

Deep SVBRDF Estimation on Real Materials

Louis-Philippe Asselin Denis Laurendeau Jean-François Lalonde
Université Laval

`louis-philippe.asselin.2@ulaval.ca, {denis.laurendeau, jflalonde}@gel.ulaval.ca`

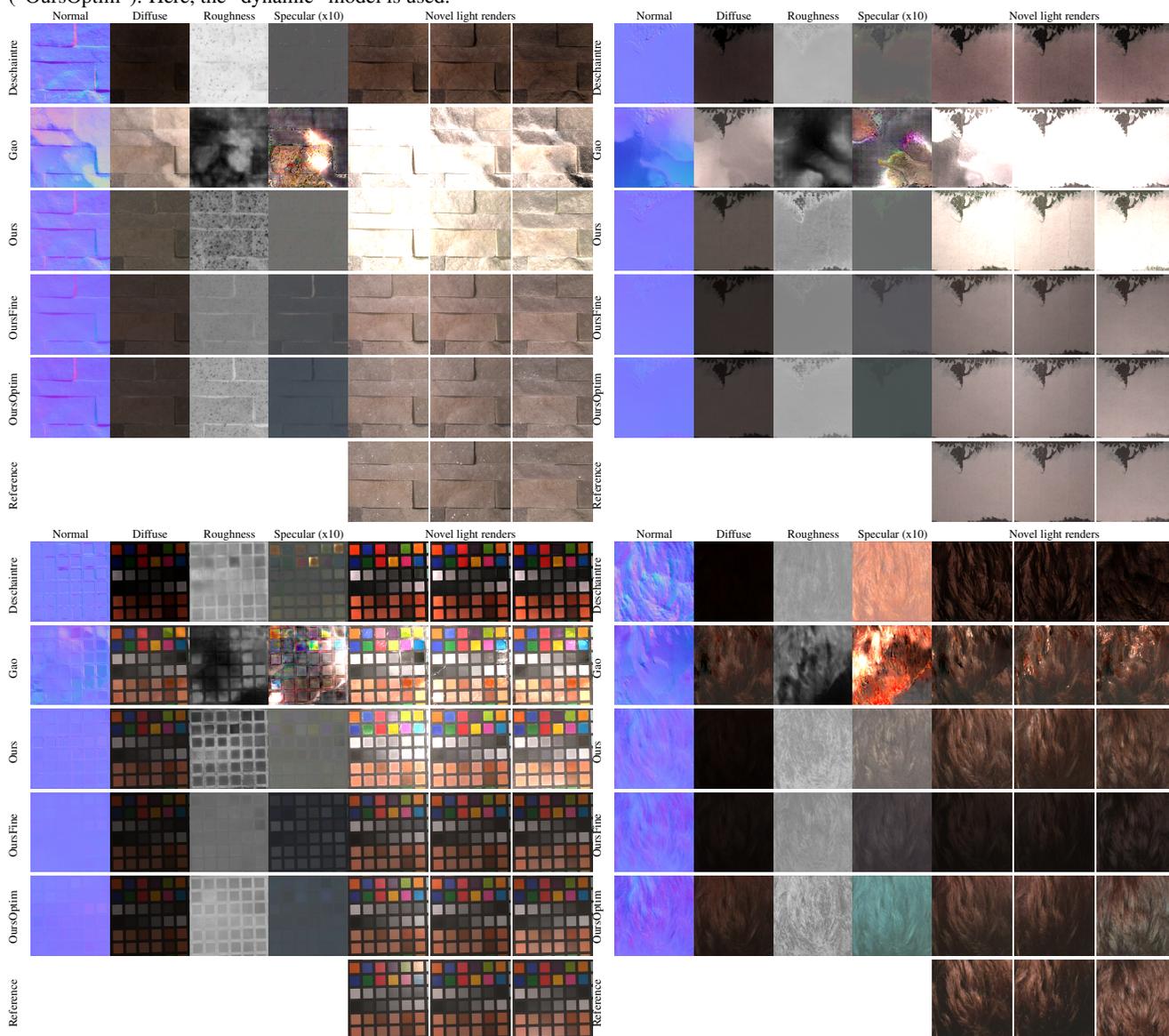
This supplementary document provides the following additional figures:

- Fig. 1: Additional real dataset samples (fig. 2 in the paper)
- Fig. 2: Qualitative comparison between our networks and previous work on the real dataset (additional samples for figs 9 and 10 in the paper)
- Fig. 3: Qualitative comparison of the number of inputs using our optimized method “OursOptim” (complements fig. 11 in the paper)
- Fig. 4: Quantitative comparison of models on the synthetic dataset (additional metrics for fig. 7 in the paper)
- Fig. 5: Quantitative comparison of models on the real dataset (additional metrics for fig. 8 in the paper)
- Fig. 6: Quantitative comparison of the number of inputs (additional metrics for fig. 11 in the paper)
- Fig. 7: Quantitative comparison of different approaches for the per-material optimization (additional metrics for fig. 12 in the paper)

Please consult the supplementary video for dynamic relighting results.

Dataset, code and supplementary video are available at <https://lvsn.github.io/real-svbrdf>.

Figure 2. Additional examples for figs 9 and 10 in the paper. Qualitative comparison between previous work (Deschaintre *et al.* [1], Gao *et al.* [2]), our model trained on synthetic data only (“Ours”), our model finetuned (“OursFine”) and after the per-material optimization (“OursOptim”). Here, the “dynamic” model is used.



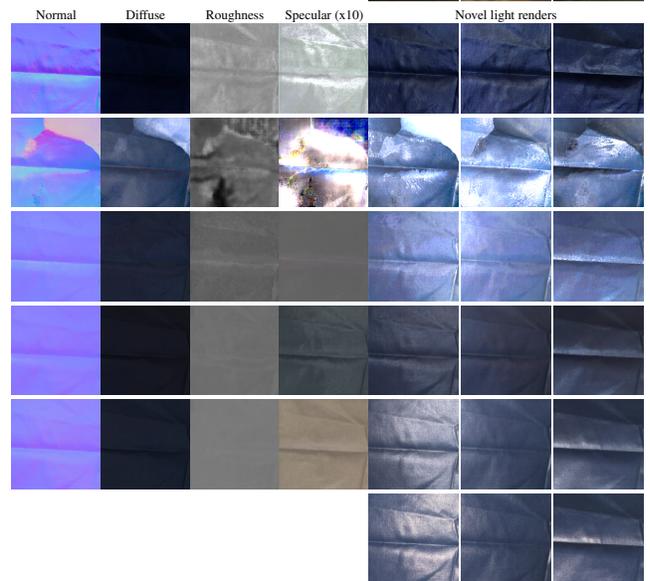
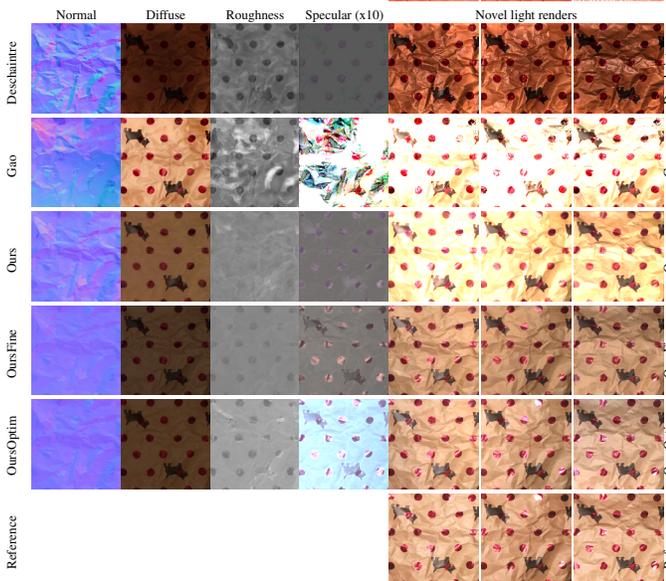
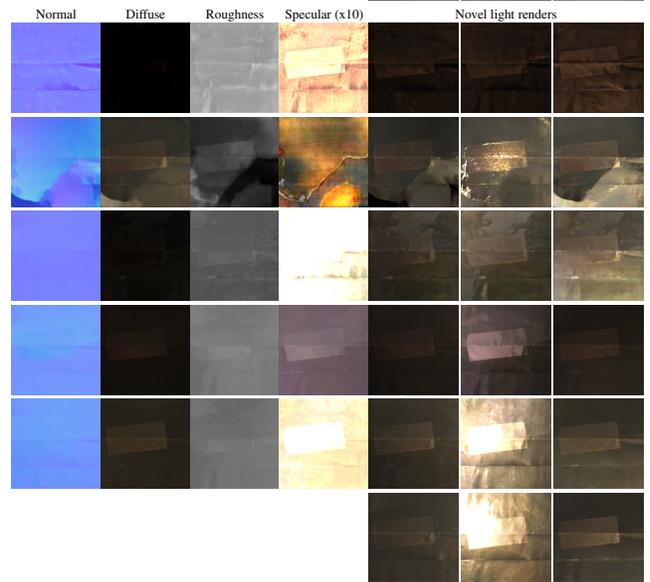
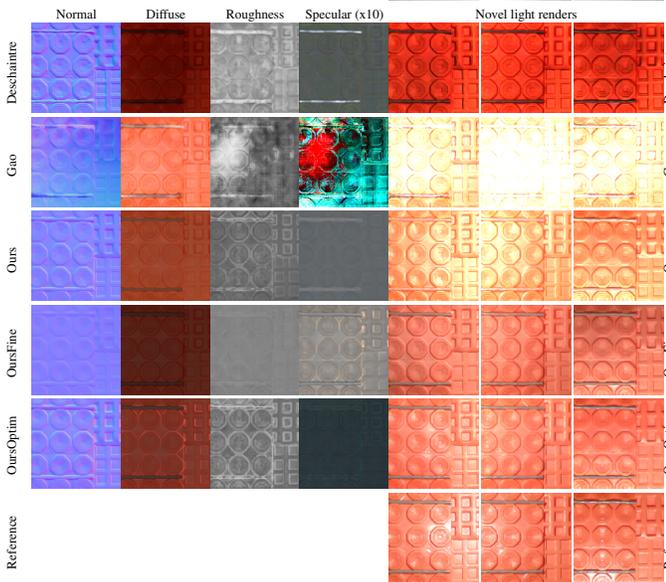
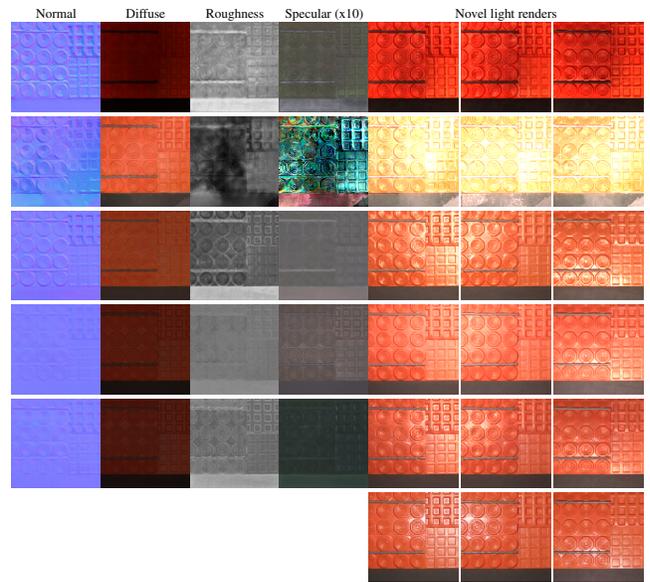
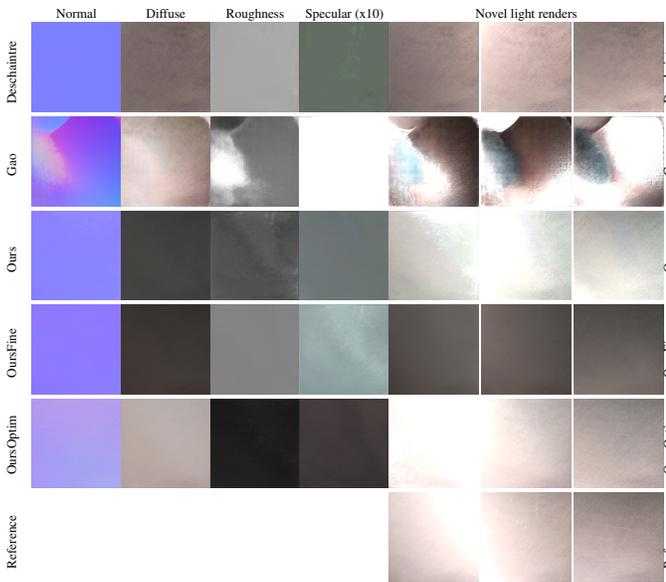
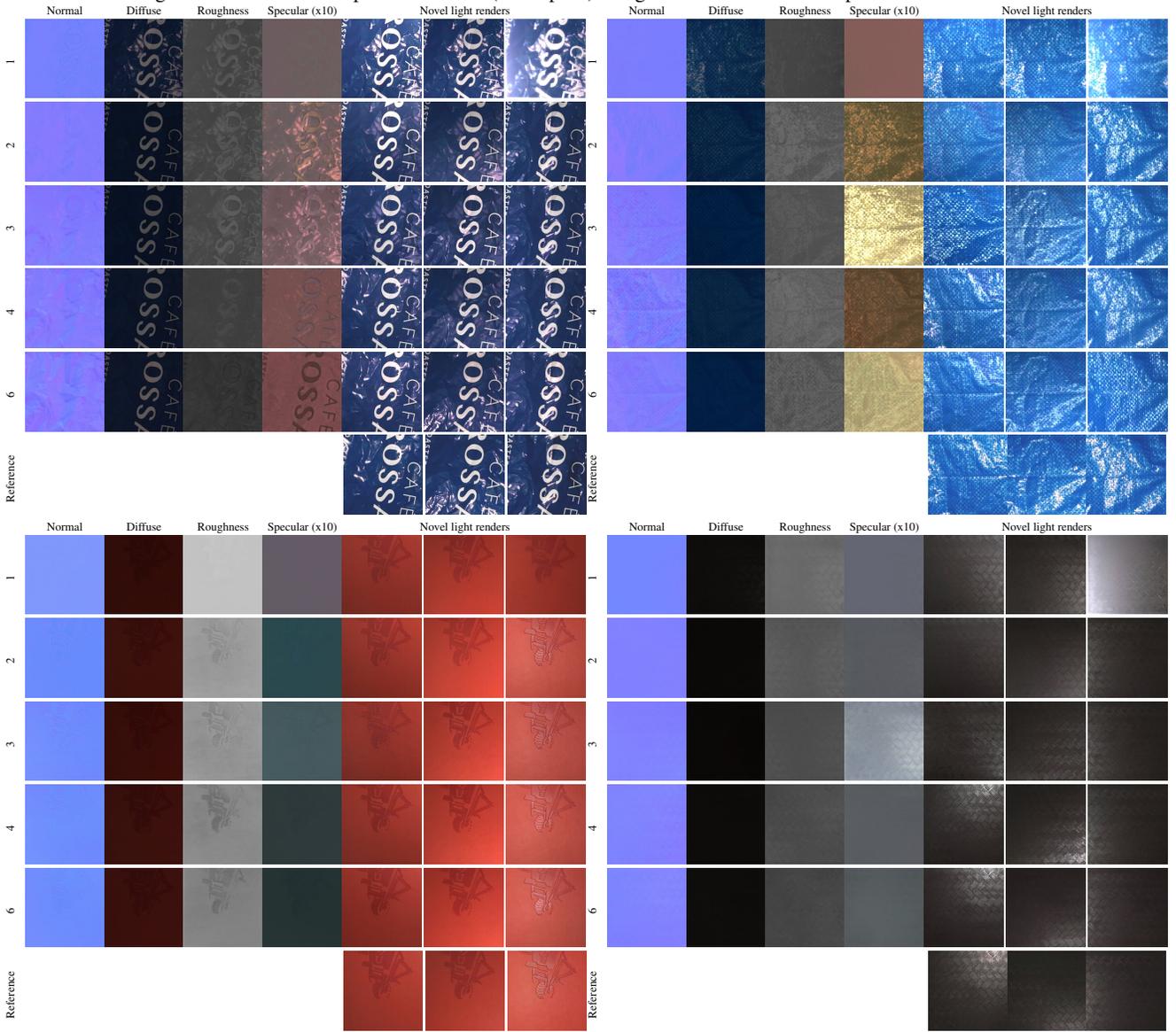
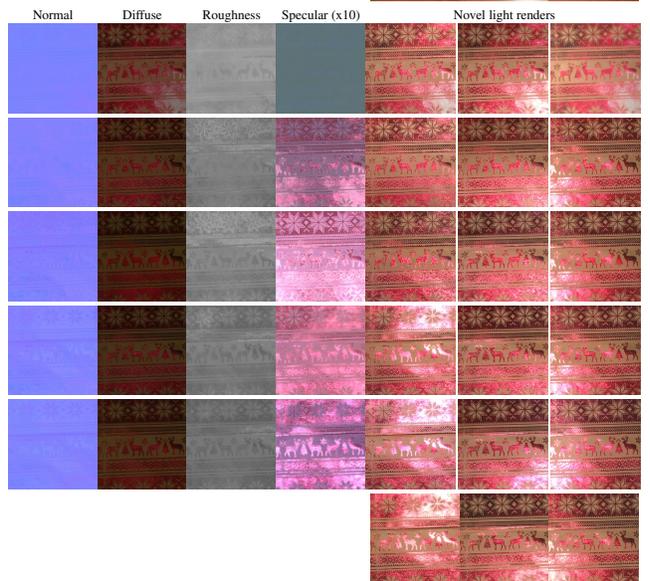
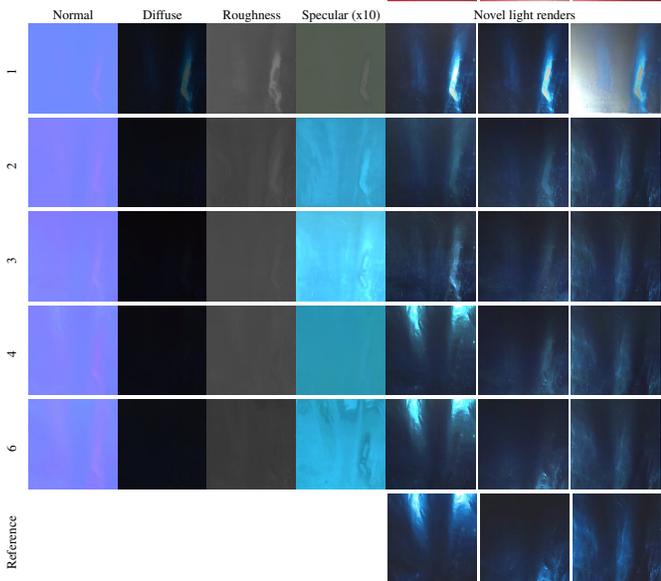
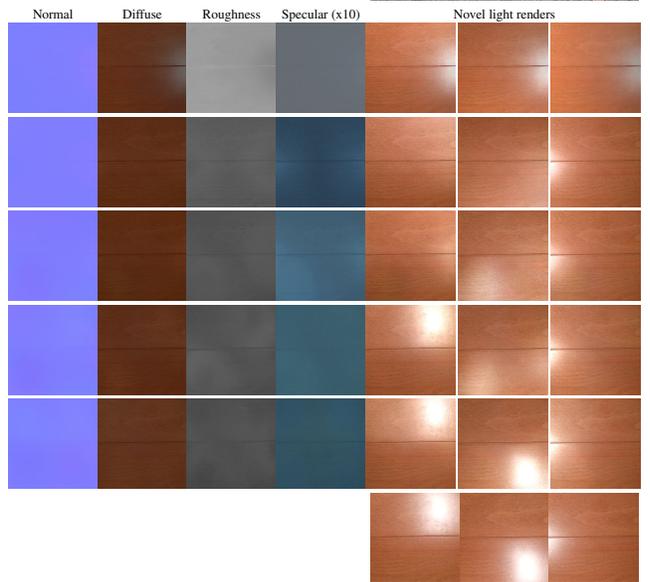
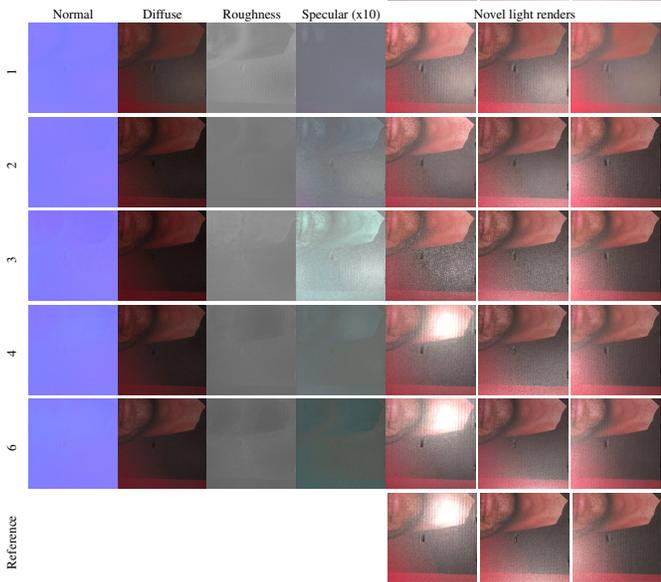
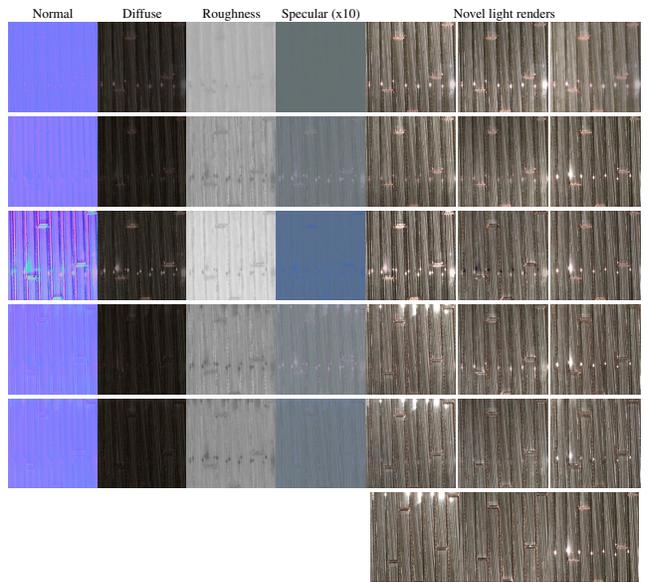
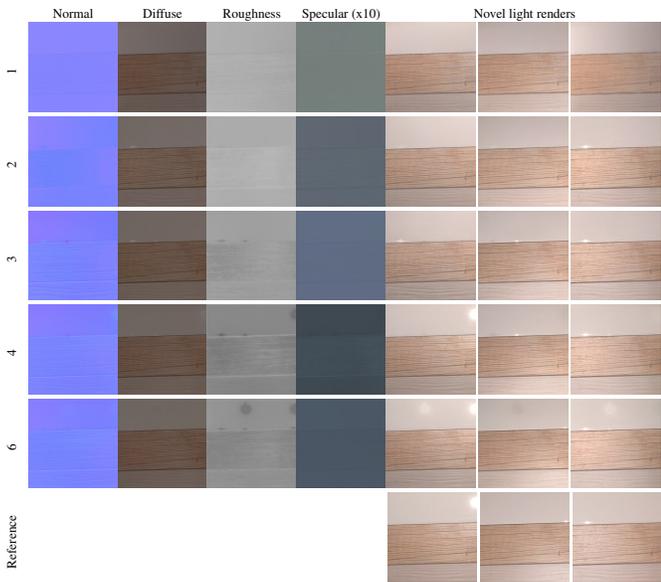


Figure 3. Results of our optimized model (oursOptim) using different numbers of inputs on real materials.





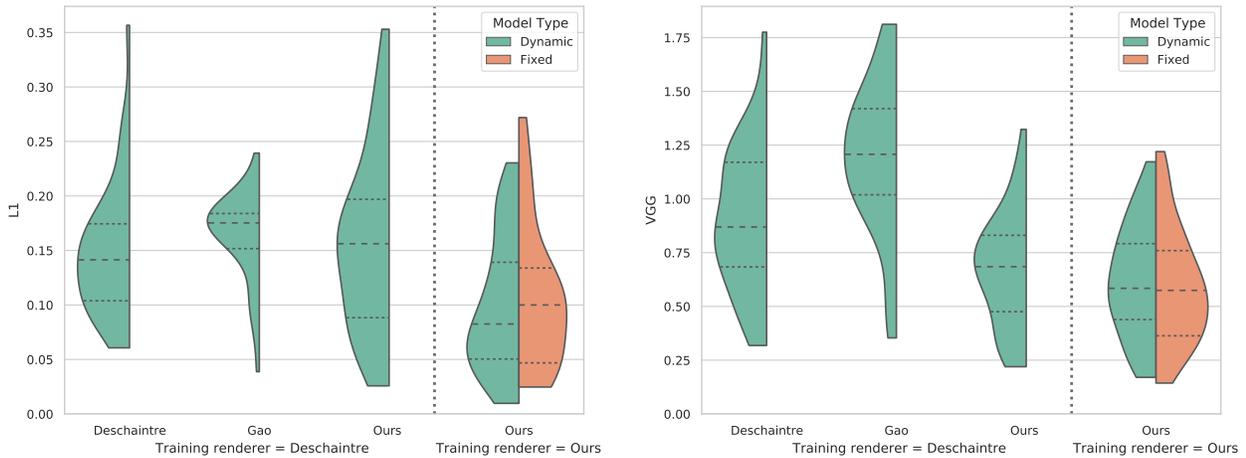


Figure 4. Additional metrics for fig. 7 in the paper. Quantitative comparison between our networks and previous work (Deschaintre *et al.* [1] and Gao *et al.* [2]) on the *synthetic* test set. The models on the left of the vertical dotted line are trained with the renderer from Deschaintre *et al.* [1]. On the right of the line are models trained with our renderer and a configuration matching the one of the capture system. All models are evaluated using our renderer and configuration. The loss is computed on novel renders. 6 input images are used by all models. The L1 loss on novel renders can be seen on the left. On the right, the VGG perceptual loss results are shown (lower is better). This VGG perceptual loss is the L1 loss of the convolutional layers features obtained by passing the images through a VGG19 model pretrained on ImageNet [3].

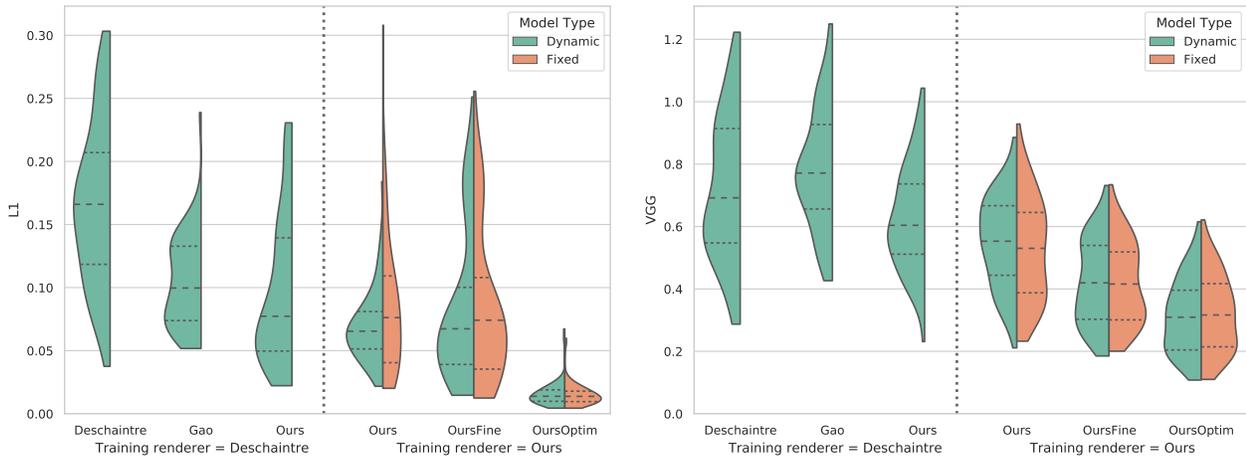


Figure 5. Additional metrics for fig. 8 in the paper. Quantitative comparison between our networks and previous work (Deschaintre *et al.* [1] and Gao *et al.* [2]) on the *real* test set. Six images are fed to the model which predicts SVBRDF parameters. Deschaintre’s renderer (left) use random unseen light position at test time, while ours (right) use the even-indexed light sources on the capture rim for training and odd-indexed lights for test. All methods are trained on synthetic images. Only “OursFine” and “OursOptim” are finetuned on real images. Metrics are L1 loss and VGG perceptual loss on novel renders (lower is better). The VGG loss is detailed in fig. 4

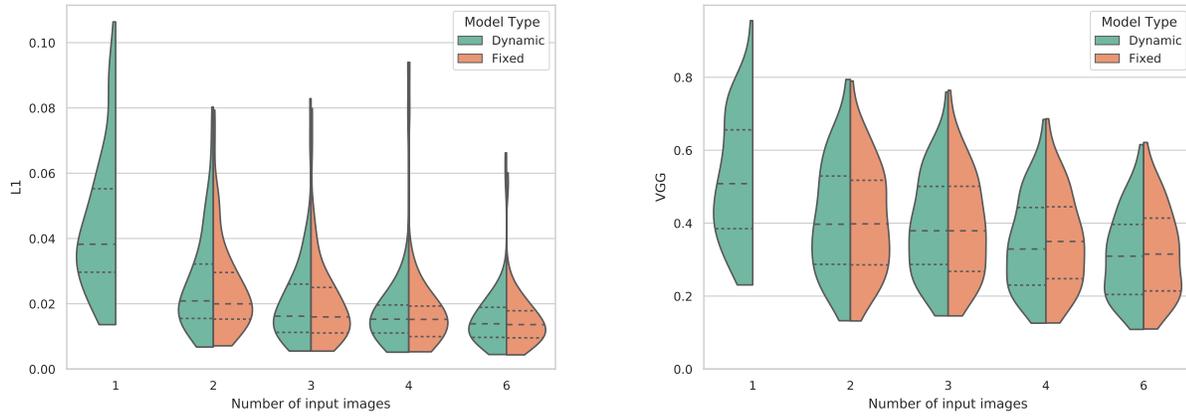


Figure 6. Additional metrics for fig. 11 in the paper. Effect of varying the number of input images for training. Here, the per-material optimization results are shown for the “dynamic” and “fixed” architectures. When the number is 1, both architectures are the same so only one result is shown. Metrics are L1 loss and VGG perceptual loss on novel renders (lower is better). The VGG loss is detailed in fig. 4

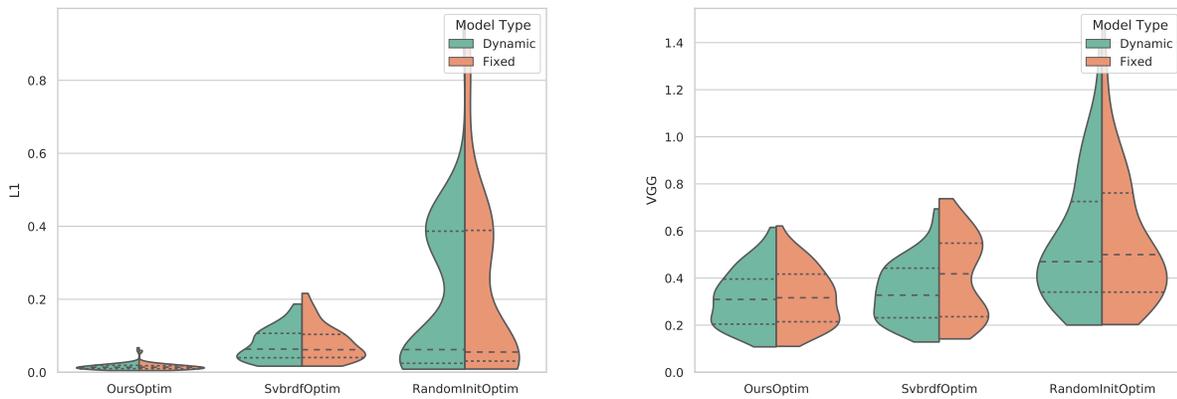


Figure 7. Additional metrics for fig. 12 in the paper. Comparison of different approaches for the per-material optimization (lower is better). All methods minimize $\mathcal{L}_{\text{render}}$ on the input images. “OursOptim” overfits the weights of a model trained on the synthetic dataset to the input images. “RandomInit” performs the same process but on a randomly initialized and untrained model. “SvbrdfRefined” optimizes the SVBRDF maps directly. Metrics are L1 loss and VGG perceptual loss on novel renders. The VGG loss is detailed in fig. 4

References

- [1] V. Deschaintre, M. Aittala, F. Durand, G. Drettakis, and A. Bousseau. Flexible SVBRDF capture with a multi-image deep network. *Computer Graphics Forum (Eurographics Symposium on Rendering)*, 38(4):13, jul 2019. [3](#), [7](#)
- [2] D. Gao, X. Li, Y. Dong, P. Peers, K. Xu, and X. Tong. Deep inverse rendering for high-resolution SVBRDF estimation from an arbitrary number of images. *ACM Transactions on Graphics*, 38(4):134, July 2019. [3](#), [7](#)
- [3] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. [7](#)