

# Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization

Chandradeep Pokhariya\*

chandradeep.pokhariya@research.iiit.ac.in  
IIIT Hyderabad  
India

Astitva Srivastava  
IIIT Hyderabad  
India

astitva.srivastava@reserach.iiit.ac.in

Shanthika Naik\*

shanthika.naik@research.iiit.ac.in  
IIIT Hyderabad  
India

Avinash Sharma  
IIIT Hyderabad  
India

asharma@iiit.ac.in

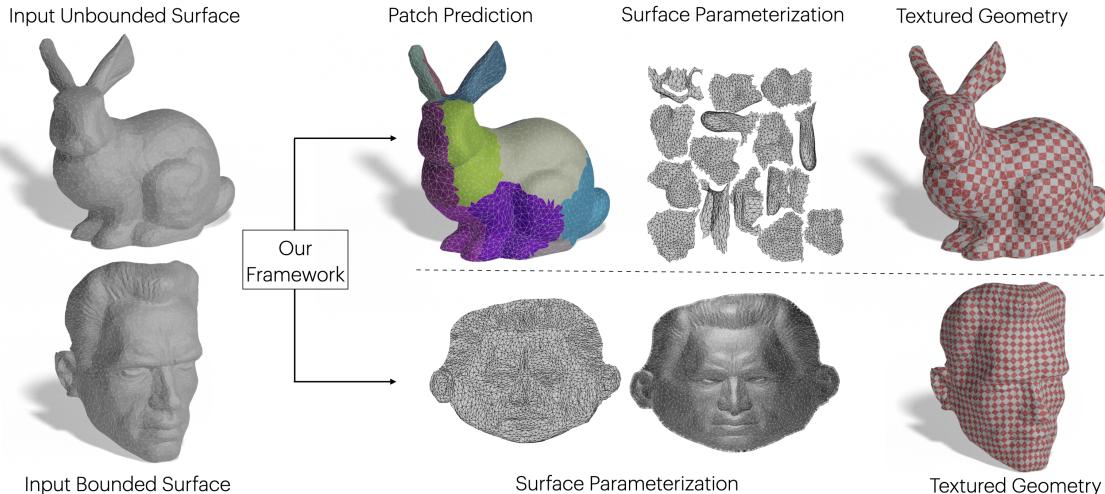


Figure 1: UV parameterization for bounded and unbounded surfaces estimated via our proposed method.

## ABSTRACT

We present a novel object-centric, self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both bounded and unbounded surfaces. Our framework leverages diffusion enabled global-to-local shape context for each vertex to first partition the unbounded surface into multiple patches using proposed self-supervised PatchNet and subsequently perform independent UV parameterization of these patches by learning forward and backward UV mapping for individual patches. Thus, our framework enables learning a discretization-agnostic parameterization at a lower resolution and then directly

infer the parameterization for a higher resolution mesh without retraining. We evaluate our framework on multiple 3D objects from publicly available SHREC dataset and report superior/faster UV parameterization over conventional methods.

## CCS CONCEPTS

- Computing methodologies → Parametric curve and surface models; Neural networks.

## KEYWORDS

UV parameterization, texture mapping, neural network, self-supervised learning, manifold, isometric, conformal, computer graphics.

### ACM Reference Format:

Chandradeep Pokhariya, Shanthika Naik, Astitva Srivastava, and Avinash Sharma. 2018. Discretization-Agnostic Deep Self-Supervised 3D Surface Parameterization. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX')*. ACM, New York, NY, USA, 5 pages. <https://doi.org/XXXXXXX.XXXXXXX>

## 1 INTRODUCTION

Estimating the UV parameterization of arbitrary 3D surfaces lies at the core of computer graphics and geometry processing domain,

\*Both authors contributed equally to this research.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

© 2018 Association for Computing Machinery.

ACM ISBN 978-1-4503-XXXX-X/18/06...\$15.00

<https://doi.org/XXXXXXX.XXXXXXX>

with a wide range of applications such as 3D modeling, texture-mapping, remeshing, simulation, etc. Formally, it is defined as the projection of vertices of a tessellated surface (polygon mesh) onto a 2D map (UV plane). Determining the aforementioned mapping is not a trivial task and demands a solution with specific properties. The estimated mapping is expected to be isometric, conformal, and non-overlapping. Existing conventional methods [4, 5, 7, 8, 10] aims to estimate an object-centric mapping with an iterative optimization process, focusing on minimizing an energy function explicitly constructed to retain the desired properties. However, they face scalability issues while dealing with high-resolution object meshes.

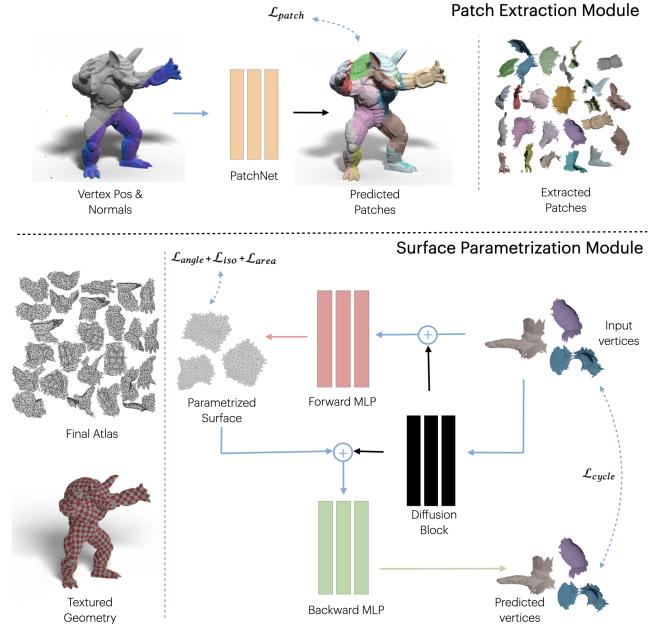
With the advent of deep learning, researchers are harnessing the power of neural networks to solve various ill-posed problems, offering tractable solutions. Neural surface parameterization has recently been attempted [1] but under supervised, data-driven settings, requiring a large amount of training data. Such supervised learning solutions get subjected to data bias and hence suffer from poor generalization to unseen, out-of-distribution samples. Additionally, their solution is restricted to bounded surfaces only and provides no provision to deal with unbounded, closed surfaces.

In this paper, we present a novel object-centric, self-supervised framework for learning the discretization-agnostic surface parameterization of arbitrary 3D objects with both bounded and unbounded surfaces as shown in Figure 1. First, to handle unbounded surfaces (e.g., a sphere) or surfaces with regions of extreme extrinsic-curvature, we propose a learning-based partitioning of the given surface into multiple bounded patches, which are independently parameterized. To this end, we employ a self-supervised network that assigns each 3D point of the surface to one of the patches, trained using losses based on local features (such as point-normals) and geodesic relationships within the patch.

Subsequently, we propose to learn the surface parameterization of an arbitrary (bounded) 3D surface to a UV plane using a *Multi-layer Perceptron* (MLP). More specifically, given a *bounded* 3D surface (patch), we train the forward MLP to predict per-point UV coordinates independently. In order to ensure a meaningful UV mapping, we enforce cycle-consistency loss between the input and reconstructed surface by learning a backward mapping (UV-to-3D) MLP. Additional losses are employed to achieve desired properties of surface parameterization, i.e., isometric, conformal, and area-preserving. A diffusion process [9] over the mesh provides multi-scale characterization of the underlying surface, entailing a global-to-local context for each of the vertex and hence DiffusionNet backbone is used for PatchNet, and similarly, respective features are appended while learning surface parameterization in order to achieve discretization-agnostic UV mapping. A key advantage of learning a discretization-agnostic parameterization is that we can learn on meshes at a lower resolution and then directly infer the parameterization for high resolution meshes without retraining, as shown in Figure 3.

## 2 RELATED WORK

Conventional methods for mesh parameterization can be categorised as *single-patch, fixed boundary* methods, e.g., harmonic parameterization [10], *single-patch, free boundary* methods, e.g.,



**Figure 2: The outline of proposed framework.**

LCSM [4]; and *global parameterization* methods, e.g., Voronoi atlas parameterization [7]. Global parameterization methods can deal with meshes of arbitrary genus. They achieve this by either partitioning the unbounded mesh into multiple bounded patches or detecting one or more seams to cut the mesh, making it bounded. Boundary-First Flattening [8] and OptCuts [5] fall into this class.

Neural parameterization methods have been gaining popularity over the past few years due to their ability to address ill-posed problems. AtlasNet [3], and DGP [11] propose a way of surface reconstruction and parameterization by training a neural network to represent a single UV chart over the reconstructed surface. Both the methods use a fixed number of patches for the surface parameterization but require a different neural network for every patch, which is an overkill and difficult to scale up. A recent work AUV-Net [2] takes a point cloud as input and learns parameterization of aligned surfaces (e.g., faces and humans in T-poses) using a cycle-loss, but requires all the meshes to have similar topology and same orientation to enable learning. Moreover, the proposed two-patch estimation method is very naive and cannot scale to an arbitrary number of patches. Another recent method [1] learns intrinsic mapping of arbitrary surfaces in a supervised setup, where a conventional method acts as the ground truth. However, it can only deal with bounded surfaces and does not provide its provision for unbounded surfaces (e.g., spheres).

## 3 METHOD

We now describe the proposed framework in detail. The input to our framework is a mesh  $\mathcal{M} = \{\mathcal{V}, \mathcal{F}, \mathcal{N}\}$ , where  $\mathcal{V}$ ,  $\mathcal{F}$  and  $\mathcal{N}$  are the sets of vertex positions, faces, and vertex-normals, respectively. Our framework consists of two modules: (i) Patch Extraction Module and (ii) Surface parameterization Module.

### 3.1 Patch Extraction Module

Handling surfaces with regions of high extrinsic curvature or unbounded topology requires the 3D manifold to be partitioned into multiple bounded patches to minimize distortion and overlap. Each patch is defined as  $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_k\}$  ( $k = 1, 2, \dots, K$ ), where  $\mathcal{V}_k \subseteq \mathcal{V}$  is the set of vertices belonging to  $\mathcal{P}_k$ ,  $\mathcal{F}_k \subseteq \mathcal{F}$  is the set of faces defined on  $\mathcal{V}_k$ , and  $\mathcal{N}_k \subseteq \mathcal{N}$  is the associated set of vertex-normals. We propose PatchNet with parameters  $\phi_{patch}$ , which learns to assign each vertex of  $\mathcal{M}$  to one of the  $K$  patches, as shown in ???. Here,  $K$  is a controllable parameter and can vary based on the acceptable amount of distortion in the input mesh. To learn the parameters  $\phi_{patch}$  we minimize the following cosine similarity constraint on the estimated patches:

$$\mathcal{L}_{cos} = \sum_{k=1}^K \left[ 1 - \frac{1}{|\mathcal{P}_k|} \left( \sum_{i,j \in \mathcal{V}_k} (\hat{n}_i^T \hat{n}_j) \right) \right] \quad (1)$$

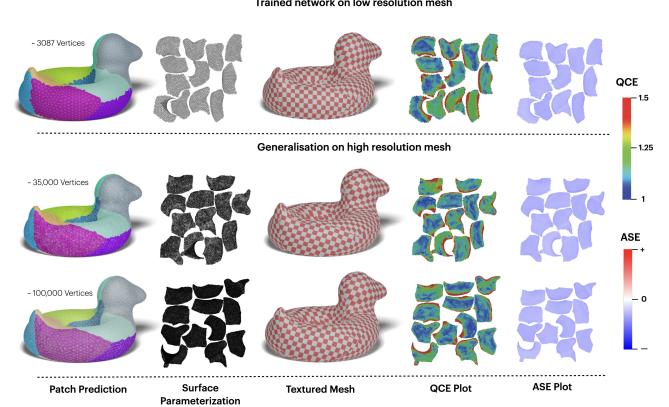
where  $i, j \in \mathcal{V}_k$  are the pair of vertices with unit normal vectors  $\hat{n}_i, \hat{n}_j \in \mathcal{N}_k$ , respectively, and  $|\mathcal{P}_k|$  is the number of vertices in that patch. The above constraint has the effect of producing locally flat patches. However, it is possible that geodesically far apart triangles with high cosine similarity will be assigned to the same patch, which is undesirable. To circumvent such disjoint assignments, we minimize the following additional constraint:

$$\mathcal{L}_{geo} = \sum_{k=1}^K \frac{1}{|\mathcal{P}_k|} \left( \sum_{i,j \in \mathcal{V}_k} g(i, j) \right), \hat{n}_j \quad (2)$$

where  $g(i, j)$  denotes the geodesic distance between the pair of vertices  $i$  &  $j$  within the patch. We model PatchNet using Diffusion-Net [9] architecture in order to achieve multi-scale characterization of the underlying surface, entailing a global-to-local context for all the vertices. Input to PatchNet is  $\mathcal{V}$  &  $\mathcal{N}$  and the output is predicted assignment probability for all the vertices to each of the  $K$  patches, given by  $\mathcal{V}_l \in [0, 1]^{|\mathcal{V}| \times K}$ . Subsequently, per-face probabilities  $\mathcal{F}_l \in [0, 1]^{|\mathcal{F}| \times K}$  are obtained by taking the mean probabilities of the corresponding face vertices. We further consolidate the per-face probabilities by taking an average over neighboring faces and then each face is assigned to the patch with the highest probability. Note that, in case of bounded surface with extrinsic curvature of low-variability, the whole mesh can be considered as a single patch. The combined objective function for patch extraction becomes  $\mathcal{L}_{patch} = \lambda_{cos} \mathcal{L}_{cos} + \lambda_{geo} \mathcal{L}_{geo}$ .

### 3.2 Surface Parameterization Module

Each patch  $\mathcal{P}_k = \{\mathcal{V}_k, \mathcal{F}_k, \mathcal{N}_k\}$  is treated as a separate bounded surface and is independently parameterized. Let  $f : \mathbb{R}^3 \rightarrow \mathbb{R}^2$  be the mapping of each vertex  $v \in \mathcal{V}_k$  to a 2D point  $u$  on the UV plane. We propose to represent  $f$  using a *forward* mapping network  $MLP_f$  with learnable parameters  $\phi_f$ . First, the set of vertices  $\mathcal{V}_k$  for the given patch is passed to the diffusion block to get a global shape encoding  $\psi \in \mathbb{R}^{128}$ . Per-vertex input given to  $MLP_f$  is  $z \in \mathbb{R}^{131}$  ( $v$  concatenated with  $\psi$ ) and the output is  $u \in \mathbb{R}^2$  (UV coordinate), i.e.  $u = MLP_f(z)$ . Since we don't have corresponding ground truth UV coordinates, we resort to a self-supervised cycle-consistency loss. We employ another  $MLP_{f^{-1}}$  with learnable parameters  $\phi_{f^{-1}}$  to represent the *backward* mapping  $f^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ .  $MLP_{f^{-1}}$  takes



**Figure 3: Discretization-agnostic UV parameterization.**

$u$  as input and predicts its corresponding 3D position, which ideally should match with the input vertex position  $v$ . We enforce this consistency by minimizing the following cycle loss:

$$\mathcal{L}_{cycle} = \frac{1}{|\mathcal{V}_k|} \sum_{v \in \mathcal{V}_k} (v - MLP_{f^{-1}}(u))^2 \quad (3)$$

Note that due to presence of non-linear activation functions in  $MLP_f$  and  $MLP_{f^{-1}}$ , the condition  $\phi_f \cdot \phi_{f^{-1}} = I$  needs not to hold. Per-vertex prediction can be noisy, resulting in an irregular UV space. Conditioning the MLPs with the diffusion-based global shape-encoding  $\psi$  regularizes the UV prediction and improves the output of  $MLP_{f^{-1}}$ . We further add losses to enforce desired properties of surface parameterization, namely,  $L_{iso}$  provides isometric behaviour,  $L_{angle}$  preserves angles of the faces and  $L_{area}$  preserves face-area (neglecting uniform scaling). The final objective function for surface parameterization is given as:

$$\mathcal{L}_{uv} = \lambda_1 \mathcal{L}_{cycle} + \lambda_2 \mathcal{L}_{iso} + \lambda_3 \mathcal{L}_{angle} + \lambda_4 \mathcal{L}_{area}. \quad (4)$$

## 4 IMPLEMENTATION DETAILS

For PatchNet, we use DiffusionNet[9] architecture with 4-blocks, channel width of 128 and 64 eigenbasis vectors for spectral acceleration. We use ReLU activations at intermediate layers and softmax function after the final output layer. The surface parameterization module uses an 8 layer MLP with  $1.3 \times 10^6$  parameters for both forward and backward MLP with LeakyReLU activations in-between the layers and tanh at the final output layer. We use the PatchNet

	BFF		OptCuts		Ours	
	QCE↓	ASE↓	QCE↓	ASE↓	QCE↓	ASE↓
Laptop	1.046	2.052	1.045	2.005	1.196	2.420
Pliers	1.112	1.909	1.128	1.391	1.274	2.895
Rabbit	1.132	2.116	1.160	2.062	1.183	0.992
Scissors	1.156	1.456	1.122	1.276	1.261	2.728
Bird	2.130	1.103	1.129	1.928	1.262	1.996

**Table 1: Comparison of QCE and ASE metrics.**

loss weights  $\{\lambda_{cos}, \lambda_{geo}\} = [1.0, 1.0]$  and the parameterization loss weights  $\{\lambda_1, \lambda_2, \lambda_3, \lambda_4\} = [1.0, 1.0, 0.001, 0.001]$ . We use ADAM optimizer with learning rate of  $10^{-3}$ , and batch size of 1 for all our experiments. The networks are implemented in PyTorch Lightning, trained on a single RTX 2080Ti GPU. We use Xatlas[] to pack the individual patches into final UV atlas.

## 5 RESULTS & EVALUATION

We compute Quasi-Conformal Error (QCE) and Area Scale Error (ASE) on the final texture atlas for quantitative and qualitative evaluation. Please refer to supplementary for their description.

**Qualitative Comparison:** We compare with BFF[8] and OptCuts[5] and our framework performs on-par with these methods on varying geometrical shapes as shown in Figure 4.

**Quantitative Comparison:** In Table 1, we compare the values of QCE and ASE on with BFF[8] and Opt-Cuts[5] on few classes of SHREC [6] dataset. For each class, we train our network on 16 meshes and compute errors on 4 test sample meshes. Please note that, instead of purely object-centric learning, here we compare on a category-specific generalized network, and our performance is comparable to other object-centric methods. Such generalization can be attributed to intrinsic characterization encoded in diffusion features used in our surface parametrization module.

More importantly, we compare the computation time of these methods on a mesh with varying resolution and report significantly superior performance in Table 2. Specifically, we train our method on the decimated Stanford’s armadillo mesh with  $2K$  vertices, and compare our inference time with the computation time of other methods at the higher resolution.

**Discretization-agnostic Learning:** Figure 3 shows the discretization-agnostic learning capability of our framework. We train on a mesh with only  $\sim 3K$  vertices and then directly infer at high resolutions ( $\sim 35K$  and  $\sim 100K$  vertices). Please note that the error values for high-resolution meshes stay close to the low-resolution mesh as can be observed in the error plots.

Please refer to supplementary for a detailed ablative studies on role of global encoding and losses.

## 6 CONCLUSION

We proposed a novel self-supervised learning based framework for surface parameterization of bounded as well as unbounded surfaces. Our framework enables discretization-agnostic learning thereby significantly improving our inference time performance on high resolution meshes.

Resolution	BFF	OptCuts	Ours
30K	17.41 sec	> 10 min	2.92 sec
100K	61.04 sec	> 10 min	5.02 sec

Table 2: Comparison of computation time.

## REFERENCES

- [1] Noam Aigerman, Kunal Gupta, Vladimir G. Kim, Siddhartha Chaudhuri, Jun Saito, and Thibault Groueix. 2022. Neural Jacobian Fields: Learning Intrinsic Mappings of Arbitrary Meshes.
- [2] Zhiqin Chen, Kangxue Yin, and Sanja Fidler. 2022. AUV-Net: Learning Aligned UV Maps for Texture Transfer and Synthesis.
- [3] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. 2018. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation.
- [4] Bruno Lévy, Sylvain Petitjean, Nicolas Ray, and Jérôme Maillot. 2002. Least Squares Conformal Maps for Automatic Texture Atlas Generation. *ACM Trans. Graph.* 21, 3 (jul 2002), 362–371.
- [5] Minchen Li, Danny M. Kaufman, Vladimir G. Kim, Justin Solomon, and Alla Sheffer. 2018. OptCuts: Joint Optimization of Surface Cuts and Parameterization. *ACM Transactions on Graphics* 37, 6 (2018).
- [6] Zhouhui Lian, Afzal Godil, Benjamin Bustos, Mohamed Daoudi, Jeroen Hermans, Shun Kawamura, Yukinori Kurita, Guillaume Lavoué, Hien Nguyen, Ryutarou Ohbuchi, Yuki Ohkita, Yuya Ohishi, Fatih Porikli, Martin Reuter, Ivan Sipiran, Dirk Smeets, Paul Suetens, Hedi Tabia, and Dirk Vandermeulen. 2011. SHREC ’11 Track: Shape Retrieval on Non-rigid 3D Watertight Meshes. , 79-88 pages.
- [7] Pedro V. Sander, John Snyder, Steven J. Gortler, and Hugues Hoppe. 2001. Texture Mapping Progressive Meshes. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’01)*. Association for Computing Machinery, New York, NY, USA, 409–416.
- [8] Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* 37, 1, Article 5 (dec 2017), 14 pages.
- [9] Nicholas Sharp, Souhaib Attaki, Keenan Crane, and Maks Ovsjanikov. 2020. DiffusionNet: Discretization Agnostic Learning on Surfaces.
- [10] He Wang, Kirill A. Sidorov, Peter Sandlands, and Taku Komura. 2013. Harmonic Parameterization by Electrostatics. *ACM Trans. Graph.* 32, 5, Article 155 (oct 2013), 12 pages.
- [11] Francis Williams, Teseo Schneider, Claudio Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. 2018. Deep Geometric Prior for Surface Reconstruction.

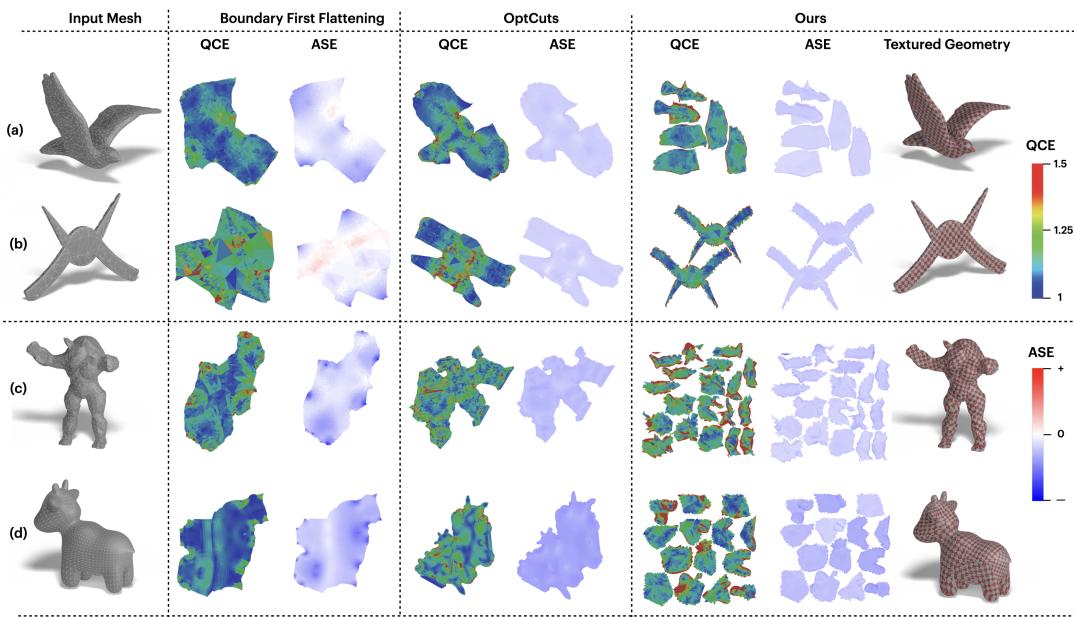


Figure 4: Comparison of error plots for QCE and ASE with other methods.