Modeling 3D Surfaces with a Locally Conditioned Atlas

Przemysław Spurek[1,4][0000−0003−0097−5521],
Sebastian Winczowski[1][0009−0009−6414−3063], Maciej Zięba[2][0000−0003−4217−7712],
Tomasz Trzciński[3,4][0000−0002−1486−8906], Kacper Kania[3][0000−0003−4177−1349],
and Marcin Mazur[1][0000−0002−3440−8173]

[1] Jagiellonian University, Faculty of Mathematics and Computer Science,
Krakow, Poland
[2] Wroclaw University of Science and Technology, Wroclaw, Poland
[3] Warsaw University of Technology, Warsaw, Poland
[4] IDEAS NCBR
{przemyslaw.spurek,marcin.mazur}@uj.edu.pl

Abstract. Recently proposed methods for reconstructing 3D objects use a mesh with an atlas consisting of planar patches that approximate the object's surface. However, in real-world scenarios, the surfaces of reconstructed objects exhibit discontinuities that degrade the mesh's quality. Therefore, conducting additional research on methods to overcome discontinuities and improve mesh quality is always advantageous. This paper proposes to address the limitation by maintaining local consistency around patch vertices. We present LoCondA, a Locally Conditioned Atlas that represents a 3D object hierarchically as a generative model. The model initially maps the point cloud of an object onto a sphere and subsequently enforces the mapping to be locally consistent on both the sphere and the target object through the use of a spherical prior. Using this method, the mesh can be sampled on the sphere and then projected back onto the manifold of the object, yielding diverse topologies that can be seamlessly connected. The experimental results demonstrate that this approach produces structurally coherent reconstructions with meshes of comparable quality to those of competitors.

Keywords: Mesh · Point cloud · Atlas · Hypernetwork.

1 Introduction

Efficient 3D object representations are crucial for various computer vision and machine learning applications, from robotic manipulation [17] to autonomous driving [32]. Modern 3D registration devices, such as LIDARs and depth cameras, produce representations in the form of sparsely sampled, unordered sets of 3D points on objects' surfaces, which are known as *point clouds* [24,25]. While a single point cloud may offer surface details for object reconstruction [12], it does not provide enough information about the neighborhood structure of 3D points.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_7

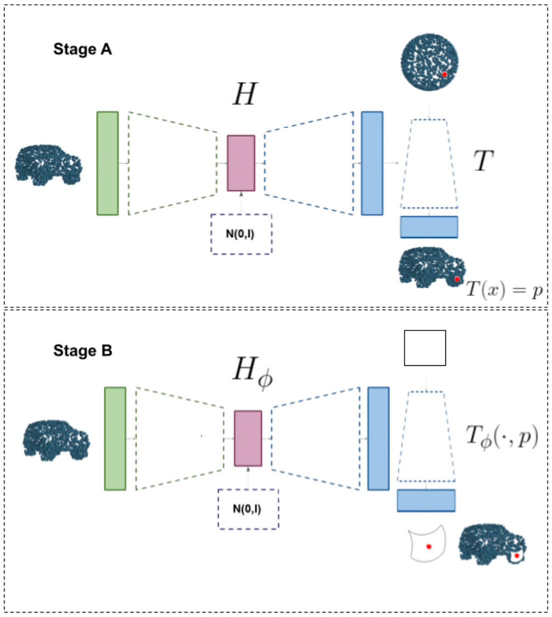

Fig. 1. Our approach extends the current hypermodels [27,28] (Stage A) by adding a mesh generation module (Stage B). First, we generate a seamless representation of the object by training a hypernetwork H that produces weights for a target network T that transforms a prior (usually uniform) distribution on the 2D unit sphere to the object surface S. This results in the ability to reach any point of S as a value of T. Next, we train another hypernetwork H_ϕ designed to produce weights for a target network T_ϕ conditioned on a given point $p = T(x) \in S$, which maps the 2D unit square to the patch on S, aligned with the neighborhood of p.

This lack of information makes it difficult to reconstruct a smooth, high-fidelity manifold for the entire object's surface.

Recently, researchers have proposed using polygonal meshes to model object surfaces for improved accuracy [14,27,36]. A mesh is a collection of vertices connected by edges to form triangles, which create the surface of an object. This representation is efficient and easily rendered, while also providing additional advantages such as the ability to sample the surface at a desired resolution and apply texturing in any 3D computer graphics software. To obtain such a representation, it is necessary to use advanced methodologies that utilize deep learning models based on an autoencoder architecture [27,28,31], or utilize an ensemble of parametric mappings from 2D rectangular patches to 3D primitives, commonly known as an *atlas* [14,36]. Previous methods were limited by the topology of the autoencoder's latent space distribution, which made them unable to model intricate structures with a non-spherical topology [27,28,31]. In contrast, atlas-based approaches offer a greater flexibility and can model virtually any surface. However, these methods often result in discontinuities in the reconstructed shapes and their deformation due to the inconsistency of individual mappings.

Although [6,11] propose adjustments that improve the quality of the outcome, their aim is to correct distortions resulting from the combination of different mappings. It is postulated that the continuous nature of neural architectures may be leveraged to prevent the formation of distortions by enforcing local consistency among patch vertices within a model's objective function. Therefore, we present a new method called *Locally Conditioned Atlas* (LoCondA) that can generate and reconstruct high-quality 3D meshes. This method enhances the existing base hypermodels [27,28] by introducing a new module for mesh generation that relies on local surface parametrization, as shown in Fig. 1. It is based on the concept of the *continuous atlas*, which provides a new approach that extends existing atlas methods and allows for adaptive sampling of any number of patches to cover any part of the reconstructed object. Specifically, we transform the point cloud embedding obtained from the base model to parameterize the bijective function represented by the multilayer perceptron (MLP) network[5]. This function aims to map a canonical 2D patch to the 3D patch on the surface of the target object. The positioning and shape of a 3D patch are conditioned using a single point from a point cloud generated by the base model. The process is repeated for each generated point while maintaining the local neighborhoods between the point cloud and the points located in the generated mesh. This procedure allows us to include the patch's stitching and reshaping in our framework's training objective, reducing the likelihood of shape discontinuities.

We summarize our contribution as follows:

- We propose a comprehensive framework for patch-based reconstruction methods that generate high-fidelity meshes from raw point clouds.
- We present the continuous atlas, a new paradigm that expands on existing atlas methods and enables adaptive sampling of any number of patches to cover any part of the reconstructed object.
- We introduce the Locally Conditioned Atlas (LoCondA), a new method for conditioning atlas-based approaches that effectively shares information between patches and resolves issues related to self-intersections and holes in reconstructed meshes.

2 Related work

The literature presents a wide range of 3D shape reconstruction models, such as dense pixel-wise depth maps [4,5], normal maps [30], point clouds [12,25,36], meshes [4,31], implicit functions [8,21], voxels [9,16], shape primitives [7,10], parametric mappings [6,14], or combinations thereof [22,23]. All of the aforementioned representations have advantages and disadvantages depending on their memory requirements and accuracy in fitting the surface.

[5] This is a neural network in which every node is connected to each node of the subsequent layer.

DOI: 10.1007/978-3-031-63751-3_7

Our paper focuses on one of the most popular representations, which is based on polygonal meshes. A mesh is a set of vertices connected by edges that allows for a piecewise planar approximation of a surface. The object's mesh can be then accessed by transforming it onto a unit sphere [27,28,31]. However, this method is limited to reconstructing objects that are topologically similar to spheres.

Patch-based approaches, proposed by [6,11], offer a greater flexibility and can model virtually any surface, even those with non-disk topology. To achieve this, parametric mappings are used to transform 2D patches into 3D shapes, also known as 2D manifolds. An example of this is FoldingNet [36], which utilizes a single patch to model an object's surface.

AtlasNet [14] is a method that simultaneously trains a number of functions, which together form an atlas, to obtain multiple patches that model a mesh. Each function transforms a unit square into a neighborhood of a point on the surface. The atlas elements are trained independently, resulting in maps that are not stitched together. This can cause discontinuities, such as holes or intersecting patches (see, e.g., Fig. 3 in [14]).

To address the aforementioned problem, many methods extend the Chamfer loss function of the basic AtlasNet by adding extra terms. In [6], the authors included terms to prevent patch collapse, reduce patch overlap, and analytically calculate the exact surface properties instead of approximating them. In [11], two new terms were introduced to improve the overall consistency of the local mappings. One of these terms uses surface normals to ensure local consistency in estimation within and across individual mappings. The other term minimizes stitching errors to enforce better spatial configuration of the mappings. Although these modifications improve the quality of the obtained results, their objective is to correct deformations after patch stitching. In this paper, we propose a different approach to solve this problem by reformulating the classical definition of an atlas to obtain maps that are correctly connected. Therefore, our method aims to prevent the issue from occurring in the first place.

Given a data set $\mathcal{X}$ containing point clouds $X_1, \ldots, X_n$, an autoencoder aims to transport the data through a latent space $\mathcal{Z} \subset \mathbb{R}^D$ while minimizing the reconstruction error. This is accomplished by identifying an encoder $\mathcal{E}\colon \mathcal{X} \to \mathcal{Z}$ and a decoder $\mathcal{D}\colon \mathcal{Z} \to \mathcal{X}$ that minimize the reconstruction error between each X_i and the reconstruction $\mathcal{D}(\mathcal{E}(X_i))$. In the generative framework, we also ensure that the data sent to the latent space is drawn from a predetermined prior distribution [18,19]. To effectively represent point clouds, it is essential to define an appropriate reconstruction loss. In the literature, two distance measures are commonly used: Earth Mover (Wasserstein) Distance [26] and Chamfer Distance [29]. Earth Mover Distance (EMD) is a metric between two distributions based on the minimum cost to transform one distribution into the other. Specifically, for two equally sized sets $X, Y \subset \mathbb{R}^3$, EMD is defined as $\mathrm{EMD}(X,Y) = \min_{\phi\colon X \to Y} \sum_{x \in X} c(x, \phi(x))$, where ϕ is a bijection and $c(x, \phi(x)) = \frac{1}{2}\|x - \phi(x)\|_2^2$. On the other hand, Chamfer Distance (CD) measures the squared distance between each point in one set and the nearest neighbor in the other set, i.e., $\mathrm{CD}(X,Y) = \sum_{x \in X} \min\limits_{y \in Y} \|x - y\|_2^2 + \sum_{x \in Y} \min_{y \in X} \|x - y\|_2^2$.

Point clouds may contain a variable number of data points that correspond to a single object and are registered at various angles. Therefore, methods that process them must be permutation and rotation invariant. The PointNet [24] framework enables the processing of 3D point clouds of varying sizes as input for neural networks. However, the output size remains a challenge. Hypernetworks [15] are a potential solution as they are neural models that generate weights for a separate target network capable of solving a specific task. Instead of producing a fixed-size 3D point cloud, a hypernetwork creates a target network to parametrize the object's surface and generate any desired number of points. The parameters of the target network are not directly optimized; only the weights of the hypernetwork are optimized during the training procedure.

3 Local parametrization of a surface

This section introduces the continuous atlas, a new approach for creating meshes from patches. It also discusses the limitations of current methods based on discrete atlas representations and presents our model as a solution to these limitations.

A set $S \subset \mathbb{R}^3$ is defined as a 2-manifold[6] (also called a surface) if for every point $p \in S$ there exists an open set U in $\mathbb{R}^2$ and an open set V in $\mathbb{R}^3$ containing p such that $S \cap V$ is homeomorphic to U. A corresponding homeomorphism is referred to as a *chart*. An atlas is a collection of charts that cover the 2-manifold.

By a *discrete atlas* of a 3D object surface S supported by a set of points $P = \{p_1, \dots, p_k\} \subset S$ we mean a collection of charts $\{\phi_1, \dots, \phi_k\}$, such that each ϕ_i maps the open square $U = (0,1) \times (0,1)$ onto a neighborhood $V(p_i)$ of a point p_i and the following condition is satisfied: $\bigcup_{i=1}^k \phi_i(U) = \bigcup_{i=1}^k V(p_i) = S$. In practice, the charts are represented by MLP networks $\phi_1(\cdot;\theta_1), \dots, \phi_k(\cdot;\theta_k)$ trained together to minimize the global reconstruction error $\mathcal{L}\left(\bigcup_{i=1}^k \phi_i(U;\theta_i), S\right)$, where $\mathcal{L}$ is either the Chamfer Distance or the Earth Mover Distance. Using such a formulation, the charts are trained to induce the *patches* $\phi_1(U;\theta_1), \dots, \phi_k(U;\theta_k)$, which together create a mesh that approximates the target object surface $\mathcal{S}$ as closely as possible. In theory, the method should produce a single seamless mesh. In practice, however, the result obtained is not ideal and requires additional post-processing to be used in real-world applications. This is because it works with a given number of charts (each generating a single patch) and does not take into account the stitching process itself, as no information is shared between patches. If a patch does not correctly cover the underlying neighborhood of a point, no other patch fixes that part, resulting in empty spaces on the object's surface. To mitigate the issues of the discrete atlas, we propose an approach that leverages the local structure of 3D objects [3,34,35,37].

By a *continuous atlas* of a 3D object surface S we define a continuous mapping $\phi\colon U \times S \to S$ which transforms the open square $U = (0,1) \times (0,1)$ and a point $p \in S$ onto a neighborhood $V(p) \subset S$ of the point p. Note that, unlike a discrete

[6] We adhere to the concept presented in [14] (see also [6,11]).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_7

atlas, instead of using a finite set of charts, we use only a transformation ϕ that locally models the surface of the object, leading to a potentially infinite number of charts $\phi(\cdot, p)$. In contrast to a traditional conditioning mechanism in the AtlasNet, p is not a global descriptor but a direct point of S. In consequence, we can produce an arbitrary number of patches $\phi(U, p)$, which may be located in any place on the object's surface.

The proposed approach overcomes the limitations of previous discrete methods. Specifically, we theoretically solve the problem of stitching partial meshes since every chart is informed about the local neighborhood. Moreover, using a continuous atlas approach we can easily fill the missing spaces in the final mesh by adding a new mapping for the region of interest. Since we can create an arbitrary number of patches, we can locate a point in the empty space neighborhood and create an additional patch using ϕ function conditioned on the selected point.

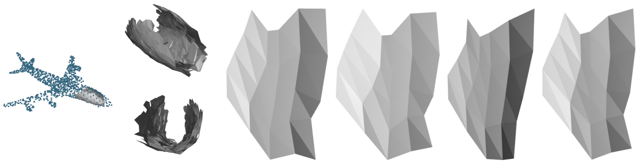

Fig. 2. Sample airplane patches created using our method, with structurally similar parts placed next to each other to create smooth surfaces.

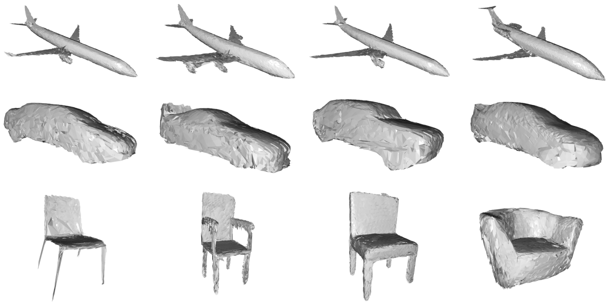

Fig. 3. Sample mesh representations generated by our method. Note that our model is capable of reconstructing complex shapes such as chair legs and backs.

We present in Fig. 2 that the resulting model can stitch patches on a sample airplane object. Note that the obtained surface is smooth due to the continuity of a continuous atlas. Additionally, Fig. 3 displays complete sample mesh represen-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_7

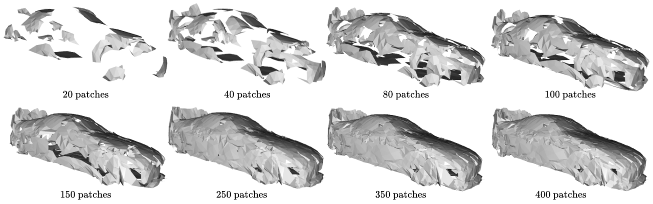

Fig. 4. Mesh representations with a different number of patches that were created by our method. Note that we need at least 200 patches to get high-quality reconstructions.

tations produced by our method. In contrast, Fig. 4 shows mesh representations generated using varying numbers of patches.

4 Local approximation of a surface

This section introduces the Locally Conditioned Atlas (LoCondA), a two-stage framework for generating and reconstructing meshes of object surfaces using the point cloud representation and the introduced notion of the continuous atlas (see Fig. 1). Both stages utilize a hypernetwork paradigm, which has an advantage over autoencoder architectures in processing inputs of varying sizes. In our approach, each point cloud is parametrized individually, unlike traditional approaches where conditioning parameters are shared among different point clouds. This can be beneficial for retrieval purposes.

Stage A: Generative autoencoder using hypernetworks As we operate directly on points located on the surfaces of 3D objects, we utilize two state-of-the-art generative autoencoder models based on hypernetworks, namely HyperCloud [27] and HyperFlow [28], instead of the traditional generative autoencoder architecture. These solutions use hypernetworks to generate weights of small neural networks that map a known prior distribution of points onto 3D objects. Thus, after training, we can implicitly sample any point $p \in S$ without accessing the actual target object surface S.

In our framework, we use a hypernetwork $H(\cdot; W_H)$ which, for a given point cloud $X \subset S$, returns the weights of the corresponding target network $T(\cdot; W_T)$. Thus X is represented by the function $T(\cdot; W_T) = T(\cdot; H(X; W_H))$. As T we use a classical MLP in the case of the HyperCloud implementation, and a Continuous Normalizing Flow (CNF) [13] in the case of the HyperFlow implementation. The hypernetwork H is pre-trained beforehand[7] using the following procedure: we take an input point cloud X and pass it to H to obtain the weights W_T of the target network T, which reconstructs X from a uniform noise $X_{S^2} \subset \mathbb{R}^3$ on the 2D unit sphere S^2, and then we minimize the reconstruction loss, expressed either by the Chamfer Distance or the Earth Mover Distance (as far as the

[7] So it does not contribute to the total training time.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_7

HyperCloud architecture is considered), or by the negative log-likelihood (as far as the HyperFlow architecture is considered).

Stage B: Locally Conditioned Atlas (LoCondA) Our LoCondA model implements the introduced approach of *continuous atlas*. Specifically, we represent the underlying mapping ϕ with a neural architecture consisting of the following parts: (1) a (small) target network $T_\phi(\cdot; W_{T_\phi})$ that transforms a sample X_U from the uniform distribution on the square $U = (0,1) \times (0,1)$ and a point p from the (reconstructed) point cloud $X \subset S$ into the object surface S, (2) a hypernetwork $H_\phi(\cdot; W_{H_\phi})$ that generates weights W_{T_ϕ} conditioned on a given point p to ensure that the image of T_ϕ forms its neighborhood $V(p) \subset X$. Note that due to a conditioning mechanism, we do not need to model each of the charts $\phi(\cdot, p)$ separately, which significantly reduces the number of parameters.

Training the LoCondA model can be done by minimizing the following loss function: $\mathcal{L}\left(T_\phi(X_U, p; H_\phi(X, p; W_{H_\phi})), V(p)\right)$, where $\mathcal{L}$ is either the Chamfer Distance or the Earth Mover Distance, $p = T(x; W_T)$ is a random point from the reconstructed point cloud X (see Stage A), and $V(p)$ is taken as the set of k elements closest to p in X (here k is treated as a hyperparameter). This leads to the point cloud (the image of T_ϕ) that reconstructs $V(p)$, which can be easily transformed into a mesh by connecting the points with edges, as in a natural mesh built on X_U. The problem however is that such a formulation causes many of the generated patches to have unnecessarily long edges, and the network folds them so that the patch fits the surface of an object. To mitigate this problem, we add an edge length regularization motivated by [31]. Specifically, we include a regularization term of the form: $l_{loc} = \sum_e ||e||_2^2$, where $\|e\|_2^2$ is the squared norm of the length of an edge e. So we get the total loss $\mathcal{L} + \lambda l_{loc}$, where λ is a hyperparameter of the model.

5 Experiments

This section presents the experimental results of the proposed method. Firstly, we evaluate the model's generative capabilities. Secondly, we provide the reconstruction results in comparison to reference approaches. Finally, we compare the quality of the generated meshes to baseline methods. The models were trained using the Chamfer Distance as $\mathcal{L}$ and λ was set to 0.0001. LoCondA-HC and LoCondA-HF are acronyms used to refer to different autoencoder architectures (see Stage A). LoCondA-HC uses HyperCloud, while LoCondA-HF uses HyperFlow. A grid search was employed to optimize the hyperparameters of all models. The source code is available at `https://github.com/gmum/LoCondA`.

It is important to note that our method has a significant advantage in its ability to generate an unlimited number of patches. To estimate the minimum number of patches required, divide the number of points in a point cloud representation of the entire object by the number of points in one patch used in training (i.e., hyperparameter k). From a practical standpoint, there is a trade-off between reconstruction quality and time performance. If we prioritize quality,

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63751-3_7

we should produce as many patches as possible. However, if we prioritize time, we should decrease the number of patches. In our experiments, we used 400 patches per object.

Generative capabilities In this experiment, the generative capabilities[8] of the LoCondA model are compared to existing reference approaches. The evaluation protocol provided in [33] is followed, and standard measures such as the Jensen-Shannon Divergence (JSD), the Coverage (COV), and the Minimum Matching Distance (MMD) are used. The last two measures are calculated separately for the Chamfer Distance (CD) and the Earth Mover Distance (EMD). The study compares the results of various existing solutions for point cloud generation, including l-GAN [1], PC-GAN [20], PointFlow [33], HyperCloud(P) [27] and HyperFlow(P) [28]. Additionally, two baselines, HyperCloud(M) and HyperFlow(M), capable of generating meshes from the unit sphere, are considered in the experiment. Each model is trained using point clouds from one out of the three categories in the ShapeNet dataset: *airplane*, *chair*, and *car*.

Table 1. (Generative capabilities) Metric scores for different generative models performed on the ShapeNet dataset according to the evaluation protocol described in [33]. To simplify the notation, MMD-CD (lower is better) scores are multiplied by 10^3, MMD-EMD (lower is better) and JSD (lower is better) scores are multiplied by 10^2, and COV (greater is better) scores are expressed on a percentage scale. HC refers to the use of the HyperCloud autoencoder in LoCondA (our), and HF refers to the use of the HyperFlow autoencoder. For the HyperCloud and HyperFlow models, we use both variants that generate point clouds (P) or meshes (M). The most successful outcomes are indicated in bold.

Method	Airplane					Chair					Car				
	JSD	MMD		COV		JSD	MMD		COV		JSD	MMD		COV	
		CD	EMD	CD	EMD		CD	EMD	CD	EMD		CD	EMD	CD	EMD
Point cloud generation															
l-GAN	**3.61**	0.26	3.29	**47.90**	50.62	2.27	2.61	7.85	40.79	41.69	2.21	1.48	5.43	39.20	39.77
PC-GAN	4.63	0.28	3.57	36.46	40.94	3.90	2.75	8.20	36.50	38.98	5.85	1.12	5.83	23.56	30.29
PointFlow	4.92	**0.21**	3.24	46.91	48.40	1.74	2.42	7.87	**46.83**	**46.98**	**0.87**	**0.91**	5.22	44.03	46.59
HyperCloud(P)	4.84	0.26	3.28	39.75	43.70	2.73	2.56	**7.84**	41.54	46.67	3.09	1.07	5.38	40.05	40.05
HyperFlow(P)	5.39	0.22	**3.16**	46.66	**51.60**	**1.50**	2.30	8.01	44.71	46.37	1.07	1.14	5.30	45.74	47.44
Mesh generation															
HyperCloud(M)	9.51	0.45	5.29	30.60	28.88	4.32	2.81	9.32	40.33	40.63	5.20	1.11	6.54	37.21	28.40
HyperFlow(M)	6.55	0.38	3.65	40.49	48.64	4.26	3.33	8.27	41.99	45.32	5.77	1.39	5.91	28.40	37.21
LoCondA-HC	16.1	0.66	4.71	30.37	32.59	4.45	3.03	8.55	42.45	38.22	1.91	1.13	5.50	**53.69**	**50.56**
LoCondA-HF	4.80	0.22	3.20	44.69	47.91	2.54	**2.23**	7.94	43.35	42.60	1.16	0.92	**5.21**	44.88	47.72

Tab. 1 presents the results. LoCondA-HF achieves results comparable to those of the reference methods dedicated to point cloud generation. The evalu-

[8] These refer to the ability of the model to produce fake samples that are indistinguishable from real data.

ated measures for HyperFlow(P) and LoCondA-HF (which uses HyperFlow(P) as a base model in Stage A) are on the same level. Incorporating an additional module dedicated to mesh generation (Stage B) does not have a negative impact on our model's generative capabilities. However, when we use HyperFlow to directly generate meshes according to the procedure described in [28] (see the results for HyperFlow(M)), the generative capabilities are significantly inferior for the evaluated metrics.

Reconstruction capabilities The goal of this experiment is to evaluate the ability of our model to encode different shapes, measured by comparing the original objects with their reconstructions. The autoencoding task was conducted on 3D point clouds from three categories in the ShapeNet dataset, namely *airplane*, *chair*, and *car*. We compare LoCondA with AtlasNet [14] (the current state-of-the-art), where the prior shape is either a sphere or a set of patches. Additionally, we compare it with l-GAN [2] and PointFlow [33]. The experimental setup of PointFlow was followed.

Table 2. (Reconstruction capabilities) Metric scores for shape reconstruction by different models obtained on the ShapeNet dataset according to the evaluation protocol described in [14]. To simplify the notation, the scores are multiplied by 10, CD (lower is better) is multiplied by 10^4, and EMD (lower is better) is multiplied by 10^2. In LoCondA (our), HC refers to the use of the HyperCloud autoencoder, while HF refers to the use of the HyperFlow autoencoder. The most successful outcomes are indicated in bold.

Dataset	Metric	l-GAN		AtlasNet		PointFlow	LoCondA-HC	LoCondA-HF	Oracle
		CD	EMD	Sphere	Patches				
Airplane	CD	1.020	1.196	1.002	**0.969**	1.208	1.135	1.513	0.837
	EMD	4.089	**2.577**	2.672	2.612	2.757	2.881	2.990	2.062
Chair	CD	9.279	11.21	**6.564**	6.693	10.120	10.382	12.519	3.201
	EMD	8.235	6.053	5.790	**5.509**	6.434	6.738	6.973	3.297
Car	CD	5.802	6.486	**5.392**	5.441	6.531	6.575	7.247	3.904
	EMD	5.790	4.780	4.587	**4.570**	5.138	5.126	5.275	3.251
ShapeNet	CD	7.120	8.850	5.301	**5.121**	7.551	7.781	-	3.031
	EMD	7.950	5.260	5.553	5.493	**5.176**	5.826	-	3.103

The results are presented in Tab. 2, including the performance in both the Chamfer Distance (CD) and the Earth Mover Distance (EMD). It is worth noting that these two metrics depend on the point clouds' scale. The 'oracle' column shows the upper bound, which is the error between two different point clouds with the same number of points sampled from the same ground truth meshes. The results show that LoCondA-HC achieves competitive results compared to reference solutions. All competitors were trained using an autoencoding framework. However, LoCondA-HC and LoCondA-HF were also able to preserve generative capabilities during the experiment.

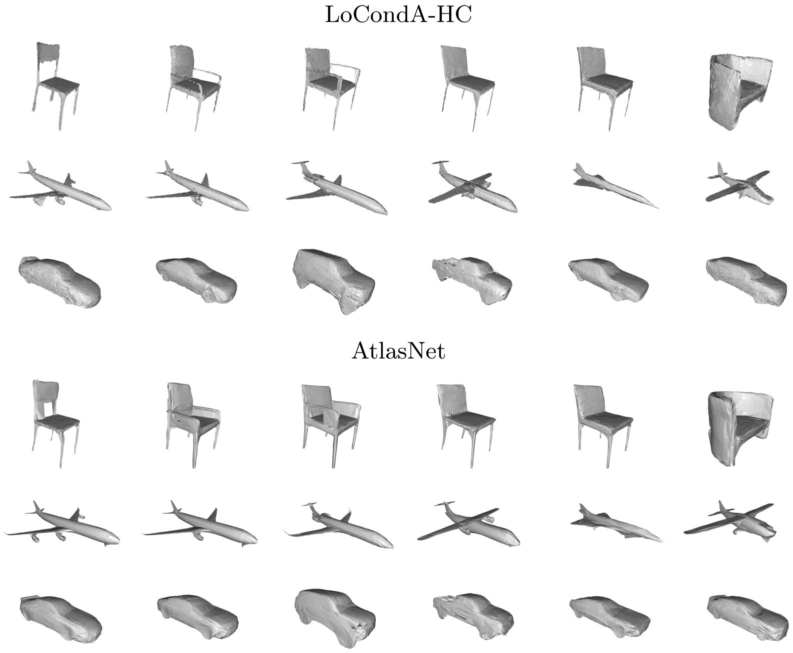

Fig. 5. AtlasNet and LoCondA-HC (our) comparison for reconstruction task. Note that both meshes have similar quality, but our method provides a more accurate description of objects that have holes, such as empty spaces in the back of chairs.

Fig. 5 presents a comparison between AtlasNet and LoCondA-HC on the reconstruction task. The quality of the meshes is similar, but our method better describes objects that contain holes, such as empty spaces in the back of chairs. On the other hand, reconstruction results obtained by decoding the linearly interpolated latent vectors of two objects from each class are shown in Fig. 6. Note that LoCondA-HC generates coherent and semantically plausible objects for all interpolation steps. However, during the middle stage of the interpolation, we encountered patches that were not properly stitched together.

Mesh quality evaluation Empirical evidence demonstrates that the proposed framework generates high-quality and seamless meshes, solving the initial problem of disjoint patches occurring in the AtlasNet model. To evaluate the continuity of the output surfaces, we recommend using the measure described below.

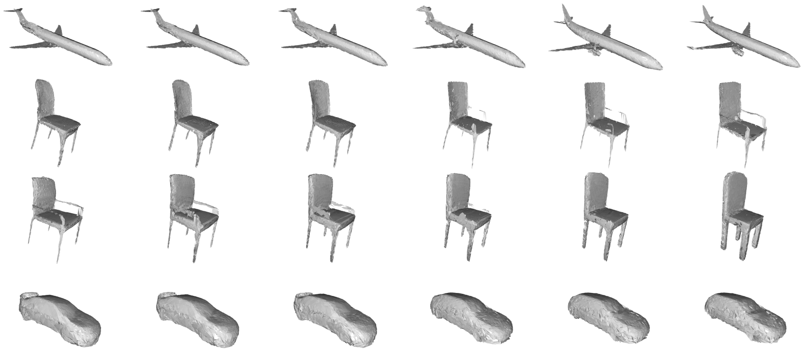

Fig. 6. Mesh interpolations generated by LoCondA-HC (our). Note that for all interpolation steps, our method produces coherent and semantically plausible objects. However, in the middle stage of the interpolation, we ran into patches that were not stitched together in the correct way.

A mesh is typically described either as watertight or not watertight, without a clear measure to define the degree of discontinuities in the object's surface. To address this issue, we propose the parity test, which provides an approximate check of whether a mesh is watertight. According to the test, for a ray cast from infinity towards the object, it must enter and leave the object. This is determined by checking the number of crossings the ray makes with all triangles in the mesh, which should be an even number for the ray to pass the parity test. The mesh is considered watertight if all rays pass the parity test.

To apply this knowledge, we express the measure of watertightness as the ratio of rays that passed the parity test to the total number of casted rays. We begin by sampling N points $p \in \hat{S}$ from all triangles of the reconstructed object $\hat{S}$. Since each point is associated with the triangle it was sampled from, we use the corresponding normal $\hat{n}$ and negate it to obtain the direction of a ray $R(\hat{S}) \ni r = -\hat{n}p$ towards the object[9]. The calculation of the number of crossings $c(r)$ with all triangles is performed. A value of 1 is assigned to each ray that passes a test, and 0 is assigned otherwise. The measure of watertightness is determined by summing the test results for all rays and dividing by the total number of rays: $\mathrm{WT}(\hat{S}) = \frac{\sum_{r \in R(\hat{S})} \mathbb{I}_{\{c(r) \bmod 2=0\}}}{|R(\hat{S})|}$, where $\mathbb{I}$ denotes a characteristic function.

Experiments were conducted to compare AtlasNet with LoCondA-HC and LoCondA-HF in terms of the watertightness of generated meshes. The results are presented in Tab. 3. Note that AtlasNet is unable to generate watertight meshes for any of the considered classes (*airplane*, *chair*, and *car*), which limits its applicability. On the other hand, LoCondA creates meshes in which all sampled rays pass the test.

[9] It is important to note that if a random direction of sampling were employed, confusion would arise due to the possible presence of tangent rays.

Table 3. Comparison of the LoCondA (our) and AtlasNet approaches in terms of watertightness (greater is better). Note that all meshes produced by our method are watertight.

Method	Airplane	Chair	Car	Average
AtlasNet (25 patches)	0.516	0.507	0.499	0.507
LoCondA-HC	**1.00**	**1.00**	**1.00**	**1.00**
LoCondA-HF	**1.00**	**1.00**	**1.00**	**1.00**

6 Conclusions

This paper introduces the Locally Conditioned Atlas (LoCondA), a new method for generating high-quality 3D meshes composed of 2D patches directly from raw point clouds. The proposed approach is based on the continuous atlas paradigm, which allows our model to produce an arbitrary number of patches to form a watertight mesh.

The framework presented here addresses the limitations of previous methods by resolving the challenge of stitching partial meshes. This is achieved by ensuring that each chart is aware of the local neighborhood. Furthermore, our approach can effectively fill in any missing regions in the final mesh by creating a new mapping for the relevant area. The empirical evaluation of LoCondA in several extensive experiments confirms its validity and competitive performance.

The main limitation of the proposed method is the use of a two-stage training procedure. In future work, we plan to extend this approach to an end-to-end training algorithm. Furthermore, we consider the possibility of working with colored meshes.

Acknowledgments. This work was supported by the National Centre of Science (Poland), grants no. 2021/43/B/ST6/01456 (P. Spurek), 2020/39/B/ST6/01511 (T. Trzciński), 2022/45/B/ST6/02817 (T. Trzciński), 2020/37/B/ST6/03463 (M. Zięba), and 2021/41/B/ST6/01370 (M. Mazur). Some experiments were performed on servers purchased with funds from the flagship project entitled "Artificial Intelligence Computing Center Core Facility" from the DigiWorld Priority Research Area within the Excellence Initiative – Research University program at Jagiellonian University in Krakow. This paper has been supported by the Horizon Europe Programme (HORIZON-CL4-2022-HUMAN-02) under the project "ELIAS: European Lighthouse of AI for Sustainability", GA no. 101120237.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 80, pp. 40–49. PMLR (10–15 Jul 2018)

2. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International conference on machine learning. pp. 40–49. PMLR (2018)
3. Ao, S., Guo, Y., Tian, J., Tian, Y., Li, D.: A repeatable and robust local reference frame for 3d surface matching. Pattern Recognition **100**, 107186 (2020)
4. Bansal, A., Russell, B., Gupta, A.: Marr revisited: 2d-3d alignment via surface normal prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5965–5974 (2016)
5. Bednarik, J., Fua, P., Salzmann, M.: Learning to reconstruct texture-less deformable surfaces from a single view. In: 2018 International Conference on 3D Vision (3DV). pp. 606–615. IEEE (2018)
6. Bednarik, J., Parashar, S., Gundogdu, E., Salzmann, M., Fua, P.: Shape reconstruction by learning differentiable surface representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4716–4725 (2020)
7. Chen, Z., Tagliasacchi, A., Zhang, H.: Bsp-net: Generating compact meshes via binary space partitioning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 45–54 (2020)
8. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5939–5948 (2019)
9. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
10. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: Cvxnet: Learnable convex decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 31–44 (2020)
11. Deng, Z., Bednařík, J., Salzmann, M., Fua, P.: Better patch stitching for parametric surface reconstruction. arXiv preprint arXiv:2010.07021 (2020)
12. Fan, H., Su, H., Guibas, L.J.: 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. pp. 605–613 (2017)
13. Grathwohl, W., Chen, R.T.Q., Bettencourt, J., Sutskever, I., Duvenaud, D.: Ffjord: Free-form continuous dynamics for scalable reversible generative models. International Conference on Learning Representations (2019)
14. Groueix, T., Fisher, M., Kim, V.G., Russell, B.C., Aubry, M.: A papier-mâché approach to learning 3d surface generation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 216–224 (2018)
15. Ha, D., Dai, A., Le, Q.V.: Hypernetworks. In: International Conference on Learning Representations (2017)
16. Häne, C., Tulsiani, S., Malik, J.: Hierarchical surface prediction for 3d object reconstruction. In: 2017 International Conference on 3D Vision (3DV). pp. 412–420. IEEE (2017)
17. Kehoe, B., Patil, S., Abbeel, P., Goldberg, K.: A survey of research on cloud robotics and automation. IEEE Transactions on automation science and engineering **12**(2), 398–409 (2015)
18. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (2014)
19. Knop, S., Spurek, P., Tabor, J., Podolak, I., Mazur, M., Jastrzebski, S.: Cramer-wold auto-encoder. Journal of Machine Learning Research **21** (2020)

20. Li, C.L., Zaheer, M., Zhang, Y., Poczos, B., Salakhutdinov, R.: Point cloud gan. In: International Conference on Learning Representations (Workshop Track) (2019)
21. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4460–4470 (2019)
22. Muralikrishnan, S., Kim, V.G., Fisher, M., Chaudhuri, S.: Shape unicode: A unified shape representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3790–3799 (2019)
23. Poursaeed, O., Fisher, M., Aigerman, N., Kim, V.G.: Coupling explicit and implicit surface representations for generative 3d modeling. In: European Conference on Computer Vision. pp. 667–683. Springer (2020)
24. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
26. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. International journal of computer vision **40**(2), 99–121 (2000)
27. Spurek, P., Winczowski, S., Tabor, J., Zamorski, M., Zieba, M., Trzciński, T.: Hypernetwork approach to generating point clouds. Proceedings of Machine Learning Research **119** (2020)
28. Spurek, P., Zieba, M., Tabor, J., Trzcinski, T.: General hypernetwork framework for creating 3d point clouds. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
29. Tran, M.P.: 3d contour closing: A local operator based on chamfer distance transformation (2013)
30. Tsoli, A., Argyros, A., et al.: Patch-based reconstruction of a textureless deformable 3d surface from a single rgb image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 0–0 (2019)
31. Wang, N., Zhang, Y., Li, Z., Fu, Y., Liu, W., Jiang, Y.G.: Pixel2mesh: Generating 3d mesh models from single rgb images. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 52–67 (2018)
32. Yang, B., Luo, W., Urtasun, R.: Pixor: Real-time 3d object detection from point clouds. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 7652–7660 (2018)
33. Yang, G., Huang, X., Hao, Z., Liu, M.Y., Belongie, S., Hariharan, B.: Pointflow: 3d point cloud generation with continuous normalizing flows. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4541–4550 (2019)
34. Yang, J., Zhang, Q., Cao, Z.: The effect of spatial information characterization on 3d local feature descriptors: A quantitative evaluation. Pattern Recognition **66**, 375–391 (2017)
35. Yang, J., Zhang, Q., Xiao, Y., Cao, Z.: Toldi: An effective and robust approach for 3d local shape description. Pattern Recognition **65**, 175–187 (2017)
36. Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 206–215 (2018)
37. Zhao, H., Tang, M., Ding, H.: Hoppf: A novel local surface descriptor for 3d object recognition. Pattern Recognition **103**, 107272 (2020)