We thank reviewers for their insightful comments. Please find below our answers to the questions.

**R1: Describe PerspectiveNet in more clear steps. Describe $g$, $K$, non-holes.** Thank you, we will add a clear overview of the algorithm as suggested and expand ln.86-94 with more concrete descriptions.

**R1 & R2: Move point-tracer from supplementary.** We agree and we will migrate the paragraph to the paper. Weighting with $\exp(-d)$ enables differentiability which is a mandatory requirement for optimizing $\ell_{cons}$.

**R1 & R3: BiGAN predictions noisy. Incorrectly trained?** During preliminary experiments, we observed "red flags" related to GANs, suggesting autoencoders are more suitable: (1) Training a state-of-the-art MSGGAN [Karnewar et al.: MSG-GAN ...] on SceneNet lead to unrealistic blurry results (fig. I). (2) Insufficient coverage of the image distribution, whose evidence was an inability to recover latent codes that lead to a correct reconstruction of arbitrary held-out images.

**R1: Hyperparam opt?** Grid search over 3 weights $\{10^{-i}\}_{i=0}^{2}$ for each of 3 losses on 100-scene subset of the train set.

**R1: Show more images**. As suggested, we will expand the supplementary with more qualitative results.

**R1: Why optimizing only non-holes of $\check{v}$ (Eq. 3)?** $\check{v}$ is a point cloud render and can contain holes. Minimizing $h(\hat{v}_{\bar{u}}, \check{v}_{\bar{u}})$ over holes $\bar{u}$ would make $\hat{v}_{\bar{u}}$ attain an unrealistic color of a hole (black by default) which is not desireable.

**R1: Blurry filling. Use GAN?** Unfortunately, as mentioned above, training a GAN lead to unrealistic blurry results.

**R1: The approach requires depth as input.** We have now implemented a method requiring ground truth (GT) depth solely at train time. We replaced the GT reference view depth with an output of a depth predictor [Laina et al.: Deeper ...] trained on the ScanNet train set. Again, PerspectiveNet outperforms other baselines (Table Ia).

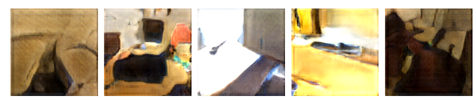
Figure I: ScanNet-trained MSGGAN samples.

**R2: Not mentioning depth as a required input.** We will update the text accordingly to avoid misleading readers.

**R2: Evaluation only on 1 dataset.** As suggested, we have now conducted evaluation on Matterport3D and SceneNet (same train/test protocol as for ScanNet). Note that SceneNet is synthetic and composed of ShapeNet objects and, hence, is more suitable for our scene-centric setting than the object-centric ShapeNet. Tables (Ib) and (Ic) contain results of our experiments. Similar to Tab. 1 in paper, PerspectiveNet outperforms other approaches. Unfortunately, due to limited amount of time, we could not finish all 3DConvNet experiments (we will include them in camera-ready).

**R2: Test GQN on real data?** We have now trained&tested GQN on ScanNet. GQN failed to learn and attained poor quantitative results - Ours/GQN: $\ell_1^{RGB}$=67.77/165.70, PSNR=13.79/6.96, LPIPS=0.434/0.687, $\ell_1^{D}$=0.109/0.513. The failure to learn probably occurs due to a greater complexity of ScanNet compared to GQNs' simplified synthetic scenes.

**R2: Is 3D ConvNet a contribution?** The 3D ConvNet was designed as a baseline we compare with.

**R2: Range/units of depth $d_u$?.** The depth is always expressed in meters. Range is rougly $[0.2, 7]$ meters.

**R2: Which layers for residuals?** $\Delta\phi^i$ were added after every "upsample&add" layer of FPN (four $\Delta\phi^i$ in total).

**R2: L209: Are the 8 views used for testing?** The 4 reference views provide all geometry and appearance conditioning. Hence, inpainting and evaluation happens only for 8 test views, for which we only know the camera parameters.

**R2: View clustering?** Given $N$ cameras, we KMeans-clustered the set of corresponding descriptors $\{\mathrm{vec}(g^i)\}_{i=1}^{N}$.

**R2: Loss weights? Train/test split?** $w(\ell_{style}, \ell_{cons}, \ell_R) = (0.1, 0.01, 0.1)$. Using official train/test split of ScanNet.

**R2: Explain perception of improvements in LPIPS / PSNR / l1.** PSNR and $\ell_1^{RGB}$ are sensitive to low-frequency image details while LPIPS better assesses image realism. Hence, the +8/-1% improvement of *PerspNet* over *PerspNet w.o. opt* in LPIPS / PSNR means that, while the local color distributions are roughly correct in both cases, adding the scene-consistent optimizer brings better image realism and an image-to-image consistent inpainting.

**R2: Performance analysis.** While PerspectiveNet brings better image quality, it is fair to admit that this comes at the cost of sub-real-time execution times (~20s per scene).

**R3: Discuss differences with [Meshry et al.].** We agree that there are similarities with the work of Meshry et al. [a] and we will cite this paper in Sec. 2. However, *our work differs substantially in*: (1) The task: While we focus on precise reconstruction of geometry and appearance of a scene given a limited amount of information in form of an image with large undefined regions, [a] is a form of stylization that aims at capturing a complete distribution of possible appearance variations of a, mostly hole-free, image. (2) Available data: [a] uses 1000s of reference images to reconstruct a scene that is later re-rendered. We use only 4 reference views, leading to large holes in new views and significantly harder inpainting problem. Furthermore, [a] requires semantic segmentation of the scene.

Finally, please note that [a] uses a BiGAN approach which we compare with in our work and outperform it significantly.

| Dataset | (a) ScanNet w/o test-time GT depth | | | | (b) SceneNet | | | | (c) Matterport3D | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | $\ell_1^{RGB}\downarrow$ | PSNR $\uparrow$ | LPIPS $\downarrow$ | $\ell_1^{D}\downarrow$ | $\ell_1^{RGB}\downarrow$ | PSNR $\uparrow$ | LPIPS $\downarrow$ | $\ell_1^{D}\downarrow$ | $\ell_1^{RGB}\downarrow$ | PSNR $\uparrow$ | LPIPS $\downarrow$ | $\ell_1^{D}\downarrow$ |
| PerspectiveNet | **93.819** | 11.193 | **0.515** | **0.505** | **51.722** | **15.442** | **0.521** | **0.214** | **38.905** | **19.108** | **0.404** | **0.226** |
| PerspectiveNet w/o opt | 94.333 | **11.224** | 0.537 | 0.516 | 61.493 | 14.950 | 0.564 | 0.280 | 42.173 | 17.722 | 0.457 | 0.384 |
| PartialConv | 96.742 | 10.948 | 0.515 | 0.606 | 80.612 | 12.218 | 0.545 | 1.984 | 46.741 | 17.119 | 0.411 | 0.647 |
| 3DConvNet | - | - | - | - | 75.942 | 12.614 | 0.614 | 0.653 | - | - | - | - |
| BiGAN | 156.958 | 7.194 | 0.715 | 0.666 | 99.358 | 11.106 | 0.637 | 0.841 | 118.614 | 9.940 | 0.613 | 1.286 |

Table I: Additional results on test sets of Matterport3D, SceneNet, ScanNet (will be included in camera-ready).