We thank the reviewers for their valuable comments and suggestions. We are excited that the reviewers identified the importance of the problem (R3,R4), appreciated the novelty and technical contributions (R2,R3,R4), acknowledged our superior experimental results (R2,R3,R4), and found the paper professional and well-written (R2,R3,R4). We believe SDF-SRN takes a significant step towards real-world 3D object reconstruction that can be learned more practically from static image datasets. R4 affirmed all the merits above, yet thought our work lacked a few explanations (clarified below). Unfortunately, R1 misunderstood the supervision settings (L223), which led to ill-founded doubts of our contributions and credibility of our experiments. We address all raised concerns in the following. In addition, we try our best to resolve the misunderstandings and sincerely hope R4 will reconsider the rating and R1 will re-evaluate at an entirety.

(R1) **Misunderstandings.** "Knowledge of CAD model correspondence" refers to the assumption that one knows which images were created from the same CAD model (L217-224), so one could supervise with effectively all the associated viewpoints for an input image. This assumption was utilized in the original DVR and SoftRas, but *not* here in the more practical setup of *training* with *individual* images. It does *not* refer to camera poses, and we have *never* claimed that no viewpoint information is needed; on the contrary, we have stated they *are* part of the required data (L183,189).

(R1) **Overclaimed/Incorrect contributions.** We strongly disagree. Previous works [14,26,27,33] can infer 3D shapes from a single image, but they must be *trained* with *multiple views*. None of them have shown the ability of *learning* (not just *inferring*) implicit 3D shapes from single-view images. SDF-SRN can be *trained* with *single-view* supervision (L10,41,57) concurred by R2, R3, and R4, and we have provided strong results. We urge R1 to recognize the distinction.

(R1) **Validity of experiments.** We trained/tested on the same data as DVR, and all methods were run using the released implementations from the authors. DVR's different performance from the original is attributed to the modified setup (see above). SRN itself is *not* a 3D reconstruction method and does not produce 3D shapes (L82-87). Other baselines either did not release source code [26,27] or provided incomplete code [14] at the time of submission, preventing us from faithfully reproducing results. We chose to compare mainly against DVR for its reproducibility in a fairer setup.

(R1,R4) **Resolving ambiguities.** Although correspondences across instances are unavailable at the *pixel* level in single-view training, they still exist at the *semantic* level. CMR has shown initial success [15] in this regard with meshes. Here, SDF-SRN learns implicit features that best explains the object semantics within the category (L232-236). In turn, ray-marching discovers and associates implicit *semantic* correspondences in 3D, such that the ray-marched surfaces are semantically interpretable across all objects. Therefore, shape/depth ambiguities *can* be resolved (albeit not 3D-perfect) by learning to recover the *appearance* (with $\mathcal{L}_{\text{RGB}}$), a classical but important cue for disambiguating 3D geometry. In SDF-SRN, f (and thus $\mathcal{S}$) would be guided by g and h in ray-marching while also explicitly optimized with $\mathcal{L}_{\text{SDF}}$. The eikonal term has little to do with this mechanism. We hope this clarifies and will include discussions in the final version.

(R4) **DVR's drawbacks.** DVR learns by encouraging *binary occupancy* randomly along the rays within the silhouettes and vice versa, so it relies on other views to "carve" the same shape. Without view-instance association, DVR would wrongly encourage occupancy at self-occluded regions (L232-236,280-283). SDF-SRN does not rely on such sampling-based loss, but rather on learning to associate semantic correspondences within category (discussed above). Positional encoding is unrelated here, and the eikonal term is specific to SDFs. We thank R4 and will clarify in the final version.

(R4) **Repetitive artifacts.** We observed that these patterns came from the positional encoding component [31], which encodes input 3D points with periodic sinusoidal functions. We leave investigation on artifact reduction to future work.

(R3) **Robustness.** We thank R3 for the great suggestion. As requested, we tried PASCAL3D+ using *estimated* cameras from keypoints (L287) and found little performance degradation. We focus on ideal cameras/silhouettes in this work as the problem is very challenging already, but we agree robustness to such noises is definitely a valuable future direction.

(R3) **More categories?** We focused on airplanes, cars, and chairs following [15,27,40,42] as they are the most common. We thank and agree with R3 that evaluating on more/general categories will add to completeness of our experiments. We do believe our emphasis on chairs speaks well to SDF-SRN's versatility since chairs exhibit high shape variations.

(R3) **Why hypernetworks?** Conditioning inputs on latent code would in fact be *more* costly for ray-marching, since all unrolled iterations would directly depend on the encoder, resulting in slower backpropagation and higher memory footprint. Hypernetworks only need one forward/backward pass, allowing much cheaper training (empirically validated).

(R2) **Silhouettes.** DVR *does* need silhouettes for the occupancy/freespace losses, so we believe the comparison is fair. Regarding practicality, one could also obtain object silhouettes in real images from off-the-shelf instance segmentation methods (*e.g.* Mask R-CNN), which was also originally utilized in CMR [15] (please also see response to R3 on noises).

(R2) **Viewpoints.** The single views of ShapeNet were randomly selected from those in L210. The improvement under the multi-view setup is mainly attributed to the increase of training data, since images were always treated individually. We did observe slightly degraded results at peculiar viewpoints (*e.g.* wings might shrink at an airplane's side view).

(R2) **Runtime.** A batched forward pass of $\mathcal{E}$ to infer the shape parameters $\boldsymbol{\theta}$ takes ~10 ms, and rendering takes ~50 ms. We chose $N = 10$ steps following SRN [38] and did not observe improvements with more. Evaluating the quality at intermediate steps is inapplicable since the surfaces only intersect with the rays between the last two steps (L160).