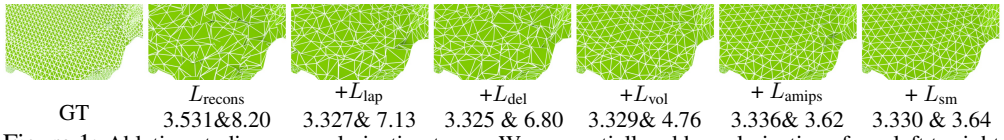

Figure 1: Ablation studies on regularization terms. We sequentially add regularizations from left to right. We show the corresponding Chamfer Distance↓ (CD) & AMIPS↓ on the bottom.

Figure 2: Scene.

We would like to thank reviewers for their detailed comments. We address questions below.

[R1,R2,R3,R4] Ablation Studies: We provide a qualitative ablation study for different terms by optimizing DEFTET to ground truth shape in Fig. 1. L_{lap} helps maintain the regularity. L_{del} encourages local deformation. L_{vol} helps avoid flips, L_{amips} helps avoid needle-like predictions, and L_{sm} helps smooth the prediction. We will add these in a revision.

R1 Impact of Resolution: We ablated different resolutions of DEFTET compared to a voxel-based representation in Fig. 4. We report the performance of DEFTET at 70 resolution in Table 1 compared to SOTA SDF-based approach [40].

R1 More convincing application: We partly disagree. Several algorithms have been proposed to differentially render an implicit function, yet these methods still use the non-differentiable marching cube algorithm to convert SDF into a 3D mesh, which impedes inference speed. DEFTET predicts meshed shape without the time-consuming post-processing.

R2 Unfair comparison to Pixel2Mesh [46] and MeshRCNN [14]: L_{vol} and L_{amips} are volume-based and not suitable for [46]&[14]. L_{lap} is also used in [46]&[14]. L_{sm} and L_{del} can also be used in [46]&[14]. We perform an additional experiment on point cloud 3D reconstruction, [46]/[14] achieved 66.06/72.92 IOU, even worse than original results.

R2 No comparison with SOTA SDF approaches: We wish to clarify that we did compare with SOTA SDF-based approaches. For point cloud 3D reconstruction (Tbl. 1, Fig. 3&4), we compared with [40] (ECCV2020), the best SDF-based approach at the time of our submission. For single image 3D reconstruction with **2D supervision** (Tbl. 2, Fig. 5), where **no** 3D ground truth is utilized, we compared with DVR [37], which is the SOTA SDF-based approach on this task. In both tasks, we outperformed the baselines. DISN is not applicable in these two tasks as it is specifically designed for single image 3D reconstruction using **3D supervision**, i.e., 3D ground truth is required during training.

R2 Worse result on single image 3D reconstruction: We first clarify that single image 3d reconstruction can be done using 3D or 2D ground truth as supervision, and we apply DEFTET in both regimes. We show results of using 2D supervision in the main paper, which has significant improvement, and we report **both quantitative and qualitative** results of using **3D supervision** in the supplement (Tbl. 2, Fig. 3). Due to the deadline rush, the results in supplement were reported with a non converged network – we apologize. We report the true performance as 2.006/19.463 Chamfer/Hausdorff Mean, which outperforms [46]&[14]. We thank the reviewer for pointing out DISN. Since we use the network backbone from OccNet, it is not fair to directly compare with DISN which uses a more powerful network structure. We will include a fair comparison in the revision.

R2 Concerns on Eq. 9∼11: We tried both and found that considering all faces converges faster than only considering the last face. If f_1 is partially occluded by f_2, the occluded part would have a much lower visibility due to Eq.10 ($m_1 < m_2$). Note that we use m instead of D to accumulate colors.

R3 Experiments on other categories: We showed cars in the supplement, and we will add more classes in a revision.

R3 #Vertices: This indeed is a limitation, as is for the voxel-based approaches, yet less so for DEFTET since we can deform the grid geometry. In the future, we plane to adaptively subdivide to get details, akin to oct-tree voxel grids.

R3 Scale to large scene: We partially agree with the reviewer, but we can increase the resolution of DEFTET for the complex scene. We show two optimization results in Fig. 2 by reconstructing multiple objects with DEFTET.

R3 Smoothness term: $\mathcal{F}$ here denotes the set of all triangular faces whose $P_s(v)$ equals to one (L185 in the paper), instead of all the faces of the tetrahedralization. We want the neighbouring boundary faces to be approximately parallel. We apologize for the typo – there should be a negative sign before cosine. We will revise.

R3 Avoid flips: We thank the reviewer for raising this question. There is a tradeoff between having no flips and performance. We can achieve having no flips by increasing the weight of L_{vol} but sacrifice the performance. Empirically, when doing optimization, we achieve 3.783 CD with 0 flip, and 3.330 CD with 1 flip over all 729162 tetrahedrons.

R3 #Vertices: The #vertices for 30^3 is 4198, and 70^3 is 47416 , the maximum resolution we can support is 150^3 with batch size 4 on a 32G V100 GPU. **R3 Related literature:** Thank you for pointing this out, we will add it.

R4 Comparison with DMC [26]: We agree that we share similarities with DMC – we discussed (L90) and compared with (Tbl. 1 & Fig. 3) DMC. We differ in the following aspects: **1. Basic element**. DMC deforms vertices in a cube and constraints the deformation to be along the edges within [0,1), while we freely deform tetrahedron's vertices, whose typical displacements are ~ 1.18x of edge length. Therefore, DMC has a fixed set of possible topologies and needs to deal with many corner cases (Fig. 2 in [26]), while we have more flexibility in representing geometric details (e.g. the curvy chair back in Fig. 3) and no corner cases. **2. Formulation.** DMC focuses on the surface and uses the "occupancy" to derive the probability over possible topologies. While our DEFTET explicitly defines the occupancy for one tetrahedron and deforms it which makes training easier than in DMC. We compared with DMC in point cloud 3D reconstruction task and single image 3D reconstruction using 3D supervision, and achieved significantly better performance. DMC is not applicable to single image 3D reconstruction using 2D supervision. The reason why DMC runs slower is that the official source code of DMC runs an iterative algorithm to get the mesh after obtaining the displacement and occupancy. We refer to the source code for details.

R4 Drawback of laplacian smoothing layer: We do not notice drawbacks. The computation is neglible.

R4 Comparison with Tet Meshing: We show that our DEFTET reaches comparable performance to traditional tetrahedral meshing methods – a sanity check experiment. We are more accurate than QuarTet in Tbl. 3. The robustness (even 1% failing rate) are very important for applications in computer graphics [17]. Note that all of traditional methods can not be plugged into deep learning, while DEFTET can easily support it. We apologize for the typo, Tbl. 3 is correct.