Learning Joint Surface Atlases

Theo Deprelle[1*] **Thibault Groueix**[2]

Noam Aigerman[2] **Vladimir G. Kim**[2] **Mathieu Aubry**[1]

[1]LIGM, Ecole des Ponts, Univ Gustave Eiffel, CNRS, [2]Adobe Research

Abstract

This paper describes new techniques for learning atlas-like representations of 3D surfaces, i.e. homeomorphic transformations from a 2D domain to surfaces. Compared to prior work, we propose two major contributions. First, instead of mapping a fixed 2D domain, such as a set of square patches, to the surface, we learn a continuous 2D domain with arbitrary topology by optimizing a point sampling distribution represented as a mixture of Gaussians. Second, we learn consistent mappings in both directions: charts, from the 3D surface to 2D domain, and parametrizations, their inverse. We demonstrate that this improves the quality of the learned surface representation, as well as its consistency in a collection of related shapes. It thus leads to improvements for applications such as correspondence estimation, texture transfer, and consistent UV mapping. As an additional technical contribution, we outline that, while incorporating normal consistency has clear benefits, it leads to issues in the optimization, and that these issues can be mitigated using a simple repulsive regularization. We demonstrate that our contributions provide better surface representation than existing baselines.

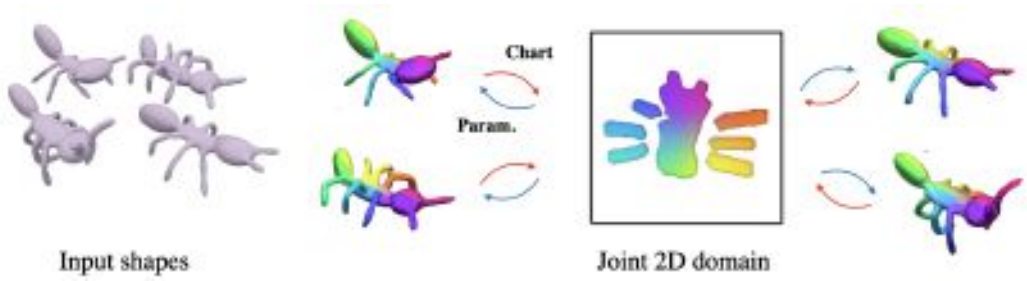

Figure 1: From a collection of shapes without annotations, we learn a 2D domain which can be used to parameterize all shapes, parametrizations (blue) and chart-mappings (red).

1 Introduction

This paper is concerned with 3D surfaces and their representation as atlases or UV maps, i.e., their mappings to and from a domain of the 2D Euclidean space. Surfaces and altases are closely related: surfaces are 2-manifolds embedded in $\mathbb{R}^3$ and are defined in topology by the existence of *charts*, homeomorphic mappings from the surface to a 2D Euclidian domain. This relation is also central to many algorithms related to surfaces: on one hand, the computation of UV maps of meshes is a highly-active research topic: on the other

hand, deep learning works have successfully used atlas-like representation and learn local parametrizations to represent 3D surfaces [4, 10]. These last techniques typically learn to map a fixed sets of 2D squares to 3D, which can approximate a 3D surface and makes it possible to use the 2D domain to compute correspondences between *predicted* surfaces [3] for which parametrizations are learned jointly.

While these parametrization-based methods produce rather-accurate 3D surface reconstructions, they do not lead to a well-defined homeomorphic map between a 2D domain and the predicted surface, which limits the scope of applications of these techniques. For examples, when mapping a set of 2D squares to 3D, AtlasNet [10] leads to many overlaps, and the chart-mappings from 3D to 2D are thus not well defined. The predicted maps from 2D to 3D might also include large amount of distortion, thus not yielding a good UV map of the generated 3D surface, and for example preventing using it to define mappings between different input surfaces. Previous works have attempted to address these limitations in various ways. Williams et al. [25] performs optimization for a single shape on local neighbourhoods with Earth Mover Distance to obtain homeomorphic parametrizations and consistent transitions maps, but this leads to an heavy optimization, with many local parametrizations, and has no obvious extension to multiple shapes. Rather than enforcing consistency between the mappings of square patches AtlasNet v2 [7] attempts to learn the 2D domain, e.g., by learning the positions of a fixed set of points, but loses the continuous aspect of the mapping and can still lead to overlapping patches. DSR [4] keeps the AtlasNet framework and its intrinsic limitations, but uses several regularizations to encourage conformal mappings, to minimize the 3D overlap between the images of the square patches and to prevent patch collapse. However, it is still limited to square patches and we found it difficult to use on complex shapes.

We argue that the two tasks of parameterizations and charts prediction are complementary to one another, and that learning the 3D shape(s) reconstruction should go together with learning a relevant 2D domain. We present an architecture for such a joint optimization, where we learn the 2D domain by learning a 2D probability distribution defined as a mixture of Gaussians and from which we sample points for reconstructing the surface. These sampled points are mapped to 3D and compared (via chamfer distance) to the target point cloud; similarly, the point cloud is mapped to 2D and compared to the sampled points. We optimize for cyclic consistency between the two mappings, as well as for geometric losses such as isometric regularization. Through experiments, we show that our method is able to better reconstruct surfaces than existing baselines, in particular leading to more meaningful parametrizations with fewer artefacts and yielding meaningful correspondences between shapes in a collection. Our code is available on our project webpage[1]

2 Related work

Our work is related to prior work in optimizing chart-mappings for UV parametrization and learning parameterization for surface reconstruction.

Optimizing charts-mappings for UV parametrization. Identifying chart is a long-standing problem in geometry processing [22]. Most prior techniques take a 2-manifold input represented as a mesh and map each point to a 2D domain. These methods typically aim to produce a bijective mapping [23], while also minimizing some distortion metric such as Dirichlet [19], ARAP [15], LSCM [14], and symmetric Dirichlet [21]. Some techniques also aim to predict consistent chart-mappings for a collection of related shapes, so that semantically-similar points on different meshes map to the same point in 2D. Doing so enables many applications such as correspondence estimation [1], morphing [12], and texture transfer [20]. Unlike our method, these techniques do not use deep learning and typically require manual input, such as a sparse set of corresponding points. By using neural networks to represent the UV map we can learn consistent charts without the user input or explicit supervision. We can also co-parameterize point clouds without knowing the underlying mesh. To the best of our knowledge, our work is the first method to use neural networks to learn consistent chart-mappings.

[1] https://imagine.enpc.fr/ deprellt/joint-surface/

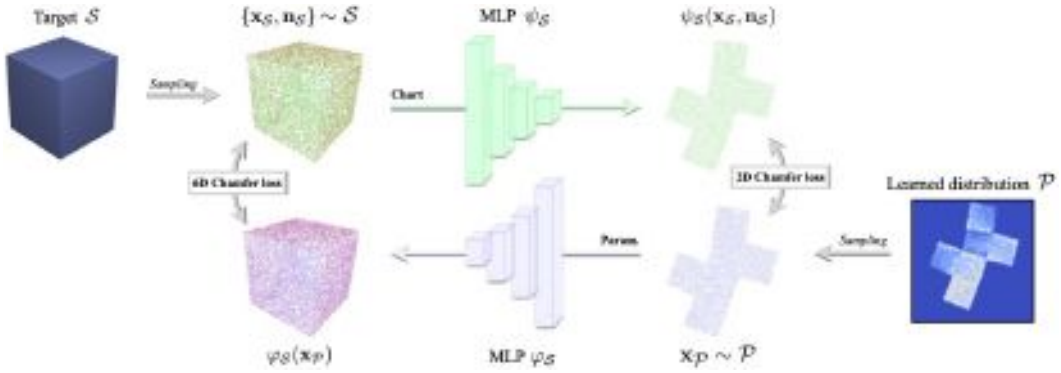

Figure 2: **Single shape architecture.** Given a shape $\mathcal{S}$, we optimize jointly a 2D probability distribution $\mathcal{P}$, a parametrization network, $\varphi_\mathcal{S}$ and chart-mapping network $\psi_\mathcal{S}$. The parameterisation network $\varphi_\mathcal{S}$ takes as input a point $\mathbf{x}_\mathcal{P} \in \mathbb{R}^2$ randomly sampled according to the density probability function $\mathcal{P}$ to produce the 6D vector (3D point and normal coordinates) $\varphi_\mathcal{S}(\mathbf{x}_\mathcal{P})$. The chart-mapping network $\psi_\mathcal{S}$ takes as input a point $\mathbf{x}_\mathcal{S}$ sampled on the shape $\mathcal{S}$, its normal $\mathbf{n}_\mathcal{S}$ and outputs a 2D points $\psi_\mathcal{S}(\mathbf{x}_\mathcal{S}, \mathbf{n}_\mathcal{S})$.

Learning surface parameterization for reconstruction. Previous learning-based techniques for surface parameterization mostly focused on modeling 2D to 3D map for reconstruction. To learn such a mapping from a collection of shape, AtlasNet [10] and FoldingNet [26] pioneered the idea of using a Multi-Layered Perceptrons (MLP) as a class of continuous parametric function that embed a 2-manifold into 3D. By conditionning the MLP on a latent code, and using a large-scale 3D shape repository [5], AtlasNet [10] demonstrated the possibility to predict the surface parameterization from an input mesh, point-cloud or even a single image. Perhaps surprisingly, Williams et al. [25] showed that parameterization did not necessarily need the regularization brought by learning on a collection of shapes, and could also be optimized on individual shapes.

Several approaches introduced novel losses to improve the parameterization. Deng et al. [6] improve the global arrangement of the different parts of the parameterization via a normal-aware reconstruction loss and a stiching loss. In DSR, Bednarik et al. [4] regularize the smoothness of the reconstructed surface by optimizing a conformal energy, based on the Jacobians of the mappings. Remarkably, having access to analytical jacobians in a higher-order differentiation procedure opens-up exciting applications. For instance, Bednarik et al. [3] achieve temporally coherent parameterizations for each frame in a video by regularizing the deformation to locally have a constant metric tensor.

Particularly relevant to us is AtlasNet-v2 [7] which jointly optimize the shape of the 2D manifold and learns the surface parameterization using two proposed strategies. The first one consists in learning a deformation of a fixed template into an elementary structure common to all shapes via an additional AtlasNet-like MLP . This strategy leads to elementary structures that can easily be meshed, but have the same topology as the initial template. The second strategy consists in sampling a fixed set of points on the template and adding them to the optimizer. After optimization, the sampled points can form a complex elementary structure, without topology constraints, but meshing it is not straight-forward. In this work, we propose a method to combine the best features of both strategies, namely the ability to learn topologically complex structures and the ability to mesh them. We achieve this via a novel differentiable layer to sample points from 2D gaussians with a learnable mean (see Sec. 3.2).

Our approach is different from AtlasNet [10] and its variants [2, 3, 4, 6, 7, 16, 25] in that: (i) we aim to jointly learn the surface parametrizations and their inverse functions, the charts-mappings; (ii) we learn a 2D domain relevant to a family of shapes by optimizing a probability density function in the 2D Euclidean plane. We also differ in two novel losses that correctly orient the normals of the reconstructed surface and fix point-collapse in 3D which is a common artifact of AtlasNet-type of approaches.

3 Method

Overview. Given a collection of shapes, our goal is to learn for all shapes surface parameterizations with their inverse chart-mappings and a join 2D domain on which the parametrizations are defined. For simplicity, we first present our approach in the case of a single shape $\mathcal{S}$: in Section 3.1 we explain how to model surface parameterization and chart-mapping, and introduce our main architectural blocks; in Section 3.2 we explain how we learn the 2D UV domain as a probability distribution; in section 3.3 we discuss our losses. Our pipeline for a single shape is illustrated in Figure 2. Finally, in Section 3.4 we explain how to train our approach jointly on a family of shapes.

Notations. We use the following notations:

- $\mathcal{S}$: 3D shape of interest
- $\mathcal{P}$: learned probability distribution in 2D
- $\phi_{\mathcal{S}} : \mathbb{R}^2 \to \mathbb{R}^3$: surface parameterization
- $\varphi_{\mathcal{S}} : \mathbb{R}^2 \to \mathbb{R}^6$: surface parameterization with normals
- $\psi_{\mathcal{S}} : \mathbb{R}^6 \to \mathbb{R}^2$: chart-mapping

We slightly abuse the brackets notation to indicate sampling a set of M points, e.g.:

- $\{\mathbf{x}_{\mathcal{S}}\}$ is a set of M points sampled on $\mathcal{S}$ using a uniform probability distribution
- $\{\mathbf{x}_{\mathcal{P}}\}$ is a set of M points sampled in $\mathbb{R}^2$ according the probability distribution $\mathcal{P}$

3.1 Parametrization and chart-mapping

We explain the two main components of our architecture for a given shape $\mathcal{S}$, a chart-mapping $\psi_{\mathcal{S}}$ and a surface parameterization network $\varphi_{\mathcal{S}}$.

Chart-mapping. We learn a single chart-mapping from the shape $\mathcal{S}$ to $\mathbb{R}^2$, using a Multi-Layer Perceptron (MLP) with both point coordinates and normals as input. We observe that naively learning an $\mathbb{R}^3$ to $\mathbb{R}^2$ mapping leads to the collapse of thin surfaces after their mapping to the 2D domain. Indeed, coordinate-based MLPs are continuous functions and Tancik et. al. [24] showed that they have a prior to learn smooth functions in the absence of positional encoding. Hence, two 3D points with very close spatial coordinates but opposite normals tend to be mapped closely in 2D. For chart-mappings, such smoothness is generally a desirable feature that should be maintained, but distant normals should be a strong cue to indicate that points are intrinsically far. To handle thin surfaces, we propose to learn a mapping $\psi_{\mathcal{S}}$ from $\mathbb{R}^6$ to $\mathbb{R}^2$ that takes as input a point $\mathbf{x}_{\mathcal{S}}$ and its associated normal $\mathbf{n}_{\mathcal{S}}$ scaled to have norm α. The parameter α is a hyperparameter of our approach which controls how much normals contribute to distances compared to the 3D positions of the points.We set it to 0.01 in all our experiments.

Surface parameterization. For a shape $\mathcal{S}$, we seek to learn the inverse function of the chart-mapping, $\psi_{\mathcal{S}}^{-1} : \mathbb{R}^2 \to \mathbb{R}^6$. We first use an MLP $\phi_{\mathcal{S}} : \mathbb{R}^2 \to \mathbb{R}^3$ to parameterize the surface $\mathcal{S}$. The function $\phi_{\mathcal{S}}$ maps a point $\mathbf{x} = (u, v)$ in $\mathbb{R}^2$ into a point $\phi_{\mathcal{S}}(\mathbf{x})$ in $\mathbb{R}^3$. The Jacobian $\mathbf{J}_{\mathbf{x}}$ of the mapping $\phi_{\mathcal{S}}$ is defined at every point $\mathbf{x}$ by:

$$\mathbf{J}_{\mathbf{x}} = \left[\mathbf{J}_{\mathbf{x},u}, \mathbf{J}_{\mathbf{x},v}\right] = \left[\frac{\partial \phi_{\mathcal{S}}}{\partial u}(\mathbf{x}), \frac{\partial \phi_{\mathcal{S}}}{\partial v}(\mathbf{x})\right] . \tag{1}$$

The two partial derivatives are trivial to compute with Pytorch auto-differentiation [18]. The normal $\mathbf{n}$ to the parametrizedsurface at $\phi_{\mathcal{S}}(\mathbf{x})$ can be computed as the normalised cross product of $\mathbf{J}_{\mathbf{x},u}$ and $\mathbf{J}_{\mathbf{x},v}$ and scaled by the same hyperparameter α used for the chart-mapping $\psi_{\mathcal{S}}$:

$$\mathbf{n} = \alpha \frac{\mathbf{J}_{\mathbf{x},u} \times \mathbf{J}_{\mathbf{x},v}}{\|\mathbf{J}_{\mathbf{x},u} \times \mathbf{J}_{\mathbf{x},v}\|}. \tag{2}$$

We define $\varphi_{\mathcal{S}} : \mathbb{R}^2 \to \mathbb{R}^6$ as the concatenation of an output point and its scaled normal:

$$\varphi_{\mathcal{S}}(\mathbf{x}) = [\phi(\mathbf{x}), \mathbf{n}] . \tag{3}$$

To summarize, the mapping $\varphi_{\mathcal{S}}$ is designed to represent the inverse of the chart-mapping $\psi_{\mathcal{S}}$. However, it is not defined for every point in $\mathbb{R}^2$, and in the next section we focus on identifying the 2D domain for which it is defined.

3.2 Learning a 2D domain as a sampling probability distribution

To parametrize a shape $\mathcal{S}$, we want to define a domain in $\mathbb{R}^2$ such that $\phi_{\mathcal{S}}$ defines a bijection from this 2D domain to $\mathcal{S}$. In practice, during training, we want to learn this domain, sample points inside it, and map them using $\varphi_{\mathcal{S}}$. Instead of handling explicitly the 2D domain geometry, e.g., points or primitives similar to [7], we take a probabilistic approach and learn the parameters of a probability distribution $\mathcal{P}$ from which to sample points. This enables us to easily deal with topological changes. We now detail how we represent this probability distribution, learn it, and use it to define a 2D domain.

Modelling 2D sampling probability as a mixture of Gaussians. We sample 2D points according to a probability distribution $\mathcal{P}$ which we model as a mixture of K 2D Gaussians with means $\mu_i \in \mathbb{R}^2$ for $i = 1, ..., K$, a fixed standard deviation $\sigma \in \mathbb{R}$ and fixed mixing coefficients equal to $1/K$:

$$\mathcal{P}(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} \mathcal{N}(\mathbf{x}|\mu_i, \sigma) \quad \text{with} \quad \sigma = \frac{1}{\sqrt{K}} \ . \tag{4}$$

During training, at every iteration, we sample N 2D points $\{\mathbf{x}_{\mathcal{P}}\}$ from $\mathcal{P}$, which are both the input of the parameterisation network $\varphi_{\mathcal{S}}$ and the target of the chart-mapping network $\psi_{\mathcal{S}}$.

Learning the Gaussian means. To allow learning the 2D domain, we make the means μ_i learnable parameters of the method. To do so, we need to see the sampled points as differentiable with respect to the μ_i. We achieve this with the pathwise gradient estimator from [11], also called the reparameterization trick. This consist in expressing a parameterized random variable via a parameterized deterministic function of a parameter-free random variable. For Gaussian Mixture Models (GMM), this simply amounts to sampling a GMM with zero means and adding the means to the sampled points, i.e., defining each point $\mathbf{x}_{\mathcal{P}}$ as the result of the following process: first selecting the id i of a mixture component using a uniform distribution; then sampling a 2D point $\mathbf{x}_{\mathcal{N}}$ from a Gaussian distribution of mean 0 and standard deviation sigma $\mathbf{x}_{\mathcal{N}} \sim \mathcal{N}(0, \sigma)$; finally, defining the point $\mathbf{x}_{\mathcal{P}}$ as $\mathbf{x}_{\mathcal{P}} = \mathbf{x}_{\mathcal{N}} + \mu_i$, which is trivially differentiable with respect to μ_i. Please see Figure 5 for examples of learned probability distributions.

From probability distribution to 2D continuous domain. Once the distribution has been optimized, we simply threshold the probability distribution function $\mathcal{P}$ to obtain a 2D domain. We can then compute a 2D triangulation of the domain, and use the parameterization network ϕ to obtain a 3D mesh (as can be seen in Figure 3).

3.3 Training losses

We now explain the loss function we optimize. We write our loss for a single shape:

$$\mathcal{L}_{single}(\varphi_{\mathcal{S}}, \psi_{\mathcal{S}}, \mu) = \lambda_{\text{6D}}\mathcal{L}_{\text{6D}} + \lambda_{\text{2D}}\mathcal{L}_{\text{2D}} + \lambda_{\text{cycle}}\mathcal{L}_{\text{cycle}} + \lambda_{\text{iso}}\mathcal{L}_{\text{iso}} + \lambda_{\text{rep}}\mathcal{L}_{\text{rep}} \ , \tag{5}$$

where $\mu = (\mu_1, ..., \mu_K)$, the λ are scalar hyper-parameters and the different loss terms are detailed bellow. To compute distances between two sets of points $\mathcal{X}$ and $\mathcal{Y}$, we base our losses on the Chamfer distance defined as:

$$\mathcal{L}_{\text{chamfer}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{|\mathcal{X}|} \sum_{\mathbf{x} \in \mathcal{X}} \min_{\mathbf{y} \in \mathcal{Y}} \|\mathbf{x} - \mathbf{y}\|^2 + \frac{1}{|\mathcal{Y}|} \sum_{\mathbf{y} \in \mathcal{Y}} \min_{\mathbf{x} \in \mathcal{X}} \|\mathbf{y} - \mathbf{x}\|^2 \ , \tag{6}$$

where $|\mathcal{X}|$ and $|\mathcal{Y}|$ are the number of points in $\mathcal{X}$ and $\mathcal{Y}$ respectively. Our set of losses is designed to enforce two objectives : (i) ensuring an accurate surface parameterization $\varphi_{\mathcal{S}}$ with low distortion, (ii) ensuring that $\varphi_{\mathcal{S}}$ and $\psi_{\mathcal{S}}$ are indeed inverse of each other.

Surface parameterization reconstruction loss. The surface parameterization $\varphi_{\mathcal{S}}$ takes as input a set of 2D points $\{\mathbf{x}_{\mathcal{P}}\}$ sampled from the probability distribution $\mathcal{P}$ and outputs a set of 6D points (3D points with scaled normals). We minimize the 6D Chamfer distance between the set of generated points $\{\varphi_{\mathcal{S}}(\mathbf{x}_{\mathcal{P}})\}$ and a set of points $\{\mathbf{x}_{\mathcal{S}}\}$ associated to normals $\{\mathbf{n}_{\mathcal{S}}\}$ sampled on the target shape $\mathcal{S}$:

$$\mathcal{L}_{\text{6D}}(\varphi, \mu) = \mathcal{L}_{\text{chamfer}}(\{\varphi_{\mathcal{S}}(\mathbf{x}_{\mathcal{P}})\}, \{\mathbf{x}_{\mathcal{S}}, \mathbf{n}_{\mathcal{S}}\}) \ . \tag{7}$$

Chart-mapping reconstruction loss. We encourage the overall 2D projection of the shape under $\psi_{\mathcal{S}}$ and the probability distribution $\mathcal{P}$ to be the same. Recall that $\psi_{\mathcal{S}}$ takes as input $\{\mathbf{x}_{\mathcal{S}}, \mathbf{n}_{\mathcal{S}}\}$ (3D points uniformely sanpled on $\mathcal{S}$ with scaled normals) and outputs a set of 2D points $\{\psi_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}, \mathbf{n}_{\mathcal{S}})\}$. We simply minimize the Chamfer distance between the generated 2D points and a set of points $\{\mathbf{x}_{\mathcal{P}}\}$ sampled from $\mathcal{P}$:

$$\mathcal{L}_{\text{2D}}(\psi, \mu) = \mathcal{L}_{\text{chamfer}}(\{\psi_{\mathcal{S}}(\mathbf{x}_{\mathcal{S}}, \mathbf{n}_{\mathcal{S}})\}, \{\mathbf{x}_{\mathcal{P}}\}) \ . \tag{8}$$

Cycle-consistency loss. We want the two mappings φ and ψ to be inverse of one another. We encourage this using a cycle-consistency loss on 2D points sampled from $\mathcal{P}$ and on 3D points sampled uniformly on $\mathcal{S}$ with their associated normals:

$$\mathcal{L}_{\text{cycle}}(\varphi, \psi, \mu) = \frac{1}{M} \sum_{\mathbf{x} \in \{\mathbf{x}_{\mathcal{P}}\}} \|\mathbf{x} - \psi \circ \varphi(\mathbf{x})\|^2 + \frac{1}{M} \sum_{(\mathbf{x}, \mathbf{n}) \sim \{\mathbf{x}_{\mathcal{S}}, \mathbf{n}_{\mathcal{S}}\}} \|\mathbf{x} - \varphi \circ \psi(\mathbf{x}, \mathbf{n})\|^2 \ . \tag{9}$$

Distortion regularization loss. We limit distortion in the parameterization with an isometric regularization. Given the Jacobian $\mathbf{J}_{\mathbf{x}}$ of the transformation ϕ at point $\mathbf{x}$ (Equation 1) and $\mathbf{I}$ the identity matrix, the isometric loss can be written:

$$\mathcal{L}_{\text{iso}}(\varphi_{\mathcal{S}}, \mu) = \frac{1}{M} \sum_{\mathbf{x} \in \{\mathbf{x}_{\mathcal{P}}\}} \|\mathbf{J}_{\mathbf{x}} \mathbf{J}_{\mathbf{x}}^T - \mathbf{I}\| \ , \tag{10}$$

where the sum is over points sampled according to $\mathcal{P}$. As already observed in [3, 9], this type of regularization has the additional benefit of making the parameterizations more consistent across shapes.

Probability distribution regularization loss. Non-uniform density is a known failure mode of the Chamfer distance, as shown in Figure 5 and also observed in [4]. In theory, a loss based on optimal transport like the Earth Mover distance would be ideal to fix this problem. However, in practice, we ran into optimisation, training time and parameter tuning issues when using EMD. On the contrary the Chamfer loss is simple to use and fast to compute. We thus use the Chamfer loss and introduce a repulsive loss between the Gaussian means defining the probability distribution $\mathcal{P}$ as a regularization:

$$\mathcal{L}_{\text{rep}}(\mu) = \frac{1}{K^2} \sum_{i,j \in [0,K]} \exp(-\frac{\|\mu_i - \mu_j\|}{\sigma}) \ , \tag{11}$$

where σ is the Gaussian standard deviations in the definition of $\mathcal{P}$. We found that this simple loss lead to much more uniform distributions of points both in the 2D plane and in the reconstructed shapes.

3.4 Joint learning on a family of shapes.

We now explain how we learn jointly atlases on a collection of N shapes $\mathcal{S}_1, ..., \mathcal{S}_N$. Since we want to share the 2D domain between the different shapes, we do not condition the probability distribution $\mathcal{P}$ on the shape, and learn a single one for all shapes. On the contrary, the parametrization and chart-mappings are expected to depend on the shape. Rather than learning them completely independently for each shape, we use the auto-decoder framework [17], where we optimize for each shape $\mathcal{S}_i$ feature vectors $\mathbf{f}_{\psi,i}$ and $\mathbf{f}_{\varphi,i}$ which we use to define respectively $\varphi_{\mathcal{S}_i}$ and $\psi_{\mathcal{S}_i}$. More precisely, we learn jointly for all shapes networks φ and ψ, and define for each shape $\mathcal{S}_i$ for all $\mathbf{x} \in \mathbb{R}^2$, $\varphi_{\mathcal{S}_i}(\mathbf{x}) = \varphi(\mathbf{x}, \mathbf{f}_{\psi,i})$ and for all $\mathbf{x} \in \mathbb{R}^6$, $\psi_{\mathcal{S}_i}(\mathbf{x}) = \psi(\mathbf{x}, \mathbf{f}_{\psi,i})$. We then optimize ψ, φ and μ by minimizing the loss:

$$\mathcal{L}_{\text{full}}(\varphi, \psi, \mu, \mathbf{f}_{\psi}, \mathbf{f}_{\varphi}) = \sum_{i=1}^{N} \mathcal{L}_{\text{single}}(\varphi_{\mathcal{S}_i}, \psi_{\mathcal{S}_i}, \mu) \ , \tag{12}$$

where $\mathbf{f}_{\psi} = (\mathbf{f}_{\psi,1}, ..., \mathbf{f}_{\psi,N})$ and $\mathbf{f}_{\varphi} = (\mathbf{f}_{\varphi,1}, ..., \mathbf{f}_{\varphi,N})$.

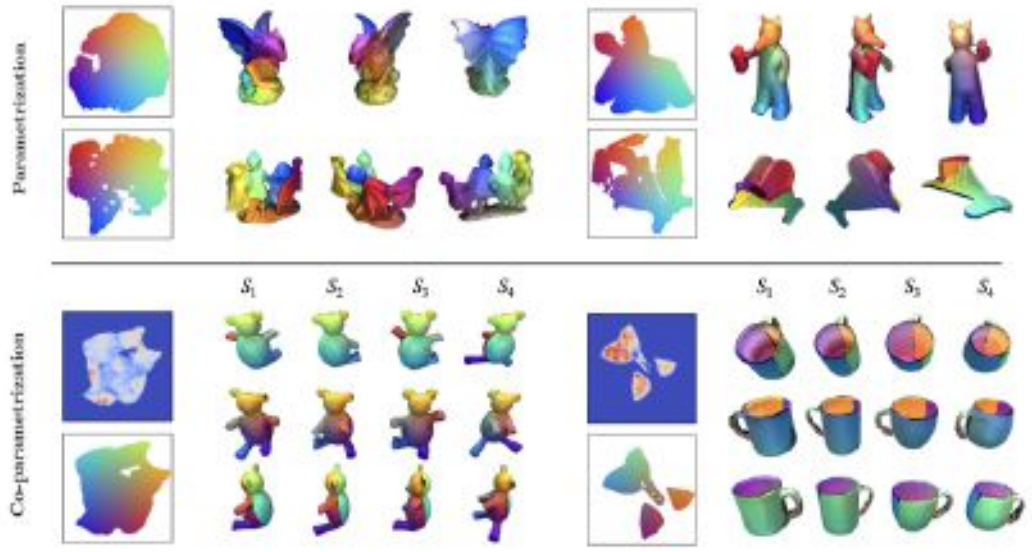

Figure 3: **Top: single shape parametrization.** On the shapes of [25], we transfer a colored mesh of the learned 2D domain (left) to 3D, which enables visualizing correspondence and cuts (right). **Bottom: co-parameterization.** For 'teddy' and 'cup' shapes of SHREC [8], we show the joint sampling probabilities and 2D domain, and the reconstructions of 4 shapes with 3 different viewpoints and consistent coloring. Note the quality of the parametrization compared to patch-based methods [4, 10]

Implementation details. We use latent codes $\mathbf{f}_{\varphi,i}$ and $\mathbf{f}_{\psi,i}$ of dimension 256. The architectures we use for φ and ψ are MLP with 5 hidden layers of size 256 and ReLU activations. We do not use batch normalization layers. $\mathcal{P}$ is defined using $K = 10,000$ mixture components. We sample $M = 10^4$ points both on the shapes and according to $\mathcal{P}$ at every training iteration. We train with $\lambda_{\text{6D}} = 1$, $\lambda_{\text{2D}} = 10^{-2}$, $\lambda_{\text{cycle}} = 1/M$, $\lambda_{\text{iso}} = 10^{-4}/M$ and $\lambda_{\text{rep}} = 1$.

4 Experiments

Datasets. We use individually the shapes of Williams et al. [25], which come from five high-resolution scans with over a million points with associated normals. We generate the manifold meshes from this data using screened Poisson Surface Reconstruction. The resulting meshes have a variety of geometric details and different topologies, providing interesting challenges for atlas-based representations. The SHREC dataset [8] contains 400 manifold meshes with sparse correspondence annotations, which enables us to quantitatively evaluate the consistency of our joint atlases. For our experiments we selected categories, *ant*, *teddy*, *cups* and *armadillo*, aiming for topological and geometric diversity, and four shapes in each.

Reconstruction Metrics. To evaluate how well our representation matches the input shape, we report two commonly used metrics: 3D Chamfer Distance and the Earth Mover's Distance (EMD). Given two sets of points $\mathcal{X}$ and $\mathcal{Y}$ with $M' = 2000$ randomly-sampled points each, we approximate the EMD as:

$$\mathcal{L}_{\text{EMD}}(\mathcal{X}, \mathcal{Y}) = \frac{1}{M'} \sum_{\mathbf{x}_i \in \mathcal{X}} \sum_{\mathbf{y}_j \in \mathcal{Y}} C_{i,j} \|\mathbf{x}_i - \mathbf{y}_j\|^2 \ , \tag{13}$$

where the association matrix C is such that $C_{i,j} = 1$ if point $\mathbf{x}_i$ is mapped to point $\mathbf{y}_j$ and $C_{i,j} = 1$, and is computed to minimize $\mathcal{L}_{\text{EMD}}$ via Hungarian Algorithm [13]. To evaluate the reconstructions of the normals of the surfaces, we also report the distances of the normals using the associations given either by the chamfer distance or the EMD using the spatial coordinates.

		Williams et al. [25]				SHREC [8]				
		Spatial		**Normal**		**Spatial**		**Normal**		**Corresp.**
		Ch. ↓	Emd ↓	Ch. ↓	Emd ↓	Ch. ↓	Emd ↓	Ch. ↓	Emd ↓	L2 ↓
Baselines										
(1)	ANv1-1 [10]	2.3±0.0	**5.0**±0.1	1.2±0.2	1.2±0.2	2.2±0.0	**5.4**±0.5	1.1±0.2	1.1±0.2	2.6±0.8
(2)	ANv2-1 [7]	**2.2**±0.0	11.1±2.1	1.2±0.1	1.3±0.1	**2.0**±0.0	7.1±1.4	1.3±0.2	1.3±0.2	1.3±0.1
(3)	ANv1-10 [10]	2.4±0.0	11.4±0.9	1.1±0.0	1.1±0.0	2.3±0.1	11.1±0.4	1.1±0.1	1.1±0.1	3.5±1.4
(4)	ANv2-10 [7]	2.2±0.0	11.1±2.1	1.2±0.1	1.3±0.1	2.1±0.0	13.4±0.5	1.0±0.1	1.1±0.1	2.8±1.6
(5)	DSR-10 [4]	4.3±0.8	18.2±0.8	1.4±0.0	1.2±0.0	3.1±1.1	10.0±2.8	1.3±0.1	1.3±0.1	2.6±1.2
Ours										
(6)	ϕ	**2.2**±0.0	15.2±5.9	1.0±0.2	1.1±0.2	**2.0**±0.0	6.8±0.9	1.1±0.2	1.1±0.2	2.0±0.8
(7)	ϕ, n	2.4±0.1	20.3±3.3	0.3±0.0	0.7±0.0	2.1±0.3	13.1±8.0	**0.2**±0.0	0.5±0.0	1.4±1.1
(8)	ϕ, n, r	**2.2**±0.1	12.9±4.4	**0.2**±0.1	0.6±0.1	**2.0**±0.1	9.6±6.6	**0.2**±0.0	**0.4**±0.1	2.8±0.9
(9)	ϕ, n, r, p	2.3±0.2	13.2±4.0	0.3±0.1	0.6±0.1	**2.0**±0.0	8.2±2.1	**0.2**±0.0	**0.4**±0.0	1.5±1.0
(10)	ϕ, ψ, n, r, p	2.6±0.0	**5.1**±0.1	**0.2**±0.0	**0.5**±0.0	2.2±0.0	**5.4**±0.2	**0.2**±0.0	**0.4**±0.0	1.0±1.0
(11)	ϕ, ψ, n, r, p, i	2.6±0.2	5.5±1.3	**0.2**±0.3	**0.5**±0.2	2.3±0.1	5.9±0.6	**0.2**±0.0	**0.4**±0.0	**0.8**±1.0

Notations n: normal - r: repulsion loss - p: Probability distribution function $\mathcal{P}$ - i: Isometric regularisation of ϕ

Table 1: **Comparison and ablation study.** We compare our method, successively adding its different componenets, to AtlasNet [10], AtlasNetV2 [7] and DSR [4]. We compute association between target and reconstructed 3D points using Chamfer distance and EMD, then report for each association the average distance between the points coordinates ('Spatial') and their normals ('Normals'). We also evaluate the quality of the correspondences when available. All the values are scaled by a factor of 10^{-2}. Please see text for details.

Correspondence Metric. Given a pair of shapes $\mathcal{S}_1, \mathcal{S}_2$ and a set of keypoints $\mathbf{p}_i \in \mathcal{S}_1$, we find the corresponding points $\mathbf{q_i} \in \mathcal{S}_2$ by compositing the chart-mapping and parameterization networks: $\mathbf{q}_i = \varphi_{\mathcal{S}_2} \circ \psi_{\mathcal{S}_1}(\mathbf{p}_i)$. For the baselines that do not have a chart-mapping network, we obtain correspondences for each point $\mathbf{p}_i$ using the following process: we sample points according to $\mathcal{P}$, map them to $\mathcal{S}_1$, select the one which image by $\varphi_{\mathcal{S}_1}$ is closest to $\mathbf{p}_i$, and map it to $\mathcal{S}_2$ using $\varphi_{\mathcal{S}_2}$. Given M'' ground truth correspondences $\mathbf{q}_i^{\text{gt}}$ for each $\mathbf{p_i}$, we measure the L2 loss on the spatial coordinates as $\frac{1}{M''} \sum_i |\mathbf{q_i} - \mathbf{q_i^{\text{gt}}}|$.

Results and analysis. Qualitative results for individual shapes of Williams et al. [25] are shown in Figure 3 (top). Joint atlases for the SHREC dataset [8], can be seen for the 'ants' are in the teaser Figure 1 and for 'cup' and 'teddy' in Figure 3 (bottom). Note how we manage to learn continuous 2D domains with complex topology, which we can mesh to obtain a high quality 3D mesh for the shapes. Also note that compared to the patch-based approaches we obtain few and meaningful cuts in the parametrization, without any overlap. Finally, note the consistency in the reconstruction of the different shapes, which can be visualized by consistent colors transfered from the joint 2D domain to all shapes.

We report quantitative results in Table 1. To account for randomness in initialization and optimization, we re-run each method four times and report mean and standard deviation. We report results for three baselines: vanilla AtlasNet [10] (rows 1 and 3) which only learns the parameterization network, AtlasNet v2-points [7] (rows 2 and 4) which learns elementary point structures in the 2D domain, and DSR [4] (row 5) which uses several regularisation losses including an isometry loss. We try the first two methods with 1 and 10 UV patches. We found that methods using a single patch typically lead to similar Chamfer distances but much lower EMD. We believe that this is because using a single patch encourages points to be more uniformly distributed on the surface. We also found that DSR [4] provided quantitativelty slightly worse results, which is consistent with what was reported in this paper that mainly aims at visual quality.

We evaluate our method by successively adding our different components. We start (row 6) by only using our parametrization network, learning a fixed set of input point positions with only 3D Chamfer distance as supervision, similar to AtlasNet v2 with 1 patch and obtained similar results.
We then add normals to the Chamfer loss (row 7), which unsurprisingly gives a very strong boost to the normal metrics. Qualitatively, the effect can be seen in the left part of Figure 4 where the orientation of the normals is color-coded for each point: without the normals in the loss, a significant part of the normals are back-facing. Adding the normals in the loss however has a second undesired effect and significantly increases the EMD. This can be understood qualitatively by looking at the results, where we see greatly varying density of points over the reconstructed shapes and in the 2D space. We interpret this as the fact that

Figure 4: **Impact of the normals and the repulsion loss.** Note the variation of the color-coded normal directions (left) and the variations in the point density (right).

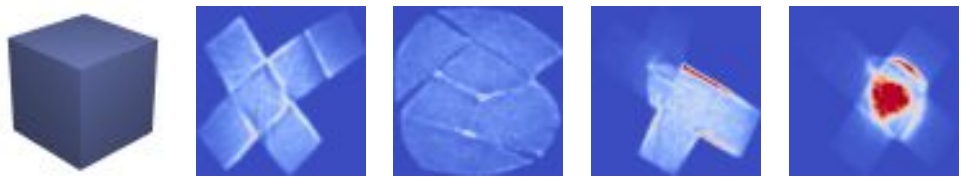

Figure 5: **Qualitative ablation results.** Given a target cube shape, we show the probability distribution $\mathcal{P}$ learned by our method and three key ablations: not using the isometric loss, not using the normals, and not using the repulsive loss.

adding normals in the loss complexifies the loss landscape and adds bad local minima.
To avoid this effect, we add our repulsive regularization to the loss (row 8). We can see this improves the EMD results without degrading the Chamfer results and normal consistency. The effect on the points density is also striking qualitatively, as visualized in the right part of Figure 4.
At this point, the parametrization is still only optimized on a set of points. To recover an interpretation of the parametrization as a continuous mapping of a 2D domain, we introduce our sampling probability distribution (row 9) which has little quantitative effect but is crucial to define a continuous 2D mapping.
We can then introduce our chart-mapping, together with the 2D reconstruction and cycle losses (row 10). It can be seen that beside its theoretical interest and benefits, it has a clear quantitative impact: both the EMD and the correspondence metrics are clearly boosted.
Finally, adding the isometric regularization (row 11) provides a small additional boost in term of correspondence quality, i.e., consistency between different jointly learned atlases.

Note that all of our regularization losses remain necessary to the success of our method, as can be visualized qualitatively in Figure 5, where we use our architecture to learn the reconstruction of a simple cube and visualize the learned 2D sampling distribution. Without the normal loss, we observe a concentration of the density on edges, an equivalent of point collapse described in [4]. Without the repulsion loss, the density is concentrated on a single face. Without the isometry loss, the shape of the cube is highly distorded. FWith our full method the net of the cube is clear and the point density relatively uniform.

5 Conclusion

We propose a novel technique for representing a shape or a collection of shapes, with two key differences with respect to prior work on atlas-like representations. First, we joinlty learn two maps, a parameterization and a chart-mapping. Second, we learn the 2D domain on which the parametrization is defined. This makes our representation much closer than previous works to be a proper atlas, i.e., one that defines an homeomorphism between 3D shape and a 2D domain. It also enables co-parameterization, where homeomorphisms are learned between several shapes and a single 2D domain, which can have applications for consistent texturing and correspondence estimation. We further offer other technical contributions, such as learning the 2D domain by optimizing a sampling probability distribution, analyzing the effect of incorporating normals in the optimization, and introducing a repulsive loss to have a more even point distribution. We believe our work is an important step towards learning consistent atlases and it will inspire future work on further improving quality of atlases, such as modeling transition maps, minimizing seams, and distortion.

Acknowledgments Thanks to F. Darmon, R. Loiseau, E. Vincent for their feedbacks on the manuscript; E. Shechtman , D. Picard, Y. Siglidis and G. Ponimatkin for inspiring discussions; and B. George for code suggestions; This work was partly supported by ANR project EnHerit ANR-17-CE23-0008, Labex Bézout, gifts from Adobe to École des Ponts and HPC resources from GENCI-IDRIS (2021-AD011011937R1).

References

[1] Aigerman, N., Poranne, R., Lipman, Y.: Seamless surface mappings. ACM SIGGRAPH (2015)

[2] Badki, A., Gallo, O., Kautz, J., Sen, P.: Meshlet priors for 3d mesh reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2849–2858 (2020)

[3] Bednarik, J., Kim, V.G., Chaudhuri, S., Parashar, S., Salzmann, M., Fua, P., Aigerman, N.: Temporally-coherent surface reconstruction via metric-consistent atlases. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10458–10467 (2021)

[4] 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)

[5] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. rep., Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)

[6] Deng, Z., Bednařík, J., Salzmann, M., Fua, P.: Better patch stitching for parametric surface reconstruction. In: 2020 International Conference on 3D Vision (3DV). pp. 593–602. IEEE (2020)

[7] Deprelle, T., Groueix, T., Fisher, M., Kim, V., Russell, B., Aubry, M.: Learning elementary structures for 3d shape generation and matching. In: Advances in Neural Information Processing Systems. pp. 7433–7443 (2019)

[8] Giorgi, D., Biasotti, S., Paraboschi, L.: Shape retrieval contest 2007: Watertight models track. SHREC competition **8**, 7 (2007)

[9] Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: 3d-coded : 3d correspondences by deep deformation. In: ECCV (2018)

[10] Groueix, T., Fisher, M., Kim, V.G., Russell, B., Aubry, M.: AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2018)

[11] Kingma, D.P., Welling, M.: Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114 (2013)

[12] Kraevoy, V., Sheffer, A.: Cross-parameterization and compatible remeshing of 3d models. ACM Transactions on Graphics (ToG) **23**, 861–869 (2004)

[13] Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**, 83–97 (1955)

[14] Lévy, B., Petitjean, S., Ray, N., Maillot, J.: Least squares conformal maps for automatic texture atlas generation. In: SIGGRAPH (2002)

[15] Liu, L., Zhang, L., Xu, Y., Gotsman, C., Gortler, S.J.: A local/global approach to mesh parameterization. In: Computer Graphics Forum. vol. 27, pp. 1495–1504. Wiley Online Library (2008)

[16] Pang, J., Li, D., Tian, D.: Tearingnet: Point cloud autoencoder to learn topology-friendly representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7453–7462 (2021)

[17] Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: Deepsdf: Learning continuous signed distance functions for shape representation. In: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 165–174 (2019)

[18] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems **32** (2019)

[19] Pinkall, U., Polthier, K.: Computing discrete minimal surfaces and their conjugates. EXPERIMENTAL MATHEMATICS **2**, 15–36 (1993)

[20] Praun, E., Sweldens, W., Schröder, P.: Consistent mesh parameterizations. In: Proceedings of the 28th annual conference on Computer graphics and interactive techniques. pp. 179–184 (2001)

[21] Rabinovich, M., Poranne, R., Panozzo, D., Sorkine-Hornung, O.: Scalable locally injective mappings. ACM Transactions on Graphics **36**, 16:1–16:16 (Apr 2017)

[22] Sheffer, A., Hormann, K., Levy, B., Desbrun, M., Zhou, K.: Nnwarp: Neural network-based nonlinear deformation. SIGGRAPH 2007 Course Notes (2007)

[23] Smith, J., Schaefer, S.: Bijective parameterization with free boundaries. ACM Trans. Graph. **34** (jul 2015). https://doi.org/10.1145/2766947, `https://doi.org/10.1145/2766947`

[24] Tancik, M., Srinivasan, P., Mildenhall, B., Fridovich-Keil, S., Raghavan, N., Singhal, U., Ramamoorthi, R., Barron, J., Ng, R.: Fourier features let networks learn high frequency functions in low dimensional domains. Advances in Neural Information Processing Systems **33**, 7537–7547 (2020)

[25] Williams, F., Schneider, T., Silva, C., Zorin, D., Bruna, J., Panozzo, D.: Deep geometric prior for surface reconstruction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10130–10139 (2019)

[26] Yang, Y., Feng, C., Shen, Y., Tian, D.: Foldingnet: Point cloud auto-encoder via deep grid deformation. In: CVPR (2018)