2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

LiLa-Net: Lightweight Latent LiDAR Autoencoder for 3D Point Cloud Reconstruction

Mario Resino[1], Borja Pérez[1], Jaime Godoy[1], Abdulla Al-Kaff[1] and Fernando García[1]

***Abstract*— This work proposed a 3D autoencoder architecture, named LiLa-Net, which encodes efficient features from real traffic environments, employing only the LiDAR's point clouds. For this purpose, we have real semi-autonomous vehicle, equipped with Velodyne LiDAR. The system leverage skip connections concept to improve the performance without using extensive resources as the state-of-the-art architectures. Key changes include reducing the number of encoder layers and simplifying the skip connections, while still producing an efficient and representative latent space which allows to accurately reconstruct the original point cloud. Furthermore, an effective balance has been achieved between the information carried by the skip connections and the latent encoding, leading to improved reconstruction quality without compromising performance. Finally, the model successfully reconstruct objects unrelated to the original traffic environment.**

I. Introduction

The perception and understanding of the environment in autonomous vehicles remain a significant challenge, involving both hardware components - such as 2D/3D sensors, control units, proximity detectors, and audio sensors - and software modules, including path planning, obstacle detection, environment classification, and automatic control. Among these technologies, LiDAR sensors have emerged as a key tool for capturing accurate and detailed 3D representations of traffic environments. However, the high volume of data generated by point clouds creates the need for models that can efficiently extract meaningful features while minimizing computational and memory demands.

Recent advances have explored Transformer-based architectures for 3D point cloud processing [1], leveraging attention mechanisms to capture spatial dependencies. While effective, these methods are often computationally expensive and resource-intensive, limiting their practical deployment in real-time systems.

To address these limitations, this work introduces LiLa-Net, a lightweight end-to-end framework for point cloud feature extraction and reconstruction. LiLa-Net learns compact and expressive latent representations that preserve the structural features of scenes, enabling efficient compression and consistent reconstruction with minimal error.

Furthermore, after validation in traffic-focused datasets, LiLa-Net demonstrated strong adaptability when applied to unrelated objects, showing promising generalization and fine-tuning capabilities even with limited additional data.

[1]Authors are affiliated with Universidad Carlos III de Madrid, Department of Systems Engineering and Automation, Autonomous Mobility and Perception Lab (AMPL), Madrid, Spain {mresino, boperezl, jgodoy}@pa.uc3m.es, {akaff, fegarcia}@ing.uc3m.es

The key contributions of this work are as follows: i) the proposal of LiLa-Net, a novel point cloud autoencoder framework that directly operates on sparse 3D points, avoiding voxelization or intermediate representations; ii) the ability to efficiently handle large and dense point clouds, enabling processing at the scene level rather than on isolated objects; iii) demonstration of effective compression and reconstruction of complex traffic environments collected from moving vehicles; iv) evidence of reasonable generalization of the learned latent representations, achieving consistent classification performance on entirely different datasets without retraining; and v) elimination of pretraining or masking strategies, resulting in a simpler, faster, and more direct training pipeline.

The remainder of this paper is organized as follows. Section II reviews the state-of-the-art (SOTA) methods in the field. Section III describes the proposed framework, detailing its encoder–decoder architecture and the integration of skip connections. Section IV presents the experimental results, including an analysis of different architectural configurations, as well as qualitative and quantitative evaluations across various tasks, such as reconstruction and cross-dataset classification, in comparison with other SOTA approaches. Finally, Section V summarizes the key findings of this work.

II. Related Works

Recent advances in 3D point cloud processing have focused on the development of architectures capable of learning meaningful latent representations directly from raw data, effectively overcoming the limitations of earlier voxel or image-based approaches [2], [3]. The dominant paradigm driving this progress is self-supervised learning (SSL) [4], [5], which enables large-scale training without the need for manual annotation.

Early SSL approaches relied on pretext tasks, such as reconstructing point clouds that had been shuffled or partially deformed, thereby forcing the network to capture the underlying geometric structure of the object. While effective, these methods have recently been outperformed by masked modeling strategies, inspired by Masked Autoencoders (MAE) [6], [7] from other domains.

The masked modeling paradigm divides the point cloud into patches, masks a high percentage of them, and trains an asymmetric Transformer-based autoencoder to reconstruct the missing parts. This approach offers significant computational efficiency, as a powerful encoder processes only the visible patches, while a lighter decoder reconstructs the

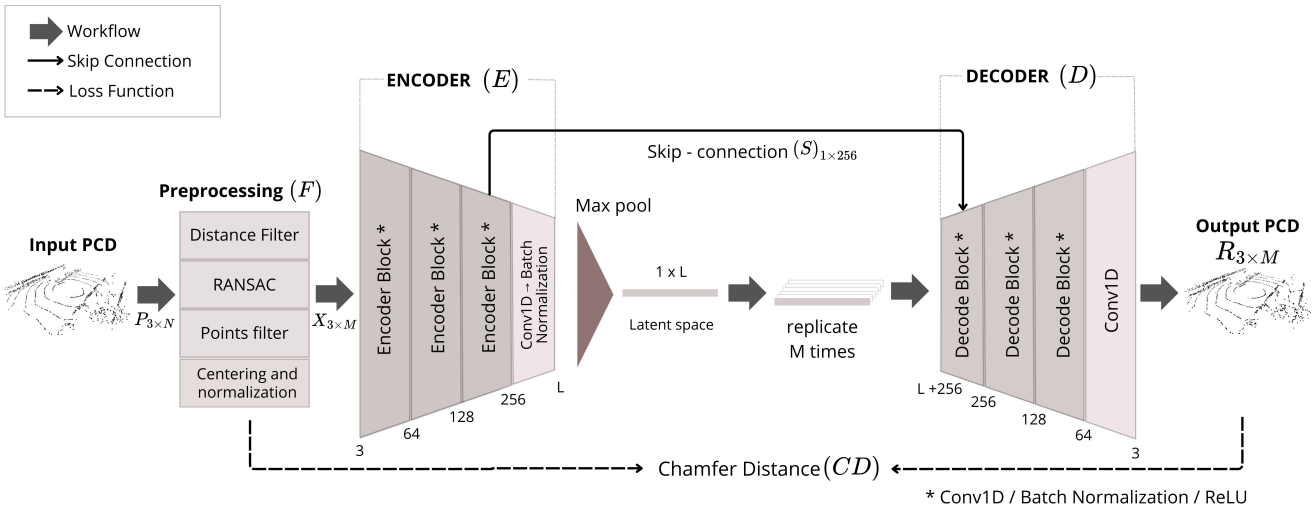


Fig. 1: Proposed LiLa-Net Architecture: point clouds are first preprocessed (F). Then, multiple encoder layers (E) progressively reduce the information to the latent space dimension (L). For reconstruction, the latent representation is concatenated with the skip connection features (S) at the lowest level, followed by a series of decoder layers (D) to obtain the final reconstruction. The loss function is the Chamfer Distance between the reconstructed output and the preprocessed point cloud.

occluded regions. Models such as Point-MAE [7], Point-BERT [8], and hierarchical variants like Point-M2AE [9] have demonstrated remarkable ability to learn robust and transferable features, which can be fine-tuned for downstream tasks, including classification and segmentation.

In parallel, generative modeling for point clouds has seen substantial progress. Classical autoencoders, such as FoldingNet [10], introduced innovative decoders that "fold" a 2D grid to reconstruct 3D objects, while Variational Autoencoders (VAEs) [11], [12] enabled the creation of probabilistic latent spaces well-suited for shape interpolation. Although, Generative Adversarial Networks (GANs) [13], [14] played an important role in early research, diffusion models have now emerged as the state-of-the-art for high-fidelity generation. These models generate highly realistic 3D shapes by reversing a noise-injection process. Hybrid architectures, such as DiffPMAE [15], combine the efficiency of MAE-based encoders with the generative power of diffusion, establishing a new benchmark in quality.

In summary, recent research highlights two main directions: methods designed to learn compact and structured latent spaces for effective representation learning, and generative approaches, such as diffusion models, that have achieved remarkable progress in 3D object synthesis. Both lines emphasize the importance of capturing geometric and semantic essence of 3D data, which continues to drive advancements across a wide range of downstream applications.

III. Methodology

The following sections describe the proposed methodology LiLa-Net, illustrated in Fig. 1. The process begins with data preprocessing (F), after which the processed point cloud (X) is passed through the encoding (E) to extract the most relevant features and generate a compact latent representation. This latent vector, together with a skip connection (S) from the encoder (E), is then fed into the decoding block (D) to reconstruct the final point cloud (R) and evaluate it with the Chamfer Distance (CD) metric. Each stage of the pipeline is described in detail in the following subsections.

A. Data Acquisition

To conduct this study, a proprietary dataset was collected at the AMPL laboratory of Universidad Carlos III de Madrid using our recording platform, an updated version of Atlas platform [16]. From this dataset, only the point clouds with spatial information ($P_{3\times N}$) corresponding to complete sequences were extracted, which were captured using a Velodyne VLP-32C LiDAR sensor around the Center for Innovation in Entrepreneurship and Artificial Intelligence (C3N-IA) at the UC3M technological Park.

B. Pipeline

1) Preprocessing: The point clouds from this dataset must be preprocessed before being fed into E due to the large proportion of points that do not contribute relevant scene information. For instance, ground points-which can dominate the input and undesirably influence the encoder's attention. To address this, the classical RANSAC [17] algorithm is employed to detect and remove ground points.

In addition to ground removal, a horizontal range filter is applied by defining a cylindrical region around the sensor. Points falling outside this parametric radius (ranging from 15 to 200 meters) are discarded. The effect of varying this radius is analyzed in the experimental results in Section IV-C.

Finally, to match the encoder's input requirements, the point cloud is randomly downsampled until get M number of points ($X_{3\times M}$).

2) Encoder: From $X_{3\times M}$, E is responsible for extracting a compact and highly rich feature representation, capturing its essential geometric features. It takes as input a tensor of

shape $B \times 3 \times M$, with batch size B. E is composed of a sequence of shared 1D convolutional layers, which operate independently on each point. These layers progressively increase the feature dimensionality from 3 to the desired latent space size (L). Each convolutional layer is followed by a Batch Normalization and ReLU activation to improve convergence and introduce non-linearity.

After the final convolution, a global feature vector is obtained by applying a max-pooling operation over the point dimension. This operation aggregates the most prominent feature across all points, resulting in a fixed-length latent vector of shape $1 \times L$.

3) Latent Feature Space: From the latent vector with L size, the framework encodes the most relevant information required for reconstructing the original point cloud from the bottleneck. This process yields a latent space that captures the global 3D structure of the scene and the semantic context of its components, while discarding less informative content. The resulting representation has a fixed size of 1×1024 dimensions. In this way, the architecture can be trained to represent 3D point maps through a fixed-dimensional latent vector, invariant to density variations, point ordering, or minor scene deformations.

These latent representations have proven highly useful not only for reconstruction but also for a wide range of downstream tasks, including classification, clustering, retrieval, and generative modeling [18], [19], [20].

4) Skip Connection: In addition to the latent space, part of the information required for reconstructing the point cloud is carried through S, which transfers features extracted from a single encoder layer directly to the corresponding decoder stage Based on the experiments described in Section IV-A, the framework was designed to retain only the skip connection from the last encoder layer. This choice ensures that the reconstruction relies primarily on the latent space while preserving the minimal complementary information needed for a good reconstruction. As a result, the latent representation becomes richer and more informative.

5) Decoder: Once all the information from the original point cloud has been encoded and the features have been successfully extracted, the final module of our framework is D, responsible for transforming a global feature vector and the skip connection features back into a set of 3D coordinates representing the reconstructed point cloud $R_{3\times M}$.

In our architecture, D is implemented as a sequence of shared 1D convolutional layers with kernel size 1, each followed by Batch Normalization and a ReLU activation, as in E. These layers progressively refine the feature maps until reaching the final output dimensionality, which corresponds to the desired number of 3D points.

C. Dataset

A study was carried out to collect multiple recordings with varying characteristics using our recording platform at the Autonomous Mobility and Perception Lab (AMPL) of Universidad Carlos III de Madrid, Spain. A total of 4,955 point clouds were initially collected. Nevertheless, to assess the robustness of the proposed architecture, we performed experiments using different maximum range thresholds and varying the number of input points N. This analysis allowed us to test the feasibility of reconstructing larger point clouds in terms of both spatial extent and point density. After that P is passing through F, which provides X as an output and split into 4,460 ($\sim 90\%$) point clouds for training and 495 ($\sim 10\%$) point clouds for testing.

To further evaluate the generalization capability of our approach and to provide a comparison with existing methods, experiments were also conducted on the publicly available ModelNet10 and ModelNet40 datasets [21], which consist of numerous point clouds of specific objects and are primarily used for classification tasks. Following the standard protocol, the official training and testing splits provided with each dataset were employed.

This combination of a proprietary dataset—captured under realistic automotive conditions—and publicly available benchmarks allows us to assess both the domain-specific reconstruction capabilities of our autoencoder and its transferability to more general 3D shape modeling tasks. Unless otherwise specified, the main configuration employed throughout the experiments—particularly for extrapolation to external datasets—was trained with 2,048 points per point cloud. However, experiments with 8,192 and 20,000 points were conducted and analyzed in section IV-C to ensure the ability to extract information and reconstruct large point clouds of automotive conditions.

D. Training Setup

With the training datasets already prepared and organized, the next step was the training step. All models were trained on a single NVIDIA RTX 4090 GPU with 24 GB of VRAM, using a workstation equipped with a 13th Gen Intel® Core™ i9-13900K and 64 GB of RAM. The model was optimized for 100 epochs using a batch size of 32 and an initial learning rate of 5×10^{-4}. Training employed the Adam optimizer [22], and the reconstruction objective was defined as the Chamfer Distance (CD) [23] between the input and the reconstructed point sets. The architecture used in the majority of our experiments, and in particular for cross-dataset extrapolation to ModelNet10 and ModelNet40 [21], consisted of three encoder and three decoder blocks with $L = 1,024$. Unless otherwise specified, the model was trained with $M = 2,048$, which we found to provide a good trade-off between computational efficiency and reconstruction accuracy.

Prior to training, all point clouds were normalized to remove the effect of global translations and variations in scale. Given $P = \{p_i \in \mathbb{R}^3\}_{i=1}^{N}$, where p_i represents the 3D coordinates of the i-th point, the centroid $\bar{p} = \frac{1}{N}\sum_{i=1}^{N} p_i$ (i.e., the mean position of all points) was first computed and subtracted from each point, effectively translating the cloud to the origin $(0, 0, 0)$. The cloud was then uniformly scaled so that its maximum distance from the origin equaled one:

$$p'_i = \frac{p_i - \bar{p}}{\max_j \|p_j - \bar{p}\|_2}, \quad i = 1, \ldots, N, \tag{1}$$

where $\mathbf{p}'_i$ is the normalized point, p_j denotes the j-th point in P, and $\|\cdot\|_2$ denotes the Euclidean norm. This normalization ensured that all input point clouds shared a consistent spatial distribution, allowing the network to focus on structural features rather than global positioning.

The reconstruction objective was the Chamfer Distance (CD) [23] between the predicted reconstructed point cloud R and the preprocessed input X, defined as:

$$CD(X,R) = \frac{1}{|X|}\sum_{p\in X}\min_{\hat{p}\in R}\|p-\hat{p}\|_2^2 + \frac{1}{|R|}\sum_{\hat{p}\in R}\min_{p\in X}\|\hat{p}-p\|_2^2. \tag{2}$$

While CD is efficient and captures local proximity between point sets, it may overlook discrepancies in global structure. To address this, we also evaluated the reconstructions using the Earth Mover's Distance (EMD) [23], defined as:

$$EMD(X,R) = \min_{\phi:X\to R}\frac{1}{|X|}\sum_{p\in X}\|p-\phi(p)\|_2, \tag{3}$$

where ϕ denotes a bijection between point sets. Unlike CD, EMD finds an optimal matching between the two sets, yielding a more faithful assessment of global shape similarity, at the expense of higher computational cost. This dual evaluation allows for a more comprehensive comparison with other models in the literature.

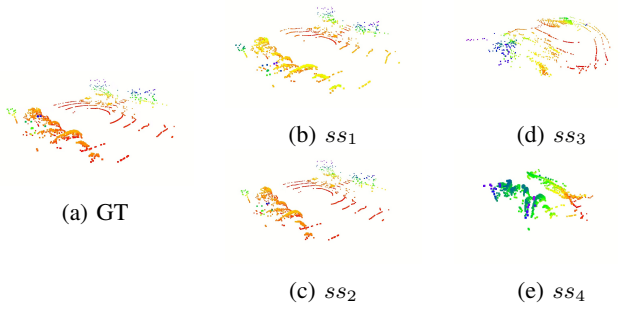
IV. Experiments and Discussion

Building on LiLa-Net, a series of experiments were conducted to evaluate the framework's performance and to refine its components. The goal was to optimize each module, ensuring a robust reconstruction of the automotive environment while producing a high-quality latent space that effectively captures the scene's structure and semantic content.

A. Architecture Refinement Experiments

From the initial architecture and with the training dataset prepared, a series of studies were conducted to refine and optimize the model design. A systematic evaluation was performed to analyze the role of S in the encoder–decoder structure, since these connections are typically present at every encoding layer. To this end, we consider various experiments to identify the optimal skip connection, such as the common skip connection setup ss_1 which connects all encoding layers with their corresponding decoding layer, ss_2 which connects only first encoding layer with last decoding layer, ss_3 which connects second encoder layer with second decoder layer or ss_4 which connects last encoding layer with first decoding layer. In order to ensure that the reconstruction relied solely on the information transmitted through S, the latent vector extracted by E was replaced with a random vector before being replicated and passed to D.

The results, summarized in Fig. 2 and Table I, show that S in early layers of E makes less relevant the encoded latent space. Reconstructions could still be achieved with reasonable quality even when the latent input to D was random, indicating limited dependence on the latent space at early layers. However, in the case of S in the last layer of E, reconstruction quality degraded significantly under the random latent space condition. However, when the random vector was replaced by the actual latent representation, reconstruction quality was restored, highlighting the critical importance of latent space information.



(a) GT (b) ss_1 (c) ss_2 (d) ss_3 (e) ss_4

Fig. 2: Comparison of R under different S configurations Using a fixed random latent space. The color gradient along the height axis highlights how the reconstruction worsens with the depth of S in the network.

These observations demonstrate two complementary findings. First, in point cloud autoencoders, acceptable reconstructions can be achieved with minimal encoding depth, relying primarily on outermost skip connections; however, in such cases the latent space carries little meaningful information and remains poor in quality. Second, when the objective is to obtain a rich and informative latent representation like in our case, the architecture can be designed with inner encoding layers while retaining only the final skip connection near the bottleneck. Table II presents a comparative analysis across different methods and our skip connection configurations (ss_1, ss_2, ss_3, ss_4), using the experimental setup described in Section III-D

Finally, since the objective of this architecture is to obtain a higher-quality latent space, the remaining experiments were carried out using the version with only the innermost skip connection setup ss_4.

TABLE I: Comparison of CD and EMD under Different S Configurations.

Metric	ss_1	ss_2	ss_3	ss_4
CD	0.003164	0.000287	0.028953	0.210549
EMD	0.049724	0.014079	0.288271	0.498255

TABLE II: Comparative Results Across Different Methods and Our Implementations; Showing the impact of the number of skip connections on model parameters, overall size, and inference speed.

Method	**# params (M)**	**Total Size (MB)**	**Inference Time (s)**
AE-EM [11]	39,650	151.51	0.00270
Point-BERT [8]	27,620	193.80	0.00450
FoldingNet [10]	1,740	177.81	0.00360
Ours-ss_1	0,698	62.73	0.00162
Ours-ss_2	0,616	62.42	0.00159
Ours-ss_3	0,628	62.47	0.00156
Ours-ss_4	0,677	62.66	0.00161

B. Evaluation with Varying Training Data

To study the impact of data volume on our architecture, we generated a custom dataset composed of subsets of the original dataset (4460 point clouds). The dataset was divided into incremental proportions with a step size of 10%. The objective of this procedure was to analyze the convergence behavior of the employed metric functions. To reduce the effect of stochastic variability, 50 independent training runs were performed for each custom dataset, resulting in a total of 500 training sessions. The outcomes of these experiments are summarized in Fig. 3, where results are presented as a hybrid between a line plot and a box plot. Since the metrics do not share the same scale, each one is represented with its own axis. The box plots illustrate the distributions obtained, whereas the line plots indicate their mean values.

C. Comparison Across Different Point Cloud Sizes

Once accurate scene reconstruction was achieved using LiLa-Net, a study was conducted to assess the impact of X size on reconstruction quality. The aim was to determine whether increasing the field of view and M would introduce inaccuracies, particularly for regions farther from the origin, or create challenges in processing larger point clouds.

To this end, three models with identical architectures were trained using different sizes of M: 2,048 points, 8,192 points, and 20,000 points. Inference was then performed on the same sequences to compare reconstruction metrics, including CD and EMD, as reported in Table III.

The results indicate that while input size has a significant effect on inference time, it does not compromise reconstruction accuracy. In fact, larger point clouds often achieve equal or slightly improved reconstruction quality due to the richer data representation. This improvement is also visually evident in Fig. 4, which shows the reconstruction of the same scene with varying point cloud sizes.

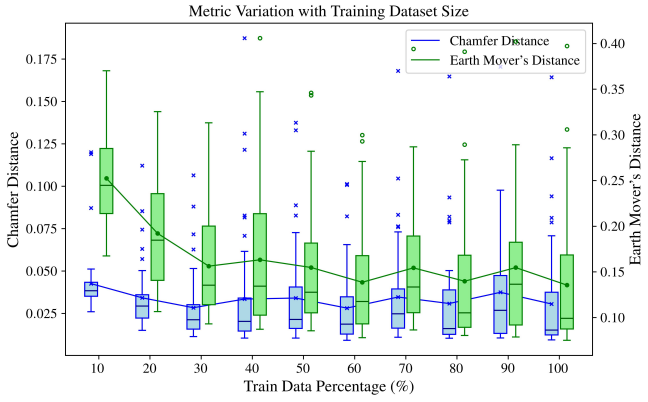


Fig. 3: Evolution of Chamfer Distance (CD) and Earth Mover's Distance (EMD) with varying training dataset size. Each point is based on 50 independent trainings per dataset size (500 in total). Line plots represent mean values (including outliers), while box plots depict the median and interquartile range.

D. Evaluation of Existing Point Cloud Models on Our Platform Recording

To further validate the complexity of our platform recording, we tested several publicly available point cloud autoencoder repositories using the same settings as our model evaluation. Specifically, we evaluated the implementations from [10], [11], [8].

Our experiments revealed that the models from [10], [11] failed to learn meaningful latent representations of our dataset, resulting in poor reconstructions and unstable training. In contrast, the implementation from [8] achieved relatively better training convergence and was able to reproduce point clouds more consistently. Nevertheless, the reconstructions remain far from satisfactory, underscoring the unique challenges posed by our platform recording compared to other standard datasets.

A detailed comparison of reconstruction quality for each model is reported in Table IV, where qualitative examples highlight the significant gap between existing approaches and the requirements for handling the complexity of our dataset.

Moreover, in Table V, we provide a quantitative evaluation based on CD and EMD. The results confirm the qualitative observations reported in Table IV. Specifically, both AE-EM [11] and FoldingNet [10] exhibit high error values, reflecting their inability to capture the geometric complexity of our point clouds. PointBERT [8] achieves significantly lower distances, indicating more consistent reconstructions. However, our proposed model further reduces both CD and EMD by a large margin, highlighting its effectiveness in learning meaningful latent representations.

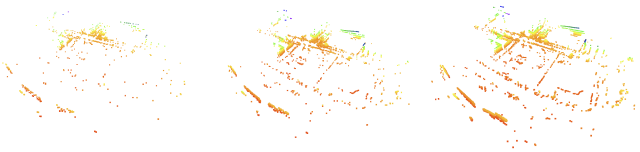
E. Cross-Dataset Extrapolation

To evaluate the generalization ability of our autoencoder LiLa-Net beyond the domain of automotive LiDAR data, we investigated its performance on the widely used synthetic ShapeNet dataset [24]. ShapeNet comprises over 50,000 3D CAD models spanning 55 object categories (e.g., airplanes, chairs, lamps), providing a diverse benchmark for 3D object representation learning.

We conducted extrapolation experiments by applying our model, trained exclusively on real-world LiDAR point

TABLE III: Comparison of Reconstruction Performance and Processing Time for Different M Values.

M	**Inf. Time (ms)**	CD **Loss**	EMD **Loss**
2,048	1.616	16.03×10^{-5}	0.126
8,192	5.612	9.50×10^{-5}	0.009
20,000	12.963	4.63×10^{-5}	0.008



(a) $M = 2{,}048$ (b) $M = 8{,}192$ (c) $M = 20{,}000$

Fig. 4: Reconstructed Point Clouds ($R_{3\times M}$) of the Same Scene Using Varying Input Cloud Sizes (M).

TABLE IV: Qualitative Comparison of Existing Point Cloud Models and Our Method on Our Platform Recording Dataset.

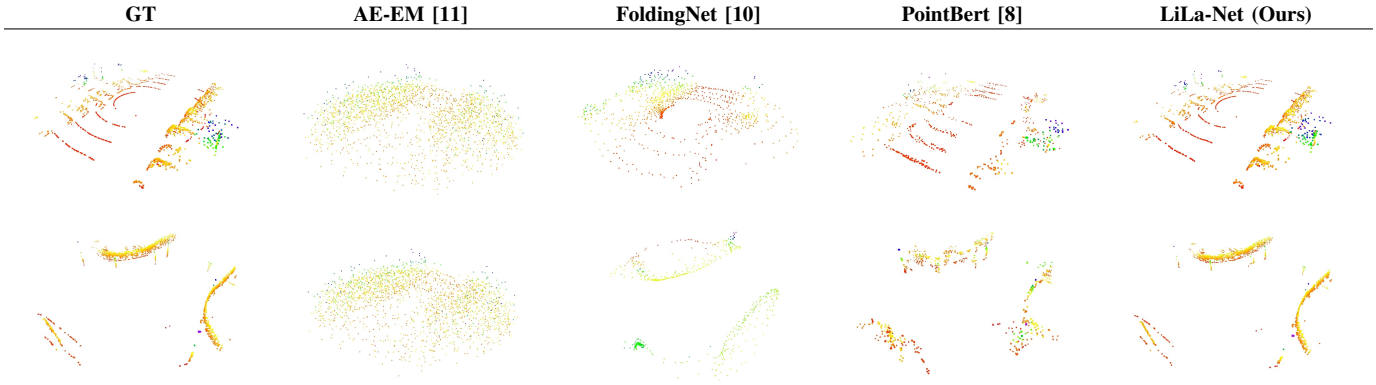


TABLE V: Comparison of CD and EMD Across Point Cloud Models on Our Own Traffic Dataset.

Metric	AE-EM [11]	FoldingNet [10]	PointBert [8]	LiLa-Net (Ours)
CD	25.03	0.397	0.0109	16.03×10^{-5}
EMD	5.430	7.493	0.136	0.012

clouds, directly to ShapeNet without any further fine-tuning. In these evaluations, we used the same processing pipeline as in the original domain, including normalization to unit sphere and fixed input size of 2,048 points, thereby assessing whether our learned latent representations can capture meaningful structure and facilitate accurate reconstruction in a drastically different setting.

Qualitative results reveal that the model is capable of recovering fine-grained geometric details across a variety of object categories, even though it was never exposed to such shapes during training. As illustrated in Table VII, we present examples from six distinct classes of the ShapeNet dataset, showcasing the model's ability to generalize across a wide spectrum of geometries.

These findings highlight the ability of the proposed architecture not only to reconstruct data from familiar scenes, but also to generalize it to unknown and structurally different domains, reinforcing its potential for implementation in scenarios where data may be limited or domain-specific.

F. Latent Space Evaluation

To further assess the quality of the latent representations learned by our model, we conducted a classification experiment on the ModelNet10 and ModelNet40 datasets [21], which are widely used benchmarks in 3D shape analysis. ModelNet10 consists of 10 object categories with 4,899 CAD models, while ModelNet40 contains 40 categories with 12,311 CAD models, providing a diverse collection of synthetic 3D objects suitable for evaluating the generalization ability of point cloud encoders.

Table VI reports the classification accuracy obtained by training a linear SVM on the latent representations extracted by our model and several baselines from the literature.

Some models specifically designed for classification or pre-trained on large-scale datasets such as ModelNet, ShapeNet, or ScanObjectNN achieve higher accuracy than ours. In contrast, our model was not pre-trained on any of these datasets and was originally developed for point cloud reconstruction rather than classification. Therefore, the purpose of this experiment is not to compete with SOA classification performance, but rather to provide a quantitative comparison of the expressiveness of the latent space.

TABLE VI: Comparison of Classification Accuracy on ModelNet10 and ModelNet40 Using Different Point Cloud Models.

Method	SVM	Acc. ModelNet10 (%)	Acc. ModelNet40 (%)
PointNet [25]	-	-	86.20
3D-GAN [13]	✓	91.00	83.30
AE-EM [11]	-	-	85.70
VIP-GAN [26]	✓	94.05	91.98
FoldingNet [10]	✓	94.40	88.40
MAP-VAE [7]	✓	94.82	90.15
Point-Flow [27]	✓	93.70	86.80
Point-BERT [8]	-	-	93.20
MaskPoint [28]	-	-	93.80
Point- M2AE [9]	✓	-	92.90
Point-MAE [7]	-	-	93.80
MAE3D [6]	-	95.50	90.60
Ours	✓	91.74	86.81

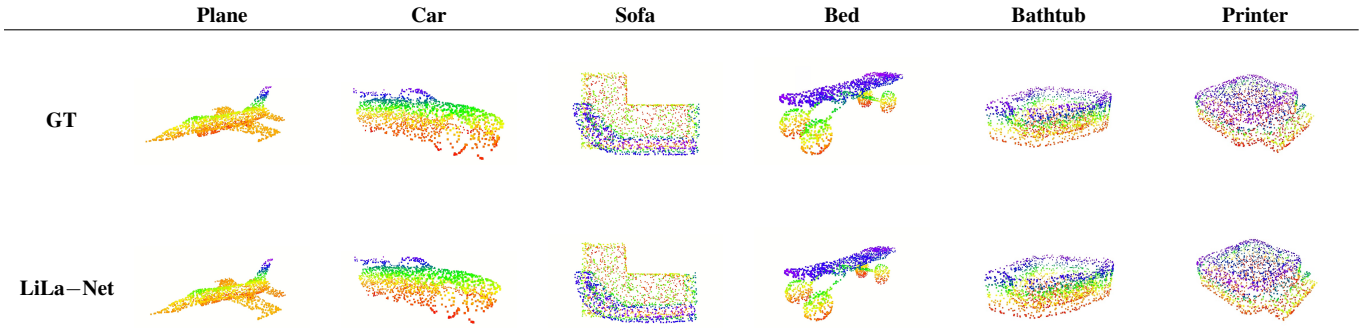
V. Conclusions

This work presented the implementation of a robust domain-shift autoencoder for point cloud generation with skip connections across different layers of the architecture to obtain a representative latent space from which the original data can be reconstructed. The proposed design achieved strong results in point cloud reconstruction, as demonstrated both by qualitative evaluations and by quantitative metrics such as accuracy, Earth Mover's Distance (EMD) and Chamfer Distance (CD). Experiments conducted on diverse datasets and point cloud representations further confirmed the reliability and precision of the approach. In summary, the results confirm that the proposed architecture provides a robust and effective framework for 3D point cloud reconstruction, consistently producing accurate and high-quality representations across diverse datasets.

Acknowledgment

This work has been supported by the Spanish Government through the projects PID2021-128327OA-

TABLE VII: Qualitative Extrapolation Results on Shape-Net Dataset. Six object categories are shown, with ground truth (GT) and model predictions.



I00, and TED2021-129374A-I00 funded by MCIN/AEI/10.13039/501100011033, by "ERDF A way of making Europe" and by the European Union NextGenerationEU/PRTR respectively.

REFERENCES

[1] Dening Lu, Qian Xie, Mingqiang Wei, Kyle Gao, Linlin Xu, and Jonathan Li, "Transformers in 3d point clouds: A survey," *arXiv preprint arXiv:2205.07417*, 2022.

[2] Bojun Liu, Yangzhi Ma, Ao Luo, Li Li, and Dong Liu, "Voxel-based point cloud geometry compression with space-to-channel context," 2025.

[3] Pin Tang, Zhongdao Wang, Guoqing Wang, Jilai Zheng, Xiangxuan Ren, Bailan Feng, and Chao Ma, "Sparseocc: Rethinking sparse latent representation for vision-based semantic occupancy prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15035–15044.

[4] Jonathan Sauder and Bjarne Sievers, "Self-supervised deep learning on point clouds by reconstructing space," 2019.

[5] Idan Achituve, Haggai Maron, and Gal Chechik, "Self-supervised learning for domain adaptation on point-clouds," 2022.

[6] Jincen Jiang, Xuequan Lu, Lizhi Zhao, Richard Dazeley, and Meili Wang, "Masked autoencoders in 3d point cloud representation learning," 2023.

[7] Yatian Pang, Wenxiao Wang, Francis E. H. Tay, Wei Liu, Yonghong Tian, and Li Yuan, "Masked autoencoders for point cloud self-supervised learning," 2022.

[8] Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu, "Point-bert: Pre-training 3d point cloud transformers with masked point modeling," 2022.

[9] Renrui Zhang, Ziyu Guo, Rongyao Fang, Bin Zhao, Dong Wang, Yu Qiao, Hongsheng Li, and Peng Gao, "Point-m2ae: Multi-scale masked autoencoders for hierarchical point cloud pre-training," 2022.

[10] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian, "Foldingnet: Point cloud auto-encoder via deep grid deformation," 2018.

[11] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas, "Learning representations and generative models for 3d point clouds," 2018.

[12] Zhizhong Han, Xiyang Wang, Yu-Shen Liu, and Matthias Zwicker, "Multi-angle point cloud-vae: Unsupervised feature learning for 3d point clouds from multiple angles by joint self-reconstruction and half-to-half prediction," 2019.

[13] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling," 2017.

[14] Salman H. Khan, Yulan Guo, Munawar Hayat, and Nick Barnes, "Unsupervised primitive discovery for improved 3d generative modeling," 2019.

[15] Yanlong Li, Chamara Madarasingha, and Kanchana Thilakarathna, "Diffpmae: Diffusion masked autoencoders for point cloud reconstruction," 2024.

[16] Miguel Ángel de Miguel, Francisco Miguel Moreno, Pablo Marín-Plaza, Abdulla Al-Kaff, Martín Palos, David Martín, Rodrigo Encinar-Martín, and Fernando García, "A Research Platform for Autonomous Vehicles Technologies Research in the Insurance Sector," *Applied Sciences*, vol. 10, no. 16, 2020.

[17] Martin A. Fischler and Robert C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[18] Emmanuel Hartman, Nicolas Charon, and Martin Bauer, "Self supervised networks for learning latent space representations of human body scans and motions," *arXiv preprint arXiv:2411.03475*, 2024.

[19] Yu Meng, Yunyi Zhang, Jiaxin Huang, Yu Zhang, and Jiawei Han, "Topic discovery via latent space clustering of pretrained language model representations," in *Proceedings of the 31st ACM International Conference on World Wide Web (WWW 2022)*. 2022, pp. 3143–3152, ACM.

[20] Yushi Lan, Shangchen Zhou, Zhaoyang Lyu, Fangzhou Hong, Shuai Yang, Bo Dai, Xingang Pan, and Chen Change Loy, "Gaussiananything: Interactive point cloud latent diffusion for 3d generation," *arXiv preprint arXiv:2411.08033*, 2024.

[21] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao, "3d shapenets: A deep representation for volumetric shapes," in *CVPR*, 2015.

[22] Diederik P. Kingma and Jimmy L. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations (ICLR)*, 2015, Poster session; also available as arXiv preprint arXiv:1412.6980.

[23] Haoqiang Fan, Hao Su, and Leonidas J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 605–613.

[24] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu, "Shapenet: An information-rich 3d model repository," *CoRR*, vol. abs/1512.03012, 2015.

[25] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[26] Zhizhong Han, Mingyang Shang, Yu-Shen Liu, and Matthias Zwicker, "Inter-prediction gan: Unsupervised representation learning for 3d shapes by learning global shape memories to support local view predictions," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2020, Accedido: 2025-09-04.

[27] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan, "Pointflow: 3d point cloud generation with continuous normalizing flows," *arXiv preprint arXiv:1906.12320*, 2019, Published in ICCV 2019.

[28] Haotian Liu, Mu Cai, and Yong Jae Lee, "Masked discrimination for self-supervised learning on point clouds," 2022.