Generating Point Cloud Augmentations via Class-Conditioned Diffusion Model

Gulshan Sharma
Indian Institute of Technology Ropar, India

Chetan Gupta
Emerging Technologies and Innovation Lab
Yamaha Motor Solutions, India

Aastha Agarwal
Emerging Technologies and Innovation Lab
Yamaha Motor Solutions, India

Lalit Sharma
Emerging Technologies and Innovation Lab
Yamaha Motor Solutions, India

Abhinav Dhall
Indian Institute of Technology Ropar, India
Monash University, Australia

Abstract

In this paper, we present a class-conditioned Denoising Diffusion Probabilistic Model (DDPM) based approach to augment point cloud data within the latent feature space. Our method focuses on generating synthetic point cloud latent embeddings, which encode both spatial and semantic information of the point cloud. By harnessing the capabilities of DDPM within a class-conditioned framework, our goal is to provide a cost-effective and practical solution for the augmentation of point cloud samples. We conduct experiments on the publicly available point cloud dataset, and our findings suggest that the proposed approach (a) effectively generates high-quality synthetic embeddings directly from the Gaussian noise and (b) improves the classification performance of the point cloud classes within limited data settings.

1. Introduction

The advancements in 3D acquisition technologies have increased the use of 3D data in scientific and engineering fields. 3D data provides detailed geometric information about the object, allowing for an overall understanding of its spatial and semantic characteristics [10, 21]. 3D data can be represented in multiple formats, such as depth images, point clouds, meshes, and volumetric grids [14]. The point cloud is a fundamental 3D data format, which consists of densely packed data points in a 3D coordinate system. Each point within this collection is represented by a set of coordinates (x, y, and z). These coordinates accurately depict the object's surface geometry and provide a comprehensive digital representation of the object [9]. Due to its efficacy in representing spatial information, point clouds have applications in urban planning, robotics, and autonomous driving.

Point clouds can be collected through various sensing technologies such as laser scanning, LiDAR, and photogrammetry. While these sensing technologies offer a non-intrusive and convenient way to collect 3D data, they come with their own set of limitations. One significant limitation is the high operational cost associated with collecting high-quality data. This cost includes equipment and data processing expenses. Acquiring the necessary equipment, such as high-fidelity scanners, requires a high initial investment. While addressing data processing costs, deep neural network (DNN) remains the state-of-the-art method for processing point cloud data [9]. DNN requires a large amount of data to achieve robust results, which can be expensive and time-consuming to collect and process [28].

To address this data scarcity and reduce the overall cost of data acquisition, Data Augmentation (DA) offers a promising solution [4, 6]. DA has demonstrated significant performance improvements in training DNN and is widely applied in domains such as image, audio, and natural language processing [5, 26]. DA artificially expands the size of a dataset by generating new samples through various transformations on the existing data. The fundamental principle behind these transformations is to introduce meaningful variations into the original data without changing its core information. This process improves the robustness of the learning algorithm and serves as an important countermeasure against overfitting.

In this paper, we present a class-conditioned DDPM-based framework for augmenting the point cloud data. Our research contributions are summarized below:

- In contrast to conventional augmentation techniques, our approach involves use of latent point cloud embeddings as descriptors for point cloud data. These embeddings serve as high-level representations of the data, and we conduct augmentation procedure based on these embeddings.
- We propose a class-conditioned Denoising Diffusion Probabilistic Model (DDPM) based framework which learns the point cloud latent representations and generates the class-specific synthetic point cloud embeddings from the Gaussian noise.
- We evaluate the class discrimination efficacy of synthetic class embeddings via applying an off-the-shelf classifier. Additionally, we compare the quality of synthetic embeddings generated via proposed framework with class-conditioned Variational Autoencoder (cVAE) and class-conditioned Generative Adversarial Network (cGAN).
- We perform a similarity check between the original and synthetic embeddings via Jensen-Shannon Divergence (JSD) scores. Furthermore, we perform a visual comparison of synthetic and original embeddings through t-distributed stochastic neighbor embedding (t-SNE) visualizations.

The rest of the paper is structured as follows: Section 2 presents a brief review on point cloud augmentation and DDPM. Section 3 introduces the dataset. Section 4 outlines the proposed framework. Section 5 details the experiments conducted, while Section 6 presents the results. Section 7 offers a discussion of the findings. Finally, Section 8 concludes the paper and outlines potential future work.

2. Background

2.1. Point Cloud Augmentation

Point cloud DA can be categorized into three categories: (a) traditional methods, (b) deep learning-based approaches, and (c) latent feature space augmentation.

2.1.1 Traditional Augmentation

Traditional augmentation methods for point cloud data involve applying geometric or statistical transformations. Geometric transformations are mostly inspired by image transformations, which include translation, rotation, and scaling operations applied uniformly or along specific axes [15]. Statistical transformations include (a) adding noise from a uniform distribution to simulate sensor inaccuracies, (b) removing a certain percentage of points to simulate missing or occluded data, (c) jittering within a local neighborhood, and (d) selecting a subset of points to reduce the point cloud's density. While these techniques are easy to implement, they struggle to capture the complex semantics of point cloud data. Additionally, manual methods demand domain expertise to select suitable transformations [15].

2.1.2 Deep Learning-based Augmentation

Deep learning-based DA techniques leverage the capabilities of DNN to create meaningful augmentations. These techniques include a wide array of strategies, such as VAE, GAN, flow-based generative models, and transformers [1]. VAE can learn compact representations, which can be interpolated to create new variations [11]. GAN can learn the underlying point cloud distribution by training generator and discriminator in a competitive process [8]. Furthermore, other image-based strategies like in-painting or outpainting can be applied where DNNs reconstruct missing parts of point clouds [23]. Self-supervised learning techniques, which exploit the intrinsic relationships within data, can also be applied to generate augmented samples [27]. In contrast to traditional methods, these methods generate realistic and contextually relevant augmentations.

2.1.3 Latent Feature Space Augmentation

The latent feature space DA method enhances datasets by generating new examples within a compact and noise-free latent space rather than the original data [20]. In this approach, meaningful latent representations are extracted from the original data either through (a) kernel functions, (b) dimensionality reduction techniques like principal component analysis, or (c) trained DNN, which maps the original data onto a semantically meaningful latent space [18]. Since these latent representations are less prone to noise, multiple augmentation techniques can be applied within this space. Moreover, the compact nature of these latent representations facilitates faster computation. In the context of point clouds, multiple classification networks [3] have been proposed, which showcase a robust capacity to learn discriminative latent features.

2.2. Denoising Diffusion Probabilistic Model

DDPM belongs to the category of score-based generative models. These models learn the inherent data distribution by approximating the gradients of the log-likelihood of the data [12]. DDPM comprises two sub-processes: forward diffusion and backward diffusion. The forward diffusion process starts with adding Gaussian noise to the input data over a series of time steps until the data transforms into

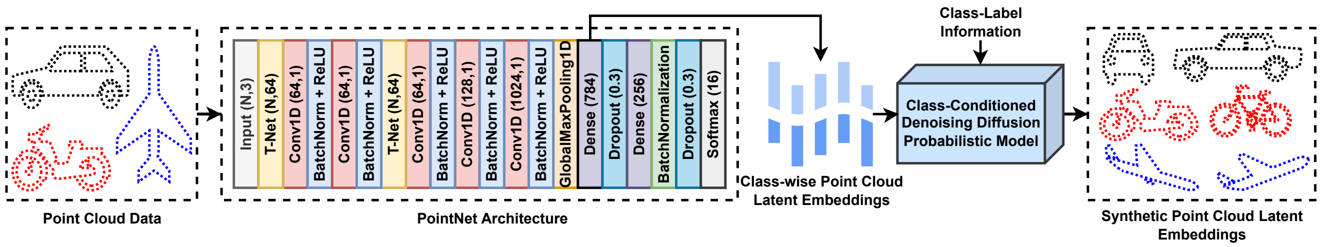

Figure 1. **Proposed framework:** PointNet generates point cloud latent embeddings, which, along with class labels are input into the class-conditioned DDPM to generate synthetic point cloud embeddings.

pure noise. During this process, the magnitude of the Gaussian noise is progressively increased at each step, which is governed by a predetermined noise scheduler. The forward process is deterministic, which implies that if both the input data and the parameters of the noise scheduler are known, one can precisely replicate the whole process. On the other hand, the backward diffusion process aims to reconstruct the original data from the noise. It begins with the noisy data obtained at the end of the forward diffusion process, and at each time step, a neural network predicts how much noise needs to be removed to return to the previous step.

DDPM can be conditioned to generate synthetic data samples belonging to a specific class. This conditioning process can be performed either with classifier guidance or with classifier-free guidance. In classifier guidance [7], the DDPM leverages the knowledge of an external classifier to guide the generation of synthetic data samples. The gradients of the classifier are injected into the backward diffusion process of the unconditional diffusion model. Another strategy for conditioning DDPM is classifier-free guidance [13], which involves mixing the score estimates of a conditional and a jointly trained unconditional diffusion model.

3. Dataset

Our proposed approach is evaluated on the ShapeNet dataset [2], which consists of richly annotated 3D shapes. ShapeNet comprises multiple versions; for our experiments, we use a publicly available subset which comprises 16 classes and is divided into train and test splits. The overall distribution of the dataset is shown in Table 1.

4. Method

Our methodology draws inspiration from the research presented in [22] and consists of two main stages: (a) Generating latent embeddings for point cloud data and (b) Developing a class-conditioned DDPM to augment these latent embeddings. Figure 1 offers a visual overview of our approach.

Table 1. Class-wise data distribution. Note the presence of a high class imbalance in the dataset.

Class Name	Train Samples	Test Samples
Cap	43	12
Rocket	44	15
Earphone	49	14
Bag	62	14
Skateboard	126	25
Mug	144	40
Motorbike	157	45
Knife	202	61
Pistol	232	42
Laptop	363	82
Guitar	604	144
Lamp	817	222
Airplane	2157	527
Car	1463	361
Chair	2988	752
Table	4202	1058

4.1. Generation of Point Cloud Embeddings

To generate the latent point cloud embeddings, we apply PointNet [3], which is a DNN designed to learn class-disentangled representations from the global and local point cloud features. In our implementation, we first conduct sample-wise normalization on the ShapeNet dataset to ensure uniform scale and range across data samples. Afterward, we train the PointNet on sparse categorical cross-entropy loss in conjunction with the Adam [16] optimizer. The learning rate is set at 0.0001. We apply a 5-fold cross-validation in order to validate that PointNet learns discriminative features across the dataset. PointNet achieves an F1 score of 0.93 ± 0.01, which confirms that PointNet learns well-discriminative features. Finally, we extract the class-specific 784-dimensional latent embedding from the dense feature representation layer, which is just after the global max pooling layer (refer to Figure 1). We reshape these 784-dimensional embeddings into 28×28-dimensional embeddings. This reshaping enhances the compatibility of the embeddings as inputs to our DDPM implementation.

4.2. Class-conditioned DDPM Development

In order to develop a class-conditioned DDPM suitable to learn the latent embeddings distribution, we implement a conditional U-Net (Figure 2) where model conditioning is based on the classifier-free diffusion guidance [13]. We generate context embeddings and time step embeddings via applying separate multi-layer perceptron networks. These networks comprise two dense layers, each with 256 neurons and GELU activation. We infuse these embeddings into the U-Net architecture to enable DDPM conditioning. During the training process, we gradually generate noisy versions of the latent point cloud embeddings by applying a predefined noise scheduler. The U-Net then learns to denoise the noisy versions of the latent point cloud embeddings. During the inference, we generate synthetic point cloud embeddings from the Gaussian noise by sequentially removing noise via trained U-Net. The architecture of U-Net is illustrated in Figure 2.

We train class-conditioned DDPM to 50 epochs with Adam optimizer [16]. The learning rate and batch size are set to 0.0001 and 32, respectively. We train the class-conditioned DDPM on the train set. We continue the training process iteratively until the model reaches its convergence point and achieves the minimum value of the loss function. Once the model achieves this criterion, we generate 1000 synthetic samples for each class.

5. Experiments & Validation Procedure

In order to validate the quality of synthetic embeddings generated via proposed approach, we conduct following experiments.

5.1. Class Discrimination Information

The quality of synthetic data can be evaluated in terms of its capacity to retain the class discrimination information which is relevant for downstream tasks [24]. To evaluate the quality of the synthetic data we perform following evaluations.

- We train a Multi-Layer Perceptron (MLP) using the *Train* set of the original embeddings and evaluate its performance on the *Test* set of the original embeddings. We report the classification performance and consider it a benchmark for evaluating the quality of the generated synthetic point cloud embeddings.

- We train the MLP on synthetically generated point cloud embeddings and evaluate its performance on both *Train* and *Test* sets of the original embeddings. If MLP demonstrates satisfactory performance on the *Train* set, it confirms the equivalence in class discrimination ability between the synthetic and original embeddings. Similarly, if MLP demonstrates satisfactory performance on the *Test* set, it confirms the potential of synthetic embeddings to generalize to unseen data.

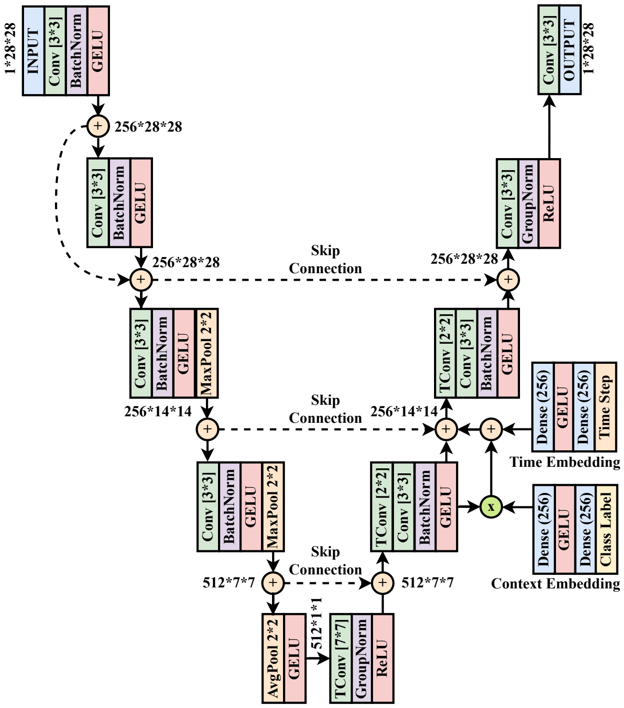

Figure 2. U-Net architecture in DDPM. Context and Time embeddings are encoded via separate dense layers.

- We combine the synthetic embeddings with the *Train* set of original embeddings and evaluate on the *Test* set of the original embeddings. If good classification performance is observed, then one can argue that synthetically generated point cloud embeddings hold meaningful and discriminative information about the data.

Furthermore, to facilitate a more detailed investigation, we conduct sub-experiments with different variations of the dataset. We divide the dataset into three subsets based on the sample count per class. We categorize these subsets as follows: *Small* (fewer than 100 samples in each class), *Medium* (between 100 to 850 samples), and *Large* (between 2100 to 4250 samples). This categorization allows us to analyze how MLP performs on point cloud classes with varying sample sizes.

5.2. Comparison with cVAE & cGAN Embeddings

We implement a class-conditioned VAE [17] and a class-conditioned GAN [8] to compare the quality of the synthetic EEG embeddings generated by the proposed DDPM. Since both VAE and GAN have been widely applied to gener-

Table 2. Classification report with macro-averaged Precision, Recall, and F1 Score. *Train Set* & *Test Set* refers to the training and test set of original embeddings, while *Syn. Embeddings* refers to the embeddings generated via class-conditioned DDPM. Details in Section 6.1.

Dataset Size	Trained on	Tested on	Accuracy	Precision	Recall	F1 Score
Complete	Train Set	Test Set	95.30±0.50	0.94±0.00	0.92±0.00	0.93±0.00
Complete	Syn. Embeddings	Train Set	95.05±0.29	0.88±0.01	0.93±0.02	0.89±0.01
Complete	Syn. Embeddings	Test Set	93.50±0.47	0.84±0.02	0.93±0.01	0.86±0.02
Complete	Syn. + Train Set	Test Set	**95.77±0.48**	**0.94±0.02**	**0.93±0.01**	**0.94±0.01**
Small	Train Set	Test Set	82.45±5.07	0.84±0.02	0.83±0.01	0.83±0.01
Small	Syn. Embeddings	Train Set	95.24±0.75	0.95±0.01	0.95±0.01	0.95±0.01
Small	Syn. Embeddings	Test Set	88.13±3.07	0.87±0.02	0.87±0.04	0.86±0.01
Small	Syn. + Train Set	Test Set	**91.36±2.54**	**0.92±0.02**	**0.91±0.01**	**0.91±0.02**
Medium	Train Set	Test Set	94.53±1.16	0.93±0.01	0.93±0.01	0.93±0.01
Medium	Syn. Embeddings	Train Set	93.60±0.55	0.90±0.02	0.94±0.01	0.92±0.01
Medium	Syn. Embeddings	Test Set	92.46±2.32	0.90±0.02	0.93±0.01	0.92±0.01
Medium	Syn. + Train Set	Test Set	**94.71±1.03**	**0.94±0.01**	**0.95±0.01**	**0.94±0.01**
Large	Train Set	Test Set	96.74±0.04	0.97±0.00	0.97±0.00	0.97±0.00
Large	Syn. Embeddings	Train Set	96.90±0.03	0.97±0.00	0.97±0.00	0.97±0.00
Large	Syn. Embeddings	Test Set	96.74±0.05	0.97±0.00	0.97±0.00	0.97±0.00
Large	Syn. + Train Set	Test Set	**96.74±0.04**	**0.97±0.0**	**0.97±0.0**	**0.97±0.0**

ate synthetic data, comparing them with the proposed approach will allow for a qualitative assessment of synthetic data quality.

5.3. Mutual Information Measure

We compute the Jensen-Shannon Divergence (JSD) score [19] between the synthetic and original embeddings. JSD quantifies the dissimilarity in the information contained within the two distributions. This quantitative evaluation provides a statistical metric for assessing how closely the synthetic embeddings match the original ones. Its value ranges from 0 to 1. A JSD value closer to 0 indicates high similarity between the distributions, whereas a value closer to 1 indicates high dissimilarity.

5.4. Visual Examination

We apply t-SNE [25] plots to evaluate how well the synthetic embedding distribution resembles the original point cloud distribution. t-SNE is a dimensionality reduction technique which maps high-dimensional data to a low-dimensional space while preserving local structures and non-linear relationships between data points. In contrast to other methods of dimensionality reduction, t-SNE addresses the issue of overlapping data points, which leads to enhanced interpretability. We create two-dimensional plots of synthetic and original embeddings for *complete*, *small*, *medium*, and *large* sets.

6. Results

6.1. Class Discrimination Information

We repeat the training process 10 times and report the outcomes in terms of mean score±variance. Table 2 presents a summary of the overall results.

- We observe that combining the synthetic embeddings with the original training set consistently improves the MLP's performance across all dataset sizes. The improvements are especially notable in the *small* and *medium* dataset sizes, where the combined training approach significantly improves the MLP's performance.
- We observe that in the *large* subset, MLP performed exceptionally well across all the cases. This suggests that with enough data samples, proposed DDPM can generate synthetic embeddings identical to the original embeddings.
- We observe that performance of MLP trained on synthetic embeddings remains stable across different dataset sizes, indicating that the proposed method generates robust synthetic embeddings.
- **Class-wise Results:** In Table 3, we present the class-wise classification results obtained by training the MLP on a combination of synthetic embeddings with *Train* set and evaluated on the *Test* set of original embeddings. Upon analyzing these results, we observe there is overall high F1 scores for most classes, which imply a good balance between precision and recall.

Table 3. Class-wise classification report on *Complete* set. MLP is trained on *Syn.+Train Set* and evaluated on *Test* set

Class Name	Precision	Recall	F1 Score
Cap	0.80	0.97	0.88
Rocket	0.89	0.70	0.78
Earphone	0.77	0.83	0.81
Bag	0.90	0.97	0.94
Skateboard	0.97	0.97	0.97
Mug	0.97	0.94	0.96
Motorbike	0.97	0.97	0.97
Knife	0.95	0.94	0.95
Pistol	0.97	0.97	0.97
Laptop	0.97	0.97	0.97
Guitar	0.96	0.97	0.96
Lamp	0.95	0.92	0.93
Airplane	0.97	0.97	0.97
Car	0.96	0.96	0.96
Chair	0.96	0.97	0.97
Table	0.97	0.97	0.97

We also observe that there is a consistency in performance across the classes, with many classes achieving F1 scores above 0.9. However, *Rocket* and *Earphone* have lower precision and recall scores, indicating challenges in correctly identifying these objects compared to other classes.

6.2. Comparison with cVAE & cGAN Embeddings

We expanded our experiments by incorporating cVAE and cGAN to augment latent point cloud embeddings. We present a summary of the results in Table 4. We observe that proposed DDPM outperforms cVAE and cGAN significantly in Complete, Small, Medium, and Large subsets. This indicates that DDPM generates a robust and accurate data representation across different subset sizes. We also observe a higher variance in cVAE and cGAN scores, which indicates that these models are more sensitive to the variations in the dataset and the training process. This sensitivity can lead to inconsistent performance across different runs or subsets of the data, making these models less reliable for real-world applications. In contrast, the proposed DDPM approach delivers good performance across all dataset sizes, with minimal variance in its scores. This consistency highlights the robustness of the proposed method.

6.3. Mutual Information Measure

The JSD scores offer valuable insights into the quality of synthetic embeddings. Lower JSD scores between the two sets indicate a significant similarity. We calculate class-specific JSD scores between the synthetic embeddings and both the *Train* and *Test* sets of the original embeddings. The obtained results are summarized in Table 5. We observe that JSD scores for cVAE-generated synthetic embeddings range from 0.19 to 0.22 for the train set and 0.19 to 0.22 for the test set. These scores suggest that the distributions of cVAE embeddings are moderately similar to both the *Train* and *Test* sets of *Original Embeddings*. In the case of cGAN synthetic embeddings, JSD scores range from 0.28 to 0.31 for both the *Train* and *Test* sets. These higher scores indicate a larger dissimilarity between cGAN embeddings and the original data distributions, suggesting that cGAN-generated embeddings are less aligned with the original data. In contrast, JSD scores for DDPM-generated embeddings are consistently lower, ranging from 0.05 to 0.09 for the *Train* set and 0.05 to 0.08 for the *Test* set. These lower scores indicate that DDPM-generated synthetic embeddings are remarkably similar to the original data distributions, demonstrating a closer match to the underlying data characteristics.

6.4. Visual Examination

We plot two-dimensional t-SNE plots of original and synthetic embeddings generated via proposed DDPM for *complete*, *small*, *medium*, and *large* subsets. These plots (Figure 3) provide insights into how these embeddings are distributed along the t-SNE space.

We observe that synthetically generated embeddings construct distinct clusters. This demonstrates the efficacy of synthetic embeddings in effectively learning discriminating class patterns. In the *small* subset, we observe that all four classes form distinct clusters, with the *Rocket* class exhibiting higher variance. This increased variability in the *Rocket* class aligns with that of the original dataset, where rocket shapes inherently have a wide range of variations. The synthetic embeddings generated for the *medium* subset effectively captured the features of the classes present in it, except for the constrained cluster of the *Lamp* class. *Lamp* class also showed a higher variance.

Furthermore, we observe that class clusters in the *large* subset are highly separable, leading to well-defined boundaries between different classes. One potential reason for this separability is the presence of a large number of data samples. The larger data samples allow the DDPM to learn diverse and complex patterns, resulting in well-defined clusters. The t-SNE visualization of the *complete* set reveals a smooth transition between distinct clusters of all classes. However, the embeddings of the *Lamp* and *Rocket* classes exhibit significant variability, resulting in highly scattered points in the plot for these classes. Apart from this, the visualization confirms the overall quality of the generated synthetic embeddings.

Table 4. Comparison of proposed DDPM approach with cVAE and cGAN. MLP is trained on *Synthetic Embeddings* and evaluated on the test set of *Original Embeddings*. Details in Section 6.2.

	Accuracy	Precision	Recall	F1 Score
cVAE (Complete)	0.68±0.25	0.80±0.12	0.84±0.13	0.73±0.20
cVAE (Small)	0.62±0.18	0.70±0.08	0.62±0.12	0.57±0.23
cVAE (Medium)	0.68±0.19	0.76±0.10	0.71±0.12	0.69±0.22
cVAE (Large)	0.90±0.01	0.90±0.00	0.89±0.01	0.89±0.00
cGAN (Complete)	0.35±0.08	0.24±0.02	0.28±0.05	0.24±0.01
cGAN (Small)	0.43±0.11	0.34±0.16	0.44±0.12	0.37±0.15
cGAN (Medium)	0.35±0.11	0.37±0.13	0.39±0.10	0.29±0.10
cGAN (Large)	0.57±0.09	0.48±0.12	0.63±0.18	0.49±0.15
DDPM (Complete)	**93.50±0.47**	**0.84±0.02**	**0.93±0.01**	**0.86±0.02**
DDPM (Small)	**88.13±3.07**	**0.87±0.02**	**0.87±0.04**	**0.86±0.01**
DDPM (Medium)	**92.46±2.32**	**0.90±0.02**	**0.93±0.01**	**0.92±0.01**
DDPM (Large)	**96.74±0.05**	**0.97±0.00**	**0.97±0.00**	**0.97±0.00**

Table 5. The JSD score is computed between *Synthetic Embeddings* and the Train/Test set of *Original Embeddings*. JSD = 0 means identical distributions; JSD = 1 means dissimilar distributions. Details in Section 6.3.

	cVAE Embeddings		cGAN Embeddings		DDPM Embeddings	
Class Name	Syn.–Train	Syn.–Test	Syn.–Train	Syn.–Test	Syn.–Train	Syn.–Test
Cap	0.20	0.20	0.29	0.29	**0.08**	**0.08**
Rocket	0.22	0.22	0.29	0.28	**0.08**	**0.07**
Earphone	0.19	0.20	0.28	0.28	**0.08**	**0.07**
Bag	0.20	0.20	0.29	0.28	**0.07**	**0.07**
Skateboard	0.21	0.21	0.29	0.29	**0.08**	**0.08**
Mug	0.19	0.19	0.29	0.29	**0.08**	**0.08**
Motorbike	0.20	0.20	0.29	0.29	**0.06**	**0.06**
Knife	0.20	0.20	0.29	0.29	**0.07**	**0.06**
Pistol	0.20	0.20	0.28	0.28	**0.09**	**0.08**
Laptop	0.20	0.20	0.29	0.29	**0.08**	**0.08**
Guitar	0.21	0.21	0.30	0.30	**0.05**	**0.05**
Lamp	0.20	0.20	0.31	0.31	**0.07**	**0.07**
Airplane	0.19	0.19	0.30	0.30	**0.07**	**0.07**
Car	0.20	0.20	0.29	0.29	**0.07**	**0.07**
Chair	0.21	0.21	0.30	0.30	**0.07**	**0.07**
Table	0.19	0.19	0.31	0.31	**0.08**	**0.08**

7. Discussions

The collection of high-quality point cloud data can be expensive, and its limited availability often poses challenges in developing robust DNN models. DA solves this problem via generating synthetic data. In our approach, we apply a class-conditioned DDPM to generate synthetic data. DDPM learns a denoising process which helps the model to learn intricate patterns within data. As the noise level decreases over iterations, the model gradually refines its understanding of the data. While comparing DDPM with other generative methods like VAE and GAN, DDPM offers several advantages such as (a) It provides explicit likelihood function for generated samples, whereas VAE approximates the likelihood and GAN does not provide at all, (b) DDPM tends to be more stable during training compared to GAN. Also, GANs may suffer from mode collapse, where they fail to explore the entire data distribution, especially when data is not uniformly distributed among the classes. DDPM are less prone to the aforementioned issues. We observe that the proposed method efficiently learns the point cloud distributions even for classes with very limited sample sizes. This can be advantageous in scenarios where obtaining point cloud data is expensive and non-trivial. An

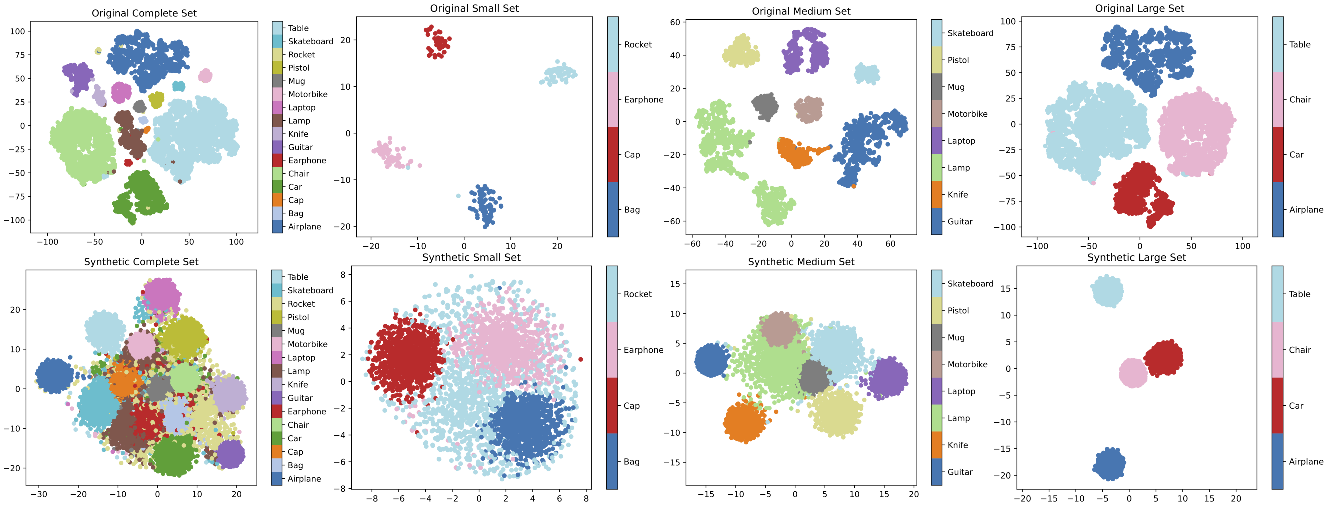

Figure 3. t-SNE visualizations of original (top row) and synthetic embeddings via class-conditioned DDPM (bottom row). The columns represent different dataset sizes: Complete, Small, Medium, and Large, from left to right. Distinguishable embeddings can be observed in the plots. For details refer to the Section 6.4.

additional benefit of the proposed approach is its capacity to generate any number of samples for each class. Furthermore, being a class-conditioned model, it eliminates the need to train multiple models for individual classes.

8. Conclusion & Future Work

Data augmentation is a widely adopted practice in deep learning due to its simplicity and effectiveness. It reduces the need for extensive data collection, allowing deep neural networks to achieve remarkable performance even with limited data. Moreover, data augmentation can be easily integrated into existing deep learning pipelines, making it a valuable tool for model generalization and cost-effectiveness across various applications. In this paper, we present a class conditioned Denoising Diffusion Probabilistic Model-based latent feature space data augmentation method for point cloud data. We aim to synthetically generate high-quality point cloud latent embeddings, which are compact feature representations encapsulating spatial and semantic information of point cloud data.

In the scope of this paper, we applied PointNet to extract latent class-wise feature representations from point clouds due to its inherent ability to extract well-discriminative feature representations. It would be intriguing to explore masked autoencoders or self-supervised learning-based methods, with an emphasis on extracting class-wise disentangled representations. We investigate our proposed approach on the synthetic dataset where objects are complete and without any background and occlusion. Objects in real-world datasets may contain background noise and may be occluded at different levels. It will be interesting to check how the proposed method works with these datasets and how the proposed approach can incorporate these intrinsic noises. Another future work will be to add semantic information aware loss functions for a better-informed reconstruction.

References

[1] Sam Bond-Taylor, Adam Leach, Yang Long, and Chris G. Willcocks. Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):7327–7347, 2022. 2

[2] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[3] R. Qi Charles, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 2, 3

[4] Lei Chen, Le Wu, Kun Zhang, Richang Hong, Defu Lian, Zhiqiang Zhang, Jun Zhou, and Meng Wang. Improving recommendation fairness via data augmentation. In *Proceedings of the ACM Web Conference 2023*, WWW '23, page 1012–1020, New York, NY, USA, 2023. Association for Computing Machinery. 1

[5] João Correia, Tiago Martins, and Penousal Machado. Evolutionary data augmentation in deep face detection. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, GECCO '19, page 163–164, New York, NY, USA, 2019. Association for Computing Machinery. 1

[6] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data augmentation for deep neural network acoustic model-

ing. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 23(9):1469–1477, sep 2015. 1

[7] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc., 2021. 3

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139–144, oct 2020. 2, 4

[9] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4338–4364, 2021. 1

[10] Benjamin Hagedorn and Jürgen Döllner. High-level web service for 3d building information visualization and analysis. GIS '07, New York, NY, USA, 2007. Association for Computing Machinery. 1

[11] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006. 2

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. 2

[13] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 3, 4

[14] Anastasia Ioannidou, Elisavet Chatzilari, Spiros Nikolopoulos, and Ioannis Kompatsiaris. Deep learning advances in computer vision with 3d data: A survey. *ACM Comput. Surv.*, 50(2), apr 2017. 1

[15] Sihyeon Kim, Sanghyeok Lee, Dasol Hwang, Jaewon Lee, Seong Jae Hwang, and Hyunwoo J. Kim. Point cloud augmentation with weighted local transformations. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 528–537, 2021. 2

[16] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 3, 4

[17] Durk P Kingma, Shakir Mohamed, Danilo Jimenez Rezende, and Max Welling. Semi-supervised learning with deep generative models. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. 4

[18] Varun Kumar, Hadrien Glaude, and William M. Campbell. A closer look at latent space data augmentation for few-shot intent classification. In *EMNLP 2019 Workshop on DeepLo*, 2019. 2

[19] J. Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151, 1991. 5

[20] Xiaofeng Liu, Yang Zou, Lingsheng Kong, Zhihui Diao, Junliang Yan, Jun Wang, Site Li, Ping Jia, and Jane You. Data augmentation via latent space interpolation for image classification. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 728–733, 2018. 2

[21] S. Ramanathan, Ashraf Kassim, Y.V. Venkatesh, and Wu Sin Wah. Human facial expression recognition using a 3d morphable model. In *2006 International Conference on Image Processing*, pages 661–664, 2006. 1

[22] Gulshan Sharma, Abhinav Dhall, and Ramanathan Subramanian. Medic: Mitigating EEG data scarcity via class-conditioned diffusion model. In *Deep Generative Models for Health Workshop NeurIPS 2023*, 2023. 3

[23] Yu Shi and Chuanchuan Yang. Point cloud inpainting with normal-based feature matching. *Multimedia Syst.*, 28(2):521–527, apr 2022. 2

[24] Joshua Snoke, Gillian M. Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. General and Specific Utility Measures for Synthetic Data. *Journal of the Royal Statistical Society Series A: Statistics in Society*, 181(3):663–688, 03 2018. 4

[25] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. 5

[26] Qingsong Wen, Liang Sun, Fan Yang, Xiaomin Song, Jingkun Gao, Xue Wang, and Huan Xu. Time series data augmentation for deep learning: A survey. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, pages 4653–4660. International Joint Conferences on Artificial Intelligence Organization, 8 2021. Survey Track. 1

[27] Yanhao Wu, Tong Zhang, Wei Ke, Sabine Süsstrunk, and Mathieu Salzmann. Spatiotemporal self-supervised learning for point clouds in the wild. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5251–5260, June 2023. 2

[28] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 1