LIST: Learning Implicitly from Spatial Transformers for Single-View 3D Reconstruction - Supplementary Material

Mohammad Samiul Arshad and William J. Beksi
Department of Computer Science and Engineering
The University of Texas at Arlington, Arlington, TX, USA
mohammadsamiul.arshad@mavs.uta.edu, william.beksi@uta.edu

This document includes supplementary material for the paper **LIST: Learning Implicitly from Spatial Transformers for Single-View 3D Reconstruction**. In Fig. 1, we show a qualitative comparison of occluded surface reconstruction. Examples of failed reconstructions are displayed in Fig. 2. More qualitative comparisons between LIST and the baseline models using the ShapeNet dataset are highlighted in Fig. 3. The results of LIST reconstructions using distinct views of the same object are provided in Fig. 4, Fig. 5, and Fig. 6. Finally, a video presents 360-degree views of the reconstructions.

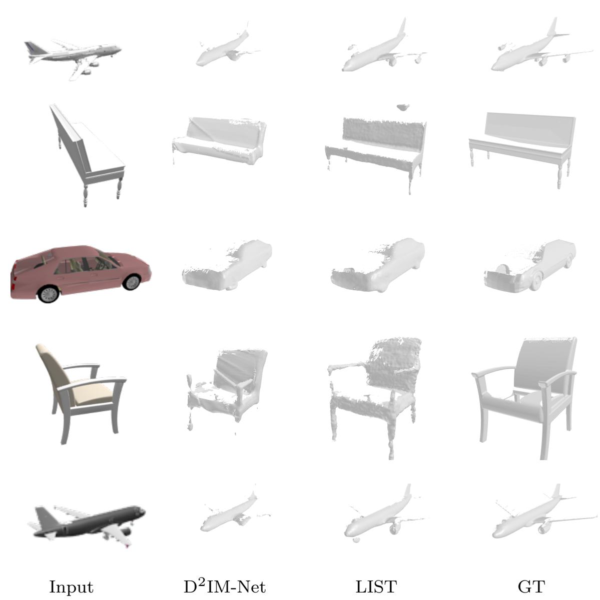

Fig. 1: A qualitative comparison between LIST and the baseline models on occluded surface reconstruction using the ShapeNet dataset. GT denotes the ground-truth objects.

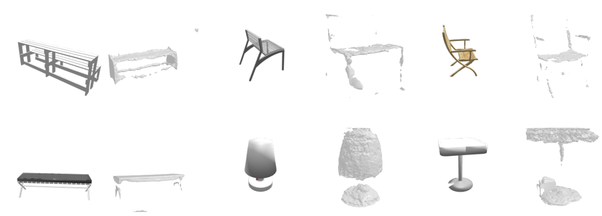

Fig. 2: Examples of failed LIST reconstructions.

1. Evaluation Metrics

Chamfer Distance (CD): The chamfer distance (CD) between two meshes is defined as

$$\mathrm{CD}(y_{\mathrm{GT}}, y_{\mathrm{pred}}) = \sum_{a \in y_{pred}} \min_{b \in y_{gt}} ||a-b|| + \sum_{b \in y_{gt}} \min_{b \in y_{pred}} ||b-a||, \tag{1}$$

where, y_{GT} and y_{pred} are two point clouds extracted from the surface of the ground-truth and reconstructed object, respectively.

Intersection over Union (IoU): The volumetric intersection over union (IoU) is defined as the quotient of the volume of the intersection of two meshes and the volume of their union,

$$\mathrm{IoU}(\mathcal{M}_{\mathrm{pred}}, \mathcal{M}_{\mathrm{GT}}) = \frac{|\mathcal{M}_{\mathrm{pred}} \cap \mathcal{M}_{\mathrm{GT}}|}{|\mathcal{M}_{\mathrm{pred}} \cup \mathcal{M}_{\mathrm{GT}}|}. \tag{2}$$

F-score: The F-score, proposed in [6] as a comprehensive scoring metric for single-view reconstruction, combines precision and recall to quantify the overall reconstruction quality. Concretely, the F-score at a distance threshold d is given by

$$F(d) = \frac{2 \cdot P(d) \cdot R(d)}{P(d) + R(d)},$$

where $P(\cdot)$ and $R(\cdot)$ represents the precision and recall, respectively. Precision quantifies the accuracy while recall assesses the completeness of the reconstruction. For the ground-truth y_{gt} and reconstructed point cloud y_{pred}, the precision of an outcome at d can be calculated as

$$P(d) = \sum_{i \in y_{\text{pred}}} [\min_{j \in y_{\text{GT}}} ||i - j|| < d].$$

Similarly, the recall for a given d may be computed as

$$R(d) = \sum_{j \in y_{\text{GT}}} [\min_{i \in y_{\text{pred}}} ||j - i|| < d].$$

To evaluate the reconstructions between LIST and the baselines we used $d = 1\%$.

2. Data Preparation

To prepare the ground truth, first the target shape was normalized into a unit cube and 50k points were sampled from the surface of the object. The query points were prepared by adding random Gaussian noise (n) to the surface points. Specifically,

$$Q_j = Q_S + n \mid n \in \mathcal{N}(0, P), \tag{3}$$

where Q_S are the sampled points and $P \in \mathbb{R}^{3\times3}$ is a diagonal covariance matrix with entries $P_{i,i} = \rho$. We empirically found that 45% of the points at $\rho = 0.003$, 44% of the points at $\rho = 0.01$, and 10% of the points at $\rho = 0.07$ achieved the best results.

3. Implementation, Training, and Inference Details

3.1. Implementation Overview

LIST was implemented using the PyTorch [4] library. To optimize the model, the Adam [3] optimizer was used with coefficients $(0.9, 0.99)$, learning rate 10^{-4}, and weight decay 10^{-5}. A pretrained ResNet [2] was employed as the image encoder in Ω and Π. We closely followed the generator in [5] to implement the coarse predictor in Ω with tree-structured convolutions. However, we empirically found that the degree values $(2, 2, 2, 2, 2, 2, 64)$ provided a better coarse estimation in our settings. We set the coarse point cloud density to $N = 4000$, and the occupancy grid resolution to $M = 128$. To generate a probabilistic occupancy with the same grid, we utilized a shallow convolutional network Γ.

We define Ξ as a convolutional neural network to map the probabilistic occupancy grid into a high-dimensional latent space. To extract the global query features and localize the query points, we used a fully-connected neural network Θ. The global image features are fused with the global query features on the 3rd layer of Θ. During training, we augment the images with random color jitter, and normalize the values to $[0, 1]$. To improve the estimation accuracy, we scale the ground-truth and predicted SDF values by 10.0. Following [1], we disentangled the query points by scaling with 2.0 and swapping the 1st and 3rd axis to extract query features from the coarse prediction. At test time, we extract the query points from a grid in the range $[-0.5, 0.5]$ with resolution 128^3.

4. Training and Inference Time

To train LIST it takes ≈ 1 s to make a forward pass on an Intel i7 machine with an NVIDIA GeForce GTX 1080Ti GPU. To fully pass through the Pix3D and ShapeNet datasets, it takes approximately 35 and 50 min, respectively. Our training process involved using 4 1080Ti GPUs for 100 epochs with a batch size of 8. To reconstruct the mesh of a single object from a corresponding RGB image, it takes $\approx$ 7 s on average at a grid resolution of 128^3.

Fig. 3: A qualitative comparison between LIST and the baseline models using the ShapeNet dataset. Our model recovers *significantly better* topological and geometric structure, and the reconstruction is not tainted by the input-view direction. GT denotes the ground-truth objects.

Fig. 4: Qualitative results of LIST reconstructions using distinct views of the same object. Odd rows represent the input and even rows represent the reconstructions.

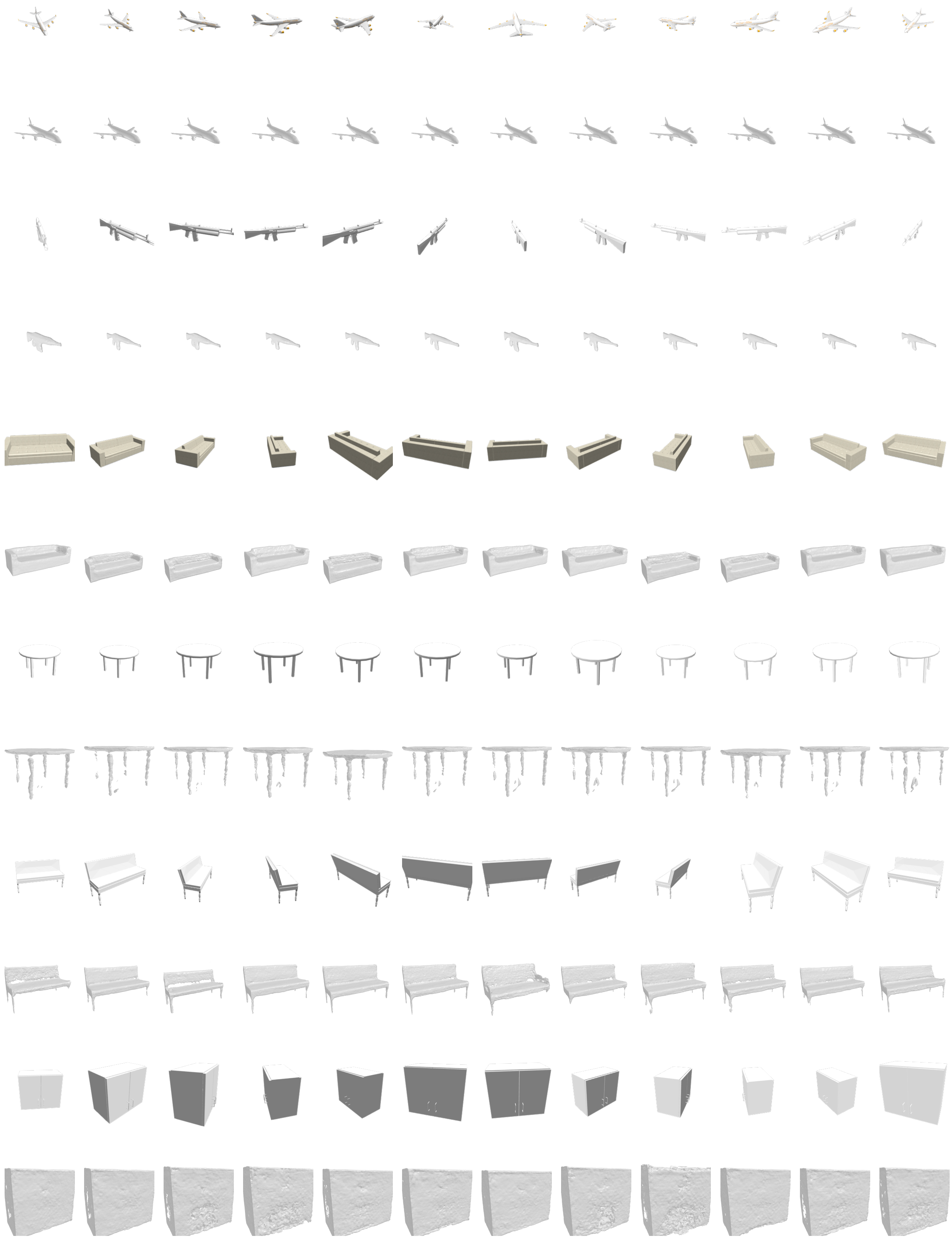

Fig. 5: Qualitative results of LIST reconstructions using distinct views of the same object. Odd rows represent the input and even rows represent the reconstructions.

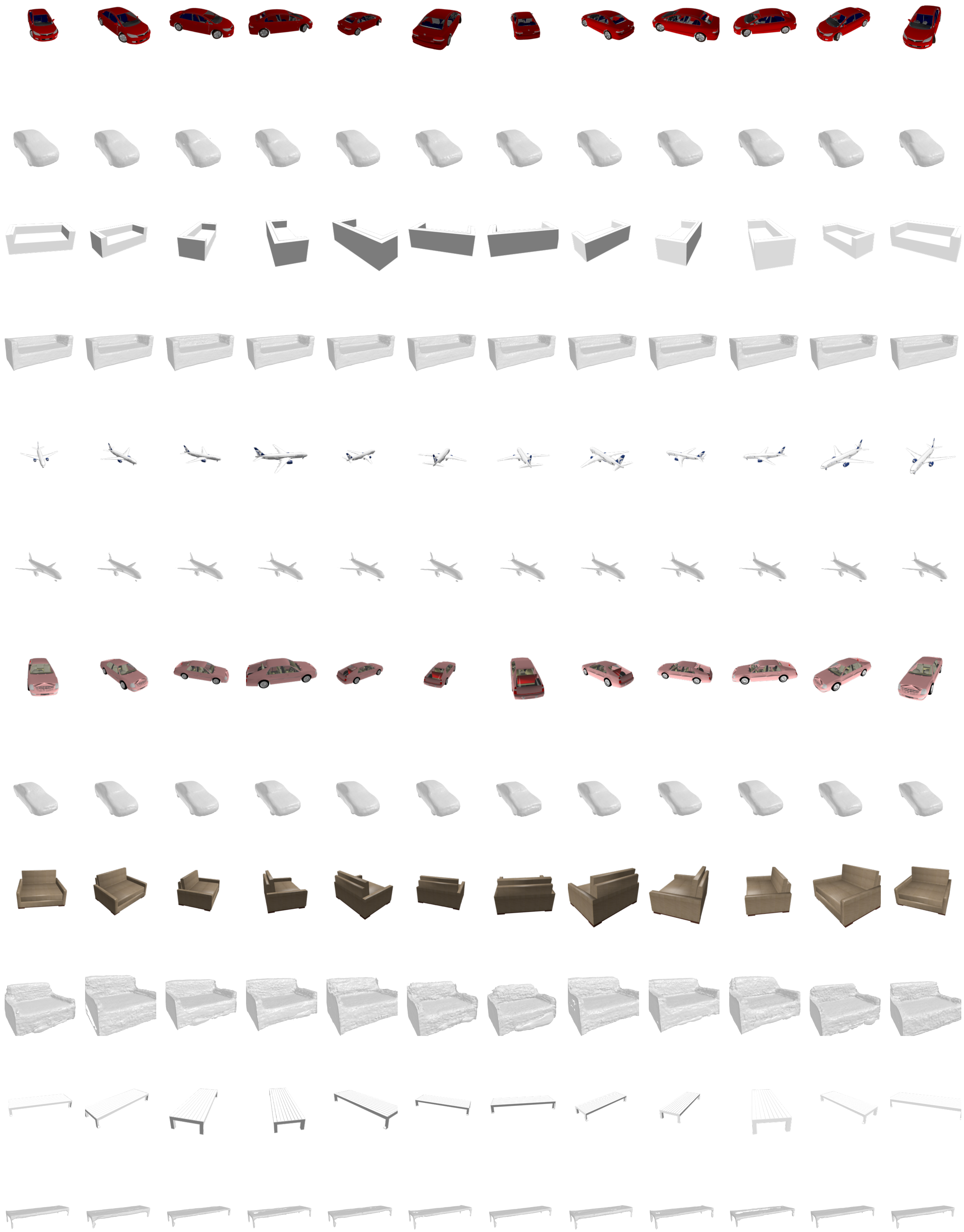

Fig. 6: Qualitative results of LIST reconstructions using distinct views of the same object. Odd rows represent the input and even rows represent the reconstructions.

References

[1] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[3] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[4] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems*, volume 32, pages 8024–8035, 2019. 2

[5] Dong Wook Shu, Sung Woo Park, and Junseok Kwon. 3d point cloud generative adversarial network based on tree structured graph convolutions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3859–3868, 2019. 2

[6] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019. 1