTS

We designed two experiments: the first one involves training the same model on datasets generated by these two methods and on a random subset of the original dataset of the same size, and then comparing the accuracy of the trained models on classification tasks. This experiment aims to verify the effectiveness of our methods.

The second experiment aims to explore whether our dataset compression method can effectively synthesize key information from each class to confirm if the direction of our work is correct. We attempt to construct datasets of different sizes (measured by IPC) using the dataset selection method and compare their performance on classification tasks with models trained on randomly selected data. If our method can effectively summarize key information, then the smaller the dataset size, the greater the relative improvement over random data should be. Since the dataset synthesis method did not perform well in the first experiment, we did not include this method in the scope of the second experiment.

To validate the effectiveness of our dataset compression methods, we conducted additional tests on a point cloud classification task. For this purpose, we used modelnet40 as the dataset for this experiment, which contains point cloud data for classification tasks. And we used PointNet++[4] as the classification model, which has been shown to perform well on point cloud data.

4.1 Results

Method	Accuracy
Random	80.2
Synthesis	70.1
Selection	82

Table 1: Result of IPC=50

The results of the first experiment are shown in Table 1. The results indicate that the dataset selection method can improve the quality of the compressed dataset, but the compressed dataset generated by the data synthesis method does not even outperform a randomly selected subset of the original data.

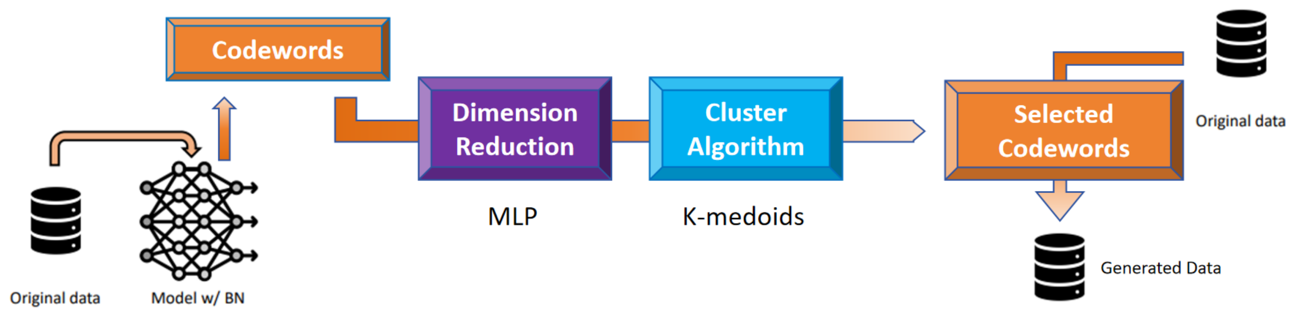

Figure 3: Data Selection Pipeline

IPC	2	3	4	5	6	7	8	9	10
Random	35.8	47.8	61.4	64.6	66.6	70.5	71.7	73.9	74.9
Selection	45.7	50.2	63.3	66.8	68.1	70.6	71.9	73.9	75.0

Table 2: Result of different IPC

The results of the second experiment are presented in Table 2. The experimental results indicate that when IPC is very small, the data we selected shows a significant improvement compared to random data. As IPC gradually increases, the gap between random data and our selected data narrows. We believe this phenomenon is reasonable: when the dataset is larger, the probability of containing more key information in random data also increases.This result suggests that our idea of making the distribution of the compressed dataset closer to that of the original dataset is largely correct.

4.2 Analysis

Both the data synthesis and dataset selection methods we employed aimed at similar optimization goals, namely, to make the compressed dataset as similar as possible to the original dataset in terms of distribution. Additionally, the data synthesis method has a much higher optimization flexibility compared to dataset selection. Therefore, theoretically speaking, data synthesis should perform better, but the experimental results are surprising.The core issue we will analyze next is: why does the data synthesis perform even worse, and even more poorly than random selection?

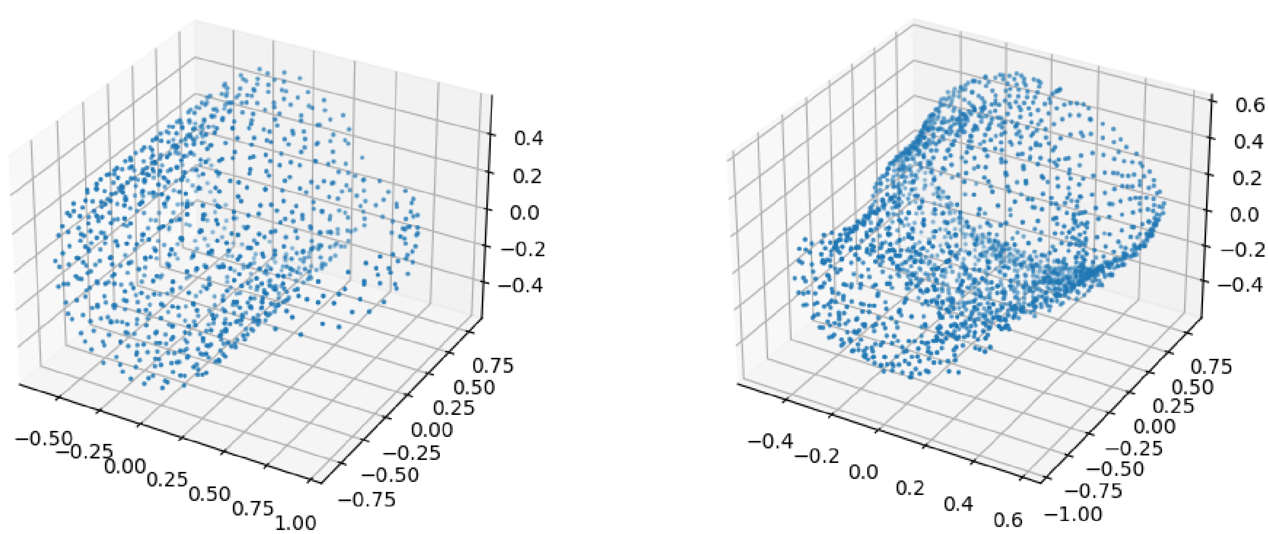

Figure 4: Original(Left) & Synthetic(Right) Data of Label "Cup"

We observed the synthesized data and compared it with the original data. We found that although the synthesized data bears a resemblance to the original data within the same class, it does not exhibit the unique characteristics of that class and may appear similar to data from several different classes. This phenomenon is due to the excessive loss of information during the encoding process. As a result, even though the distribution of the synthesized features is similar to that of the original data, the decoded data from different classes has much less discrimination than the original data.

Conversely, although dataset selection involves fewer manipulations, it selects the original data without loss of information, which may more vividly reflect the characteristics of the class to which they belong. In summary, we believe that within a certain range, the integrity of information is more significant for dataset compression work than the comprehensiveness of information, especially for types of data like point clouds where key information is more complex.

5. CONCLUSION

Our work decouples feature extraction from dataset compression to reduce VRAM usage and explores two different compression methods. One of them achieved good results, especially when the dataset size is very small; the other method, based on dataset distillation, did not yield the expected results, but we analyzed the reasons and summarized the experience in this regard.

Our target for compression is 3D point cloud data, which, compared to the two-dimensional image data commonly used as subjects for dataset distillation methods, is more complex and has key information hidden deeper. The difficulty of obtaining high-quality datasets through dataset compression is greater for 3D point clouds.

Our innovation lies in successfully designing a method that processes 3D point cloud data after feature extraction, rather than the raw data, using the k-means clustering algorithm to achieve dataset compression with a smaller VRAM consumption. Furthermore, we identified the shortcomings of traditional data distillation methods when there is information loss and the original data is complex, providing our analysis on this phenomenon.This phenomenon may also provide some inspiration for future research.

ACKNOWLEDGMENTS

We would like to express our sincere gratitude to Professors Cewu Lu and Yonglu Li for their invaluable guidance and mentorship throughout this work. Their expertise and support have greatly contributed to the development of this research. We would also like to thank Teaching Assistant Yue Xu for his continuous guidance and assistance, which has been crucial to the successful completion of this study. Additionally, we appreciate the insightful suggestions provided by Jiarui Xue, which have significantly enhanced the quality of this work.

REFERENCES

[1] Wang, T., Zhu, J.-Y., Torralba, A., and Efros, A. A., "Dataset distillation," (2020).

[2] Yang, Y., Feng, C., Shen, Y., and Tian, D., "Foldingnet: Interpretable unsupervised learning on 3d point clouds," *arXiv* (Dec. 2017).

[3] Liu, H., Xing, T., Li, L., Dalal, V., He, J., and Wang, H., "Dataset distillation via the wasserstein metric," *CoRR* **abs/2311.18531** (2023).

[4] Qi, C. R., Yi, L., Su, H., and Guibas, L. J., "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413* (2017).