ROAD: Learning an Implicit Recursive Octree Auto-Decoder to Efficiently Encode 3D Shapes (Supplementary)

Sergey Zakharov, Rareş Ambruş, Katherine Liu, Adrien Gaidon

Toyota Research Institute

1 Latent Vector Fusion

In this section we provide additional details to complement the *Latent Vector Fusion* experiments presented in the main paper, with results summarized in Table 2 in the main paper.

Recall the latent subdivision function, $\phi : \mathbb{R}^D \rightarrow \mathbb{R}^{8D}$, defined as $\phi(\boldsymbol{z}^m) = \{\boldsymbol{z}_i^{m+1}\}_{i=1}^8$ (cf. Equation 1 in the main paper). Thus, ϕ directly regresses the next level latents.

Addition. We define $\phi_{add} : \mathbb{R}^D \rightarrow \mathbb{R}^{8D}$ as: $\phi_{add}(\boldsymbol{z}^m) = \{\boldsymbol{z}^m + \boldsymbol{z}_i^{m+1}\}_{i=1}^8$. ϕ_{add} maintains the dimensionality of the latent space during traversal, as does the original formulation of ϕ, i.e. $\boldsymbol{z}^m \in \mathbb{R}^D$ and $\boldsymbol{z}_i^{m+1} \in \mathbb{R}^D$.

Concatenation. In this case, the signature of the octree traversal function in latent space changes from one LoD to the next. This introduces significant modifications to the underlying neural network architecture, requiring specialized networks at each LoD. Specifically, at LoD m, we define $\phi_{concat} : \mathbb{R}^{m \times D} \rightarrow \mathbb{R}^{(m+1) \times 8D}$ as: $\phi_{concat}(\boldsymbol{z}^m) = \{\text{concatenate}(\boldsymbol{z}^m, \boldsymbol{z}_i^{m+1})\}_{i=1}^8$, i.e. $\boldsymbol{z}^m \in \mathbb{R}^{m \times D}$ and $\boldsymbol{z}_i^{m+1} \in \mathbb{R}^D$.

2 Evaluation Details

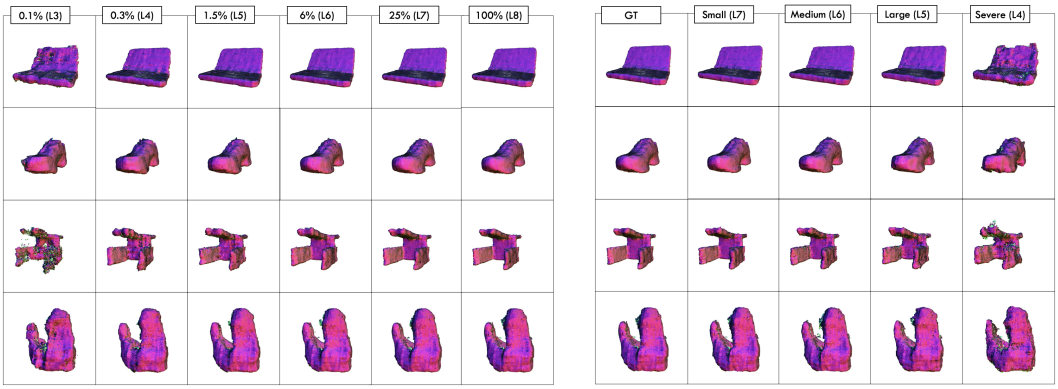
Following NGLOD's implementation [1], we uniformly sample $2^{17} = 131072$ points within the unit cube for gIoU computation. To get occupancy estimates, we recover an object mesh using Poisson surface reconstruction [2] as implemented in Open3D [3]. Similarly, we sample 2^{17} points on the ground truth mesh for Chamfer distance computation. We use Pytorch3D [4] Chamfer distance implementation. We also use the original NGLOD implementation as well as NGLOD's re-implementations of presented baselines: DeepSDF, FFN, SIREN, and Neural Implicits.

3 Extended Generalization Experiment

We extend the generalization experiment to include multiple grades of sparsity and noise. We use a network trained on 512 dense objects of the Google Scanned Objects dataset [5]. We then optimize latent vectors to fit unseen objects from the generalization experiment. Given a ground truth dense unseen object point cloud we apply sparse supervision computed from the dense GT point cloud to estimate the surface geometry. We optimize the pre-trained ROAD to a lower LoD, i.e. provide a coarser supervisory signal than the network was trained to. We then extract the surface to the highest LoD. LoDs 3 through 7 represent approximately 0.1%, 0.3%, 0.15%, 6%, and 25% of the supervision at LoD8. We observe that even in the case of optimizing only to LoD3, our method is still able to converge to reasonable shapes (see Fig. 1a).

We additionally demonstrate how noise affects the generalization performance. Similarly to the sparsity experiments above, we optimize the pre-trained ROAD network to a lower LoD and additionally we randomly perturb SDF annotations at the final LoD of interest with a uniform noise distribution scaled by the voxel size. This procedure corresponds to adding different levels of metric

6th Conference on Robot Learning (CoRL 2022), Auckland, New Zealand.



(a) Sparsity ablation

(b) Noise ablation

Figure 1: Extended Generalization Experiment

noise at LoD 7 (small), 6 (medium), 5 (large), and 4 (severe). We then use a network trained on unperturbed data (from the same set of 512 Google Scanned Objects as in the experiment above) to fit to the occupancy and noisy surface annotations at a particular LoD; for all LoDs below the query LoD we supervise only on occupancy. Finally, we visualize the fully extracted object (i.e. at LoD 8). Once again, we observe that our method is robust to introduced perturbations and is able to faithfully reconstruct objects even when noise is introduced (see Fig. 1b).

4 Ground Truth Labels

In our experiments, we extract ground truth labels from meshes and pointclouds, and generally require dense surface points to obtain accurate labels. In practice, the occupancy label of a voxel at a particular LoD is determined by querying whether a point exists within the voxel of interest, and the SDF value and normal are extracted from the nearest neighbor to the voxel center. We observe that these same quantities could also be extracted from an object represented by an SDF. Additionally, we pre-compute and store these annotations once per dataset over all LoDs.

5 Additional Qualitative Results

Below, we plot additional qualitative reconstruction results for Google Scanned Objects [5] and Thingi32 [6] datasets. Please refer to the supplementary video for further qualitative results, latent space visualizations, and our method's architecture review.

Figure 2: Example reconstruction results on the Thingi32 dataset.

Figure 3: Example reconstruction results on the Google Scanned Objects dataset.

References

[1] T. Takikawa, J. Litalien, K. Yin, K. Kreis, C. Loop, D. Nowrouzezahrai, A. Jacobson, M. McGuire, and S. Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *CVPR*, 2021.

[2] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *SGP*, 2006.

[3] Q.-Y. Zhou, J. Park, and V. Koltun. Open3d: A modern library for 3d data processing. *arXiv*, 2018.

[4] N. Ravi, J. Reizenstein, D. Novotny, T. Gordon, W.-Y. Lo, J. Johnson, and G. Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv*, 2020.

[5] L. Downs, A. Francis, N. Koenig, B. Kinman, R. Hickman, K. Reymann, T. B. McHugh, and V. Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. *arXiv*, 2022.

[6] Q. Zhou and A. Jacobson. Thingi10k: A dataset of 10,000 3d-printing models. *arXiv*, 2016.