MS3D: High-Quality 3D Generation via Multi-Scale Representation Modeling

Supplementary Material

In this appendix, we first provide more details of our method, including network designs of the geometry model, texture model, and multi-scale feature extractor in Appendix A. Details related to experiments, including dataset setup, baselines, and metrics are documented in Appendix B. We further demonstrate our method's versatility through extensive visualizations and comparisons for single-view reconstruction and text-to-3D generation in Appendix C and accompanying videos.

A. Model Details

A.1. Geometry Modeling

Our geometry modeling network employs a customized decoder-only architecture that combines sparse 3D convolutions [48, 78] with view-aware cross-attention [37, 92] to generate a hierarchical structured latent representation for signed distance modeling (Fig. S10). Below we detail key components of our architecture:

Hierarchical Structure Latent Representation. The model begins with 4096 learnable tokens organized in a low-resolution (16^3) voxel grid and progressively refines the representation through four levels of pruning and subdivision, ultimately reaching sparse high-resolution (256^3) voxels. While similar to octree-based approaches [71, 72, 74], our method distinctively aggregates voxels from all levels for signed distance modeling, as detailed in Sec. 3.1 in the main paper.

Transformer. At the coarsest level, the learnable tokens (low-resolution voxel grid) are processed through a series of transformer blocks to capture semantic and global structural information (Fig. S10, upper part). Each transformer block integrates view-aware cross-attention [37, 92], self-attention [68], and MLP layers.

Decoder. We progressively up-sample the low-resolution voxel grid via a U-Net-like decoder architecture similar to [26], enhanced with view-aware cross-attention modules after sparse convolution layers for multi-view feature integration. For each level, we append dedicated branches for structure prediction and signed distance estimation.

Structure Prediction. The structure prediction branch outputs binary classification scores for each voxel, determining whether it should be subdivided or pruned in the next resolution level. This process maintains hierarchical consistency by ensuring finer voxel regions remain within the coverage of coarser voxels.

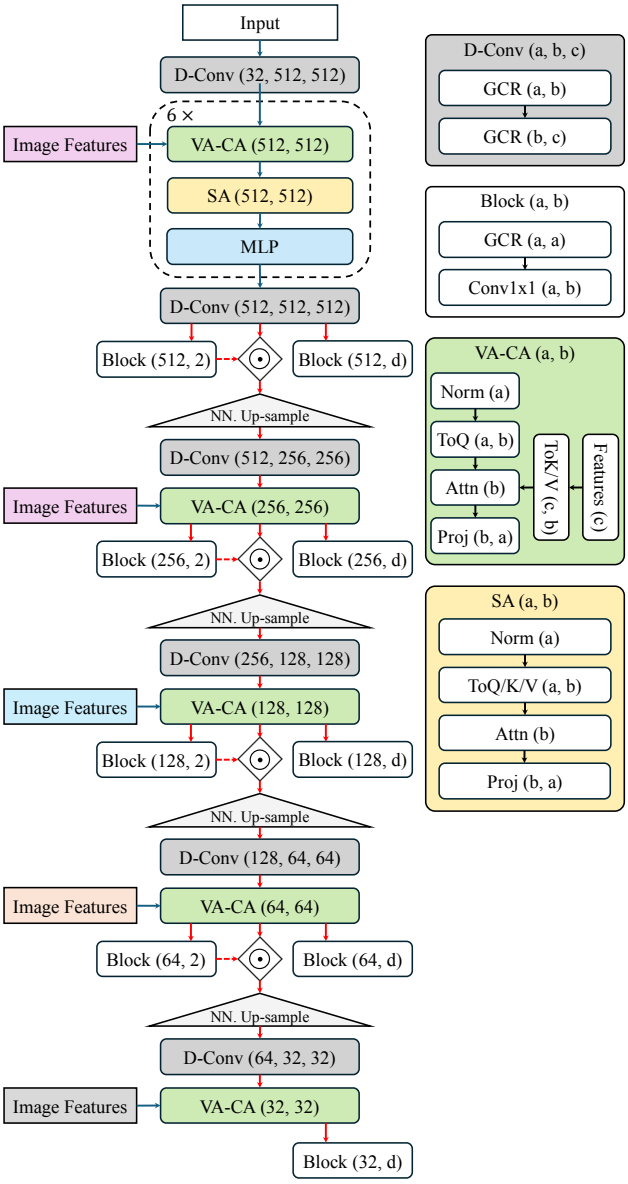


Figure S10. **Architecture for the geometry model.** 'GCR' means a sequential of GroupNorm, Convolution, and ReLU activation. '⊙' denotes voxel masking using the structure prediction results. Different colors represent image features from different scales.

Signed Distance Decoder. For point-wise signed distance prediction, we adopt the architecture from [25, 26], where features from all containing voxels across scales are trilinearly interpolated and concatenated. We enhance spatial awareness by incorporating positional encoding of relative coordinates to the nearest voxel, ensuring translation-invariant predictions.

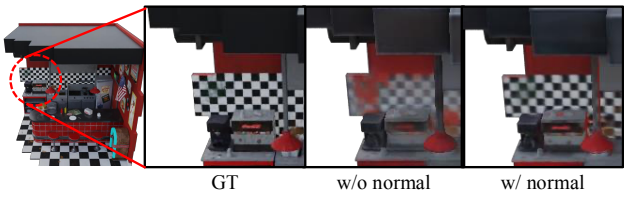


Figure S11. Ablation of normal in texture decoder.

A.2. Texture Modeling

Our texture modeling network employs a customized auto-encoder architecture that combines sparse 3D convolutions [48, 78] with view-aware cross-attention [37, 92] for effective multi-view color aggregation and propagation to unseen regions (Fig. S12). The network consists of several key components:

Geometry-Aware Input Encoding. We first voxelize the reconstructed mesh into sparse voxels, with each voxel encapsulating a local surface region. For each voxel, we compute its normal through dense sampling and averaging, which is then concatenated with the voxel center's positional encoding to form the initial feature vector.

Multi-scale Feature Processing. We employ an auto-encoder architecture for multi-scale feature processing. Our encoder adopts alternating view-aware cross-attention, sparse convolution layers, and max pooling operations to gradually increase the receptive field while coarsening the voxel representation (Fig. S12, upper part). Our decoder reverses this process through sparse convolutions, view-aware cross-attention, and nearest-neighbor upsampling operations, with skip connections facilitating the fusion of low-level and high-level features (Fig. S12, lower part). Unlike the geometry network, we align the decoded voxels directly with the encoded hierarchy, eliminating the need for structure prediction.

Texture Prediction. The final color prediction utilizes an MLP with ReLU activations [1], taking as input the trilinearly interpolated features from the finest level concatenated with local surface normals. Such design is crucial for accurate texture reconstruction, particularly in challenging cases such as thin structures. As demonstrated in Fig. S11, when reconstructing a thin wall with different textures on each side (pure red versus black-white checkboard), incorporating normal information enables the network to correctly differentiate and produce the detailed textures, while omitting normals leads to color bleeding between opposite sides.

A.3. Multi-Scale Extractor

Our multi-scale feature extraction network builds upon Segment Anything V2 [53], modified to accommodate both

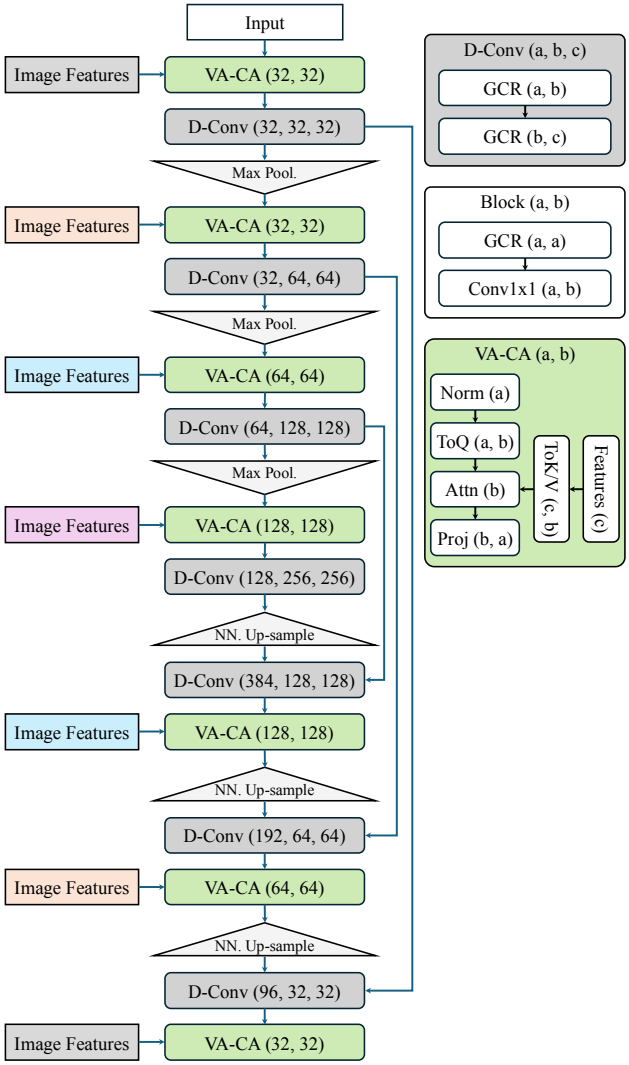


Figure S12. **Architecture for the texture model.** We use an auto-encoder architecture with skip connections. Different colors represent image features from different scales.

RGB and normal inputs (Fig.S13). The architecture processes patchified and projected images through a series of windowed self-attention modules. Feature downsampling is achieved through Q-Pooling [57] at layers 3, 6, and 22, while global structural relationships are captured via global self-attention at layers 13, 17, and 21. Unlike the original architecture, we maintain separate feature maps at four distinct scales, which are injected into corresponding levels of our sparse voxel hierarchy through view-aware cross-attention (see geometry branch Fig. S10 and texture branch Fig. S12).

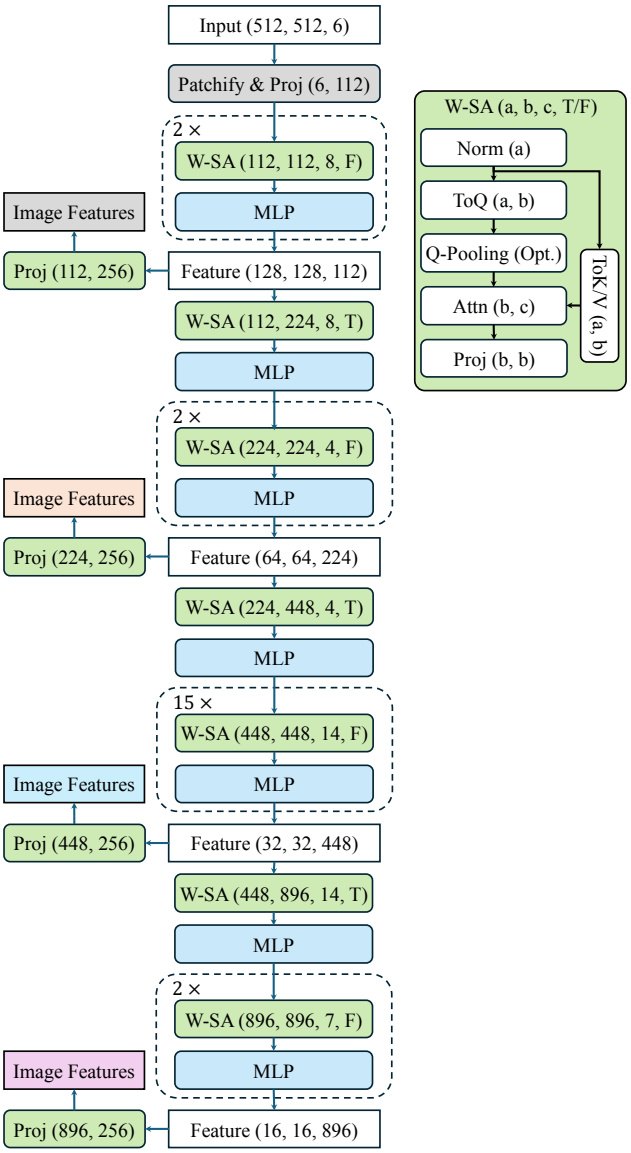


Figure S13. **Architecture for the multi-scale feature extractor.** W-SA (a,b,c,T/F) represents a windowed self-attention module with a window size of c, takes the features with channel a, outputs features with channel b, and applies Q-Pooling [57] to downsample tokens by 2 when the 4th parameter is T.

A.4. Training and Inference

We train the geometry and texture reconstruction, separately. In the geometry branch, the model progressively refines voxel features and predicts a sparse voxel hierarchy, optimized by minimizing the difference between the extracted and ground-truth meshes. The texture branch takes ground-truth meshes as input and predicts surface colors with direct supervision. During inference, we first extract a mesh from the geometry model, which then serves as input for the texture model to generate the final textured mesh.

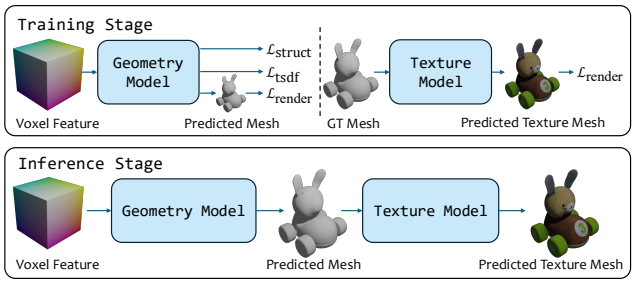


Figure S14. **Training and inference stage.**

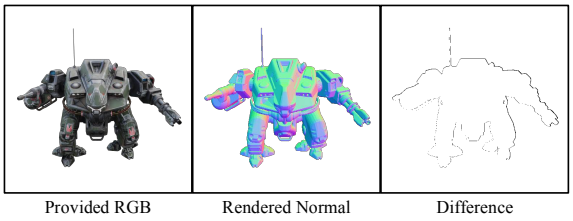


Figure S15. **Visual comparisons of remeshing operation.** The RGB image is rendered using the original mesh, and the normal image is rendered using the result of remeshing operation. Difference image indicates the difference around the object silhouette.

B. Experiment Details

B.1. Hyperparameters

Dataset Preparation. For Objaverse [15] meshes, we implement a crucial preprocessing step to ensure water-tight models with clean internal structures, as accurate SDF is essential for robust reconstruction. Following [89], our preprocessing pipeline computes unsigned distance fields and determines inside/outside regions through visibility checking, followed by isosurface extraction to obtain clean, water-tight meshes. While this remeshing process may introduce minor topological modifications and slight misalignments between 3D meshes and 2D renderings, our visual analysis (Fig. S15) demonstrates that such discrepancies are negligible and do not affect reconstruction quality.

Training Objectives. Our training process employs different loss combinations for geometry and texture stages. The geometry model is supervised by a weighted combination of structure, TSDF, and rendering losses:

$$\mathcal{L}_{\text{g}} = \lambda_{\text{struct}}\mathcal{L}_{\text{struct}} + \lambda_{\text{tsdf}}\mathcal{L}_{\text{tsdf}} + \lambda_{\text{mse}}\mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} \quad (6)$$

where $\lambda_{\text{struct}} = 20$, $\lambda_{\text{tsdf}} = 50$, $\lambda_{\text{mse}} = 10$, and $\lambda_{\text{lpips}} = 1$. The higher weight on TSDF loss emphasizes accurate geometry reconstruction, while the structure loss guides the progressive refinement process. For the texture model, we focus solely on rendering quality through MSE and perceptual losses:

$$\mathcal{L}_{\text{t}} = \lambda_{\text{mse}}\mathcal{L}_{\text{mse}} + \lambda_{\text{lpips}}\mathcal{L}_{\text{lpips}} \quad (7)$$

with weights $\lambda_{\text{mse}} = 10$ and $\lambda_{\text{lpips}} = 1$, balancing pixel-wise accuracy and perceptual quality.

B.2. Metrics

ICP Alignment. To ensure fair comparison across methods using different coordinate systems, we carefully align the output mesh with the ground-truth mesh using the Iterative Closest Points (ICP) algorithm [93]. Specifically, we begin by roughly aligning the scaling using a bounding sphere. Then, we sample the scaling factor uniformly between $[1/1.5, 1.5]$, along with 3D rotations uniformly distributed over the rotation space. For each object, we perform 500 trials with different initializations and select the optimal transformation based on minimal Chamfer distance.

Evaluation Protocol We employ both 2D and 3D metrics for comprehensive evaluation of our reconstruction quality. For 2D assessment, we generate renders from 24 viewpoints (two elevation levels with 12 evenly distributed cameras each) to compute PSNR, SSIM, and LPIPS [90] scores for both color images C and normal images N. For 3D evaluation, we transform the aligned meshes to a normalized space $[-0.5, 0.5]$ and sample 20k points X for computing Chamfer distance and F-Score, where predicted and ground-truth results are denoted with subscripts pd and gt respectively.

Chamfer Distance. The Chamfer distance d_C measures bidirectional point-to-surface distances:

$$
\begin{aligned}
d_C &= \text{Comp.} + \text{Acc.}, \\
\text{Comp.} &= \frac{1}{|X_{\text{gt}}|} \sum_{x_{\text{gt}} \in X_{\text{gt}}} \min_{x_{\text{pd}} \in X_{\text{pd}}} ||x_{\text{gt}} - x_{\text{pd}}||, \\
\text{Acc.} &= \frac{1}{|X_{\text{pd}}|} \sum_{x_{\text{pd}} \in X_{\text{pd}}} \min_{x_{\text{gt}} \in X_{\text{gt}}} ||x_{\text{pd}} - x_{\text{gt}}||.
\end{aligned}
$$

F-Score. The F-Score f_S evaluates reconstruction completeness and accuracy:

$$
f_S = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}
$$

where

$$
\begin{aligned}
\text{Precision} &= \frac{|\{x_{\text{pd}} \in X_{\text{pd}} | \min_{x_{\text{gt}} \in X_{\text{gt}}} ||x_{\text{gt}} - x_{\text{pd}}|| \xi\}|}{|X_{\text{pd}}|}, \\
\text{Recall} &= \frac{|\{x_{\text{gt}} \in X_{\text{gt}} | \min_{x_{\text{pd}} \in X_{\text{pd}}} ||x_{\text{pd}} - x_{\text{gt}}|| \xi\}|}{|X_{\text{gt}}|}.
\end{aligned}
$$

Following prior work [37, 81], we set the distance threshold $\xi = 0.05$.

C. More Results

We provide more visualization results for various applications in Fig. S16, Fig. S17 and Fig. S18.

Figure S16. **Qualitative comparisons of single-view to 3D on GSO dataset.** Best viewed in 2× zoom.

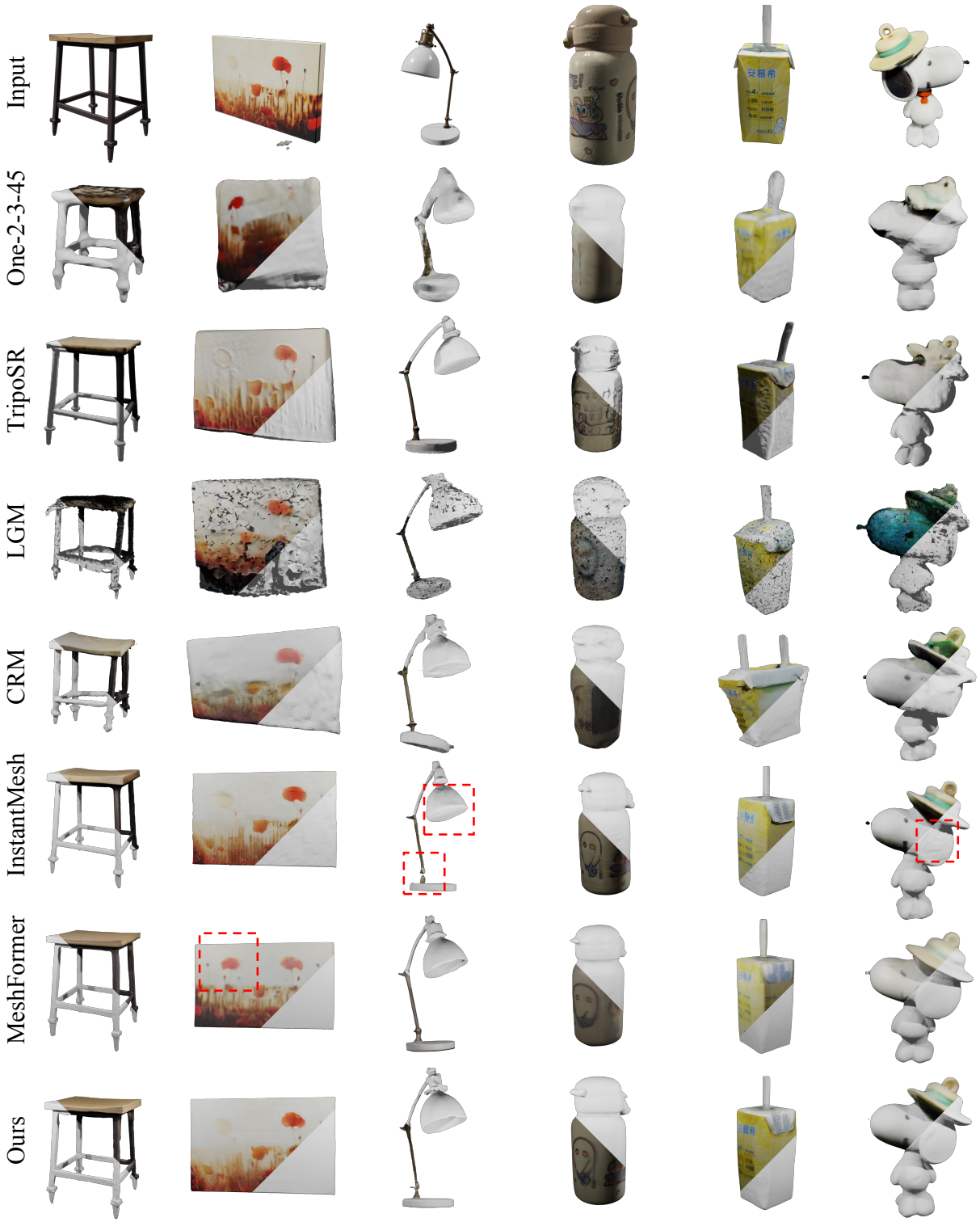
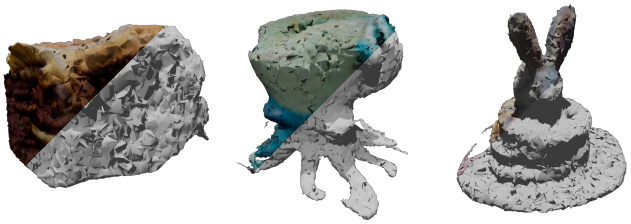


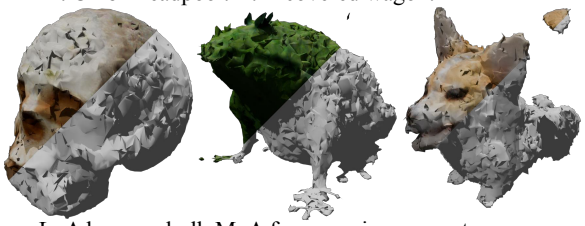
Figure S17. **Qualitative comparisons of single-view to 3D on ABO dataset and OmniObject3D dataset.** Best viewed in 2× zoom.

LGM

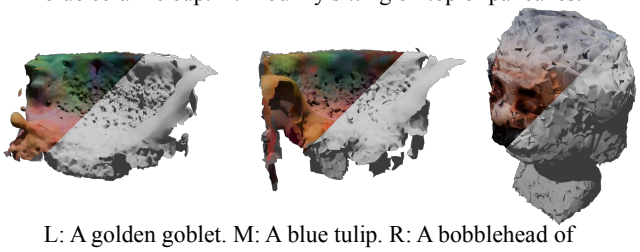
L: A blue jay standing on rainbow macarons.
M: Chibi Deadpool. R: A covered wagon.



L: An overstuffed pastrami sandwich. M: An octopus holding a blue ceramic cup. R: A bunny sitting on top of pancakes.



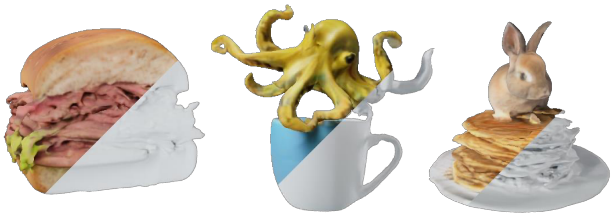
L: A human skull. M: A frog wearing a sweater.
R: A corgi puppy.



L: A golden goblet. M: A blue tulip. R: A bobblehead of Albert Einstein.

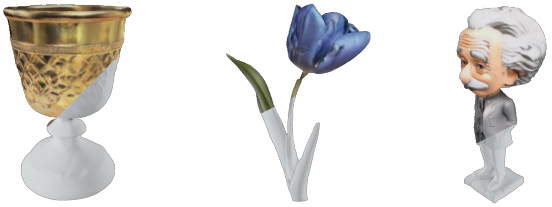
Ours

L: A blue jay standing on rainbow macarons.
M: Chibi Deadpool. R: A covered wagon.



L: An overstuffed pastrami sandwich. M: An octopus holding a blue ceramic cup. R: A bunny sitting on top of pancakes.

L: A human skull. M: A frog wearing a sweater.
R: A corgi puppy.



L: A golden goblet. M: A blue tulip. R: A bobblehead of Albert Einstein.

Figure S18. **Qualitative comparisons of text to 3D.** Best viewed in 2× zoom.