Shape Reconstruction by Learning Differentiable Surface Representations
Supplementary Material

Jan Bednařík Shaifali Parashar Erhan Gündoğdu Mathieu Salzmann Pascal Fua

CVLab, EPFL, Switzerland

{firstname.lastname}@epfl.ch

We provide more details on the training and evaluation of Single-View 3D Shape Reconstruction (**SVR**) on the TDS dataset in Section 1; we show additional results for the **SVR** task on the synthetic ShapeNet dataset in Section 2; we perform an ablation study of the components of the deformation loss $\mathcal{L}_{\text{def}}$ in Section 3; we analyze thoroughly the deformation properties of the predicted patches in Section 4; and finally we compare the precision of analytically and approximately computed normals in Section 5.

1. Training and Evaluation of SVR on TDS

As described in [1], the TDS dataset was recorded as a set of video sequences. Therefore, it is necessary to split the dataset properly into training, validation and testing subsets so that the testing samples do not leak into the training set. Furthermore, to follow the evaluation protocol introduced in [1], data preprocessing and postprocessing steps are needed.

1.1. Dataset Splits

As described in Section 5.2 of the paper, we selected two object categories for which the most data samples are available, a piece of a cloth and a T-Shirt. We use 85% of the samples for training, 5% for validation, and 10% for testing. For the cloth object, one full video sequence (`Lr_top_edge_3`) was removed from the training set, and split into validation (the first 500 frames) and testing (remaining 529 frames). There are not enough T-shirt sequences to do the same. We therefore split one sequence (`Lr_front`) and exploited the first 250 frames for training, the following 98 frames for validation and the remaining 200 for testing.

1.2. Data Preprocessing and Postprocessing

To evaluate the reconstruction quality of AN and **OURS** for SVR on the TDS dataset, some preprocessing and postprocessing steps are necessary.

The TDS dataset samples are centered around point $\mathbf{c} = \begin{bmatrix} 0 & 0 & 1.1 \end{bmatrix}^\top$, which is out of reach of the activation function *tanh* that AN uses in its last layer. Therefore, we translated all the data samples by $-\mathbf{c}$.

In [1], which introduced the TDS dataset, the authors align the predicted sample with its GT using Procrustes alignment [5] before evaluating the reconstruction quality. Since we do not have correspondences between the GT and predicted points, we used the Iterative Closest Point (ICP) [3] algorithm to align the two point clouds. This allows rigid body transformations only.

2. Single-view Reconstruction on ShapeNet

For the sake of completeness, we ran the experiments for the **SVR** task not only on the real-world TDS dataset, as presented in Section 5.6 of the paper, but also on the synthetic ShapeNet dataset. We trained both AN and **OURS** using 25 patches, 2500 points randomly sampled from the GT, and the same number is predicted by the models. As before, we trained both models separately on the object categories airplane, chair, car, couch and cellphone, and jointly on all the categories. We used the same synthetic renderings as in [4].

We report our results in Table 1. Similarly to the **PCAE** on ShapeNet experiments, **OURS** delivers comparable CHD precision but significantly higher quality in terms of the predicted normals, number of collapsed patches and amount of overlap, as is further demonstrated in Figure 1.

3. Deformation Loss Term Ablation Study

We have seen that the deformation loss term defined as $\mathcal{L}_{\text{def}} = \alpha_E \mathcal{L}_E + \alpha_G \mathcal{L}_G + \alpha_{\text{sk}} \mathcal{L}_{\text{sk}} + \alpha_{\text{str}} \mathcal{L}_{\text{str}}$ prevents the predicted patches from collapsing. Here we perform an ablation study of the individual components $\mathcal{L}_E, \mathcal{L}_G, \mathcal{L}_{\text{sk}}$ and $\mathcal{L}_{\text{str}}$ and show how each of them affects the resulting deformations that the patches undergo.

We carry out all the experiments on **SVR** using the cloth object from the TDS dataset and the same training/validation/testing splits as before. We employ **OURS** and the original loss function $\mathcal{L} = \mathcal{L}_{\text{CHD}} + \alpha_{\text{def}} \mathcal{L}_{\text{def}} + \alpha_{\text{ol}} \mathcal{L}_{\text{ol}}$ (with $\alpha_{\text{def}} = 0.001$ and $\alpha_{\text{ol}} = 0.1$, as before).

Table 1. **OURS vs AN trained for SVR on ShapeNet.** Both models were trained individually on 5 ShapeNet categories (plane, chair, car, couch, cellphone) and jointly on all of them (all). While CHD is comparable for both methods, **OURS** delivers better normals and lower patch overlap.

obj.	method	CHD	m_{ae}	$m_{\text{olap}}^{(0.01)}$	$m_{\text{olap}}^{(0.05)}$	$m_{\text{olap}}^{(0.1)}$	m_{col}
plane	AN	**2.43**	27.12	9.43	16.34	18.93	0.104
	OURS	2.76	**24.36**	**4.26**	**8.99**	**12.02**	**0.000**
chair	AN	8.65	41.54	8.30	13.32	16.00	4.320
	OURS	**7.67**	**41.17**	**2.77**	**5.99**	**8.41**	**0.000**
car	AN	10.40	40.09	4.10	8.63	11.60	0.010
	OURS	**4.36**	**22.76**	**2.20**	**4.51**	**6.76**	**0.000**
couch	AN	**6.33**	28.73	6.69	13.16	17.36	0.576
	OURS	6.64	**26.01**	**3.04**	**6.56**	**9.74**	**0.000**
cellphone	AN	**3.90**	15.20	7.60	16.73	20.01	0.221
	OURS	4.07	**13.73**	**2.93**	**6.53**	**9.13**	**0.000**
all	AN	10.09	37.92	9.56	15.9	18.43	3.570
	OURS	**9.42**	**34.51**	**3.6**	**7.7**	**10.44**	**0.000**

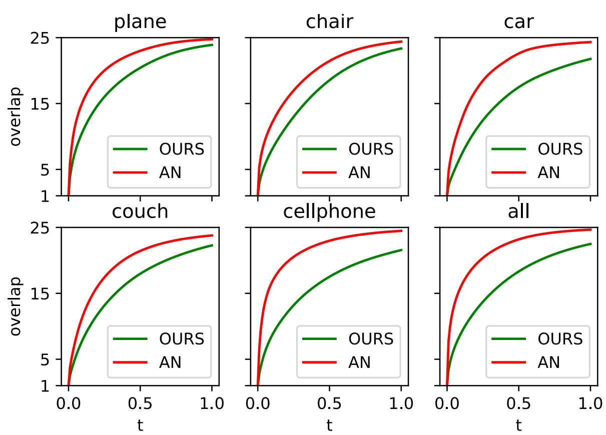


Figure 1. **Patch overlap for OURS and AN** trained for **SVR** on the ShapeNet dataset. We plot $m_{\text{col}}^{(t)}$ as a function of t.

Table 2. **Configurations of the ablation study.** The components of the $\mathcal{L}_{\text{def}}$ loss are either turned on or off using their corresponding hyperparameters.

Experiment	α_E	α_G	α_{sk}	α_{str}
free	0	0	0	0
no collapse	**1**	**1**	0	0
no skew	**1**	**1**	**1**	0
no stretch	**1**	**1**	0	**1**
full	**1**	**1**	**1**	**1**

To identify the contributions of the components of $\mathcal{L}_{\text{def}}$, we switch them on or off by setting their corresponding hyperparameters $\alpha_E, \alpha_G, \alpha_{\text{sk}}$ and α_{str} to either 0 or 1, and for each configuration we train **OURS** from scratch until convergence. We list the individual configurations in Table 2.

Fig 2 depicts the qualitative results for all 5 experiments on 5 randomly selected test samples. We discuss the individual cases below:

Free: The $\mathcal{L}_{\text{def}}$ term is completely switched off, which results in high distortion mappings and many 0D point collapses and 1D line collapses.

No collapse: We only turn on the components $\mathcal{L}_E$ and $\mathcal{L}_G$, which by design prevent any collapse and encourage the amount of stretching along either of the axes to be uniform across the whole area of a patch. However, the patches still tend to undergo significant stretch along one axis (light red patch) and/or display a high amount of skew (light blue and light orange patch).

No skew: Adding the $\mathcal{L}_{\text{sk}}$ component to $\mathcal{L}_E$ and $\mathcal{L}_G$ (but leaving out $\mathcal{L}_{\text{str}}$) prevents the patches from skewing, resulting in strictly orthogonal rectangular shapes. However, the patches tend to stretch along one axis (light blue and light red patch). If skew is needed to model the local geometry, the patches stay rectangular and rotate instead (dark blue patch).

No stretching: Adding the $\mathcal{L}_{\text{str}}$ component to $\mathcal{L}_E$ and $\mathcal{L}_G$ (but leaving out $\mathcal{L}_{\text{sk}}$) results in a configuration where the patches prefer to undergo severe skew (cyan and dark green patch), but preserve their edge lengths.

All: Using the full $\mathcal{L}_{\text{def}}$ term, with all its components turned on, results in strictly square patches with minimum skew or stretching.

4. Distortion Analysis

In the previous section, we showed that the individual types of deformations that the patches may undergo—-stretching, skewing and in extreme cases collapse—-can be effectively controlled by suitable combination of the components of the loss term $\mathcal{L}_{\text{def}}$. In this section, we present a different perspective on the distortions which the patches undergo. We focus on a texture mapping task where we show that using the $\mathcal{L}_{\text{def}}$ to train a network helps learn mappings with much less distortion. Furthermore, we inspect each patch individually and analyze how the distortion distributes over its area.

4.1. Regularity of the Patches

We experiment on **PCAE** using the ShapeNet dataset, on which we train AN and **OURS** as in Section 5.6., i.e., using the full loss function $\mathcal{L} = \mathcal{L}_{\text{CHD}} + \alpha_{\text{def}}\mathcal{L}_{\text{def}} + \alpha_{\text{ol}}\mathcal{L}_{\text{ol}}$ with $\alpha_{\text{def}} = 0.001$ and $\alpha_{\text{ol}} = 0.1$ and with $\alpha_E = \alpha_G = \alpha_{\text{sk}} = 1, \alpha_{\text{str}} = 0$. Furthermore, we train one more model, ***OURS-strict***, which is the same as **OURS** except that we set $\alpha_{\text{str}} = 1$. In other words, **OURS-strict** uses the full $\mathcal{L}_{\text{def}}$ term where even stretching is penalized.

To put things in perspective, when considering the ablation study of Section 3, AN corresponds to the *free* configuration, **OURS** to the *no skew* configuration and **OURS-strict** to the *full* configuration.

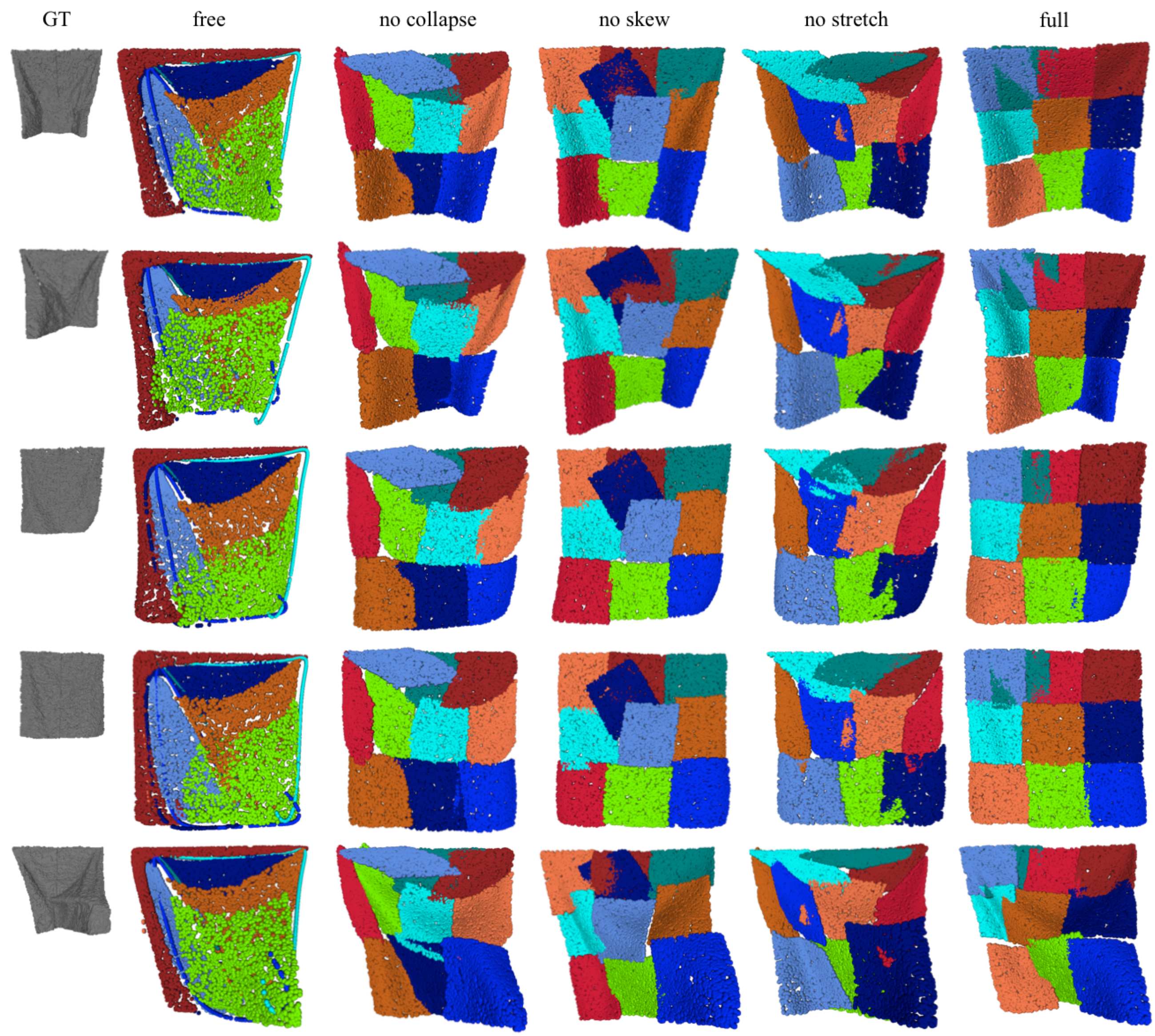


Figure 2. **Qualitative results of the ablation study.** Each row depicts a randomly selected sample from the test set and each column corresponds to one experimental configuration. See the text for more details.

Figs. 3 and 4 depict qualitative reconstruction results for various objects from ShapeNet, where we map a regular checkerboard pattern texture to every patch. Note that while AN produces severely distorted patches, **OURS** introduce a truly regular pattern elongated along one axis (since stretching is not penalized) and **OURS-strict** delivers nearly isometric patches.

Note, however, the trade-off between the shape precision and regularity of the mapping (i.e., the amount of distortion). When considering the two extremes, AN delivers much higher precision than **OURS-strict**. On the other hand, **OURS** appears to be the best choice as it brings the best of both worlds — it delivers high precision reconstructions while maintaining very low distortion mappings.

4.2. Intra-patch Distortions

To obtain more detailed insights into how the patches deform, we randomly select a test data sample from the ShapeNet plane object category and analyze the individual types of deformations that each patch predicted by AN and **OURS** undergoes. We are interested in 4 quantities $D_E, D_G, D_{\text{sk}}, D_{\text{str}}$, which are proportional to the components $\mathcal{L}_E, \mathcal{L}_G, \mathcal{L}_{\text{sk}}, \mathcal{L}_{\text{str}}$ of the deformation loss term $\mathcal{L}_{\text{def}}$.

Fig. 5 depicts the spatial distribution of the values coming from all these 4 quantities over all 25 patches predicted by AN and **OURS**. Note that while the patches predicted

Figure 3. **Qualitative results of ShapeNet objects plane and chair reconstruction by AN, OURS and OURS-strict.**

by AN are subject to all the deformation types and yield extremely high values, which change abruptly throughout each predicted patch, the patches predicted by **OURS** undergo very low distortions, which are mostly constant throughout the patches.

The exception is the D_{str} quantity, which has high values for all the patches. This is due to the fact that **OURS** does not penalize stretching. This can be seen in Fig. 6, which depicts the distribution of the values of the terms E and G coming from the metrics tensor $g = \begin{bmatrix} E & F \\ F & G \end{bmatrix}$ across all the patches predicted by **OURS**. All the patches corresponding to E yield high values while the ones corresponding to G low values. This means that the patches prefer to stretch only along the u-axis in the 2D parametric UV space (recall that $E = \left\| \frac{\partial f_{\mathbf{w}}}{\partial u} \right\|^2$).

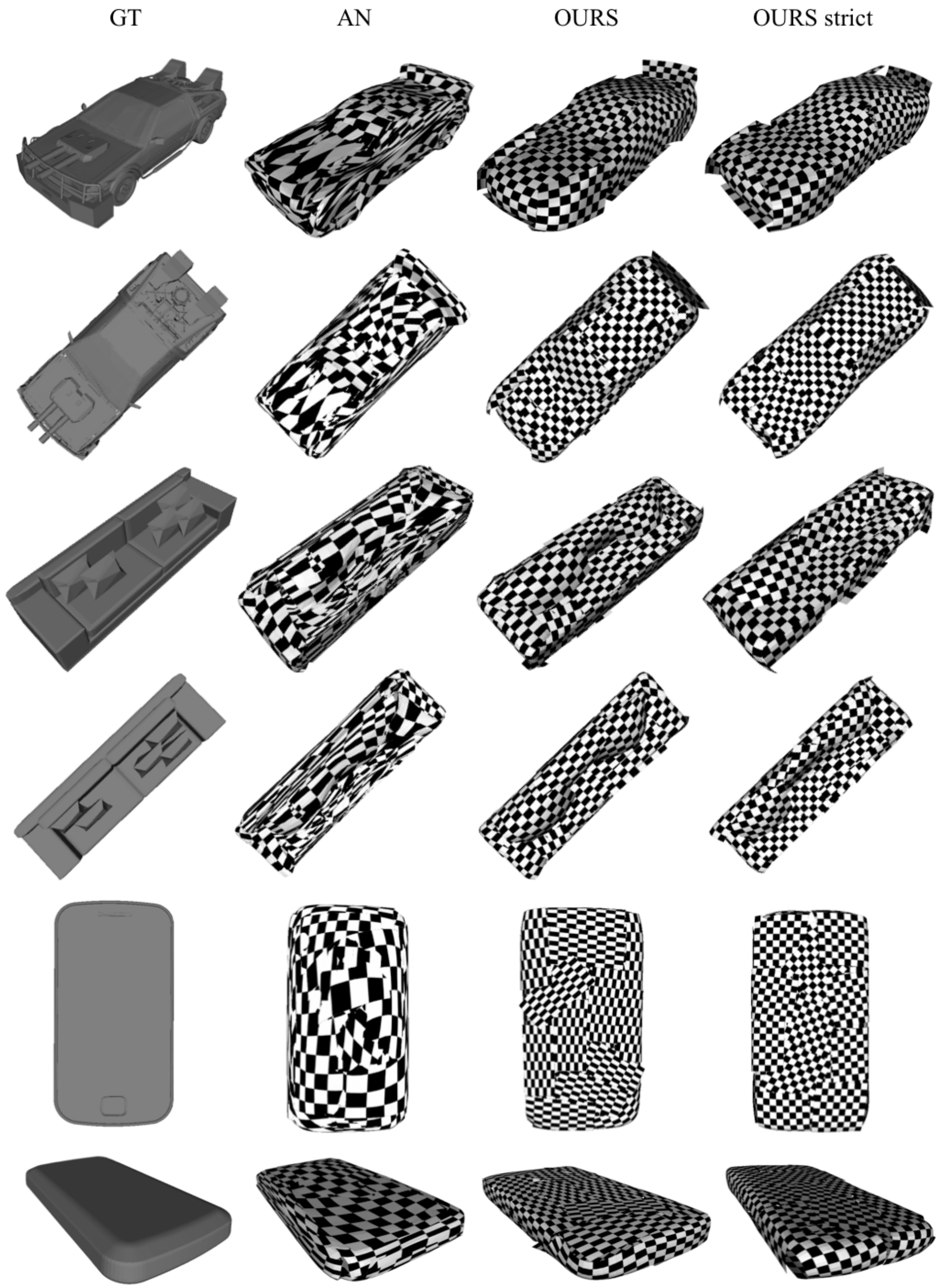
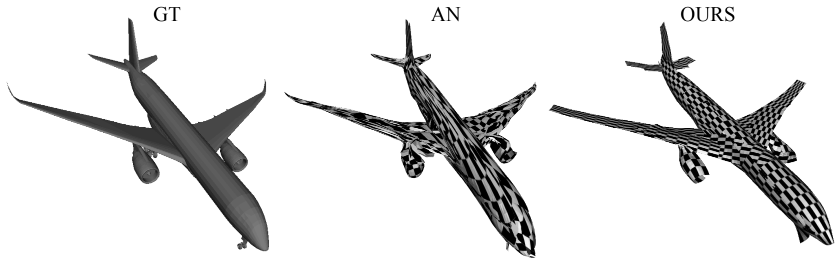


Figure 4. **Qualitative results of ShapeNet objects car, couch and chair reconstruction by AN, OURS and OURS-strict.**

5. Approximate Normal Estimation

As discussed in Section 5.6, an alternative to exact analytical computation of the per-point normals consists of estimating the normals approximately, e.g., using the popular covariance-based method [2], which computes a covariance matrix on a point neighborhood, performs an eigendecomposition of this matrix and takes the eigenvector corresponding to the smallest eigenvalue as normal estimate.

The point neighborhood can be represented either as a set of k nearest points or as a set of points lying within a given distance. These two methods rely on the hyperparameters $k \in \mathbb{N}$ and $r \in \mathbb{R}$, respectively, and will be referred to as COV-kNN and COV-radius.



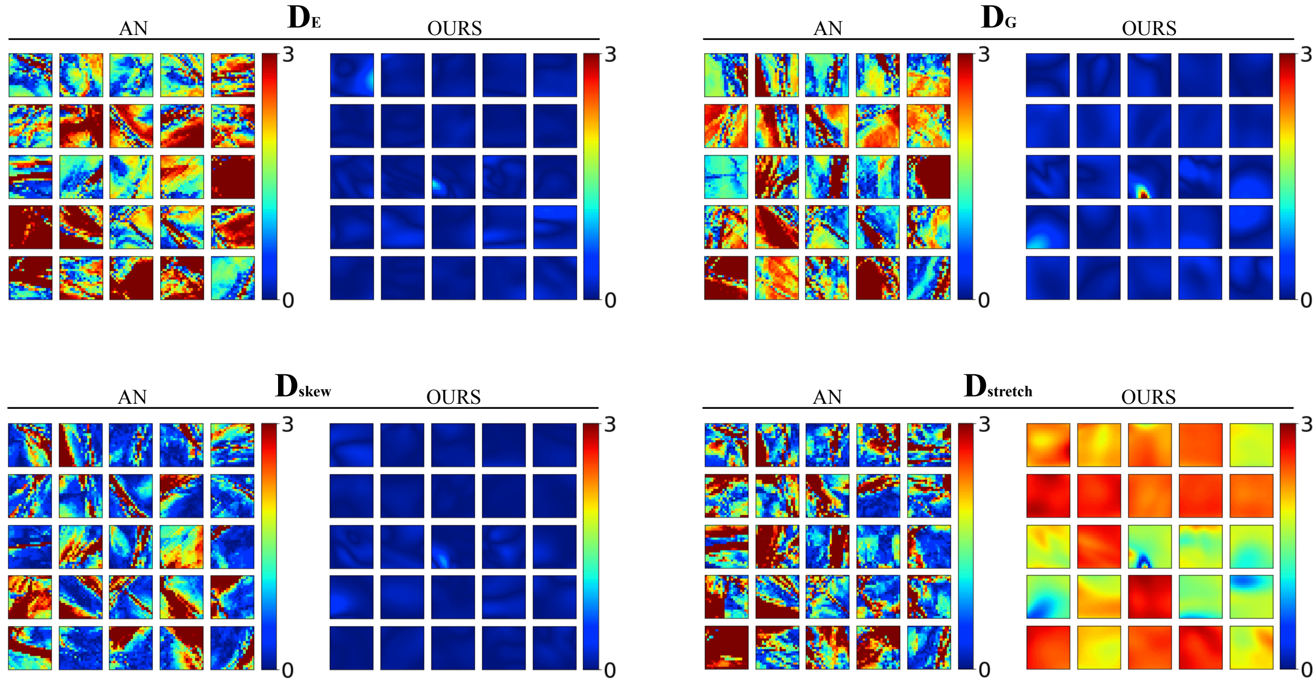


Figure 5. **Spatial distribution of the quantities $D_E, D_G, D_{\text{sk}}, D_{\text{str}}$ across all the 25 patches predicted by AN and OURS for a single test data sample from ShapeNet dataset.**

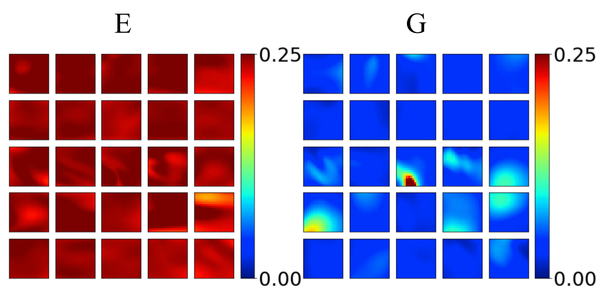


Figure 6. **Spatial distribution of metric tensor g quantities E and G over all the 25 patches predicted by OURS on a single data sample from ShapeNet dataset.**

For fair and complete comparison, we use the covariance method [2] to compute the approximate normals from the point clouds predicted by AN in both the **PCAE** on ShapeNet and **SVR** on TDS experiments. Since the selection of the neighborhood method and the correct value for the corresponding hyperparameter strongly affects the precision of the normals estimate, we ran a grid search on a validation set separately for all the object categories in both experiments. The hyperparameter values corresponding to the lowest validation error are reported in Table 3 and used for subsequent evaluations.

Table 4 provides the resulting angular errors m_{ae} for both methods, COV-kNN and COV-radius, ran on the predictions of AN and compares them to the m_{ae} evaluated on the analytically computed normals on the prediction of both AN and **OURS** (which are reported in Tables 3 and 4 in the main paper). **OURS** outperforms COV-kNN in all experiments and COV-radius in all but one. This further motivates the use of our framework, which by allowing for analytical normal computation, not only yields higher precision but also avoids the necessity of tedious and costly hyperparameter search and the need for an extra post-processing step.

Table 3. Values of the hyperparameters k and r corresponding to the lowest m_{ae} error found on a validation set separately for each object category.

	PCAE on ShapeNet					**SVR on TDS**	
method	**plane**	**chair**	**car**	**couch**	**cellphone**	**cloth**	**tshirt**
COV-kNN (k)	40	50	20	20	20	100	100
COV-radius (r)	0.1	0.25	0.15	0.15	0.2	0.075	0.075

Table 4. Comparison of the m_{ae} metric evaluated for every object category using the approximate normals predicted by the COV-kNN and COV-radius methods using the hyperparameters listed in Table 3 and using the analytically computed normals (AN and **OURS**).

	PCAE on ShapeNet					**SVR on TDS**	
method	**plane**	**chair**	**car**	**couch**	**cell.**	**cloth**	**tshirt**
COV-kNN	19.01	25.95	18.29	19.59	16.70	46.12	39.84
COV-radius	19.28	27.00	20.57	22.23	16.86	22.78	**19.79**
AN	21.26	24.49	18.08	16.83	10.29	47.42	42.12
OURS	**17.90**	**23.06**	**17.75**	**14.90**	**9.64**	**20.06**	20.52

References

[1] J. Bednarík, M. Salzmann, and P. Fua. Learning to Reconstruct Texture-Less Deformable Surfaces. In *International Conference on 3D Vision*, 2018. 1

[2] J. Berkmann and T. Caelli. Computation of surface geometry and segmentation using covariance techniques. *PAMI*, 1994. 5, 6

[3] P. Besl and N. Mckay. A Method for Registration of 3D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992. 1

[4] T. Groueix, M. Fisher, V. Kim, B. Russell, and M. Aubry. Atlasnet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Conference on Computer Vision and Pattern Recognition*, 2018. 1

[5] M. B. Stegmann and D. D. Gomez. A Brief Introduction to Statistical Shape Analysis. Technical report, University of Denmark, DTU, 2002. 1