Multiresolution Deep Implicit Functions for 3D Shape Representation (Supplementary Material)

Zhang Chen[1,2,*] Yinda Zhang[1] Kyle Genova[1] Sean Fanello[1] Sofien Bouaziz[1]
Christian Häne[1] Ruofei Du[1] Cem Keskin[1] Thomas Funkhouser[1] Danhang Tang[1]
[1] Google [2] ShanghaiTech University

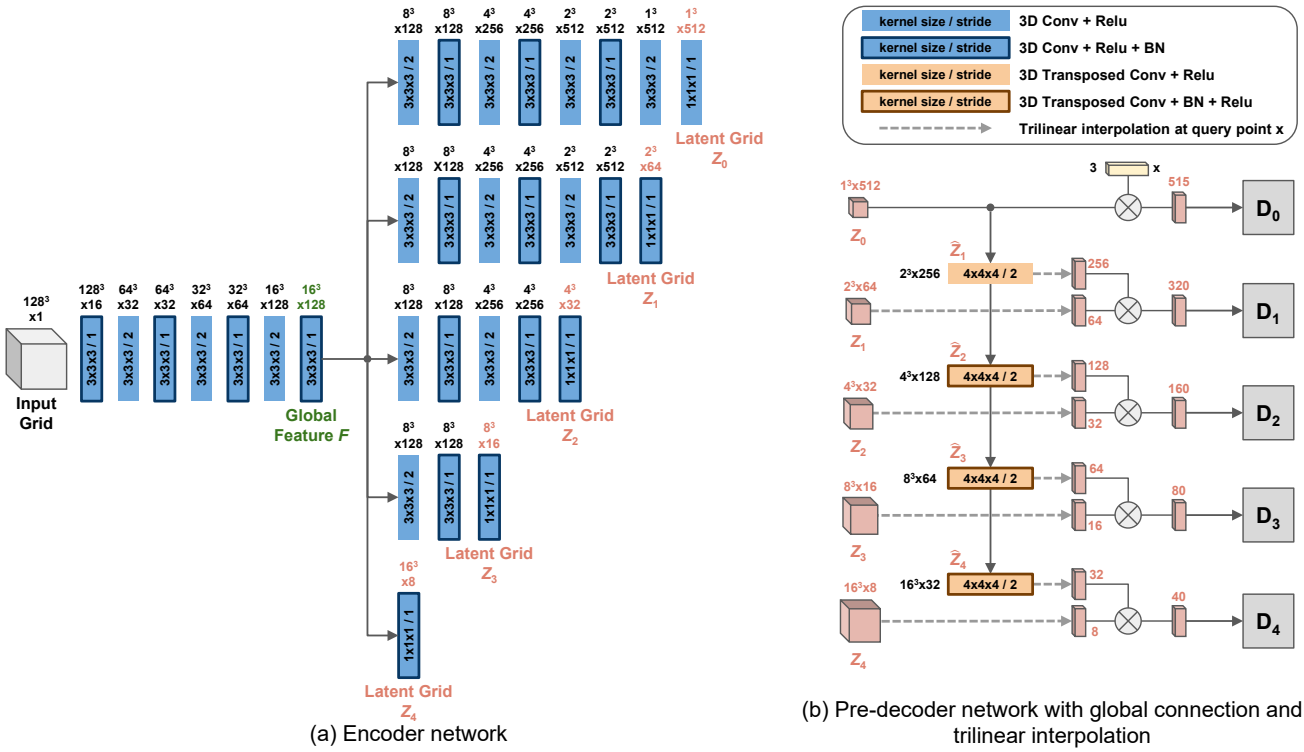
6. Supplementary Material

6.1. Implementation Details

Detailed network architecture. Figure 13 shows the detailed architecture of our network. On the left, Figure 13 (a) is the encoder network that is used in training and encoder-decoder inference. It takes 3D grid as input and outputs the latent grid Z_n of each level. For the voxel super-resolution experiment (Section 4.5), since the input is only 32^3, we accordingly remove the first 4 convolution layers along with their activation and normalization layers.

On the right, Figure 13 (b) is the pre-decoder network. With latent grids $\{Z_n\}$ as input, it includes global connection and trilinear interpolation. The global connection consists of 3D transposed convolution layers to propagate global context from level 0 to other levels. Trilinear interpolation is utilized to obtain the latent codes at each query point, which are then fed into the decoders at each level. For level 0, the 3D position of query point is also fed into the decoder. For the decoder modules, we use the same IM-Net [3] architecture for each level, with the only difference in the input dimension.

*Work done while the author was an intern at Google.



(a) Encoder network

(b) Pre-decoder network with global connection and trilinear interpolation

Figure 13: Detailed architecture of our network.

Hyperparameters. We implement our method in TensorFlow. During training, we set batch size as 8 and train our network end-to-end. We use Adam as optimizer, with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and a learning rate of $1e{-}4$. The latent grid dropout rate is set as 0.5 for the models that need to carry out decoder-only latent optimization while it is set as 0 for the models that only run encoder-decoder inference (*e.g.*, the models for point cloud completion and voxel super-resolution).

During decoder-only latent optimization, we optimize over $Z_n, n = 0, 1, ..., 4$ and keep other parameters fixed. We use Adam with the same configuration of β_1, β_2 as training, but at a higher learning rate of $1e{-}2$ to accelerate convergence. In all our experiments, we only run latent optimization for 1000 steps. For each step during auto-encoding, we randomly draw 2048 points. For each step during shape completion, we randomly draw 2048 camera-observable points, along with 1024 occluded points for the *global consistency loss*.

Experiment details. For the training data, we use the watertight ShapeNet meshes from OccNet [22] and normalize into bounding box with side length 1.28. We also truncate SDF values at 0.05.

For the auto-encoding experiment (Section 4.3), as mentioned in the paper, IF-Net [4] originally uses high-resolution latent grids which contain more parameters than the input grid. We therefore constrain IF-Net to only use latent grids with dimensions: $[8^3 \times 22, 16^3 \times 8]$. The resulting total number of parameters in the latent grids is the same as MDIF .

For the point cloud completion (Section 4.4) and voxel super-resolution (Section 4.5) experiments, unlike auto-encoding, the goal is to infer missing data rather than learn a compact latent space. Therefore, in these experiments, we use the original implementation of IF-Net which exploits high-resolution latent grids. Similarly, for MDIF in these experiments, we additionally interpolate features at query points from high-resolution feature grids and feed into the decoders.

6.2. Encoder-Decoder vs. Decoder-Only Inference

In Figure 14, we show qualitative auto-encoding results of MDIF using encoder-decoder inference and decoder-only latent optimization. Compared with encoder-decoder inference, decoder-only latent optimization already produces more accurate reconstruction with only 200 optimization steps. More steps further lower the error.

6.3. Illustration of Ablation Baselines

In Figure 15, we illustrate the baselines that we ablate in Table 1 and Table 2.

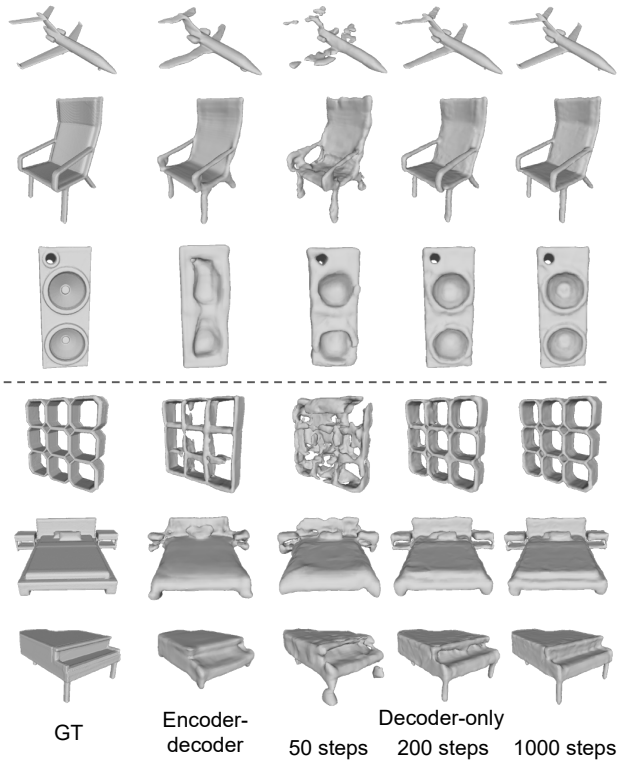


Figure 14: **Encoder-decoder vs. decoder-only inference.** Auto-encoding results of MDIF under encoder-decoder inference and decoder-only latent optimization. Top 3 rows: objects in 3D-R^{2}N^2 test split. Bottom 3 rows: objects in unseen categories.

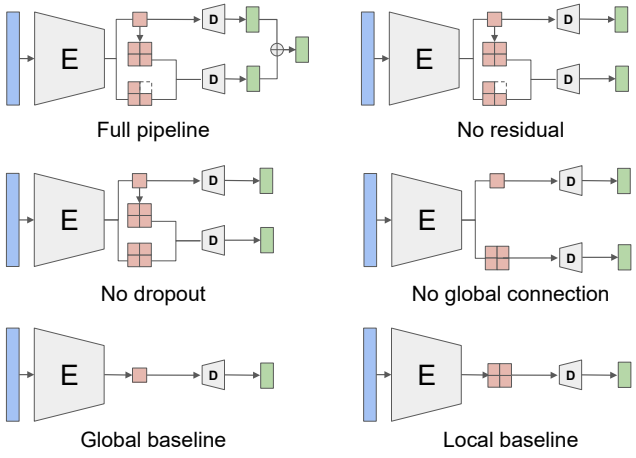


Figure 15: **Illustration of ablation study baselines.** E: encoder; D: decoder.

6.4. Comparison of Dropout and Consistency Loss

To further analyze the different contribution of *latent grid dropout* and *global consistency loss* on shape completion,

Method	Shape Completion Chamfer (↓)	F-Score (↑)
Full pipeline	**1.34**	**66.5**
No consistency loss	1.43	64.7
No dropout (leave-one-out)	1.43	63.9

Table 6: Quantitatively ablate the impacts of consistency loss and latent grid dropout on shape completion from depth image.

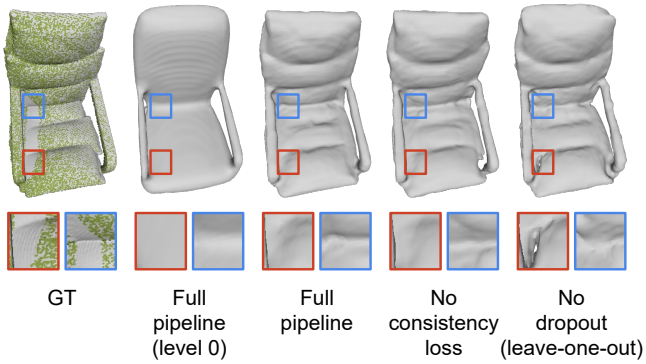


Figure 16: **Ablation on latent grid dropout and consistency loss for the task of shape completion.** Green dots are observed depth points. Compared to the global consistency loss which regularizes regions far from observed points, latent grid dropout reduces noisy residuals and enables plausible detail synthesis on regions that are close to the observed part.

we carry out a leave-one-out ablation on dropout where the only difference with full pipeline is the removal of latent grid dropout. Same as the baselines in Table 2, this ablation is conducted on the chair category of ShapeNet. In Table 6, we show that the removal of dropout leads to slightly larger decrease in quantitative performance than the removal of consistency loss. Meanwhile, dropout impacts qualitative results in a different way than the consistency loss. As shown in Figure 16, when dropout is applied (the third and fourth columns from the left), the model is able to synthesize plausible details on the unobserved regions that are close to the observed part (see insets at the bottom). On the contrary, without dropout (the rightmost column), the model tends to produce noisy residuals (red inset) or add no detail due to the consistency loss (blue inset).

6.5. Failure Cases

Figure 17 shows our failure cases under decoder-only latent optimization for auto-encoding and shape completion from depth image. For objects with very complex geometry or thin structures, our approach still faces challenges. For auto-encoding, such problems could be alleviated by using

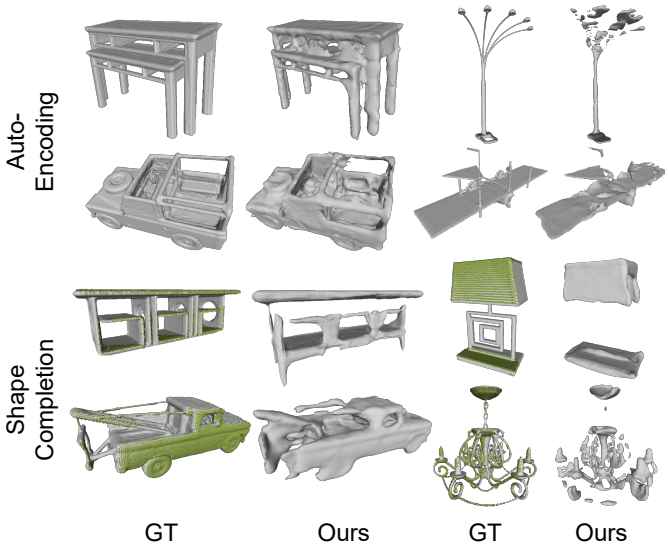


Figure 17: **Failure cases.** Row 1 and 2: auto-encoding; Row 3 and 4: shape completion from depth image.

	Ours	Ours-6	Ours-7	Ours-8	Ours	Ours-6	Ours-7	Ours-8
Chamfer	0.19	0.13	0.13	0.12	0.17	0.14	0.13	0.13
F-Score	93.0	96.5	96.7	97.5	92.8	96.3	97.1	97.3

Table 7: **Auto-encoding accuracy with more levels.** Middle columns: 3D-R^{2}N^2 test set. Right columns: unseen categories. "Ours" stands for 5 levels and "Ours-N" stands for N levels.

more levels and higher resolution latent grids. For shape completion, when an unobserved part (*e.g.*, the lamp body in row 3, column 3) is completely missing in the coarse prediction from level 0, our approach is unable to synthesize such delicate structures.

6.6. Additional Ablation for Number of Levels

In the paper, we use 5 levels as it is a good balance between accuracy and efficiency. But as previously indicated, MDIF is flexible to use other number of levels. In Figure 9, we showed progressive refinement rate-distortion for levels 1-5. Here in Table 7, we further show the auto-encoding accuracy under encoder-decoder inference with up to 8 levels.

6.7. Interpolation and Retrieval in Latent Space

Figure 18 shows linear interpolation in latent space. The latent codes for the two ends are obtained with encoder-decoder auto-encoding. Figure 19 shows results for object retrieval based on latent codes (top-2 retrievals for each query object).

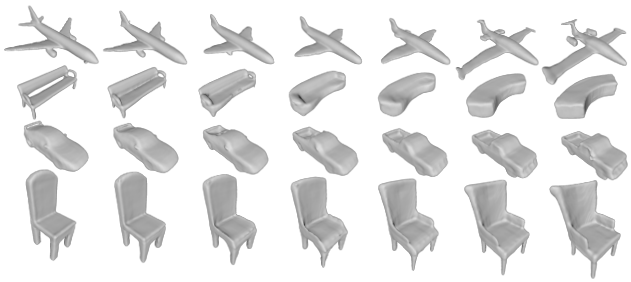


Figure 18: **Linear interpolation in latent space.**

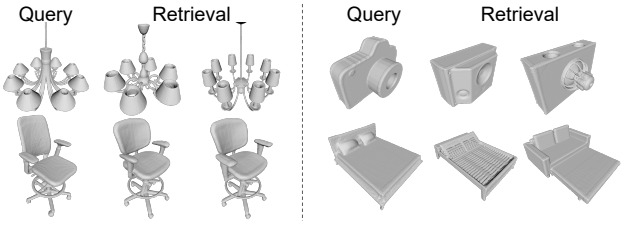


Figure 19: **Object retrieval.** Queries are from test set (left) and unseen categories (right). Retrieved objects are from training set.

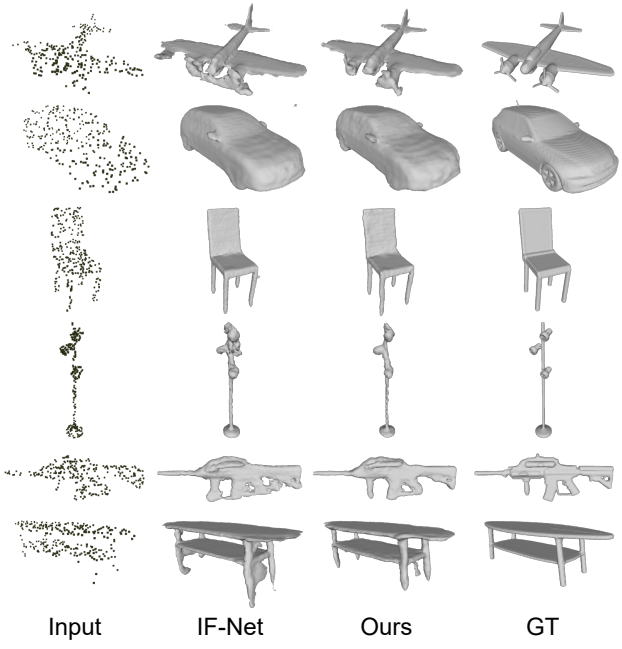


Figure 20: **Point cloud completion**. Additional qualitative results.

6.8. Additional Qualitative Results

Figure 20 and Figure 21 show additional qualitative comparisons on point cloud completion and voxel super-resolution. Compared to IF-Net, our method generally produces cleaner reconstructions with less artifacts.

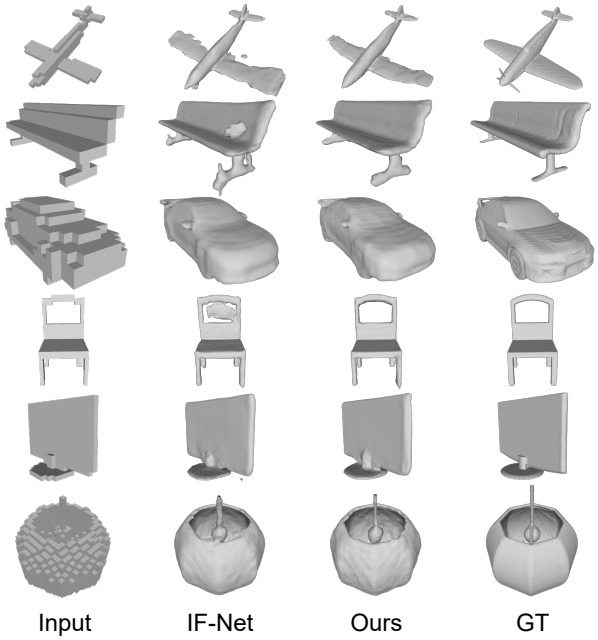


Figure 21: **Voxel super-resolution**. Additional qualitative results.

6.9. Detailed Quantitative Results

Table 8 and Table 9 show per-category quantitative results (Chamfer L2 distance and F-Score) on auto-encoding. For encoder-decoder inference, we compare MDIF with OccNet ("Occ.") [22], SIF [13], LDIF [12] and IF-Net ("IF.") [4]. For decoder-only latent optimization, we compare MDIF with OccNet ("Occ.") [22], IM-Net ("IM.") [3] and a local baseline (resembles [18, 1]). Table 10 shows per-category quantitative results (Chamfer L2 distance / F-Score) on point cloud completion and voxel super-resolution, where we compare MDIF with IF-Net [4] under encoder-decoder inference.

In these experiments, MDIF has lower Chamfer errors for most categories and higher overall F-Score.

6.10. Shape Completion User Study

First, in Table 11, we compare quantitative results of MDIF and competing methods on shape completion from depth image. In this comparison, we also include a MDIF model ("Ours") that uses encoder-decoder inference. This model has the same architecture as the MDIF model in the point cloud completion experiment, and is retrained from scratch to take voxelized depth points (depth points voxelized into a 128^3 grid) as input. In terms of metrics, we additionally use Asymmetric Chamfer to measure the reconstruction accuracy in observed regions. It is computed as one-directional Chamfer L2 distance from depth points to reconstruction.

Category	Chamfer (↓)									F-Score (↑, %)								
	Occ.	SIF	LDIF	IF.	Ours	Occ.*	IM.*	Local*	Ours*	Occ.	SIF	LDIF	IF.	Ours	Occ.*	IM.*	Local*	Ours*
airplane	0.16	0.44	0.10	0.52	**0.05**	0.25	0.13	0.044	**0.028**	87.8	71.4	96.9	94.4	**97.2**	89.8	91.7	98.5	**98.6**
bench	0.24	0.82	0.17	0.31	**0.08**	0.34	0.22	0.121	**0.052**	87.5	58.4	**94.8**	92.6	92.4	85.2	88.6	**96.0**	**96.0**
cabinet	0.41	1.10	0.33	**0.11**	0.29	0.32	0.23	0.063	**0.051**	86.0	59.3	92.0	**93.0**	91.5	83.2	89.2	**96.6**	**96.6**
car	0.61	1.08	**0.28**	0.30	0.29	0.58	0.26	0.090	**0.088**	77.5	56.6	87.2	**87.4**	86.6	69.3	82.7	**93.1**	93.0
chair	0.44	1.54	0.34	**0.10**	**0.10**	0.38	0.43	0.042	**0.035**	77.2	42.4	90.9	**94.5**	93.8	80.2	82.5	**97.7**	97.6
display	0.34	0.97	0.28	**0.07**	0.08	0.35	0.20	0.043	**0.019**	82.1	56.3	94.8	**96.1**	95.1	82.3	89.4	98.6	**98.7**
lamp	1.67	3.42	1.80	1.17	**0.90**	1.47	2.76	**0.795**	**0.795**	62.7	35.0	84.0	**89.1**	87.1	62.9	73.8	**93.5**	**93.5**
rifle	0.19	0.42	0.09	1.07	**0.05**	0.39	0.55	0.060	**0.057**	86.2	70.0	**97.3**	93.5	96.2	86.1	81.1	**96.9**	**96.9**
sofa	0.30	0.80	0.35	0.13	**0.11**	0.31	0.16	0.208	**0.037**	85.9	55.2	92.8	92.5	**93.5**	85.2	89.3	98.3	**98.4**
speaker	1.01	1.99	0.68	**0.14**	0.27	0.38	0.17	0.065	**0.044**	74.7	47.4	84.3	**90.2**	90.1	78.1	89.4	**97.3**	**97.3**
table	0.44	1.57	0.56	0.17	**0.13**	0.31	0.30	0.107	**0.046**	84.9	55.7	92.4	93.4	**93.7**	87.2	88.6	96.5	**97.6**
telephone	0.13	0.39	0.08	0.08	**0.06**	0.19	0.11	0.043	**0.010**	94.8	81.8	98.1	**98.8**	98.3	88.9	96.5	**99.6**	**99.6**
watercraft	0.41	0.78	0.20	0.90	**0.10**	0.35	0.39	0.075	**0.067**	77.3	54.2	93.2	92.7	**93.7**	80.3	84.7	**97.4**	97.2
mean	0.49	1.18	0.40	0.39	**0.19**	0.43	0.46	0.135	**0.102**	81.9	59.0	92.2	92.9	**93.0**	81.4	86.7	96.9	**97.0**

Table 8: **Per-category auto-encoding accuracy for objects in 3D-R^{2}N^2 test set of ShapeNet.** For each metric, left columns compare methods under encoder-decoder inference while right columns compare under decoder-only latent optimization. $*$: decoder-only latent optimization.

Category	Chamfer (↓)									F-Score (↑, %)								
	Occ.	SIF	LDIF	IF.	Ours	Occ.*	IM.*	Local*	Ours*	Occ.	SIF	LDIF	IF.	Ours	Occ.*	IM.*	Local*	Ours*
bed	1.30	2.24	0.68	**0.10**	0.16	0.87	0.43	0.052	**0.045**	59.3	32.0	81.4	**94.7**	90.9	67.1	77.8	96.8	**97.0**
birdhouse	1.25	1.92	0.75	0.31	**0.11**	0.72	0.49	**0.036**	**0.036**	54.2	33.8	76.2	90.4	**92.1**	61.3	74.3	97.6	**97.7**
bookshelf	0.83	1.21	0.36	0.30	**0.20**	0.99	0.60	0.103	**0.091**	66.5	43.5	86.1	**93.5**	88.3	59.0	73.0	**95.1**	94.2
camera	1.17	1.91	0.83	0.27	**0.16**	0.45	0.58	**0.047**	0.050	57.3	37.4	77.7	**95.0**	94.0	70.2	75.9	**98.6**	**98.6**
file	0.41	0.71	**0.29**	0.35	0.30	0.38	0.25	0.054	**0.041**	86.0	65.8	93.0	**95.7**	94.4	84.3	90.0	97.6	**97.7**
mailbox	0.60	1.46	0.40	1.18	**0.20**	0.51	0.74	**0.102**	**0.102**	67.8	38.1	87.6	81.4	**93.5**	80.0	85.2	**98.5**	**98.5**
piano	1.07	1.81	0.78	0.34	**0.08**	0.91	0.71	0.034	**0.030**	61.4	39.8	82.2	**96.7**	94.8	62.2	77.3	**98.3**	**98.3**
printer	0.85	1.44	0.43	**0.15**	**0.15**	0.48	0.31	**0.035**	**0.035**	66.2	40.1	84.6	**94.9**	94.3	74.9	82.3	98.2	**98.3**
stove	0.49	1.04	0.30	0.55	**0.22**	0.37	0.25	0.107	**0.040**	77.3	52.9	89.2	91.3	**93.5**	78.6	87.4	**97.7**	**97.7**
tower	0.50	1.05	0.47	0.44	**0.14**	0.53	0.30	**0.060**	0.070	70.2	45.9	85.7	90.3	**91.8**	73.9	81.7	**96.9**	**96.6**
mean	0.85	1.48	0.53	0.40	**0.17**	0.62	0.47	0.063	**0.054**	66.6	43.0	84.4	92.4	**92.8**	71.1	80.5	**97.5**	**97.5**

Table 9: **Per-category auto-encoding accuracy for objects in unseen categories of ShapeNet.** For each metric, left columns compare methods under encoder-decoder inference while right columns compare under decoder-only latent optimization. $*$: decoder-only latent optimization.

Category	Point Cloud Completion		Voxel Super-Resolution	
	IF-Net	Ours	IF-Net	Ours
airplane	2.37 / 89.7	**0.08 / 93.3**	1.51 / 78.3	**1.02 / 80.7**
bench	1.22 / 84.5	**0.18 / 86.0**	1.88 / 59.1	**1.09 / 59.5**
cabinet	1.65 / **87.1**	**0.84** / 83.8	0.65 / 60.6	**0.60 / 60.8**
car	1.96 / 79.4	**0.19 / 80.9**	0.40 / **75.8**	**0.30 / 75.8**
chair	2.02 / **81.3**	**0.33** / 80.5	1.02 / 62.6	**0.82 / 63.4**
display	1.09 / 88.5	**0.30 / 88.6**	1.04 / 62.0	**0.74 / 62.1**
lamp	2.03 / 76.3	**1.76 / 78.0**	8.14 / 58.3	**3.97 / 60.9**
rifle	2.19 / 85.3	**0.05 / 95.9**	2.09 / 78.0	**0.34 / 81.3**
sofa	0.71 / **88.2**	**0.18** / 86.8	0.68 / 56.2	**0.48 / 57.5**
speaker	1.52 / **78.4**	**0.65** / 75.9	0.73 / 56.1	**0.65 / 58.0**
table	1.70 / 84.7	**0.25 / 85.1**	2.72 / 53.5	**1.87 / 55.7**
telephone	0.98 / 95.7	**0.06 / 96.5**	0.77 / 77.9	**0.67 / 78.2**
watercraft	1.51 / 87.2	**0.14 / 88.4**	2.05 / 71.7	**0.69 / 73.6**
mean	1.61 / 85.0	**0.39 / 86.1**	1.82 / 65.4	**1.02 / 66.7**

Table 10: Per-category quantitative results (Chamfer L2 distance / F-Score) for point cloud completion and voxel super-resolution.

When comparing under encoder-decoder inference ("OccNet", "LDIF", "Ours"), MDIF is only slightly worse than LDIF in F-Score while performs the best in the other two metrics. This reveals that when using encoder-decoder inference, MDIF can produce completion results similarly close to the groundtruth as LDIF. Meanwhile, the large margin in Asymmetric Chamfer compared with OccNet and LDIF demonstrates the better capability of MDIF to preserve details in observed regions, even under encoder-decoder inference. For the MDIF model that uses decoder-only latent optimization ("Ours*"), although it has worse performance in Chamfer distance and F-Score, it can reduce the error in Asymmetric Chamfer even much further. This indicates that it performs much better on the observable parts and the source of error mostly comes from the unobserved parts. As illustrated in the paper (Figure 12), although different from

Category	Chamfer (↓)				F-Score (↑, %)				Asym. Chamfer (↓)			
	OccNet	LDIF	Ours	Ours*	OccNet	LDIF	Ours	Ours*	OccNet	LDIF	Ours	Ours*
airplane	0.47	**0.17**	0.26	0.46	70.1	89.2	**90.1**	73.2	0.246	0.054	0.022	**0.007**
bench	0.70	**0.39**	0.45	0.96	64.9	81.9	**82.5**	56.9	0.281	0.108	0.049	**0.012**
cabinet	1.13	0.77	**0.73**	1.35	70.1	**77.9**	73.8	60.4	0.109	0.052	0.070	**0.009**
car	0.99	0.51	**0.41**	1.04	61.6	72.4	**74.3**	64.2	0.138	0.054	0.043	**0.011**
chair	2.34	1.02	**0.91**	1.42	50.2	69.6	**72.5**	67.0	0.785	0.270	0.053	**0.012**
display	0.95	0.62	**0.56**	1.69	62.8	**80.0**	76.7	55.4	0.312	0.217	0.056	**0.007**
lamp	9.91	2.15	**1.26**	3.26	44.1	66.4	**70.5**	54.6	10.80	1.429	0.160	**0.110**
rifle	0.49	**0.14**	0.31	0.62	66.4	**92.3**	91.5	75.9	0.246	0.048	0.022	**0.005**
sofa	1.08	0.83	**0.70**	1.19	61.2	**71.7**	71.4	62.1	0.155	0.074	0.059	**0.007**
speaker	3.50	1.48	**1.45**	3.73	52.4	**67.3**	64.6	49.8	0.280	0.115	0.077	**0.020**
table	2.49	1.14	**0.94**	1.11	66.7	**78.0**	77.8	61.5	0.784	0.339	0.065	**0.015**
telephone	0.35	**0.19**	0.21	1.05	86.1	**92.0**	89.4	55.9	0.089	0.046	0.046	**0.002**
watercraft	1.15	0.50	**0.45**	0.69	54.5	77.5	**78.3**	67.2	0.684	0.148	0.033	**0.020**
mean	1.97	0.76	**0.67**	1.43	62.4	**78.2**	78.0	61.9	1.147	0.227	0.058	**0.018**

Table 11: **Shape completion from depth image.** Quantitative comparisons on Chamfer distance, F-Score and Asymmetric Chamfer distance. "Ours*" achieves the lowest error on the observed part, measured by the Asymmetric Chamfer distance. Its worse Chamfer and F-Score results are caused by the unobserved part. See our user study for more in-depth analysis. $*$: decoder-only latent optimization.

the groundtruth, the unobserved parts of its results still look plausible.

To prove our point, we conducted a user study to compare human subjective verdicts and F-Score. We recruited 88 participants who were at least 18 years old. All participants had no prior knowledge of this project. Each participant was given 32 pairs of examples, one from MDIF (with decoder-only latent optimization) and one from LDIF [12]. Order of the examples is fully counterbalanced and randomized. Each example was shown in two different views: one observed (input view) and one unobserved. Participants were then asked to choose which example was the more plausible reconstruction given the input. If both examples looked similarly plausible, they were allowed to choose *cannot decide*.

Examples were chosen in this way. The worst results in F-Score were filtered, since both human and F-Score tend to agree on those cases. Then examples with unmatched input views were removed. We then randomly picked 32 examples from the rest.

The results of user study are summarized in Figure 22. In contrast to F-Score, 54.2% of the participants chose in favor of MDIF results, whilst 31.9% thought LDIF results were better. In addition, 13.9% could not decide between MDIF and LDIF. Moreover, when compared with the quantitative metrics, 68.1% disagree with Chamfer L2 distance, and 51.4% disagree with F-Score. All the 32 examples and itemized results are shown in Figure 23, Figure 24, Figure 25 and Figure 26.

The conclusion of this user study aligns with previous work [32], where Chamfer distance has been argued as not suitable for evaluating completion tasks due to its sensitivity to outliers. Moreover, this study also shows that, although more robust, F-Score only tells us how different the reconstruction of the unobserved part is from the groundtruth, but not how *plausible* it is, which is what humans ultimately care about.

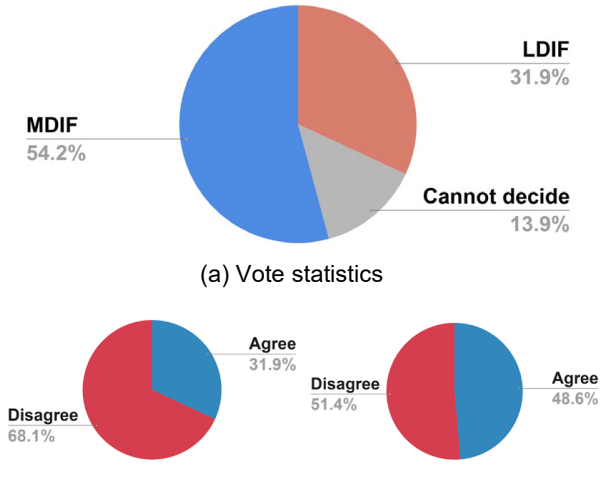


Figure 22: **Summary of user study.** Participants were asked which reconstruction was more plausible. 54.2% chose MDIF while 13.9% cannot decide between the results. Moreover, 68.1% of the votes disagree with Chamfer L2 distance, and 51.4% disagree with F-Score. Refer to Figure 23 to Figure 26 for itemized results.

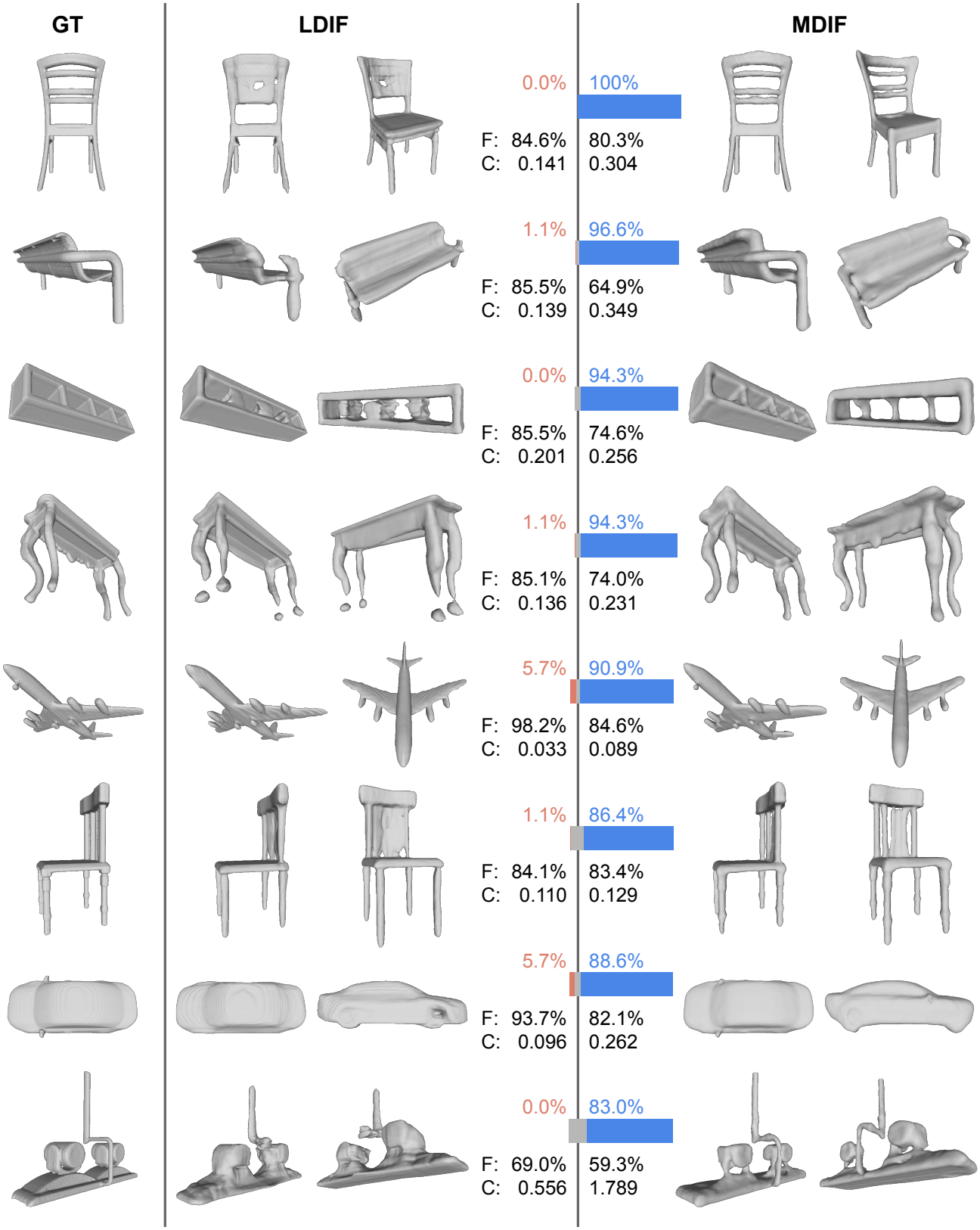


Figure 23: **Itemized user study results.** For each example, we show the groundtruth mesh under input view, and the reconstruction results under two views: one observed view same as input and one unobserved view. The bar chart shows the percentages of votes. Red: prefer LDIF; Blue: prefer MDIF; Gray: Cannot decide; F: F-Score; C: Chamfer L2 distance.

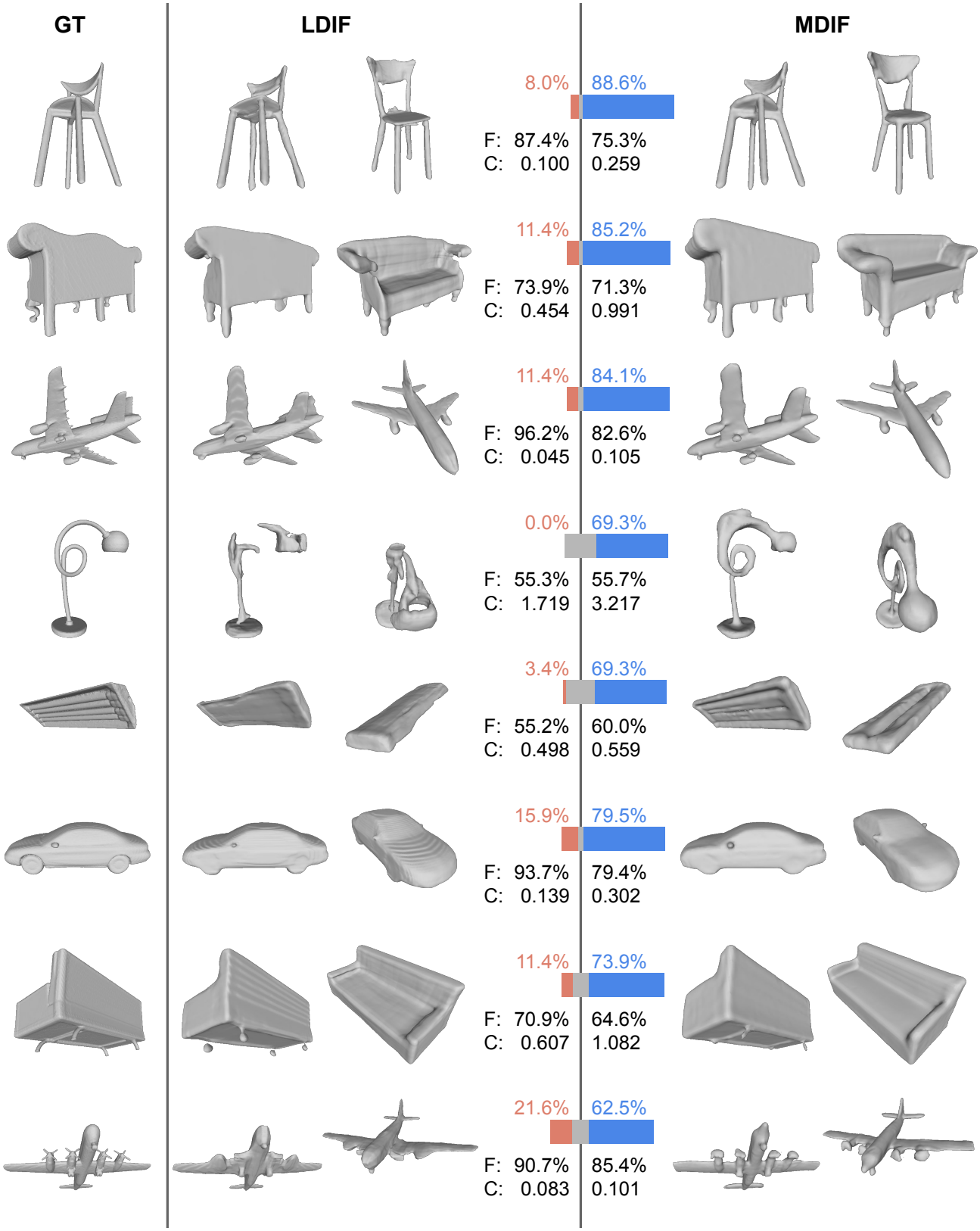


Figure 24: **Itemized user study results.** For each example, we show the groundtruth mesh under input view, and the reconstruction results under two views: one observed view same as input and one unobserved view. The bar chart shows the percentages of votes. Red: prefer LDIF; Blue: prefer MDIF; Gray: Cannot decide; F: F-Score; C: Chamfer L2 distance.

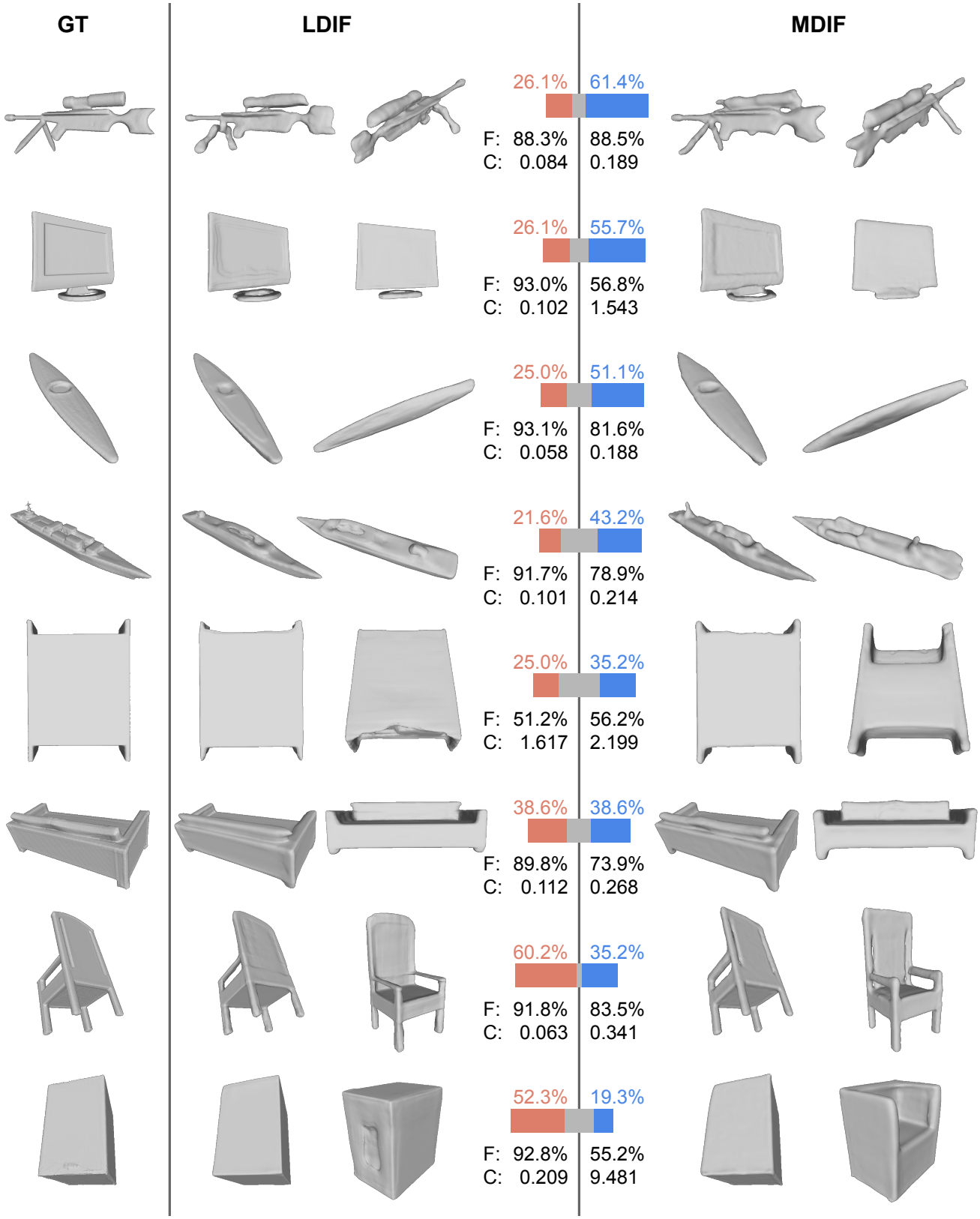


Figure 25: **Itemized user study results.** For each example, we show the groundtruth mesh under input view, and the reconstruction results under two views: one observed view same as input and one unobserved view. The bar chart shows the percentages of votes. Red: prefer LDIF; Blue: prefer MDIF; Gray: Cannot decide; F: F-Score; C: Chamfer L2 distance.

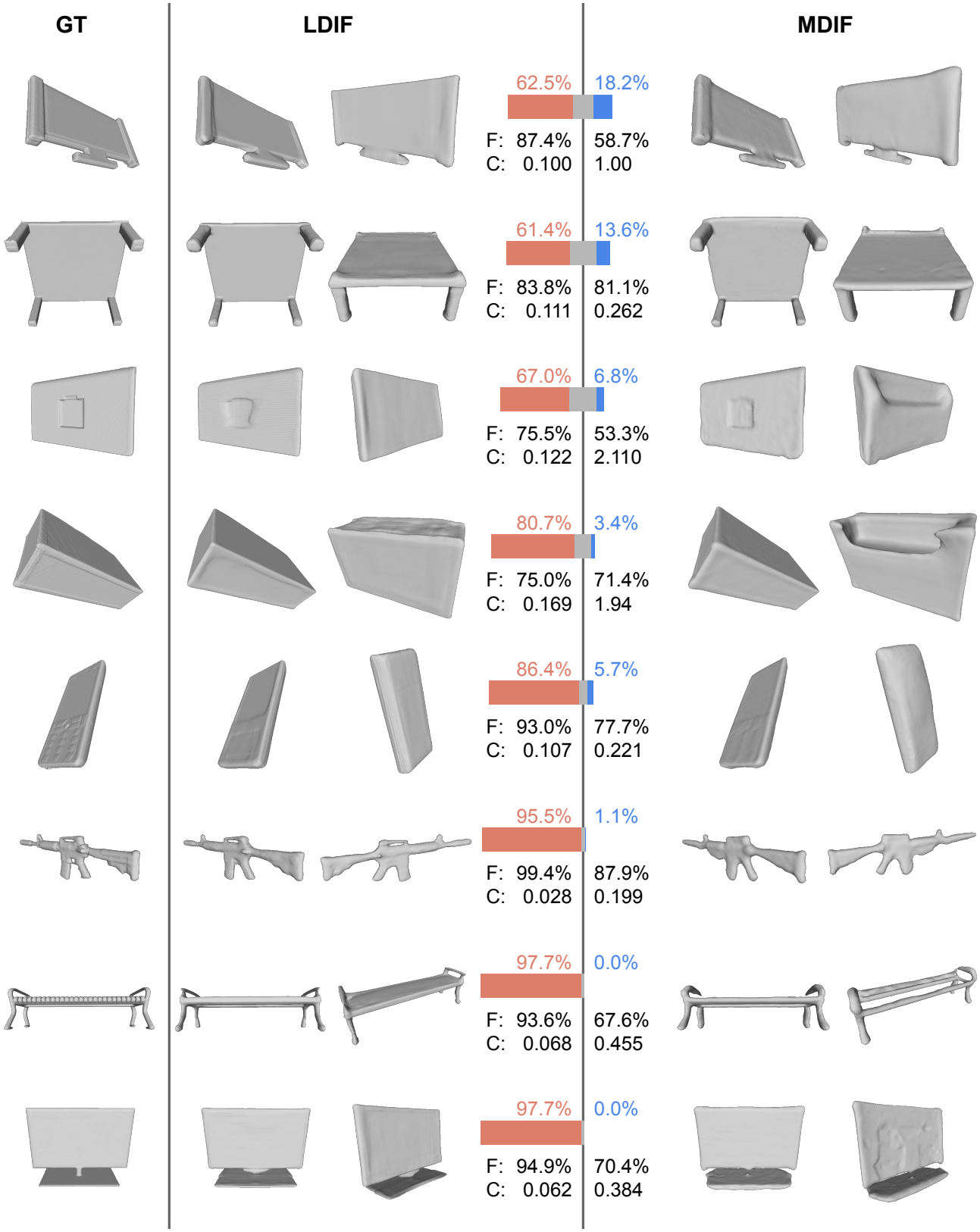


Figure 26: **Itemized user study results.** For each example, we show the groundtruth mesh under input view, and the reconstruction results under two views: one observed view same as input and one unobserved view. The bar chart shows the percentages of votes. Red: prefer LDIF; Blue: prefer MDIF; Gray: Cannot decide; F: F-Score; C: Chamfer L2 distance.

References

[1] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *Eur. Conf. Comput. Vis.*, pages 608–625. Springer, 2020.

[2] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[3] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5939–5948, 2019.

[4] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6970–6981, 2020.

[5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Eur. Conf. Comput. Vis.*, pages 628–644. Springer, 2016.

[6] Charles K Chui. *An introduction to wavelets*. Elsevier, 2016.

[7] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[8] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 5868–5877, 2017.

[9] Mingsong Dou, Philip Davidson, Sean Ryan Fanello, Sameh Khamis, Adarsh Kowdle, Christoph Rhemann, Vladimir Tankovich, , and Shahram Izadi. Motion2fusion: Real-time volumetric performance capture. *ACM TOG (SIGGRAPH Asia)*, 2017.

[10] Mingsong Dou, Sameh Khamis, Yury Degtyarev, Philip Davidson, Sean Ryan Fanello, Adarsh Kowdle, Sergio Orts Escolano, Christoph Rhemann, David Kim, Jonathan Taylor, Pushmeet Kohli, Vladimir Tankovich, and Shahram Izadi. Fusion4d: real-time performance capture of challenging scenes. *ACM Trans. Graph.*, 35(4):114, 2016.

[11] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In *Eur. Conf. Comput. Vis.*, pages 51–67. Springer, 2020.

[12] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *IEEE Conf. Comput. Vis. Pattern Recog.*, June 2020.

[13] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Int. Conf. Comput. Vis.*, pages 7154–7164, 2019.

[14] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Eur. Conf. Comput. Vis.*, pages 484–499. Springer, 2016.

[15] Christian Häne, Sohubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction. *IEEE Trans. Pattern Anal. Mach. Intell.*, 42(6):1348–1361, 2019.

[16] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. *ACM Trans. Graph.*, 38(4):1–12, 2019.

[17] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In *Proc. UIST*, 2011.

[18] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 6001–6010, 2020.

[19] Yue Jiang, Dantong Ji, Zhizhong Han, and Matthias Zwicker. Sdfdiff: Differentiable rendering of signed distance fields for 3d shape optimization. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1251–1261, 2020.

[20] Marian Kleineberg, Matthias Fey, and Frank Weichert. Adversarial generation of continuous implicit shape representations. *arXiv preprint arXiv:2002.00349*, 2020.

[21] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. *arXiv preprint arXiv:2007.11571*, 2020.

[22] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4460–4470, 2019.

[23] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. *arXiv preprint arXiv:1901.06802*, 2019.

[24] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *Eur. Conf. Comput. Vis.*, 2020.

[25] Richard A Newcombe, Dieter Fox, and Steven M Seitz. Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[26] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 165–174, 2019.

[27] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. Octnetfusion: Learning depth fusion from data. In *International Conference on 3D Vision (3DV)*, pages 57–66. IEEE, 2017.

[28] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33, 2020.

[29] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features

let networks learn high frequency functions in low dimensional domains. *Advances in Neural Information Processing Systems*, 2020.

[30] Danhang Tang, Saurabh Singh, Philip A Chou, Christian Hane, Mingsong Dou, Sean Fanello, Jonathan Taylor, Philip Davidson, Onur G Guleryuz, Yinda Zhang, Shahram Izadi, Andrea Tagliasacchi, Sofien Bouaziz, and Cem Keskin. Deep implicit volume compression. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1293–1303, 2020.

[31] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2088–2096, 2017.

[32] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 3405–3414, 2019.

[33] Ayush Tewari, Ohad Fried, Justus Thies, Vincent Sitzmann, Stephen Lombardi, Kalyan Sunkavalli, Ricardo Martin-Brualla, Tomas Simon, Jason Saragih, Matthias Nießner, Rohit Pandey, Sean Fanello, Gordon Wetzstein, Jun-Yan Zhu, Christian Theobalt, Maneesh Agrawala, Eli Shechtman, Dan B Goldman, and Michael Zollhoefer. State of the art on neural rendering. In *Eurographics*, 2020.

[34] Hao Wang, Nadav Schor, Ruizhen Hu, Haibin Huang, Daniel Cohen-Or, and Hui Huang. Global-to-local generative model for 3d shapes. *ACM Trans. Graph.*, 37(6):1–10, 2018.

[35] Peng-Shuai Wang, Yang Liu, Yu-Xiao Guo, Chun-Yu Sun, and Xin Tong. O-cnn: Octree-based convolutional neural networks for 3d shape analysis. *ACM Trans. Graph.*, 36(4):72:1–72:11, July 2017.

[36] Peng-Shuai Wang, Yang Liu, and Xin Tong. Deep octree-based cnns with output-guided skip connections for 3d shape and scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 266–267, 2020.

[37] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1912–1920, 2015.

[38] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems*, pages 492–502, 2019.