Supplementary Material for IRISformer: Dense Vision Transformers for Single-Image Inverse Rendering in Indoor Scenes

Rui Zhu[1] Zhengqin Li[1] Janarbek Matai[2] Fatih Porikli[2] Manmohan Chandraker[1]
[1]UC San Diego [2]Qualcomm AI Research
{rzhu,zhl378,mkchandraker}@eng.ucsd.edu {jmatai,fporikli}@qti.qualcomm.com

1. More Results

1.1. OpenRooms

We include more examples of material, geometry and lighting estimation on OpenRooms in Fig. 1. In general we demonstrate more spatially consistent material estimation (*e.g*. the textured floor in bright and shadowed areas in sample 1, or the surface of the sink table in sample 2), better geometry in challenging lighting (*e.g*. the shape of the sink bowl in sample 2, consistency of the flat wall in samples 3 and 4), as well as fewer artifacts in re-rendered results as a result of overall better estimation of all factors.

1.2. Real World Images

We include more examples of material, geometry and lighting estimation on real world images from Garon *et al*. [5] in Fig. 2. We arrive at similar observations on comparisons of geometry, material and lighting as in the previous subsection, and we prove that our methods generalize well to real world indoor images in the wild.

1.3. IIW

We include more examples of albedo estimation on IIW in Fig. 3 to showcase our state-of-the-art albedo estimation from our models finetuned on IIW, with better spatial consistency and rich details.

1.4. NYUv2

We include more examples of depth and normal estimation on NYUv2 in Fig. 4 to show improved geometry estimation compared to previous works in multi-task inverse rendering setting.

1.5. Object Insertion

We show in Fig. 5 more samples of virtual object insertion where we achieve with more photorealistic results. In particular, we show more physically plausible lighting with better spatial consistency (*e.g*. lighting on the bunnies which sit against major light sources to the camera in sample 2 and 4), strong and accurate directional lighting (*e.g*. the bunnies around the lighting sources in sample 1 and 2 where they are properly lit up or darkened according to their relative position and orientation w.r.t. the kitchen/desk lamps).

1.6. Material Editing

We show in Fig. 6 an additional material editing result. We observe that our method can recover spatially-varying lighting, with the rendered result similar to that of prior state-of-the-art [7].

1.7. Attention Maps

We include more examples of the attention maps of selected patches on real world images in Fig. 7. Our model learns to attend to various semantic regions within the image (*e.g*. the entire object, other objects, area of highlights or shadows, *etc*.) to update the token (feature) for a patch, without explicit supervision of semantic regions. This attention across potentially long-range interactions results in better disambiguation of the shape, material and lighting factors.

2. Model Design Details

2.1. Detailed Architecture

The modules in IRISformer mostly follow the design of DPT-hybrid [9] as detailed in Fig. 8. We denote the convolutional operation as `conv(k,s,p,c)` where `k` is the convolutional kernel size, `s` is stride, `p` is padding, and `c` is output channels. `BN` is for batch normalization, `upsample` is for 2x bilinear interpolation. All convolution layers are followed with *ReLU* activation unless otherwise stated, and with Batch Normalization except in `Head` where BN is only optionally applied to the second convolution layer (further comparison included in Subsection 2.2). We use the same residual attention module in Transformer layers as in DPT [11] or ViT [3] where each layer is followed by *GELU* activation [6] and Layer Normalization [1].

Head design. We use BN in all heads of **BRDFGeoNet** as we found it to be useful to stabilize training. However

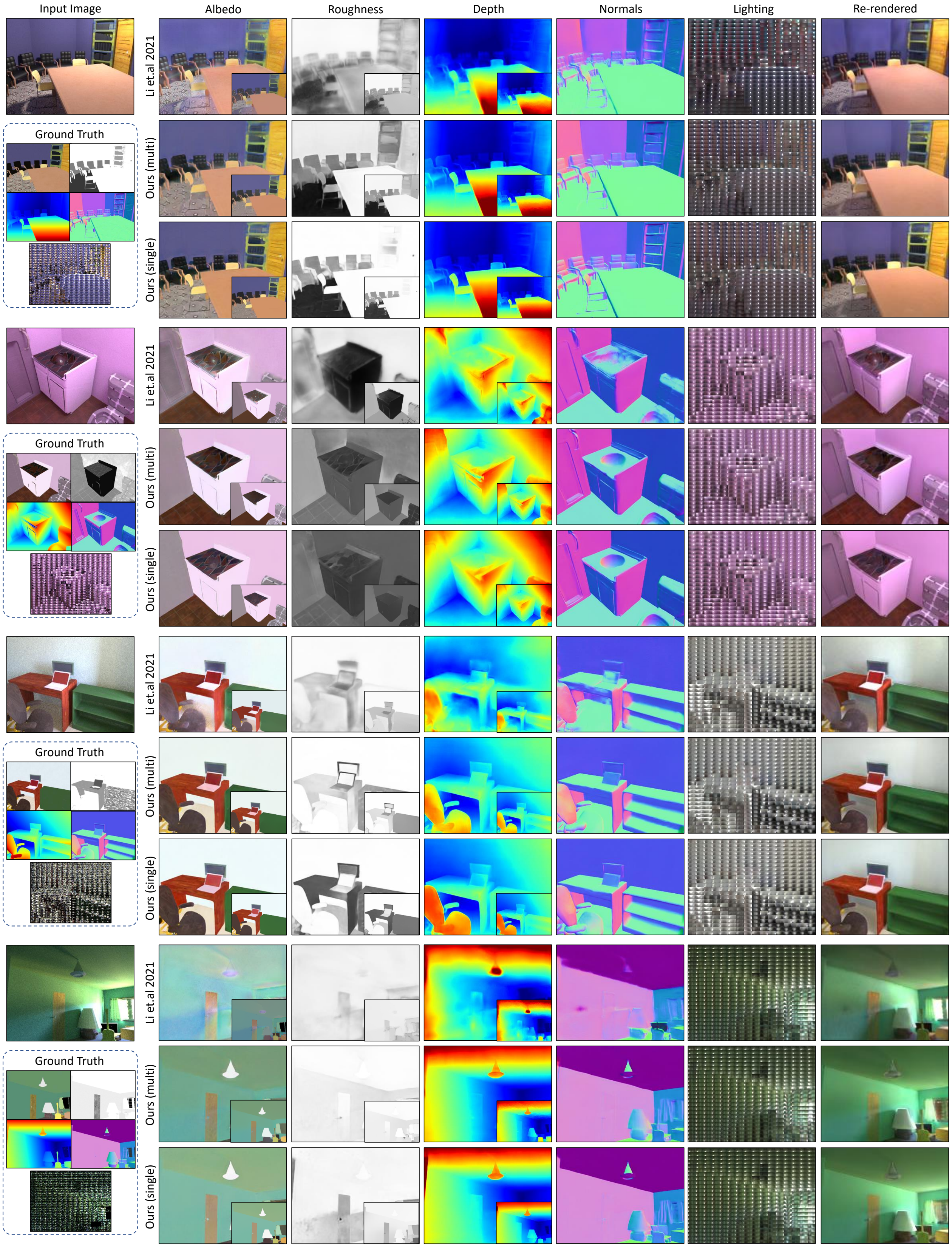

Figure 1. Additional BRDF, geometry and lighting estimation on OpenRooms. Small insets (best viewed when enlarged in PDF version) are estimations processed with bilateral solvers (BS).

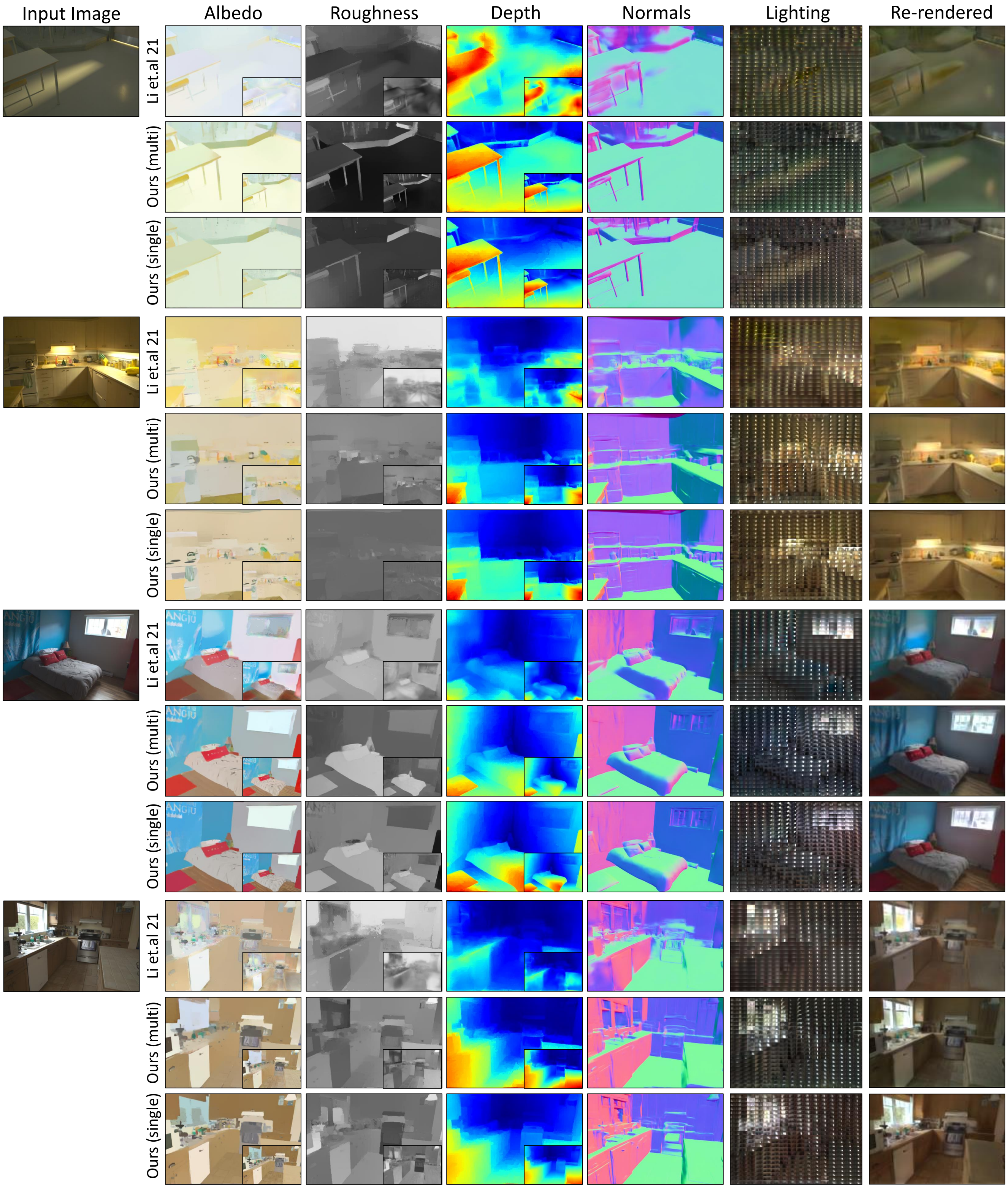

Figure 2. Additional BRDF, geometry estimation, per-pixel lighting and re-rendering results on Garon *et al*. [5] (after BS). Insets are results before BS.

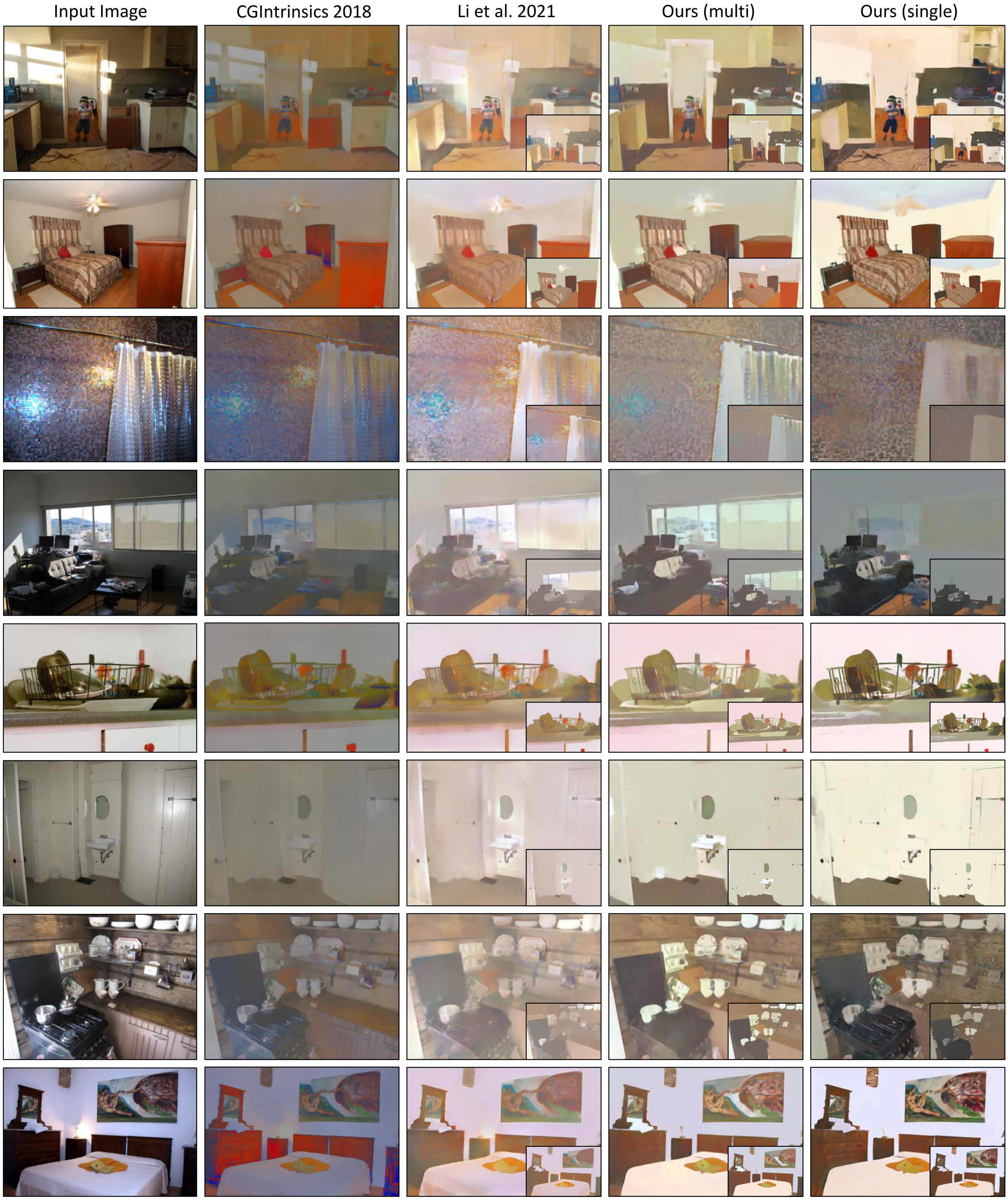

Figure 3. Additional intrinsic decomposition results on IIW [2] (all before BS). The inset figure within each result is the result after BS.

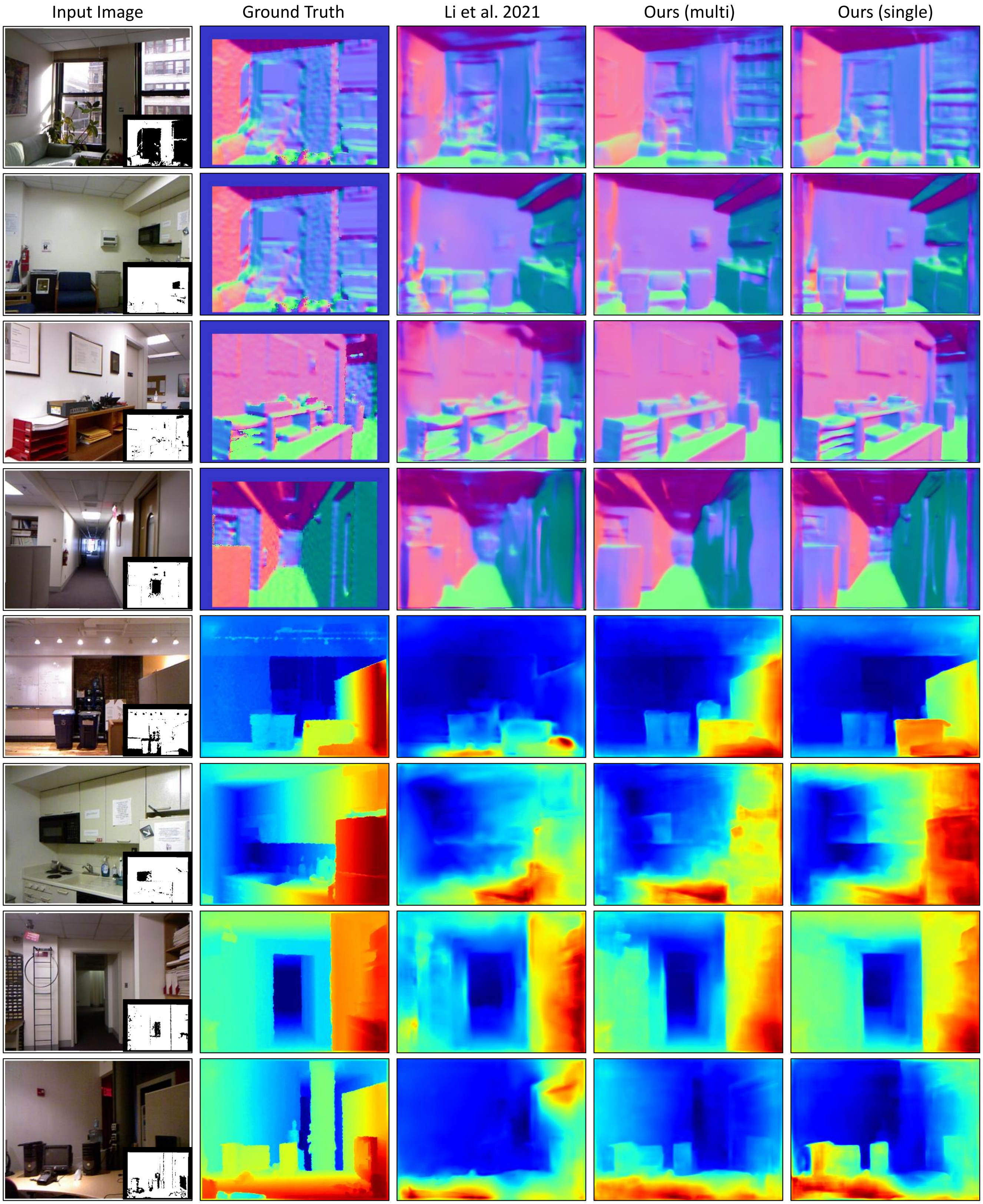

Figure 4. Additional geometry estimation results on NYUv2 [12] (all without BS).

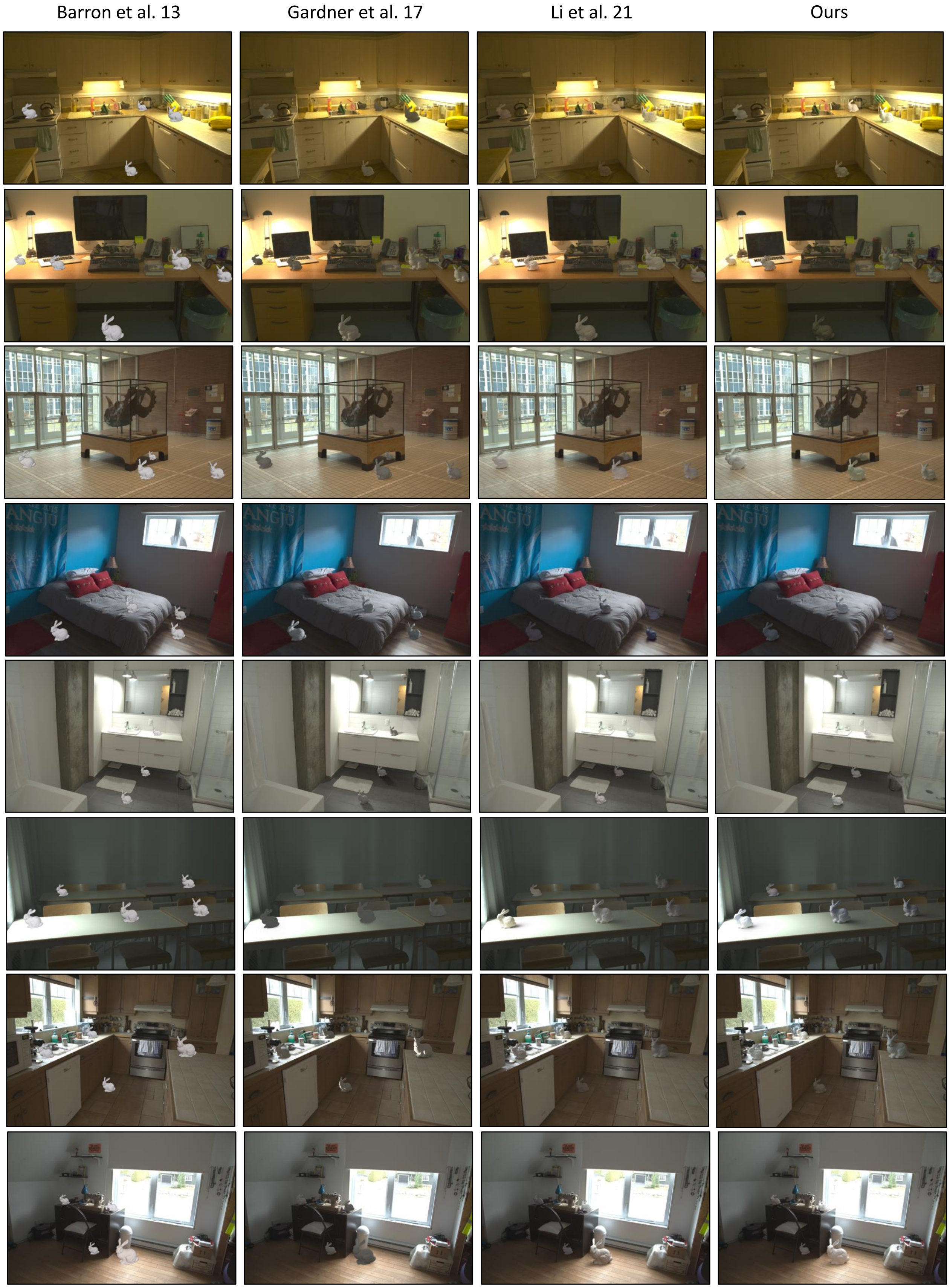

Figure 5. Additional virtual object insertion results.

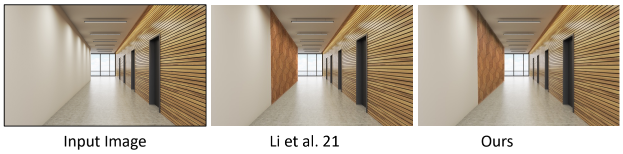

Figure 6. An additional material editing result.

we choose not to use BN in **LightNet** heads as it will tend to force the model to converge to a local minimum which produces blurry results. The choice of BN is mostly empirical but we include additional comparison on BN in Table 3 where we disable BN for **BRDFGeoNet** or enable BN for **LightNet** and compare with previous results to support our choice.

All heads share similar architecture as illustrated in Fig. 8 except for the `Post-process` layer where non-trainable normalization or activation operations are applied depending on the specific domain. For albedo, normals and roughness we use *tanh* activation so that the output is constrained in a closed range of [-1, 1] due to the physical nature of those properties. Albedo and roughness prediction are later rescaled to [0, 1] while normals are normalized to have unit L2 norm. For depth, we also use *tanh* activation plus scaling to predict inverse depth within (0, 1] and then invert to linear depth space.

For **LightNet** heads of bandwidth of SGs $\{\lambda_k\}_{k=1}^K$ and intensities $\{\mathbf{f}_k\}_{k=1}^K$ we again use *tanh* and re-scale output to [0, 1]. For estimating axes $\{\xi_k\}_{k=1}^K$, we use the same rule as estimating normals via *tanh* plus normalization.

2.2. Ablation Study on Model Design

We include comparison on several design choices and report the performance. The comparisons include (1) sharing or not sharing decoders in **BRDFGeoNet** in multi-task setting, where we provide comparison on joint estimation of albedo+roughness (material tasks) in Table 1 or albedo+normal (material and geometry jointly estimated) in Table 2, considering the full multi-task model with 4 independent decoders does not fit into our hardware for training; (2) BN or no BN, where we compare in single-task estimation of depth, albedo and lighting in Table 3; (3) 4 layers in encoder/decoder vs. 6 layers, in single-task models in Table 4 as an addition to Table 5 in the main paper. We conclude that, (1) there is not a significant performance drop by sharing decoders, or using 4 layer in Transformers, but there are benefits for significant memory saving; (2) BN works better for **BRDFGeoNet** heads while no BN is preferred for **LightNet** heads. We demonstrate that, by empirically comparing those choices we arrive at our final architecture, which achieves reasonable trade-off between memory cost and accuracy.

al+ro	sharing decoders	not sharing decoders
Model Size (MB)	1,206	1.606
Inference (ms)	34.4	42.3
$L_{\mathbf{A}}$	**0.50**	0.51
$L_{\mathbf{R}}$	1.91	**1.88**

Table 1. Analysis on whether to share decoders in multi-task joint estimation of albedo and roughness: comparison on model sizes, inference speed and losses.

al+no	sharing decoders	not sharing decoders
Model Size (MB)	1,206	1.606
Inference (ms)	34.4	42.3
$L_{\mathbf{A}}$	**0.51**	**0.51**
$L_{\mathbf{N}}$	1.88	**1.85**

Table 2. Analysis on whether to share decoders in multi-task joint estimation of albedo and normals: comparison on model sizes, inference speed and losses.

	BN	no BN
$L_{\mathbf{A}}$ in single-task albedo estimation	**0.43**	0.51
$L_{\mathbf{N}}$ in single-task normal estimation	**1.89**	1.92
$L_{\mathbf{light}}$ in multi-task lighting estimation	13.23	**12.54**

Table 3. Analysis on whether to use BN in output heads in single-task estimation of albedo and normals, as well as multi-task estimation of lighting: comparison on losses.

	single-4	single-6
$L_{\mathbf{A}}$	0.48	**0.43**
$L_{\mathbf{R}}$	1.93	**1.91**
$L_{\mathbf{D}}$	1.43	**1.42**
$L_{\mathbf{N}}$	**1.89**	**1.89**

Table 4. Analysis on 4-layer encoder-decoder design vs. 6-layer versions in single-task estimation of albedo, roughness, normals and depth, as well as multi-task estimation of lighting: comparison on losses.

3. Training Details

We train our entire pipeline in two stages: (1) train **BRDFGeoNet** with full supervision on albedo, roughness, depth and normals, (2) freeze **BRDFGeoNet**, feed the output to **LightNet** and train **LightNet** with full supervision on the estimated lighting map and re-rendered image. We additionally use binary masks on pixels of objects $\mathbf{M}_o \in \mathbb{R}^{h \times w}$ or $\mathbf{M}'_o \in \mathbb{R}^{h \times w \times 3}$ (excluding windows), and masks on pixels of valid materials and lighting $\mathbf{M}_l \in \mathbb{R}^{h \times w}$ or $\mathbf{M}'_l \in \mathbb{R}^{h \times w \times 3}$ (excluding surface of lit-up lamps and windows).

For the first stage, as stated in Sec.3.1, we use scale-invariant L2 loss [7, 10] for albedo and depth (log space) and L2 loss for roughness and normals. Specifically, for losses on roughness and normals, we have:

$$L_{\mathbf{R}} = ||(\mathbf{R} - \hat{\mathbf{R}}) \cdot \mathbf{M}_l||_2^2, \qquad (1)$$

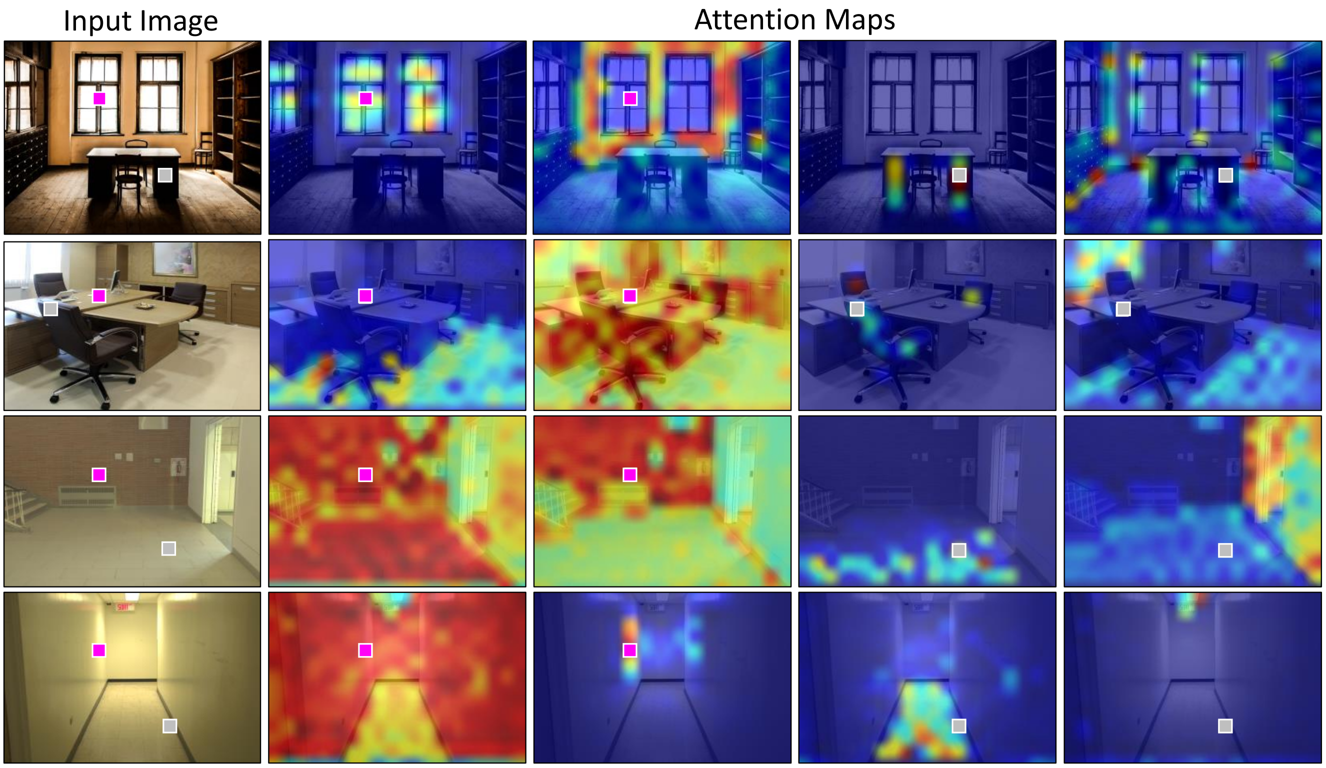

Figure 7. More attention maps learned by the single-task model for albedo, on real world images. We pick 4 samples of real world images, and for each image show two attention maps for each of two selected token locations.

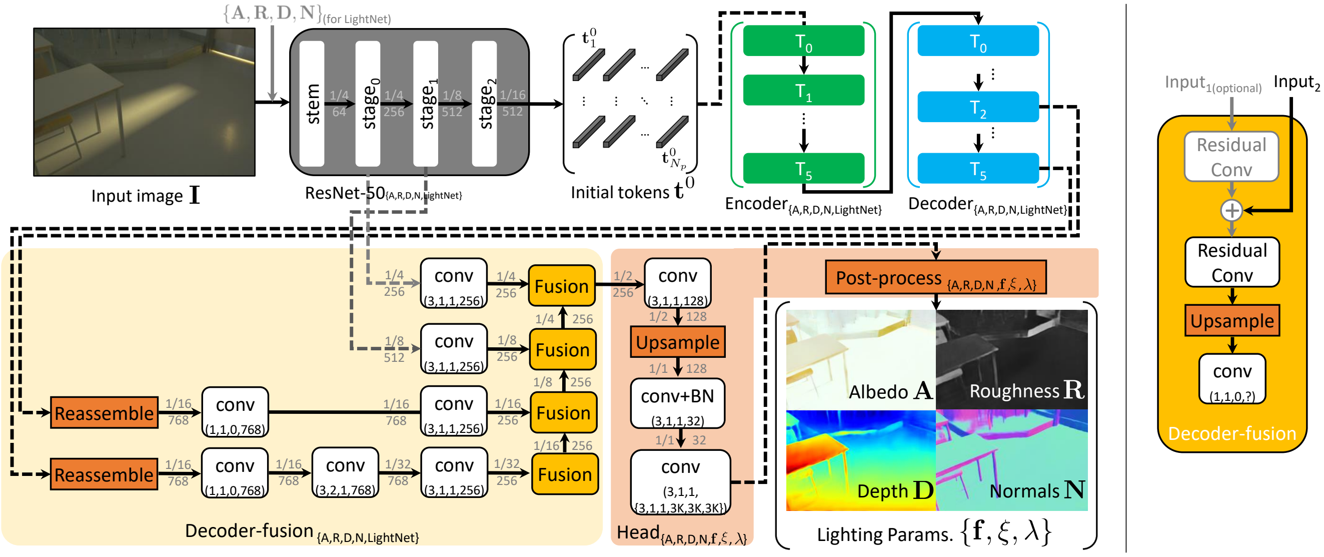

Figure 8. Details of the modules of the single-task and multi-task models (left) and the Decoder-fusion module (right). Reassemble operation is introduced in DPT [9]. Details of the Post-process operation for different modalities can be found in text. The ratio and integer in gray font around each arrow are respectively the output tensor size w.r.t. original input size to the network, and output tensor channels.

$$L_{\mathbf{N}} = ||(\mathbf{N} - \hat{\mathbf{N}}) \cdot \mathbf{M}'_o||_2^2. \tag{2}$$

For albedo, the loss is

$$L_{\mathbf{A}} = ||(\mathbf{A} - \hat{\mathbf{A}}') \cdot \mathbf{M}'_l||_2^2, \tag{3}$$

where $\hat{\mathbf{A}}'$ is the estimated albedo aligned to the ground truth from a least-squares solution [7]. For depth, the loss is computed in log space:

$$L_{\mathbf{D}} = ||(\log \mathbf{D} - \log \hat{\mathbf{D}}') \cdot \mathbf{M}_o'||_2^2 \quad (4)$$

where similarly $\hat{\mathbf{D}}'$ is the estimated depth aligned in log space to the ground truth from a least-squares solution [7].

Given we train the entire pipeline in two stages, we break L_{all} into two losses; $L_{\text{all}} = L_{\text{BRDFGeo}} + L_{\text{light}}$. In training **BRDFGeoNet** in multi-task setting, the loss L_{BRDFGeo} is a weighted combination of losses on albedo, roughness, depth and normals:

$$L_{\text{BRDFGeo}} = \lambda_{\mathbf{A}} L_{\mathbf{A}} + \lambda_{\mathbf{R}} L_{\mathbf{R}} + \lambda_{\mathbf{D}} L_{\mathbf{D}} + \lambda_{\mathbf{N}} L_{\mathbf{N}}, \quad (5)$$

where $\lambda_{\mathbf{A}} = 1.5$, $\lambda_{\mathbf{R}} = 0.5$, $\lambda_{\mathbf{D}} = 0.5$, $\lambda_{\mathbf{N}} = 1.0$.

In single-task settings, the loss for each task will simply be the corresponding loss term weighted by the corresponding weight. In the second stage of training **LightNet**, the loss is a combination of the lighting map reconstruction loss and the image-space re-rendering loss:

$$L_{\text{light}} = \lambda_{\mathbf{L}} L_{\mathbf{L}} + \lambda_{\mathbf{I}} L_{\mathbf{I}} \quad (6)$$

with $\lambda_{\mathbf{L}} = 10.0$ and $\lambda_{\mathbf{I}} = 1.0$.

All models are trained with a learning rate of $1e-5$ and a batch size of 8 with Adam optimizer without weight decay, over the entire training set of OpenRooms for 40 epochs until convergence.

Finetuning details. For finetuning on IIW and NYUv2, we follow Li *et al*. [7] on losses and finetuning strategy for a fair comparison. In each finetuning step, we draw one batch of size 8 from IIW/NYUv2, do a full feed-forward pass and back-propagation with a learning rate of 1e-5, using relative loss on albedo or full supervised loss on normals and depth as described in Li *et al*. [7]. Then we draw another batch from OR of size 8, do the same training step as what is done in pre-training the models on OR. We finetune on IIW/NYU2 for 10 epochs in all finetuning experiments.

4. User Study Details

For the user study, we employ users from Amazon Mechanical Turk to determine the photorealism of an image with inserted bunnies. We compare object insertion results from a set of methods including Gardner'17 [4], Garon'19 [5], Li'21 [8], ours and results rendered using ground truth lighting. In each task, we ask the user to do an A/B test where a pair of images from both our method (multi-task setting) and one baseline method are presented. The user is asked to pick the one with 'better photorealism' based on how well all the inserted objects blend into the image. Each pair of images is presented to 20 users and for each method we use all 20 images from Garon *et al*. [5] with inserted bunnies. We interpret the results as follows: for each comparison of ours against a baseline method, the percentage of users who consider results from the baseline method as better than ours, by averaging 20 feedbacks from each comparison.

5. Statement on Potential Negative Impacts

As stated in the main paper, one possible negative impact of our method is the vulnerability to misuses including Deepfake [13]. While there is no way to prevent our method to be used by a third-party once it is open-sourced, a way to mitigate the potential negative impact is to employ techniques like Yu *et al*. [14] to embed fingerprinting into the model and hence the results, so that tracing of accountability of improperly edited or generated results will be made possible to countermeasure malicious use.

Another impact is due to the nature of the transformer architecture we use. Due to the fact that transformers are relatively new and understudied especially for computing efficiency on current dedicated deep learning hardware, as well as their larger computing cost compared to previous CNN-based models on similar tasks, the new models may result in increased carbon footprint if deployed on a large scale. Thus it is important to dedicate more research effort to discover improvements to the architecture and hardware implementation, so that the potential energy and environmental concerns can be mitigated while enabling the use of transformers in inverse rendering.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 1

[2] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014. 4

[3] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1

[4] Marc-André Gardner, Kalyan Sunkavalli, Ersin Yumer, Xiaohui Shen, Emiliano Gambaretto, Christian Gagné, and Jean-François Lalonde. Learning to predict indoor illumination from a single image. *ACM Transactions on Graphics (TOG)*, 36(6):1–14, 2017. 9

[5] Mathieu Garon, Kalyan Sunkavalli, Sunil Hadap, Nathan Carr, and Jean-François Lalonde. Fast spatially-varying indoor lighting estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6908–6917, 2019. 1, 3, 9

[6] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 1

[7] Zhengqin Li, Mohammad Shafiei, Ravi Ramamoorthi, Kalyan Sunkavalli, and Manmohan Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and svbrdf from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2475–2484, 2020. 1, 7, 9

[8] Zhengqin Li, Ting-Wei Yu, Shen Sang, Sarah Wang, Meng Song, Yuhan Liu, Yu-Ying Yeh, Rui Zhu, Nitesh Gundavarapu, Jia Shi, Sai Bi, Zexiang Xu, Hong-Xing Yu, Kalyan Sunkavalli, Miloš Hašan, Ravi Ramamoorthi, and Manmohan Chandraker. OpenRooms: An end-to-end open framework for photorealistic indoor scene datasets. In *CVPR*, 2021. 9

[9] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021. 1, 8

[10] René Ranftl, Katrin Lasinger, David Hafner, Konrad Schindler, and Vladlen Koltun. Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. *arXiv preprint arXiv:1907.01341*, 2019. 7

[11] Soumyadip Sengupta, Jinwei Gu, Kihwan Kim, Guilin Liu, David W Jacobs, and Jan Kautz. Neural inverse rendering of an indoor scene from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8598–8607, 2019. 1

[12] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012. 5

[13] Mika Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9(11), 2019. 9

[14] Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14448–14457, 2021. 9