Spatially-Varying Outdoor Lighting Estimation from Intrinsics: Supplementary Material

Yongjie Zhu[1 †] Yinda Zhang[2] Si Li[1 *] Boxin Shi[3, 4 *]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[2]Google [3]NELVT, Department of Computer Science and Technology, Peking University
[4]Institute for Artificial Intelligence, Peking University

In this supplementary material, we provide more details about our implementation details, synthetic data generation, network architectures, and additional results on synthetic data and real images in the wild.

A. Implementation Details

To train SOLID-Net, we use SOLID-Img augmented with random flip and crop. Our framework is implemented in PyTorch [3] and Adam [2] optimizer is used with default parameters. We first train I-Net using a batch size of 8 for 20 epochs until convergence, and then train P-Net with a batch size of 4 for 60 epochs on an RTX2080 GPU. We find that an end-to-end fine-tuning does not improve the performance. The learning rate is initially set to 5×10^{-4} and halved every 5 epochs for both networks. Training convergence takes roughly 24 hours.

B. Additional Details in Data Generation

The data generation pipeline for **SOLID-Img** dataset has been introduced in Section 3.1 of the main paper. Here, we introduce additional details about the **camera setting** step.

For each road block, we select a set of cameras with diverse views seeing most objects in the context, to provide comprehensive information for lighting estimation, as shown in Figure 2(a) of the main paper. Our process starts by selecting the "best" camera [6] for each of the six horizontal view direction sectors in every road block. For each horizontal view direction, we sample a dense set of cameras on a 2D grid with 1m solution, choosing a random camera viewpoint within each grid cell, a random horizontal view direction within the 60° sector, a random height of 1.55 ± 0.05m above the floor uniformly, and a pitch angle within 10° around horizontal direction. Then for each camera, we render an item buffer and count the number of visible pixels according to Z-buffer and the number of objects. For each horizontal view direction in each road block, we select the view with the highest percentage pixel coverage, as long as it has more than three object categories.

Table A: Numerical results and MAE errors on estimated sky environment map.

Methods	$\xi_{azimuth}$	$\xi_{elevation}$	ξ_{HDR}
$\mathbf{LENet_{reg}}$	37.0°	16.2°	0.508
$\mathbf{LENet_{ae}}$ [5]	34.0°	16.0°	0.542
$\mathbf{LENet_{sky}}$ [1]	22.3°	11.0°	0.609
Ours ($w/o\ \mathcal{L}_{\text{dif}}$)	19.1°	11.4°	0.491
Ours	$\mathbf{12.6}^\circ$	$\mathbf{8.5}^\circ$	**0.478**

C. Details about Network Architectures

In this section, we introduce the detailed network architectures of baseline models as shown in Figure A.

The first baseline model, denoted as $\mathbf{LENet_{reg}}$, is a regression model that directly regresses the global sky environment map from the input limited-FOV image. The second baseline model, denoted as $\mathbf{LENet_{ae}}$, is a two-stream convolution network used to estimate sun position and normalized HDR panorama from a LDR panorama [5]; we modify the input as a single limited-FOV LDR image to adapt our task. The last baseline model, denoted as $\mathbf{LENet_{sky}}$, learns to estimate both the sun azimuth and the sky parameters from two image encoders and uses an autoencoder to learn the space of outdoor lighting by compressing an HDR sky image to a 64-dimensional latent vector and reconstructing it to a HDR sky environment map [1]. In particular, $\mathbf{LENet_{ae}}$ learns azimuth estimation as a regression task, while $\mathbf{LENet_{sky}}$ treats it as a classification task.

Numerical results about these baseline models compared with ours are shown in Table A ($\xi_{azimuth}$ is the sun azimuth angular error, $\xi_{elevation}$ is the sun elevation angular error, and ξ_{HDR} is the MAE error of normalized sky environment map). Training a single model to estimate a sky environment map with azimuth rotation is proved to be difficult (see Row 1 in Table A). By separating this task into the estimation of a normalized sky environment map (sun

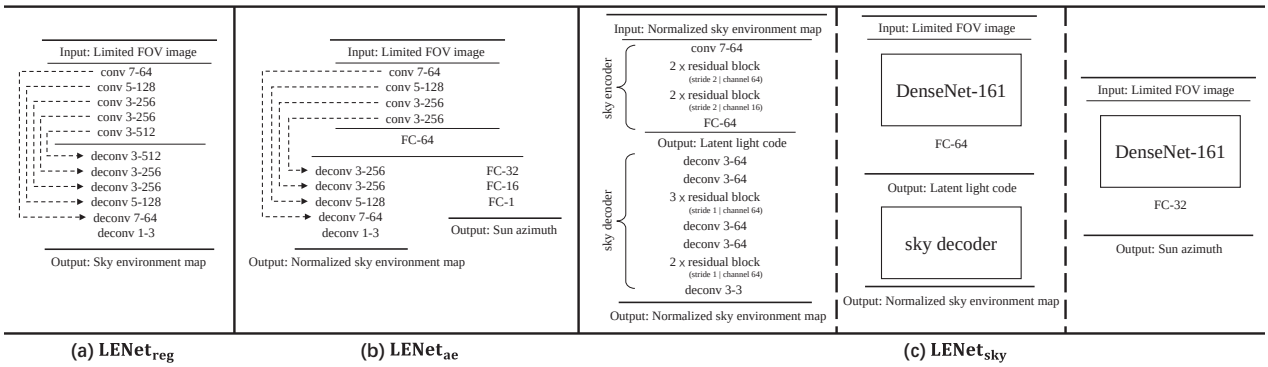

Figure A: Structures of baseline models.

in the middle position along the horizontal direction) and sun azimuth angle, the results of single model lighting estimation get improved (see Row 2 in Table A). Further using three sub models to solve the whole problem results in more accurate estimation than using single models (see Row 3 in Table A). But the sun position and sky environment map are closely related, we prove that the estimation accuracy to both could be improved by our jointly training with intrinsic constraints being integrated in deep models (see Row 4-5 in Table A).

D. Qualitative Results on Synthetic Data

More results of image decomposition and lighting estimation using SOLID-Img test dataset are shown in Figure B, Figure C, and Figure D. Given an input image, our estimated albedo, normal, plane distance, shadow, and shading show close appearance to the ground truth (shown as insets) as shown in Figure B. In Figure C, we can see that global lighting estimation resuls of SOILD-Net are closer to ground truth than three baseline models (we rotate the lighting of $\mathbf{LENet_{ae}}$ and $\mathbf{LENet_{sky}}$ according to the sun azimuth for better comparison). In addition, the relighted bunnies using our estimated lighting display accurate cast shadows, while other models fail to render.

As shown in Figure D, our method can recover more accurate local lighting than NeurIllum [4] even for the reflection of the ground and some unseen parts (typical examples could be found in local lighting 1 in row 1, local lighting 3 in row 4, and local lighting 4 in row 5). We conjecture this is because our multi-input (global lighting for lighting information, shadow for occlusion information) module provides more useful information.

E. Qualitative Results on Real Data

More results of local lighting estimation on real dataset are shown in Figure E. Compared with NeurIllum [4], the lighting estimation results of our method are more similar to the ground truth in terms of overall structure, and our relighted bunnies show more realistic rendering apperances.

References

[1] Yannick Hold-Geoffroy, Akshaya Athawale, and Jean-François Lalonde. Deep Sky Modeling for Single Image Outdoor Lighting Estimation. In *Proc. of Computer Vision and Pattern Recognition*, 2019. 1

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 1

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of Neural Information Processing Systems*, 2019. 1

[4] Shuran Song and Thomas Funkhouser. Neural Illumination: Lighting Prediction for Indoor Environments. In *Proc. of Computer Vision and Pattern Recognition*, 2019. 2

[5] Jinsong Zhang and Jean-François Lalonde. Learning High Dynamic Range from Outdoor Panoramas. In *Proc. of International Conference on Computer Vision*, 2017. 1

[6] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-Based Rendering for Indoor Scene Understanding Using Convolutional Neural Networks. In *Proc. of Computer Vision and Pattern Recognition*, 2017. 1

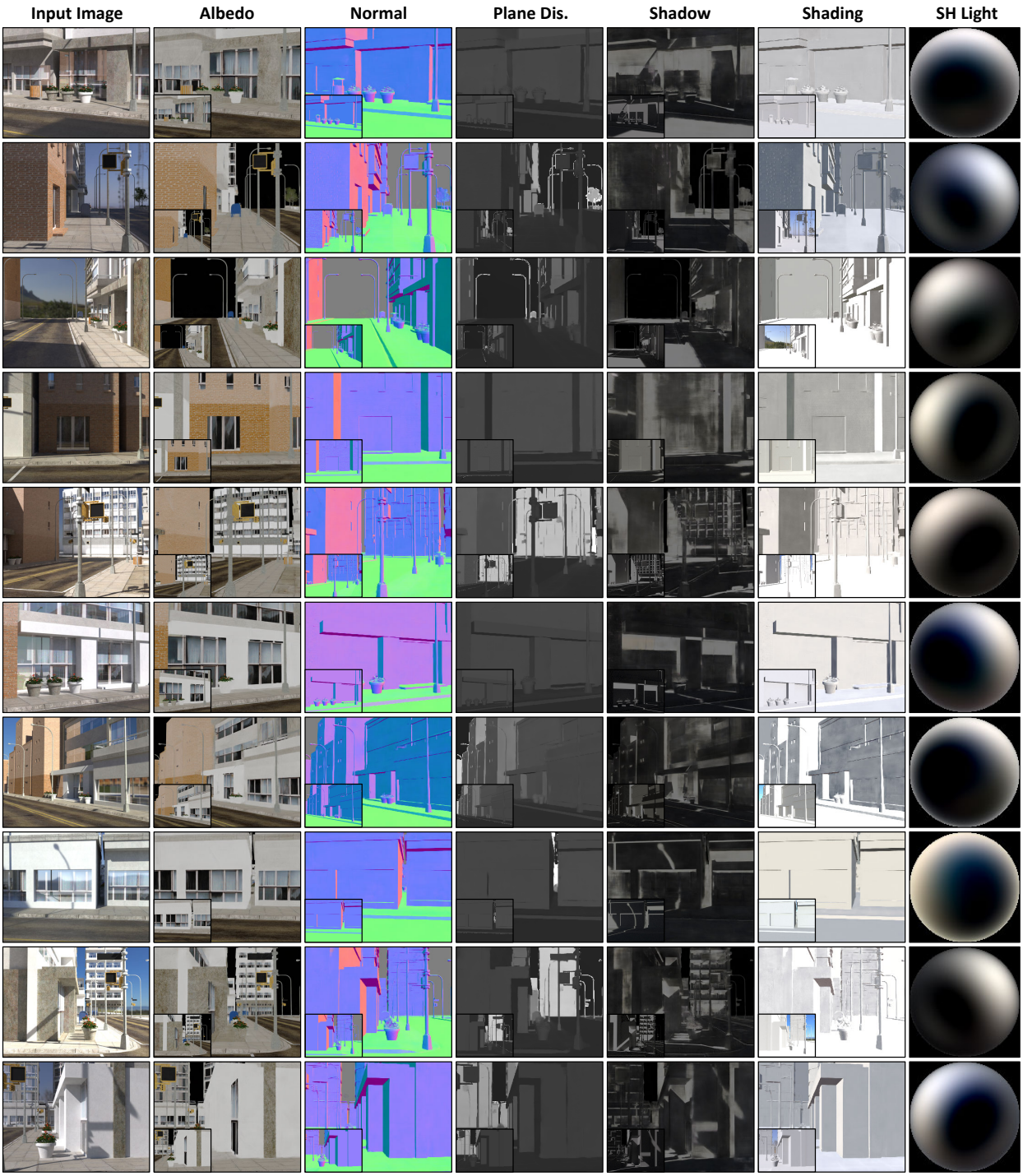

Figure B: Intrinsic decomposition results and ground truth (shown as insets) on SOLID-Img test dataset.

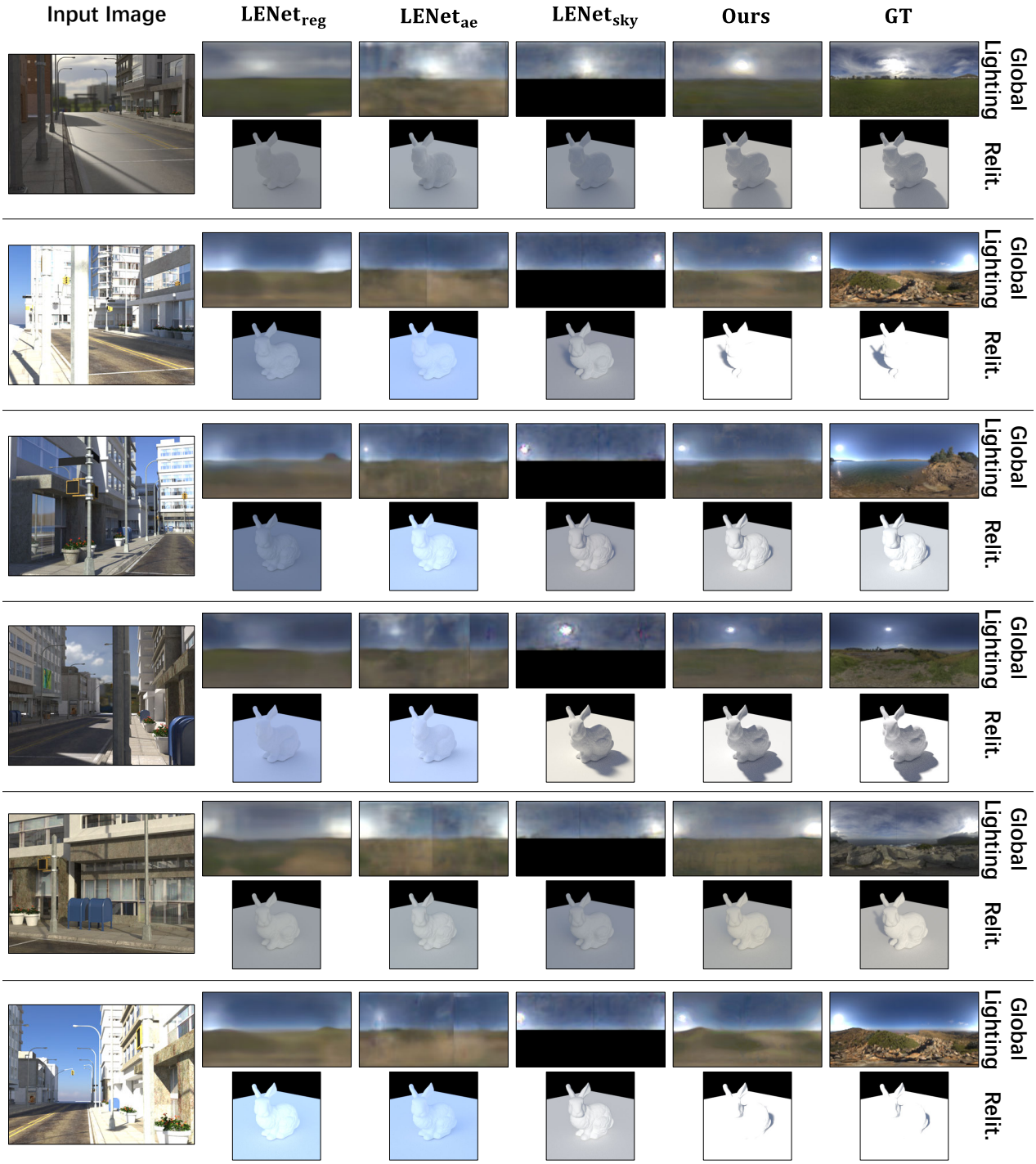

Figure C: Global lighting estimation and relighting results on SOLID-Img test dataset.

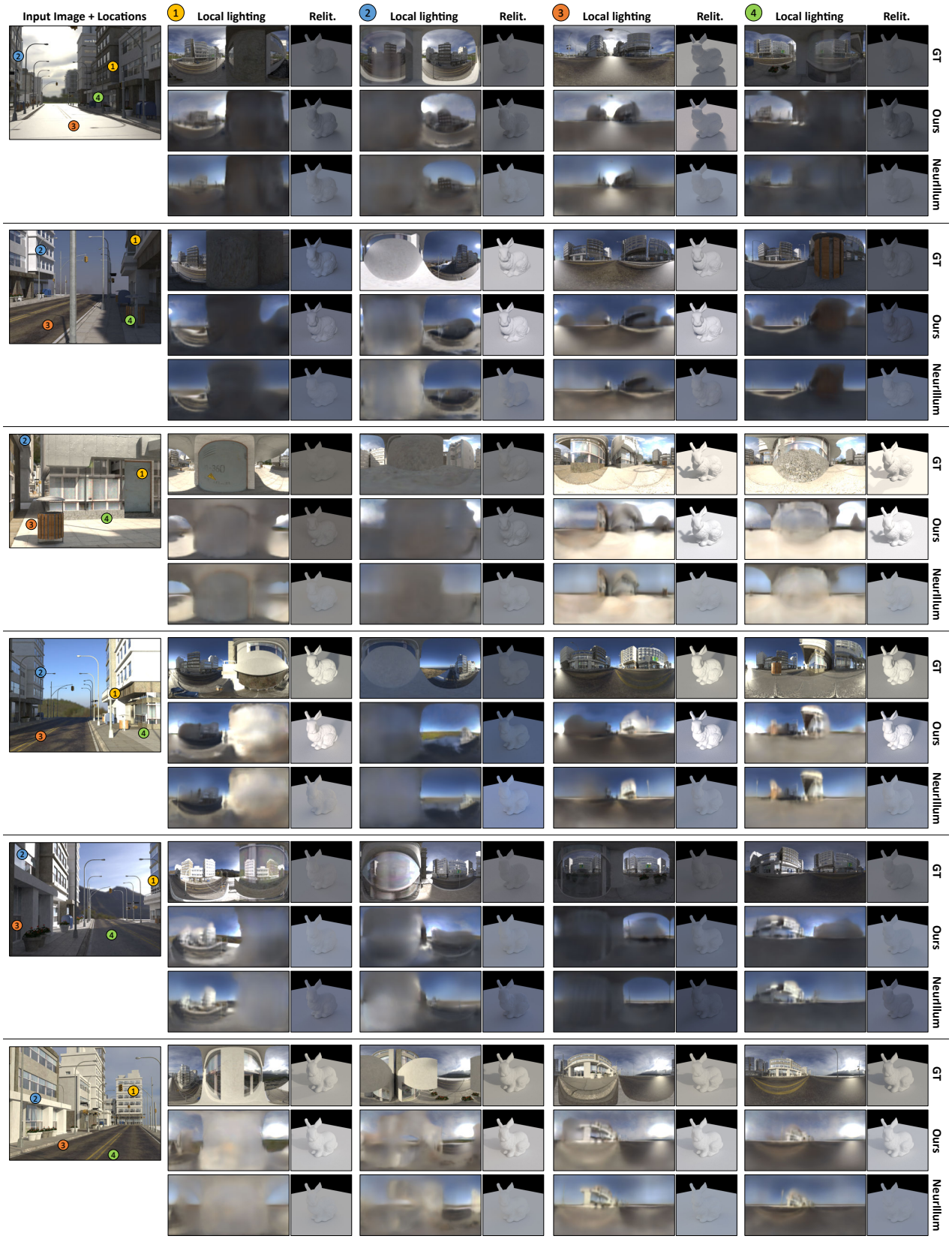

Figure D: Local lighting estimation and relighting results on SOLID-Img test dataset.

Figure E: Spatially-varying lighting estimation and relighting results on real dataset.