# Decoupled Illumination Priors for Spatially Controllable Multi-View Indoor Scene Relighting –Supplementary Material–

## 1 Additional Results on ScanNet++

We further present additional qualitative results on real scenes from ScanNet++ [2] in Figures 2–7. These examples provide a broader evaluation of our method on real indoor scenes beyond the representative cases shown in the main paper. For each scene, we show the input images, the corresponding spatial lighting conditions, and our relit outputs across multiple synchronized views. The results demonstrate that our method generalizes well to real-world captured environments, producing spatially controllable illumination effects while maintaining plausible shading, cast shadows, and multi-view consistency under diverse room layouts and lighting configurations.

## 2 User-controllable Lighting Interface

To enable explicit spatial control over scene illumination, we implement a Blender-based interactive interface that converts reconstructed scene geometry into an editable 3D lighting workspace, as illustrated in Figure 1. Rather than specifying illumination through ambiguous 2D cues, the user directly manipulates light sources in a coarse scene proxy that is aligned with the input images and recovered camera poses. The edited scene state is then rendered into the spatial lighting condition used by our framework, providing an intuitive bridge between user interaction and model conditioning.

### 2.1 Interface Overview

Our interface follows a simple workflow. Given a set of input views, we first recover per-view depth and camera parameters [1], and convert them into a coarse geometric proxy of the scene. This proxy is imported into Blender together with the corresponding camera poses, allowing the user to inspect the scene from the original viewpoints. The user can then add and adjust explicit light sources in 3D space, including their positions, orientations, and strengths. Finally, the edited scene is rendered to obtain the target lighting condition, which is used as the spatial control input for relighting.

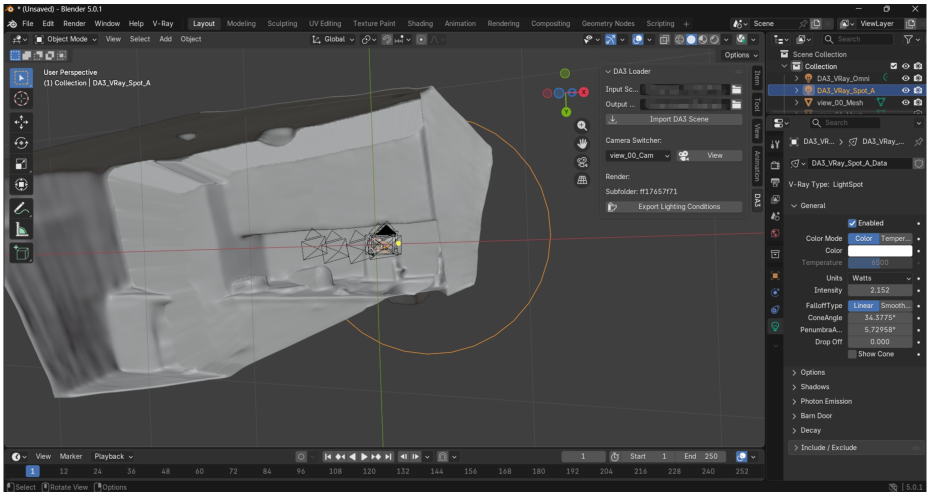

**Fig. 1:** Overview of our Blender-based user-controllable lighting interface. The interface converts reconstructed coarse scene geometry into an editable 3D workspace aligned with the recovered input cameras, allowing the user to place and adjust explicit light sources directly in 3D space. It supports interactive editing of Omni, Spot, and Rectangular Area lights, and renders the edited scene state into the spatial lighting condition used by our framework.

### 2.2 Coarse Geometry and Camera Initialization

To initialize the editable scene, we first construct a coarse geometric proxy from the predicted depth maps. For each input image, we unproject the depth map into 3D using the recovered camera intrinsics, producing a point cloud in camera coordinates. We then transform the points into a common world coordinate system using the predicted extrinsics and export both a colored point cloud and a triangle mesh representation. To improve geometric stability, we apply bilateral filtering to the predicted depth before unprojection, and remove triangles that span large relative depth discontinuities. This filtering suppresses spurious connections across object boundaries while preserving the dominant room layout and major occlusion structures.

In addition to the mesh, we also export the recovered camera metadata, including image resolution, intrinsics, and camera-to-world poses. The Blender interface reads these files and reconstructs the corresponding cameras inside the scene. As a result, the user edits illumination in a 3D workspace that is explicitly registered to both the coarse geometry and the source images.

### 2.3 Interactive Light Editing

Once the scene proxy and cameras are loaded, the user can interactively place and edit light sources directly in Blender. Our interface supports the canonical light primitives including Omni, Spot, and Rectangular Area lights. These light types provide complementary illumination behaviors, ranging from localized hard shadows to smoother area-based lighting. For each light, the user can modify its spatial position, orientation, intensity, and other type-specific attributes, such as spotlight direction or area-light extent.

### 2.4 Rendering and Lighting Condition Export

After the user finishes editing, we render the current scene state to obtain the final lighting control signal. The rendering can be performed for the full set of aligned input cameras. Since all cameras share the same edited 3D lighting configuration, the exported lighting conditions are naturally synchronized across views.

## References

1. Lin, H., Chen, S., Liew, J.H., Chen, D.Y., Li, Z., Shi, G., Feng, J., Kang, B.: Depth anything 3: Recovering the visual space from any views. arXiv preprint arXiv:2511.10647 (2025) 1
2. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 12–22 (2023) 1

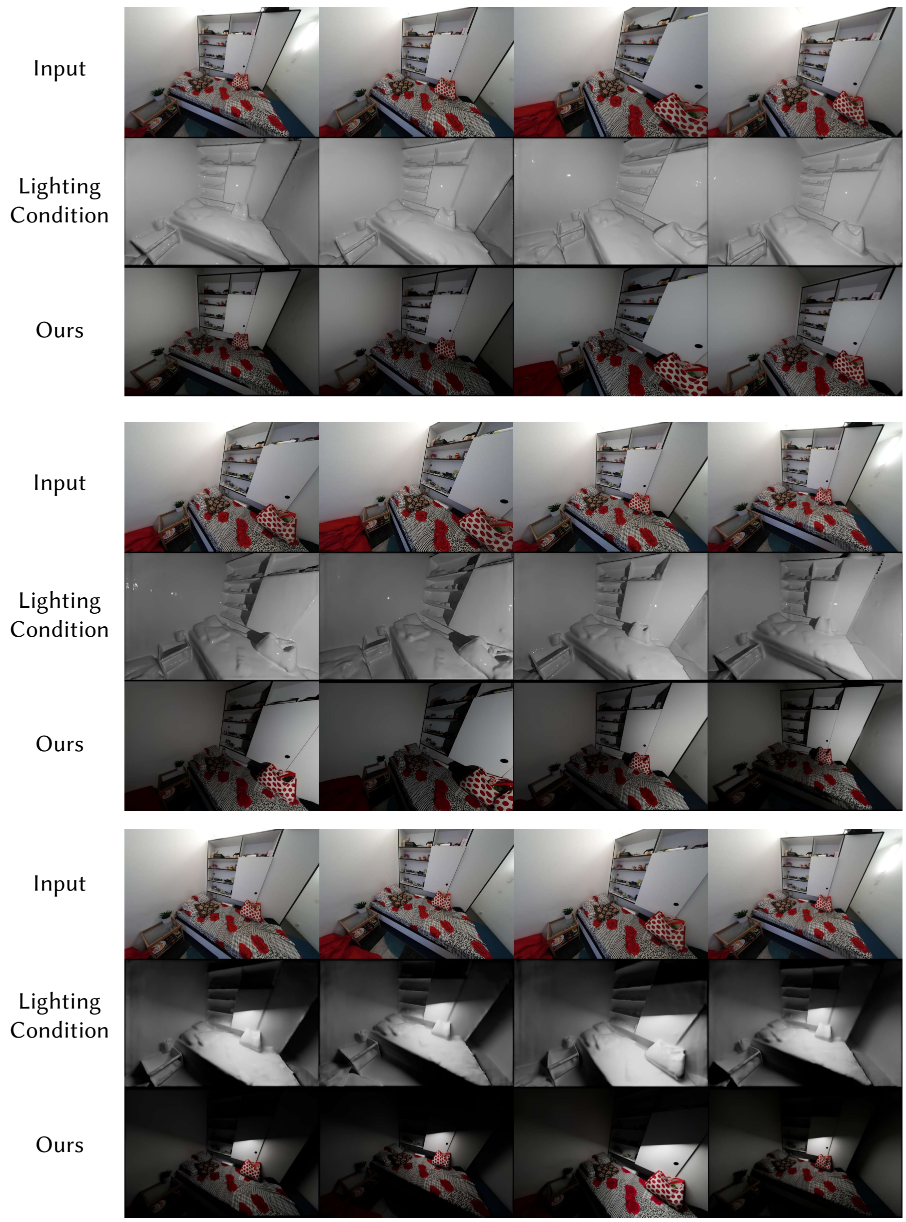


**Fig. 2:** Our relighting results on ScanNet++ (I).

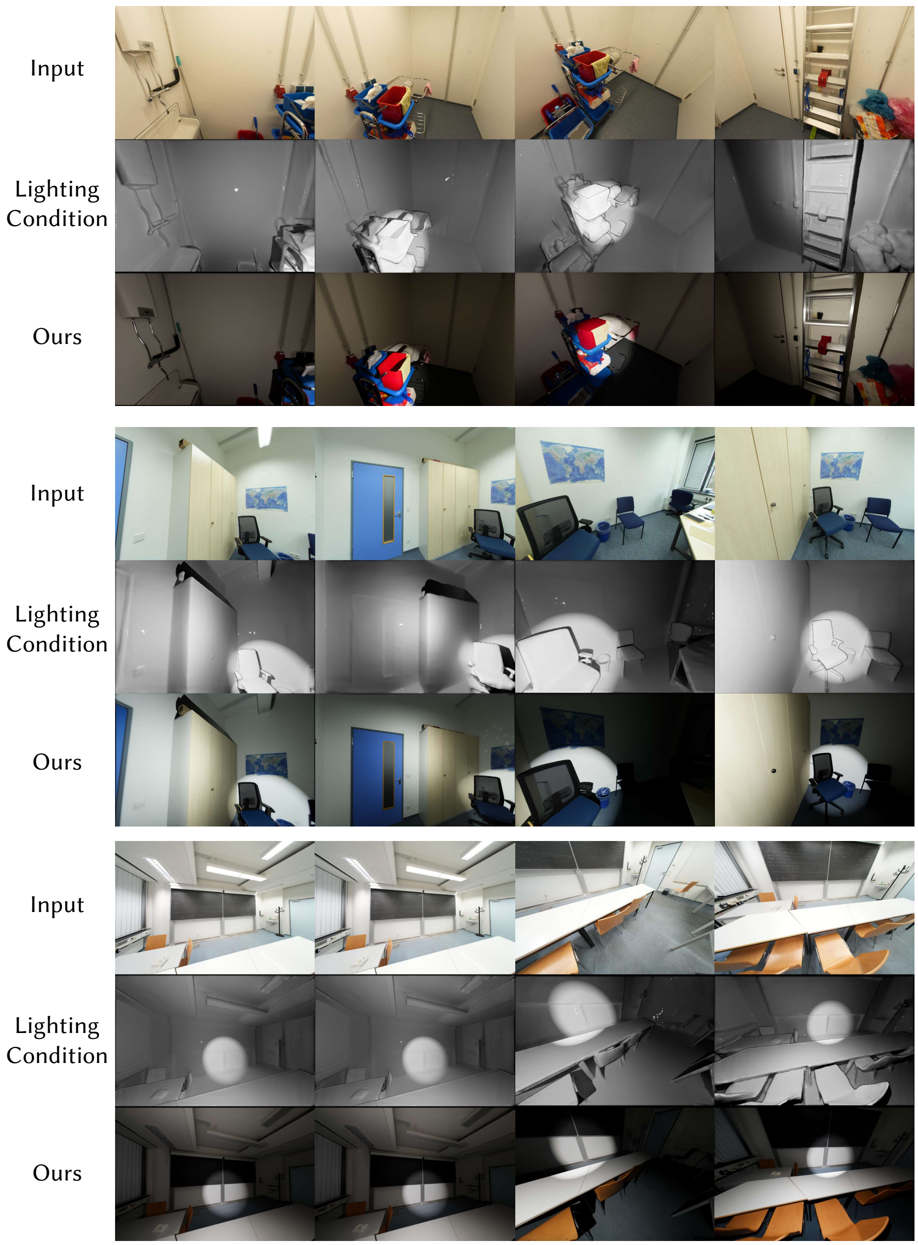


**Fig. 3:** Our relighting results on ScanNet++ (II).

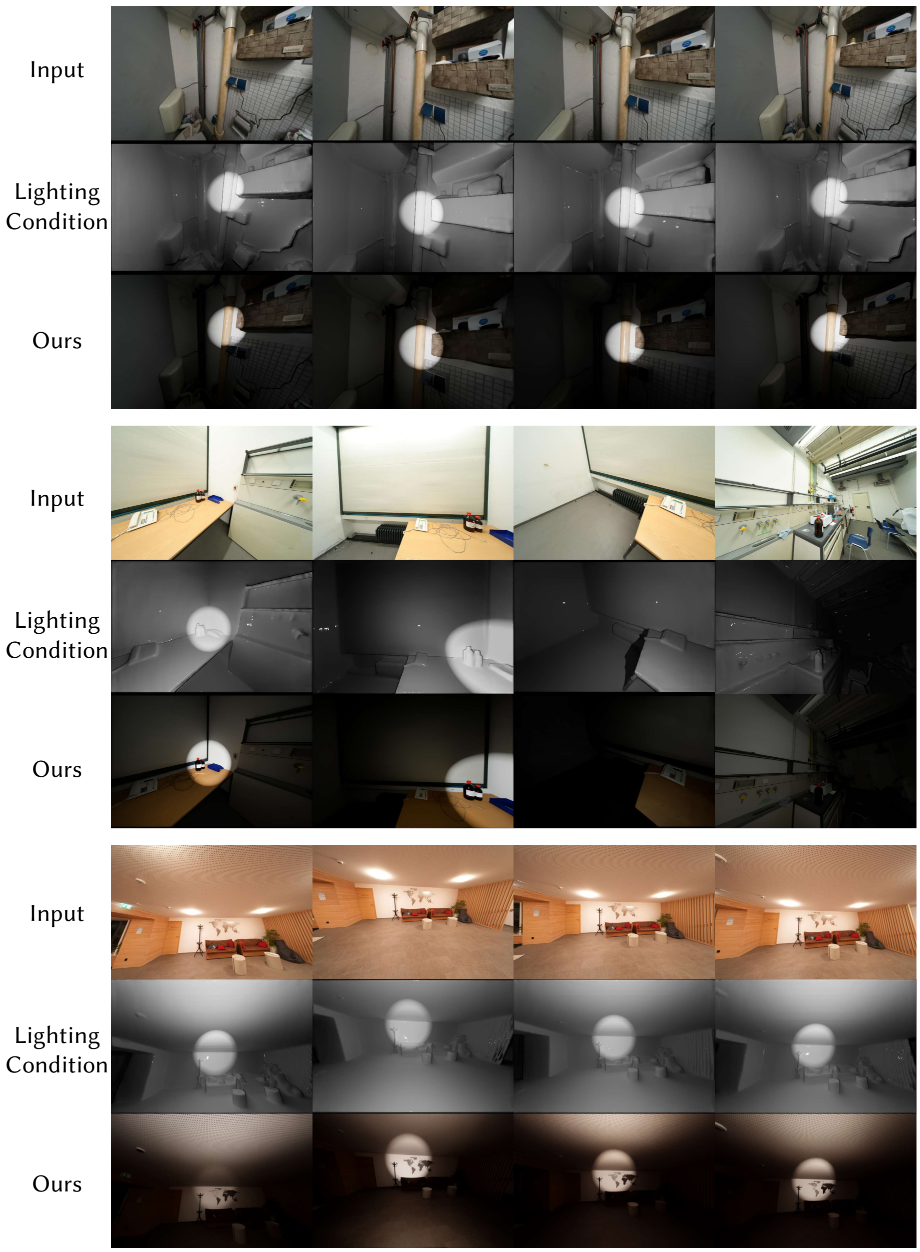


**Fig. 4:** Our relighting results on ScanNet++ (III).

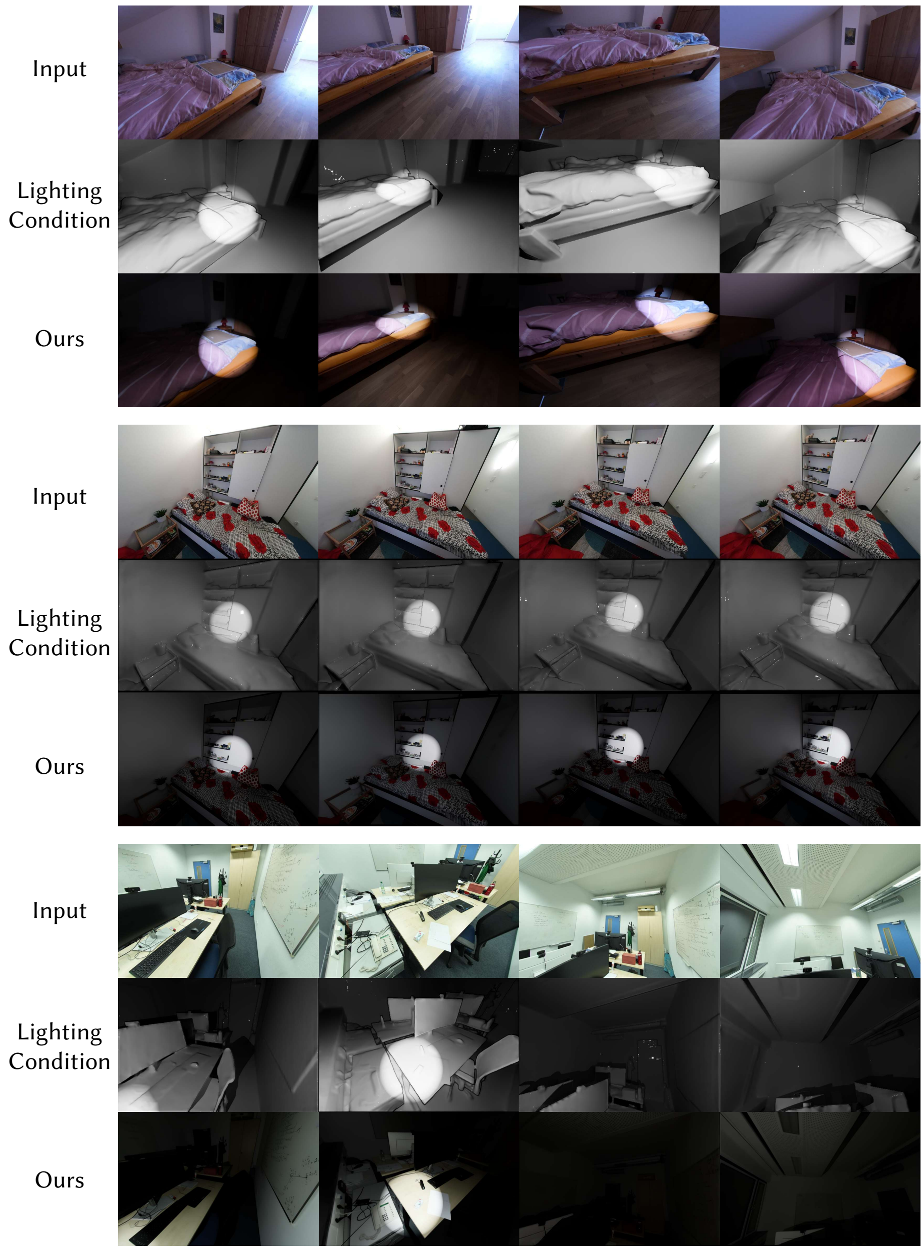


**Fig. 5:** Our relighting results on ScanNet++ (IV).

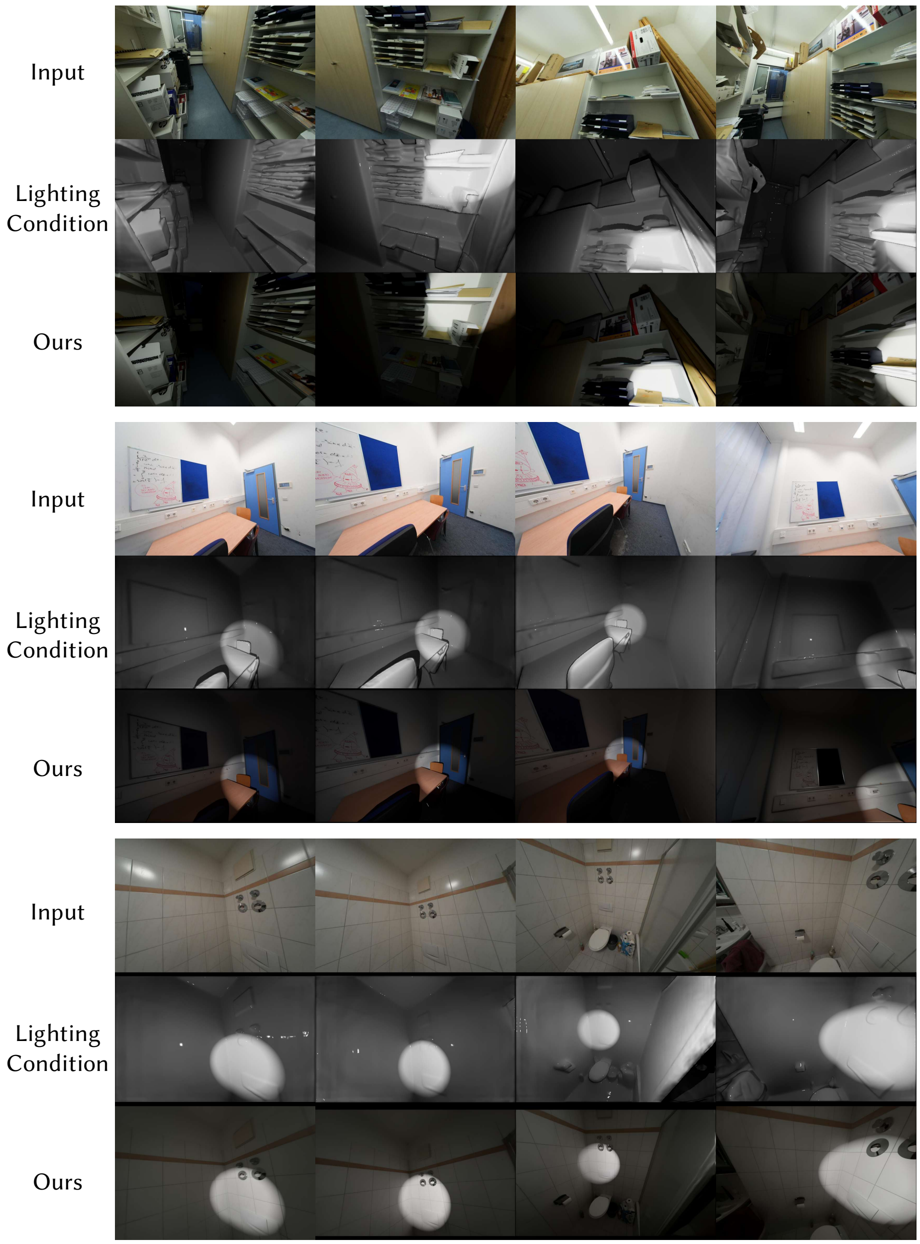


**Fig. 6:** Our relighting results on ScanNet++ (V).

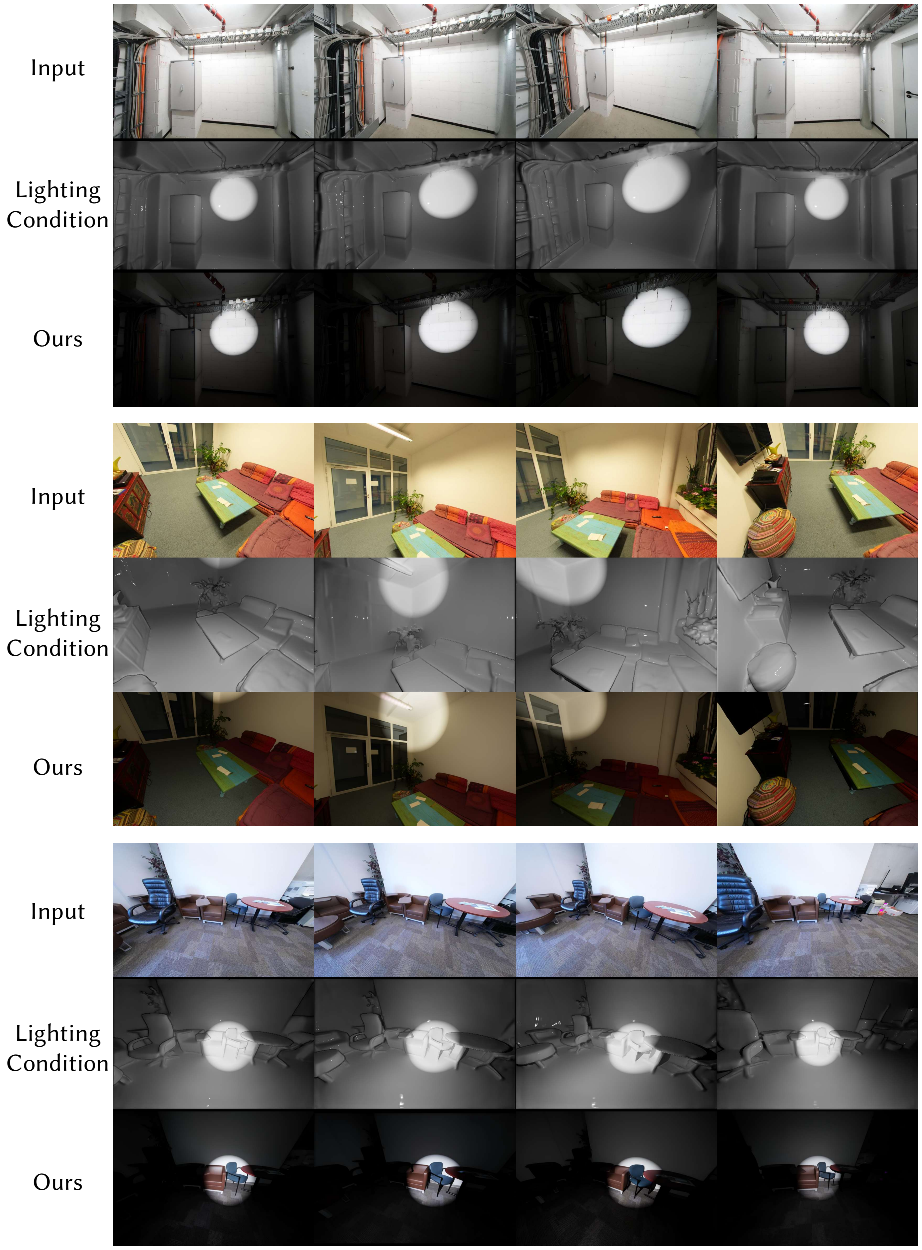


**Fig. 7:** Our relighting results on ScanNet++ (VI).