Latent-space Laplacian Pyramids for Adversarial Representation Learning with 3D Point Clouds

Vage Egiazarian[1], Savva Ignatyev[1], Alexey Artemov[1],
Oleg Voynov[1], Andrey Kravchenko[2], Youyi Zheng[3], Luiz Velho[4] and Evgeny Burnaev[1]

[1]*Skolkovo Institute of Science and Technology, Moscow, Russia*

[2]*DeepReason.ai, Oxford, U.K.*

[3]*State Key Lab, Zhejiang University, China*

[4]*IMPA, Brazil*

Keywords: Deep Learning, 3D Point Clouds, Generative Adversarial Networks, Multi-scale 3D Modelling.

Abstract: Constructing high-quality generative models for 3D shapes is a fundamental task in computer vision with diverse applications in geometry processing, engineering, and design. Despite the recent progress in deep generative modelling, synthesis of finely detailed 3D surfaces, such as high-resolution point clouds, from scratch has not been achieved with existing learning-based approaches. In this work, we propose to employ the latent-space Laplacian pyramid representation within a hierarchical generative model for 3D point clouds. We combine the latent-space GAN and Laplacian GAN architectures proposed in the recent years to form a multi-scale model capable of generating 3D point clouds at increasing levels of detail. Our initial evaluation demonstrates that our model outperforms the existing generative models for 3D point clouds, emphasizing the need for an in-depth comparative study on the topic of multi-stage generative learning with point clouds.

1 INTRODUCTION

A point cloud is an ubiquitous data structure that has gained a strong presence in the last few decades with the widespread use of range sensors. Point clouds are sets of points in 3D space, commonly produced by range measurements with 3D scanners (*e.g.* LI-DARs, RGB-D cameras, and structured light scanners) or computed using stereo-matching algorithms. A key use-case with point clouds is 3D surface reconstruction involved in many applications such as reverse engineering, cultural heritage conservation, or digital urban planning.

Unfortunately, for most scanners the raw 3D measurements often cannot be used in their original form for surface/shape reconstruction, as they are generally prone to noise and outliers, non-uniform, and incomplete. Whilst constant progress in scanning technology has led to improvements in some aspects of data quality, others, such as occlusion, remain a persistent issue for objects with complex geometry. Thus, a crucial step in 3D geometry processing is to model full 3D shapes from their sampling as 3D point clouds, inferring their geometric characteristics from incom-
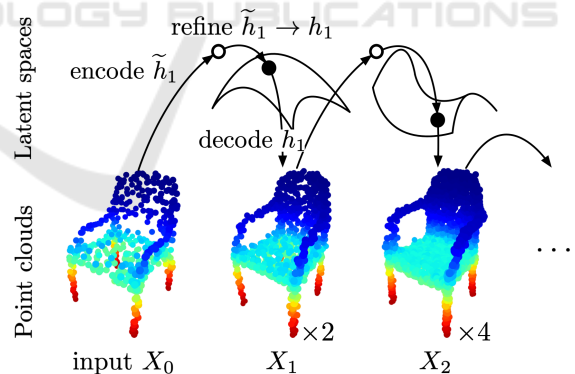

Figure 1: The proposed latent-space Laplacian GAN operates on latent representations, bypassing the need to process large-scale 3D point clouds. Through using mutiple stages of the pyramid, the sample resolution can be iteratively increased.

plete and noisy measurements. A recent trend in this direction is to apply data-driven methods such as deep generative models (Achlioptas et al., 2018; Li et al., 2018a; Chen et al., 2019).

However, most known deep models operate directly in the original space of raw measurements, which represents a challenging task due to the om-

nipresent redundancy in raw data; instead, it might be beneficial to encode characteristic shape features in the latent space and further operate with latent representations. Another shortcoming is that most models only operate with coarse (low-resolution) 3D geometry, as high-resolution 3D shapes are computationally demanding to process and challenging to learn from.

In this work, we consider the task of learning 3D shape representations given their 3D point cloud samples. To this end, we develop a novel deep cascade model, Latent-Space Laplacian Pyramid GAN, or LSLP-GAN, taking advantage of the recent progress in adversarial learning with point clouds (Achlioptas et al., 2018) and deep multi-scale models (Denton et al., 2015; Mandikal and Radhakrishnan, 2019). Our proposed model (schematically represented in Figure 1) has several important distinctions: (i) it is generative and can be used to produce synthetic shapes unseen during training; (ii) it is able to produce high-resolution point clouds via a latent space Laplacian pyramid representation; (iii) it is easy to train as it operates in the space of latent codes, bypassing the need to perform implicit dimensionality reduction during training. We train our model in an adversarial regime using a collection of 3D point clouds with multiple resolutions.

In this work, our contributions are as follows:

1. We propose LSLP-GAN, a novel multi-scale deep generative model for shape representation learning with 3D point clouds.

2. We demonstrate by the means of numerical experiments the effectiveness of our proposed method for shape synthesis and upsampling tasks.

2 RELATED WORK

Neural Networks on Unstructured 3D Point Clouds. Despite deep convolutional neural networks (CNNs) have proved themselves to be very effective for learning with 2D grid-structured images, until very recently, the same did not hold true with unstructured 3D point clouds. The basic building blocks for point-based architectures (equivalents of the spatial convolution) are not straightforward to implement. To this end, MLP-based (Qi et al., 2017a; Qi et al., 2017b), graph convolutional (Wang et al., 2018), and 3D convolutional point networks (Hua et al., 2018; Li et al., 2018c; Atzmon et al., 2018) have been proposed, each implementing their own notion of convolution, and applied to classification, semantic labelling, and other tasks (see (Uy et al., 2019) for a survey). We adopt the building blocks of (Qi et al.,

2017a; Achlioptas et al., 2018) as a basis for the architecture of our autoencoders. Building on top of the success of point convolutions, auto-encoding networks have been proposed (Achlioptas et al., 2018; Yang et al., 2018; Li et al., 2018b) to learn efficient latent representations.

Generative Neural Networks For 3D Modelling. Literature on generative learning with 3D shapes shows instances of variational auto-encoders (VAEs) (Kingma and Welling, 2013) and generative adversarial networks (GANs) (Goodfellow et al., 2014) applied to 3D shape generation. On the one hand, VAEs have been demonstrated to efficiently model images (Kingma and Welling, 2013), voxels (Brock et al., 2016), and properties of partially-segmented point clouds (Nash and Williams, 2017), learning semantically meaningful data representations. On the other hand, GANs have been studied in the context of point set generation from images (Fan et al., 2017), multi-view reconstruction (Lin et al., 2018), volumetric model synthesis (Wu et al., 2016), and, more recently, point cloud processing (Achlioptas et al., 2018; Li et al., 2018a). However, neither of the mentioned approaches provides accurate multi-scale modelling of 3D representation with high resolution, which has been demonstrated to drastically improve image synthesis with GANs (Denton et al., 2015). Additionally, in several instances, for point cloud generation, some form of input (*e.g.*, an image) is required (Fan et al., 2017). In contrast, we are able to generate highly detailed point clouds unconditionally.

Multi-scale Neural Networks on 3D Shapes. For 2D images, realistic synthesis with GANs has first been proposed in (Denton et al., 2015) with a coarse-to-fine Laplacian pyramid image representation. Surprisingly, little work of similar nature has been done in multi-scale point cloud modelling. (Mandikal and Radhakrishnan, 2019) propose a deep pyramid neural network for point cloud upsampling. However, their model accepts RGB images as input and is unable to synthesise novel 3D shapes. We, however, operate directly on unstructured 3D point sets and provide a full generative model for 3D shapes. (Yifan et al., 2019) operate on patches in a progressive manner, thus implicitly representing a multi-scale approach. However, their model specifically tackles the point set upsampling task and cannot produce novel point clouds. In contrast, our generative model purposefully learns mutually coherent representations of full 3D shapes at multiple scales through the use of a latent-space Laplacian pyramid.
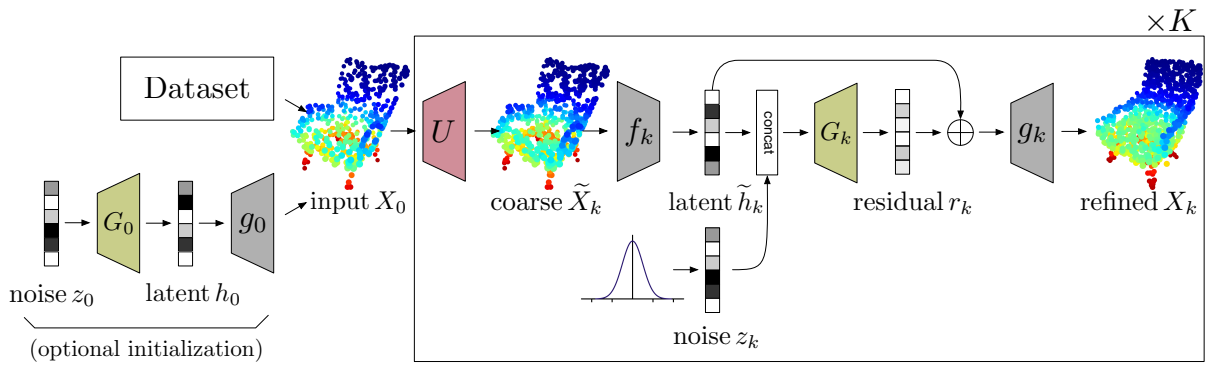
Figure 2: Full architecture of LSLP-GAN model. The network either accepts or generates an initial point cloud X_0 and processes it with a series of K learnable steps. Each step (1) upsamples its input using a non-learnable operator U, (2) encodes the upsampled version into the latent space by f_k, (3) performs correction of the latent code via a conditional GAN G_k, and (4) decodes the corrected latent code using g_k.

3 FRAMEWORK

Our model for learning with 3D point clouds is inspired by works on Latent GANs (Achlioptas et al., 2018) and Laplacian GANs (Denton et al., 2015). Therefore, we first briefly describe these approaches.

3.1 Latent GANs and Laplacian GANs

The well-known autoencoding neural networks, or autoencoders, compute *latent codes* $h \in \mathbb{R}^d$ for an input object $X \in \mathbb{X}$ through the means of an encoding network $f(X)$. This latent representation can later be decoded in order to reconstruct the original input, via a decoding network $g(h)$. *Latent GANs* (Achlioptas et al., 2018) are built around the idea that real latent codes commonly occupy only a subspace of their enclosing space $\mathbb{R}^d$, *i.e.* live on a manifold embedded in $\mathbb{R}^d$. Thus, it should be possible to synthesise samples by learning the manifold of real latent codes, presumably an easier task compared to learning with the original high-dimensional representations. A GAN model is thus used in addition to an autoencoder model, yielding artificial latent codes obtained as a result of an adversarial game. Such models were recently applied to learning with point clouds, where they compare favourably with GANs that operate on the raw input (Achlioptas et al., 2018).

Laplacian GANs (Denton et al., 2015) increase image resolution in a coarse-to-fine manner during synthesis, aiming to produce high quality samples of natural images. This is motivated by the fact that standard GANs give acceptable results for generation of low-resolution images but fail for images of higher resolution. Laplacian GANs overcome this problem by a cascading image synthesis with a series of gen-

erative networks $G_0, \ldots, G_n$, where each network G_k learns to generate a high-frequency residual image $r_k = G_k(U(I_{k+1}), z_k)$ conditioned on the upsampled image I_k provided by G_{k-1}. Thus, an image at stage k is represented via[1]:

$$I_k = U(I_{k+1}) + G_k(U(I_{k+1}), z_k), \tag{1}$$

where $U(\cdot)$ is an upsampling operator and z_k is a noise vector. For point clouds, a change in image resolution transforms to resampling, where subsets of points may be selected to form a low-resolution 3D shape, or reconstructed for a higher resolution 3D shape.

3.2 Representation Learning with Latent-space Laplacian Pyramids

Spaces of 3D Point Clouds. We start with a series of 3D shape spaces $\mathbb{R}^{n_0 \times 3}, \mathbb{R}^{n_1 \times 3}, \ldots, \mathbb{R}^{n_K \times 3}$ with $n_0 < n_1 < \ldots < n_K$ (specifically, we set $n_k = 2^k \cdot n_0$). A 3D shape can be represented in the space $\mathbb{R}^{n_k \times 3}$ with a 3D point sampling of its surface $X_k = \{x_i\}_{i=1}^{n_k}$. If this sampling maintains sufficient uniformity for all k, then the sequence of 3D point clouds $X_0, X_1, \ldots$ represents a progressively finer model of the surface. Our intuition in considering the series of shape spaces is that modelling highly detailed 3D point clouds represents a challenge due to their high dimensionality. Thus, it might be beneficial to start with a low-detail (but easily constructed) model X_0 and decompose the modelling task into a sequence of more manageable stages, each aimed at a gradual increase of detail.

---

[1]Note that here, larger values of index k correspond to coarser pyramid levels (low resolution residuals), while in our relation (2) below k indexes shapes with increasing resolution.

Training Auto-encoding Point Networks on Multiple Scales. Learning the manifold of latent codes has been demonstrated to be beneficial in terms of reconstruction quality (Achlioptas et al., 2018). Motivated by this observation, we use 3D shape spaces $\{\mathbb{R}^{n_k \times 3}\}_{k=1}^{K}$ and construct a series of corresponding latent spaces $\{\mathbb{R}^{d_k}\}_{k=1}^{K}$ by training point autoencoders $\{(f_k, g_k)\}_{k=1}^{K}$. Note that an autoencoder (f_k, g_k) is trained using the resolution n_k of 3D point clouds, which grows as k increases. As our method strongly relies on the quality of autoencoders, we evaluate reliability of their mappings from 3D space into latent space in Section 4. After training the autoencoders, we fix their parameters and extract latent codes for shapes in each of the 3D shape spaces.

Laplacian Pyramid in The Spaces of Latent Codes. For what follows, it is convenient to assume that we are given as input a point cloud $X_{k-1} \in \mathbb{R}^{n_{k-1} \times 3}$. We aim to go from X_{k-1} to X_k, *i.e.* to increase resolution from n_{k-1} to $n_k = 2n_{k-1}$, generating additional point samples on the surface of an underlying shape. Figure 3 illustrates our reasoning schematically.
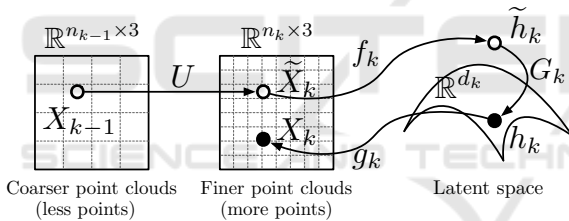

Figure 3: A detailed operation scheme of our latent-space Laplacian pyramid (see the accompanying text).

We start by processing the input point cloud by a simple upsampling operator $U(\cdot)$, obtaining a coarse point cloud $\widetilde{X}_k = U(X_{k-1})$: for each point $x \in X_{k-1}$ we create a new instance $\widetilde{x} = \frac{1}{m} \sum_{i \in \mathrm{NN}(x)} x_i$ where $\mathrm{NN}(x)$ is a set of m nearest Euclidean neighbours of x in X_{k-1} (we use $m = 7$ neighbours, including x), and add it to the point cloud. This procedure represents a simple linear interpolation and forms exactly n_k points located in the vicinity of the real surface. However, the computed point cloud $\widetilde{X}_k$ generally contains perturbed points, and we view it only as a rough approximation to our desired X_k.

We map the coarse point cloud $\widetilde{X}_k$ by f_k into a latent code $\widetilde{h}_k = f_k(\widetilde{X}_k)$, that we assume is offset by a small delta from the manifold of latent representations due to an interpolation error in $\widetilde{X}_k$. To compensate for this offset in the latent space, we compute an additive correction r_k to $\widetilde{h}_k$ using a generator network G_k, resulting in a corrected code $h_k = \widetilde{h}_k + r_k =$ $\widetilde{h}_k + G_k(\widetilde{h}_k, z_k)$. Decoding h_k by g_k, we obtain a refined point cloud $X_k = g_k(h_k)$ with resolution n_k.

Putting together the full procedure in the space of latent representations leads to a series of relations

$$h_k = f_k(U(X_{k-1})) + G_k(f_k(U(X_{k-1})), z_k), \quad (2)$$

which is a latent-space equivalent of (1). Hence, we call the resulting series $\{h_k\}_{k=0}^{K}$ of hierarchical representations a *latent-space Laplacian pyramid* (LSLP).

Training Latent GANs on Multiple Scales. To perform meaningful corrections of the rough latent code $\widetilde{h}_k$, each generator G_k faces a challenge of learning the subtle differences between the latent codes of true high-resolution 3D point clouds and those of coarsely upsampled ones. Thus, we train a series of latent GANs $\{(G_k, D_k)\}_{k=1}^{K}$ by forcing the generator G_k to synthesise residuals r_k in the latent space conditioned on the input $\widetilde{h}_k$, and the discriminator D_k to distinguish between the real latent codes $h_k \in \mathbb{R}^{d_k}$ and the synthetic ones $\widetilde{h}_k + G_k(\widetilde{h}_k, z_k)$. Note that as each (but the first) latent GAN accepts a rough latent code $\widetilde{h}_k$, they may be viewed as conditional GANs (CGANs) (Mirza and Osindero, 2014).

Two Execution Modes: Synthesis and Upsampling. In the text above, we assumed an initialiser $X_0 \in \mathbb{R}^{n_0 \times 3}$ to be given as an input, which is the case in particular applications, such as upsampling or shape completion. However, our framework can as easily function in a purely generative mode, sampling unseen high-resolution point clouds on the fly. To enable this, we start with an (unconditional) latent GAN G_0 and produce a point cloud $X_0 = g_0(G_0(z_0))$, which serves as an input to the remaining procedure.

An overview of our architecture is presented in Figure 2.

Architectural and Training Details of our Framework. The architecture of all our networks is based on the one proposed in (Achlioptas et al., 2018), where the autoencoders follow the PointNet (Qi et al., 2017a) encoders design and have fully-connected decoders, and GANs are implemented by the MLPs.

When training the autoencoders, we optimise the Earth Mover's Distance (EMD) given by:

$$d_{\mathrm{EMD}}(X, Y) = \min_{\phi: X \to Y} \sum_{x \in X} ||x - \phi(x)||_2$$

where ϕ is a bijection, obtained as a solution to the optimal transportation problem involving the two point sets $X, Y \in \mathbb{R}^{n_k \times 3}$. Training the GANs is performed by optimising the commonly used objectives (Goodfellow et al., 2014; Mirza and Osindero, 2014).

4 EVALUATION AND APPLICATIONS

4.1 Setup of our Evaluation

Training Datasets. For all our experiments, we use the meshes from the ShapeNet dataset (Chang et al., 2015). We have perfomed experiments using separate *airplane, table, chair, car,* and *sofa* classes in the Shapenet dataset, as well as using a *multi-class* setup. We train three stages of autoencoders and generative models on resolutions of $n_0 = 512$, $n_1 = 1024$, and $n_2 = 2048$ points, respectively, using a training set of 3046, 7509, 5778, 6497, and 4348 3D shapes for the classes *airplane, table, chair, car,* and *sofa,* respectively, and 9000 shapes for our *multi-class* setup. We have used Adam (Kingma and Ba, 2014) optimisers to train both autoencoders and GANs. All autoencoders have been trained for 500 epochs with the initial learning rates of 5×10^{-4}, $\beta_1 = 0.9$ and a batch size of 50; GANs have been trained for 200 epochs with the initial learning rates of 10^{-4}, $\beta_1 = 0.9$ and a batch size of 50.

Metrics. Along with the Earth Mover's Distance (EMD), we assess the point cloud reconstruction performance using the Chamfer Distance (CD) as a second commonly adopted measure, given by

$$d_{CD}(X,Y) = \sum_{x \in X} \min_{y \in Y} ||x-y||_2^2 + \sum_{y \in Y} \min_{x \in X} ||x-y||_2^2.$$

To evaluate the generative models, we employ the Jensen-Shannon Divergence (JSD), coverage (COV), and Minimum Matching Distance (MMD) measures, following the scheme proposed in (Achlioptas et al., 2018). JSD is defined over two empirical measures P and Q as $\text{JSD}(P \parallel Q) = 1/2\text{KL}(P \parallel M) + 1/2\text{KL}(Q \parallel M)$ where $\text{KL}(P \parallel Q)$ is the Kullback-Leibler divergence between the measures P and Q, $M = 1/2(P+Q)$, and measures P and Q count the number of points lying within each voxel in a voxel grid across all point clouds in the two sets A and B, respectively. COV measures a fraction of point clouds in B approximately represented in A; to compute it, we find the nearest neighbours in B for each $x \in A$. MMD reflects the fidelity of the set A with respect to the set B, matching every point cloud of B to the one in A with the minimum distance and computing the average of distances in the matching.

4.2 Experimental Results

Evaluating Autoencoders. We first evaluate our point autoencoders to validate their ability to compute efficient latent representations with increasing resolutions of input 3D point clouds. To compute the reconstructions, we encode into the latent space and decode back the 3D shapes from the test split of the respective class unseen during training. In Table 1 we display the reconstruction quality of our auto-encoders for the three levels of resolution, using CD and EMD measures. As the sampling density increases, both measures improve as expected.
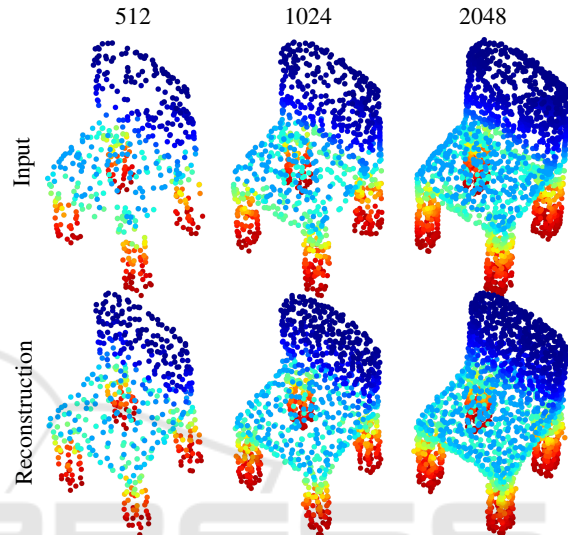

Figure 4: Inputs and reconstructions using our autoencoders at resolutions $n_i \in \{512, 1024, 2048\}$ of the 3D point cloud.

Table 1: Reconstruction quality using our autoencoders at resolutions $n_i \in \{512, 1024, 2048\}$ of the 3D point cloud.

Shape class	CD $\times 10^{-3}$			EMD $\times 10^{-3}$		
	512	1024	2048	512	1024	2048
chair	0.16	0.10	0.07	60.2	53.5	48.3
airplane	0.57	0.38	0.29	39.4	34.5	30.8
table	1.41	0.96	0.67	56.9	50.1	45.6

Figure 4 demonstrates the ground-truth and decoded 3D point clouds, respectively, at all stages in our autoencoders. We conclude that our models can represent the 3D point clouds at multiple resolutions.

Evaluating Generative Models. We further evaluate our generative models using the MMD-CD, COV, and JSD measures, in both single-class and multi-class setups. To this end, we train our LSLP-GAN using the latent spaces obtained with the previously trained autoencoders. Table 2 compares our LSLP-GAN and the L-GAN model (Achlioptas et al., 2018). We consistently outperform the baseline L-GAN across all object classes according to the quality metrics defined above.
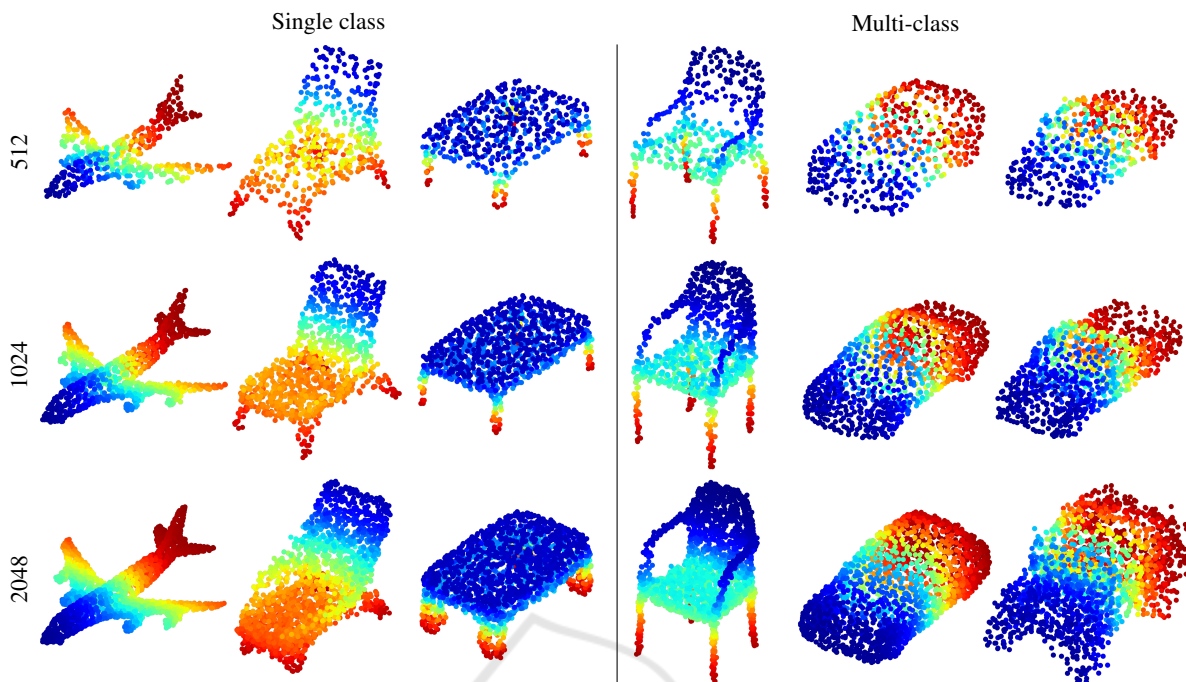
Single class   Multi-class

Figure 5: Examples of shapes synthesised using our LSLP-GAN model. *Left:* airplanes, chairs, and tables synthesised using our single-class models. *Right:* samples of 3D shapes synthesised using our multi-class model, note that the overall geometry of the shape changes slightly due to averaging over many classes. The rightmost figure displays a failure mode for our model.

Table 2: Performance evaluation of our proposed LSLP-GAN model as compared to the baseline L-GAN model (Achlioptas et al., 2018).

Shape class	MMD-CD $\times 10^{-3}$		COV-CD, %		JSD $\times 10^{-3}$	
	L-GAN	Ours	L-GAN	Ours	L-GAN	Ours
car	0.81	**0.71**	23.5	**32.1**	28.9	**24.2**
chair	1.79	**1.71**	44.9	**47.8**	13.0	**10.1**
sofa	1.26	**1.23**	43.9	**46.3**	9.6	**9.3**
table	1.93	**1.77**	39.7	**47.8**	19.9	**10.1**
airplane	0.53	**0.51**	41.7	**44.0**	17.1	**13.8**
multiclass	1.66	**1.55**	41.4	**45.7**	14.3	**9.8**

To demonstrate examples of novel 3D shape syntheses using our framework, we sample z_0 and process it with our framework, obtaining 3D point clouds X_0, X_1, X_2, which we display in Figure 5. Our framework can synthesise increasingly detailed 3D shapes, gradually adding resolution using a series of generative models.

Point Cloud Upsampling. Generative models such as L-GAN and our proposed LSLP-GAN are a natural fit for the task of 3D point set upsampling, as they learn to generate novel points given the lower resolution inputs. Thus, we model an upsampling task using the low-resolution 3D shapes from the Shapenet dataset. We supply LSLP-GAN with a low-resolution point cloud from the test split of the multi-class dataset and increase its resolution four-fold from
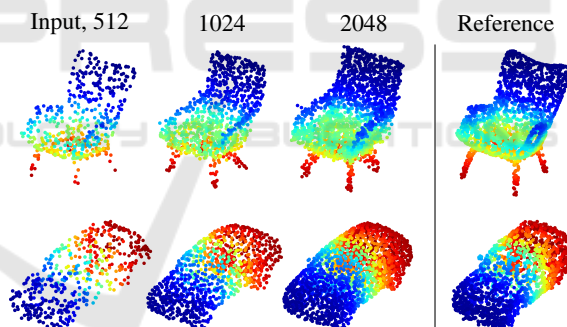
Input, 512   1024   2048   Reference

Figure 6: 3D point clouds upsampling results using our model, initialised with the input shape.

$n_0 = 512$ to $n_2 = 2048$ points, performing conditional generation using G_1 and G_2. Figure 6 displays 3D shapes upsampled using our multi-class model. Note that model has not been trained to perform upsampling directly, *i.e.* to preserve the overall shape geometry when producing novel points, hence the subtle changes in 3D shapes as the upsampling progresses.

5 CONCLUSION AND FUTURE WORK

We have presented LSLP-GAN, a novel deep adversarial representation learning framework for 3D point

clouds. The initial experimental evaluation reveals the promising properties of our proposed model. However, further investigation into the multi-scale generative learning methods is needed, including adoption of more recent deep architectures (Zhang et al., 2019; Uy et al., 2019). Other promising directions include exploring the limits of the Laplacian pyramid representation and a more extensive experimental evaluation of our approach. To this end, we plan to (i) further extend our work, considering deeper pyramid levels and larger upsampling factors (*e.g.* ×64), and (ii) conduct a comparative investigation of our framwork using more challenging tasks such as shape completion, using deep learning methods, *e.g.*,. (Yu et al., 2018b; Yifan et al., 2019; Yu et al., 2018a; Li et al., 2019; Mandikal and Radhakrishnan, 2019; Chen et al., 2019).

ACKNOWLEDGEMENT

REFERENCES

Achlioptas, P., Diamanti, O., Mitliagkas, I., and Guibas, L. (2018). Learning representations and generative models for 3d point clouds. In *ICML*, pages 40–49.

Atzmon, M., Maron, H., and Lipman, Y. (2018). Point convolutional neural networks by extension operators. *ACM ToG*, 37(4):71.

Brock, A., Lim, T., Ritchie, J. M., and Weston, N. (2016). Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*.

Chang, A. X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al. (2015). Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*.

Chen, X., Chen, B., and Mitra, N. J. (2019). Unpaired point cloud completion on real scans using adversarial training. *arXiv preprint arXiv:1904.00069*.

Denton, E. L., Chintala, S., Fergus, R., et al. (2015). Deep generative image models using a laplacian pyramid of adversarial networks. In *NIPS*, pages 1486–1494.

Fan, H., Su, H., and Guibas, L. J. (2017). A point set generation network for 3d object reconstruction from a single image. In *CVPR*, pages 605–613.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. In *NIPS*, pages 2672–2680.

Hua, B.-S., Tran, M.-K., and Yeung, S.-K. (2018). Pointwise convolutional neural networks. In *CVPR*, pages 984–993.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Li, C.-L., Zaheer, M., Zhang, Y., Poczos, B., and Salakhutdinov, R. (2018a). Point cloud gan. *arXiv preprint arXiv:1810.05795*.

Li, J., Chen, B. M., and Hee Lee, G. (2018b). So-net: Self-organizing network for point cloud analysis. In *CVPR*, pages 9397–9406.

Li, R., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. (2019). Pu-gan: a point cloud upsampling adversarial network. In *CVPR*, pages 7203–7212.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., and Chen, B. (2018c). Pointcnn: Convolution on x-transformed points. In *NIPS*, pages 820–830.

Lin, C.-H., Kong, C., and Lucey, S. (2018). Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI*.

Mandikal, P. and Radhakrishnan, V. B. (2019). Dense 3d point cloud reconstruction using a deep pyramid network. In *WACV*, pages 1052–1060. IEEE.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Nash, C. and Williams, C. K. (2017). The shape variational autoencoder: A deep generative model of part-segmented 3d objects. In *Computer Graphics Forum*, volume 36, pages 1–12. Wiley Online Library.

Qi, C. R., Su, H., Mo, K., and Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, pages 652–660.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. (2017b). Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NIPS*, pages 5099–5108.

Uy, M. A., Pham, Q.-H., Hua, B.-S., Nguyen, T., and Yeung, S.-K. (2019). Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *CVPR*, pages 1588–1597.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. (2018). Dynamic graph cnn for learning on point clouds. *CoRR*, abs/1801.07829.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NIPS*, pages 82–90.

Yang, Y., Feng, C., Shen, Y., and Tian, D. (2018). Foldingnet: Point cloud auto-encoder via deep grid deformation. In *CVPR*, pages 206–215.

Yifan, W., Wu, S., Huang, H., Cohen-Or, D., and Sorkine-Hornung, O. (2019). Patch-based progressive 3d point set upsampling. In *CVPR*, pages 5958–5967.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. (2018a). Ec-net: an edge-aware point set consolidation network. In *ECCV*, pages 386–402.

Yu, L., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. (2018b). Pu-net: Point cloud upsampling network. In *CVPR*, pages 2790–2799.

Zhang, Z., Hua, B.-S., and Yeung, S.-K. (2019). Shell-net: Efficient point cloud convolutional neural networks using concentric shells statistics. In *CVPR*, pages 1607–1616.