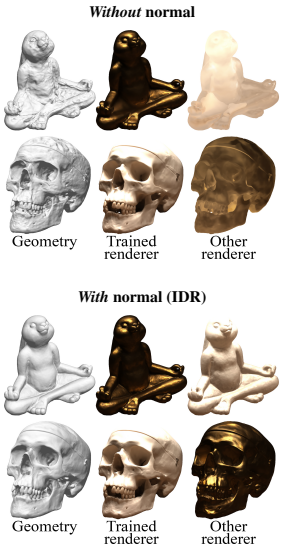
We thank the reviewers for their insightful comments. We next address questions and comments raised in the reviews.

R1, R3: There is no ability to disentangle lighting and material, the paper is misleading in that aspect. We will emphasize in the text that we only disentangle geometry and appearance, and that the appearance, which consists of material (BRDF) and light, is not further factored. We will also add to the conclusions section a statement that lighting and material are not separated, and mark it as an interesting future work. Therefore (and addressing the specific request of R1), we suggest changing the paper title to: **Multiview Neural Surface Reconstruction via Disentangling Geometry and Appearance**. In section 3.2 we will focus on approximating the surface radiance function $L(\boldsymbol{x}, \boldsymbol{n}, \boldsymbol{v})$ as a function of $\boldsymbol{x}$ (surface location), $\boldsymbol{n}$ (surface normal), and $\boldsymbol{v}$ (view direction). The rendering equation will only be used to motivate the dependence of L on $\boldsymbol{n}$.

R1, R2: For fixed geometry, surface light fields are defined only in terms of location ($\boldsymbol{x}$) and view direction (v) and are arbitrarily powerful, surface normals are therefore not necessary. In section 3.2 we will clearly state that, in theory, incorporating the surface normal ($\boldsymbol{n}$) is not necessary for producing a general surface light field. However, it is necessary for learning a general renderer (M) that is independent from *any specific geometry*. We will incorporate an empirical evidence, such as the inset, that shows the effect of incorporating normal in the renderer M. In this experiment we took two trained models (consists of geometry network f and renderer M), trained on two different DTU scenes, once without and once with normals in the renderer. The inset shows: the reconstructed geometry (left column); novel views using the trained renderer (middle column); and novel views rendered using the renderer from the other scene (right column). Note that using the normals in the renderer provides a better geometry-appearance separation: an improved surface geometry approximation as well as correct rendering of different geometries.



R2,R3: IDR cameras optimization vs. bundle-adjustment (e.g., Colmap SFM). Our method is a step towards end-to-end, simultaneous dense surface reconstruction and camera optimization using 2D image supervision. This is in contrast to MVS pipelines (e.g., Colmap MVS) that cannot handle noisy cameras without a pre-processing step of bundle-adjustment. In some cases, such as the "Fountain" scene, our method can go beyond bundle-adjustment accuracy.

R1, R2: Training and inference times are missing. Timings were accidentally dropped. Training time are: 6.5 or 8 hours for 49 or 64 images, respectively, on a single Nvidia V100 GPU. Rendering (inference) time: 30 seconds for 1200×1600 image with 100K pixel batches. All relevant details will be added to the text.

R2, R3: Unexpected lighting effects in the supplied video. In the original views (training images), the camera body occasionally occludes some light sources, casting temporary shadows on the object; see inset. Notice that in the video we move the camera (i.e., viewing direction) and not the object, therefore when the camera moves near such a view we see the projected shadow in the generated rendering.

R1, R2: Clarify architecture details, supply data and code. We will add exact architecture details to clarify how equation 3 is implemented, and will make the code and data available as well.

R2: "How were the 15 scans used in the results chosen?". They were chosen arbitrarily to span a wide range of different geometries and appearances. We did *not* cherry-pick results from a larger group of scans.

R2: "It would be stronger result to show PSNR on a held-out set of images". Indeed, in our experiments the PSNR was computed over training images. Following the reviewer's question we performed an experiment where we held-out 10% of the images as test views. Overall, we got 23.34 / 22.55 mean PSNR accuracy on the train / test images, respectively. We will report the per-scene accuracies in the paper.

R4: Comparison with other neural rendering methods. In this paper we focus on reconstruction of geometry. The mentioned papers focus on novel view generation. As the first inset above shows this is a different task. We therefore chose to compare to methods with a similar objective; we will nevertheless add these references to our previous work section.

R4: Results on data of complex scenes. It would be a very interesting future direction to handle non masked scenes.

Additional answers for R3: (1) DeepSDF and Occupancy papers do not learn geometry with 2D image supervision. (2) We show qualitative comparisons with DVR in Figure 4, second column. (3) As commonly assumed in calibrated SFM, we assume known intrinsics; we will clarify this in the text. (4) Intuition for Lemma 1 is given in lines 127-130.