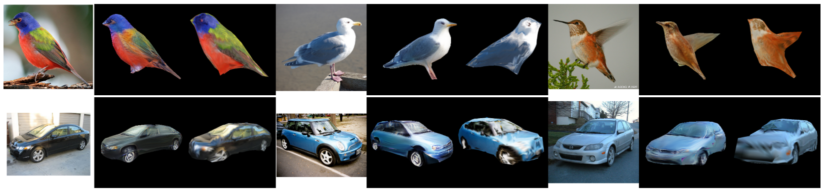

Input Ours CMR Input Ours CMR Input Ours CMR

Figure 1: Qualitative comparisons for prediction from real images. Top is CUB-bird dataset, bottom is PASCAL3D+ car dataset

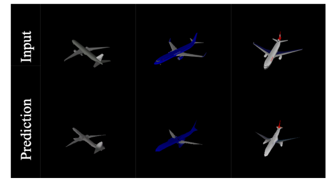

Figure 2: Qualitative results on Airplane.

We would like to thank reviewers for their detailed comments. We have attempted to address all concerns below.

Contribution [R4] We wish to emphasize that although barycentric interpolation has been used before in a forward rendering pipeline, it is non-differentiable due to rasterization, our contribution is its **differentiable reformulation**. More importantly, while barycentric interpolation only affects foreground pixels, we further employ a global aggregation method to obtain a probabilistic silhouette which handles background pixels, making DIB-Renderer a solution for all the pixels in the image. To the best of our knowledge, DIB-Renderer is the first analytical differentiable renderer which supports all common elements in a rendering pipeline, while also supporting optimization over large shape deformation. We also want to clarify that supporting texture mapping and illumination is **not a trivial engineering problem**. SoftRas-Mesh [18] and SoftRas-Color [19] do not support texture and lighting of this form. N3MR [12] only supports face sampling based texture mapping and Lambertian illumination. However, this texture mapping samples the same number of pixels per face, resulting in inaccuracy for large faces. Both models cannot be easily extended to support advanced illumination and texture models due to specific design choices in their differentiable rendering process which would require a major algorithm reformulation. In contrast, our formulation which, due to intentional similarities and parallels to standard OpenGL texture mapping, naturally supports texture and lighting models.

Additional Experiments [R2,R3,R4] To further demonstrate the effectiveness of our DIB-Renderer, we show additional 3D reconstruction results for real images and a new ShapeNet class. For **real images**, we follow Learning Category-Specific Mesh Reconstruction from Image Collections (CMR [11]), and use two datasets: the CUB bird dataset and the PASCAL3D+ car dataset. We predict geometry and texture from a single-view image. Results are shown in Fig. 1 and Tab. 1, where we demonstrate more faithful geometry and more realistic textures. For **new ShapeNet class**, examples on the Airplane class are shown in Fig. 2 and we will put more results in an updated supplementary.

Models	Texture	2D IOU	Key Point
CMR [11]	0.043	0.262	0.930
Ours	0.043	**0.243**	**0.972**

Table 1: Results on CUB bird dataset. Texture and 2D IOU show L-1 loss and 2D IOU loss between predictions and GT. (Lower is better) Key point evaluates percentage of predicted key points lying in the threshold of 0.1. (Higher is better)

Related Work MCRT [R2,R4] We agree that the Monte-Carlo Ray Tracing based method [15] supports more advanced features. However, there are 2 main benefits to our method. Firstly, DIB-Renderer is faster in terms of running time since we do not need to sample millions of rays. To demonstrate this, we render a 3D human model with 6890 vertices and 13776 faces into a 256x256 image with a Titan-V GPU with the same camera settings and calculate the gradients for all operations. We show the results of this comparison in Tab. 2, which shows a roughly 7.7x speed increase, over MCRTs fastest optimization setting. Secondly, MCRT has not yet been demonstrated to work for machine learning applications where the initialization does not already lie very close to the target.

Ours	MCRT-4	MCRT-16	MCRT-64	MCRT -256
0.0234	0.1819	0.4485	1.6408	6.1288

Table 2: Running time (seconds) for one iteration (forward + backward). For MCRT, the more rays you sample, the higher running time it is. We are 7.7x faster than MCRT even when only 4 rays are sampled.

Softras [R5] We wish to emphasize that we treat Softras-Mesh [18] and Softras-Color [19] differently since they are released in Jan, 2019 and April, 2019, respectivley. Thus, we view Softras-Color as concurrent work. Our treatment of background pixels is inspired by Softras-Mesh, as stated in our paper. We agree that soft-z-buffer is a promising direction for occlusion, however our choice of complete face aggregation for background pixels also implicitly handles occlusion. In the first experiment (Sec. 5.1), we compare our method with Sofrtas-Mesh, which does not have a z-buffer. For the difference between use of the exponential and sigmoid function, we show the comparisons in the supplement. The exponential function is a better approximation of silhouettes, removing dim lines near face edges (Sec. 2 and Fig. 2), allowing our method to outperform Softras-Mesh on geometry prediction.

Other mentioned papers [R3,R4,R5] Petersen et al. estimates 3D geometry only and neglects lighting and texture. Szabo et al. only supports per-vertex color and approximates the gradient near boundary with blurring, which produces wired effects and can not cover the full image. Both produce objectively worse results than ours (Fig. 10 in Petersen et al. and Fig. 8 in Szabo et al. v.s. Fig. 3,4,5 in our paper). Insafutdinov et al. focus on rendering of point cloud and adopts a differentiable reprojection loss to constrain the distribution of predited point clouds, which loses point connectivity and cannot handle texture and lighting. We will included the suggested citations in a revision.

Explanation [R2,R3,R5]: R2. Z-buffer We recompute the z-buffer for each iteration. **R2. Camera optimization in Fig.2 (f)** We optimize camera parameters (rotation and translation matrices) via gradient descent, while fixing other parameters (vertex, light). **R2. Alpha channel** Besides RGB channel, we create an alpha channel that stores A_i, which represents the probability that pixel i is covered by the mesh. **R3. Quantitative results for lighting & texture** We compare lighting and texture in Tab. 3 of our paper. **R5. 3D GAN** Line 223 is a straight-forward GANs formulation, we have a second discriminator operating on texture map directly and this produces better results in line 299. **R5. Line 166 and figures** It should be "light, normal, reflectance and eye". We will improve the figures in a refined version.