Supplementary Material: Universal Photometric Stereo Network using Global Lighting Contexts

Appendix A. Implementation Details

Architecture details: In the main paper, an overview of our universal photometric stereo network was given but some important details were omitted due to the space limit. In this section, we detail the "basic + pre-fusion" configuration of our universal photometric stereo network. It should be noted that the hyper parameters in our architecture are all selected empirically, so it is quite possible that there are parameters that will give better performance.

The image-wise feature extraction network (*i.e*. Swin-S variant of SwinTransformer [10]) and subsequent multi-scale feature fusion with the feature pyramid network (*i.e*. UPerNet [14]) in our encoder were implemented on MMSegmentation [2]. The updates from the original codes are mainly two. First, we input a mask image in addition to an RGB image. Second, following the suggestions in [13], we modified the original mlp-based patch embedding (*i.e*., local information embedding during the reduction of the image resolution to 1/4 of the canonical resolution) to the CNN-based one with five convolutional layers to capture the local shading variations. The number of different scales was four; hence, given the canonical resolution of 256×256, sizes of the multi-scale feature maps were 64×64×96, 32×32×192, 16×16×384 and 8×8×768, which were fused to 64×64×256 global lighting contexts.

The feature communication in our encoder and aggregation in our decoder were pixelwisely applied to feature vectors under different lighting conditions in similar to our previous work [8]. As illustrated in Fig. 1, the feature communication step built upon a single Transformer layer where input feature vectors were firstly projected to query, key and value vectors whose dimensions were same with the input ones. They were then passed to a multi-head self-attention (the number of heads is 8) and a multi-layer perceptron (MLP) with the pre-layer normalization [15] and dropout ($p = 0.1$). Though the MLP doubled the original feature dimension, the feature dimension and number of feature vectors in a set did not change between the input and output of the feature communication step.

The feature aggregation step input q sets of vectors $\mathrm{cat}\{I(\boldsymbol{x}), \mathcal{G}(\mathcal{S}(\boldsymbol{x}))\}_{\{1\ldots q\}} \in \mathbb{R}^{q\times(256+3)}$ where each vector was composed of raw pixel values and the interpolated global lighting context. Then the input set was passed to three Transformer layers and a PMA [9] where the number of elements in a set was shrunk from q to one. The surface normal predictor was a MLP with one hidden layer whose feature dimension shrank as $384 \rightarrow 192 \rightarrow 3$ and the norm of the output vector was normalized to be a unit surface normal vector at the location.

Competitor details: It should be noted again that all the algorithms (ours, GCNet [6], MPM [11] and Variational [7]) took the object mask as input. To ensure a fair comparison, we applied the same center crop to input images, which means that the input of all the algorithms were exactly same (*i.e*., crops of images and an object mask). For a fair evaluation, we used the authors' official implementations for competitors. Since there is a binary ambiguity left in the surface normal recovered by MPM (*i.e*. signs of x, y, z directions), we manually solved it so as to be quantitatively optimal in the quantitative experiments and most visually plausible in the qualitative evaluation. As for GCNet, we used the pre-trained model provided by authors since our training dataset was not available for their model due to the fact that GCNet requires the supervision of directional lightings. In addition, we found that GCNet [5] didn't work at all for our raw test images without the proper image normalization (The data normalization is also important for the DiLiGenT [12] evaluation), therefore we empirically performed the linear image normalization dividing each image by $0.1 \cdot \max(I)$ so that the pixel values in each image ranged between 0 and 0.1. Unlike others, Variational [7] is actually an algorithm for perspective images and it requires the focal length of images as input. So we approximated test images as perspective ones by using the unit focal length (*i.e*. $f = 256$ for 256×256 image) for our PS-Wild test dataset and using ones from Exif-Tags in the real evaluation. We note that empirically, the small differences of the focal length didn't show any significant difference in results. Unfortunately, MPM [11] is a quite computationally expensive algorithm whose computational complexity is $O(h^2w^2)$ and we confirmed that it didn't work for images whose sizes were bigger than 512×512. For a fair comparison, we used 256×256 crops in both quantitative and qualitative comparison because we confirmed that MAE didn't significantly depend on the input image resolution.

Appendix B. PS-Wild Dataset and Training Details

Renderer details: PS-Wild was rendered with the *Cycles* engine in Blender2.93 [1]. For a full global illumination rendering using a path tracing integrator with direct light sampling, we used 256 rendering samples with 10 max ray bounces. Each BRDF material in both training and test 3-D assets consisted of 2-D texture maps of the base color, roughness, metalness which were directly fed to the diffuse and specular BRDFs of the Cycles engine (*i.e*. we used Principled BSDF shader [4]). In Fig. 2, we illustrated some examples of rendered images and 3-D assets in our PS-Wild training dataset. Each row corresponds to one object from 10,099 objects in total. As for the test dataset, we illustrate the entire 50 objects and corresponding results in Fig.3-52. As mentioned, our textures are classified into three types of

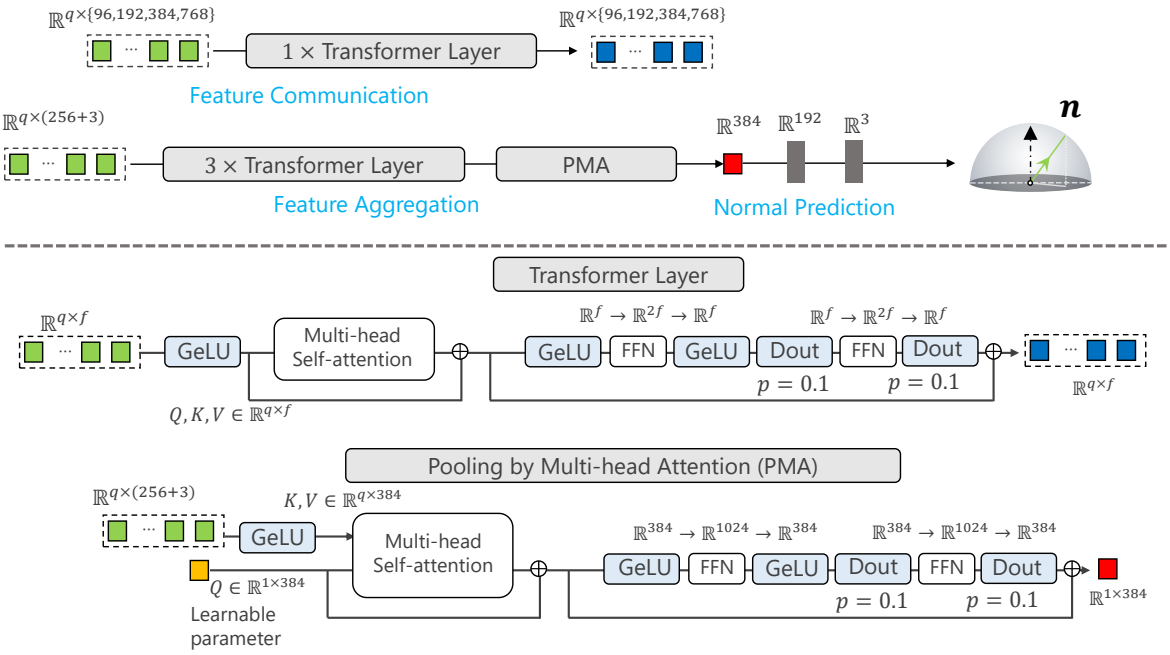

Figure 1. Implementation details of the feature communication and aggregation steps. Our feature communication step is composed of a single Transformer layer and our feature aggregation step is composed of three Transformer layers followed by PMA layer.

materials (six as categorized in ShareTextures [3]); diffuse (Fabric, Concrete), specular (Wood, Floor, Ground) and metallic (Metal). The rendering pipeline was exactly same as one for the training dataset.

Training details: We augmented the dataset during the training to bring more variations in training examples. Concretely, we randomly flipped images horizontally or vertically, and randomly rotated images by 90 degrees. In addition, we also performed the random color swapping for each image since our task didn't include the surface reflectance recovery. We used $p = 0.5$ for all the augmentations.

Appendix C. Complete Quantitative Comparison

We illustrated the complete results on our PS-Wild test datasets in Fig.3-52. As described in the main paper, we observed that GCNet worked well for images under the directional lighting, however had problems in handling more complicated lighting conditions. We also observed that GCNet basically produced more blurry output than other methods due to the image-wise operations such as convolutional neural networks. MPM and Variational could produce sharper results but had problems in handling non-Lambertian, non-convex objects.

References

[1] Blender. https://www.blender.org/. 1

[2] MMSegmentation. https://github.com/open-mmlab/mmsegmentation. 1

[3] ShareTextures. https://www.sharetextures.com/. 2

[4] B. Burley. Physically-based shading at disney, part of practical physically based shading in film and game production. *SIGGRAPH 2012 Course Notes*, 2012. 1

[5] G. Chen, K. Han, B. Shi, Y. Matsushita, and K. K. K. Wong. Self-calibrating deep photometric stereo networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8731–8739, 2019. 1

[6] Guanying Chen, Michael Waechter, Boxin Shi, Kwan-Yee K Wong, and Yasuyuki Matsushita. What is learned in deep uncalibrated photometric stereo? In *European Conference on Computer Vision*, pages 745–762. Springer, 2020. 1

[7] Bjoern Haefner, Zhenzhang Ye, Maolin Gao, Tao Wu, Yvain Quéau, and Daniel Cremers. Variational uncalibrated photometric stereo under general lighting. pages 8539–8548, 2019. 1

[8] S. Ikehata. Ps-transformer: Learning sparse photometric stereo network using self-attention mechanism. In *BMVC*, 2021. 1

[9] J. Lee, Y. Lee, J. Kim, A. Kosiorek, S. Choi, and Y. W. Teh. Set transformer: A framework for attention-based

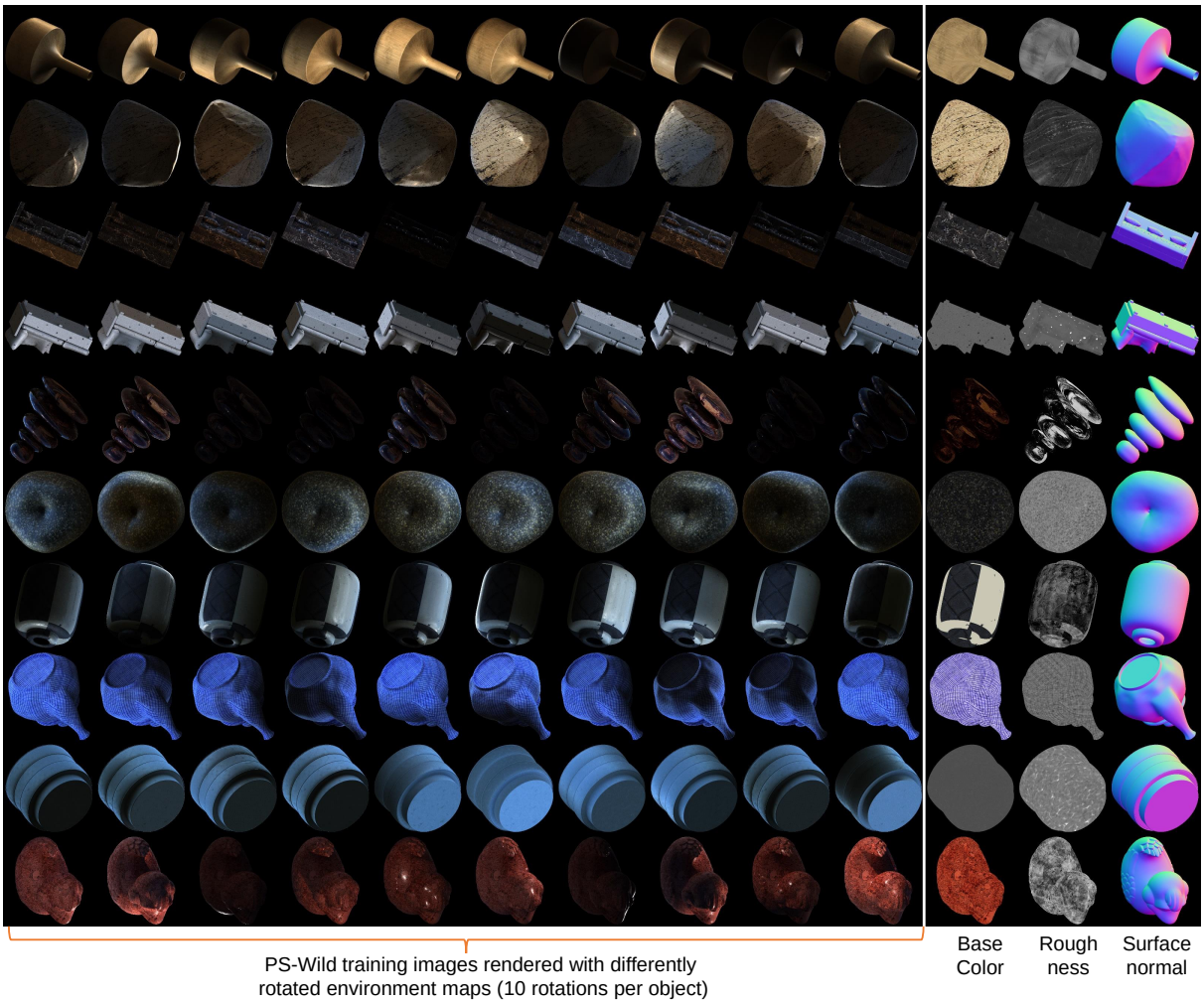

Figure 2. Examples of images and BRDF parameter maps in our PS-Wild training dataset (metallic map is omitted).

permutation-invariant neural networks. In *ICML*, pages 3744–3753, 2019. 1

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1

[11] Zhipeng Mo, Boxin Shi, Feng Lu, Sai-Kit Yeung, and Yasuyuki Matsushita. Uncalibrated photometric stereo under natural illumination. pages 2936–2945. IEEE Computer Society, 2018. 1

[12] B. Shi, Z. Mo, Z. Wu, D.Duan, S-K. Yeung, and P. Tan. A benchmark dataset and evaluation for non-lambertian and uncalibrated photometric stereo. *IEEE TPAMI*, page (to appear), 2018. 1

[13] Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34, 2021. 1

[14] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, pages 418–434, 2018. 1

[15] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pages 10524–10533. PMLR, 2020. 1

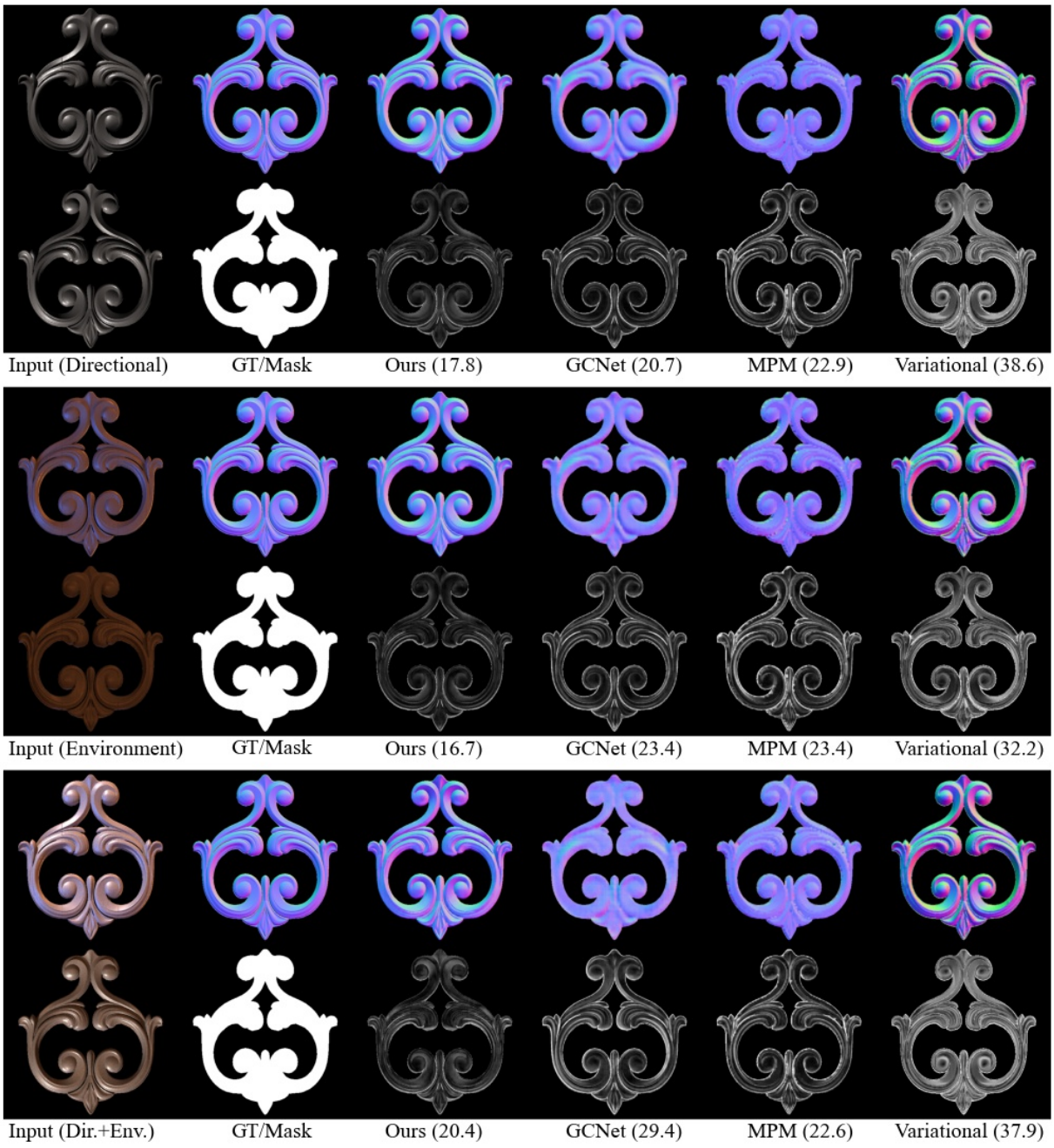

Figure 3. Results on object ID 1 (accessory, dark-wooden-parquet [Wood]). MAEs (in degrees) are shown next to the name of the method.

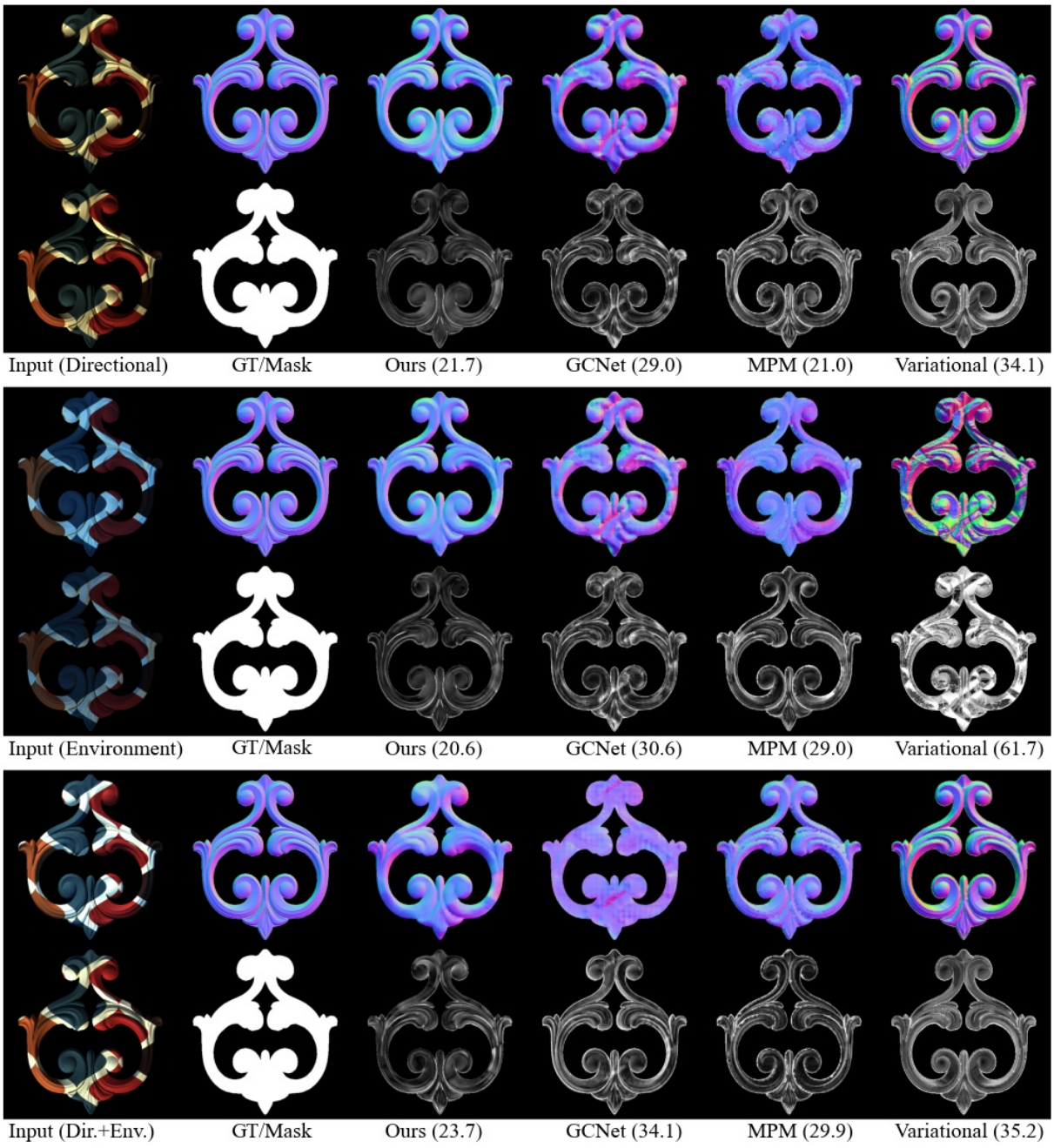

Figure 4. Results on object ID 2 (accessory, fabric-85 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

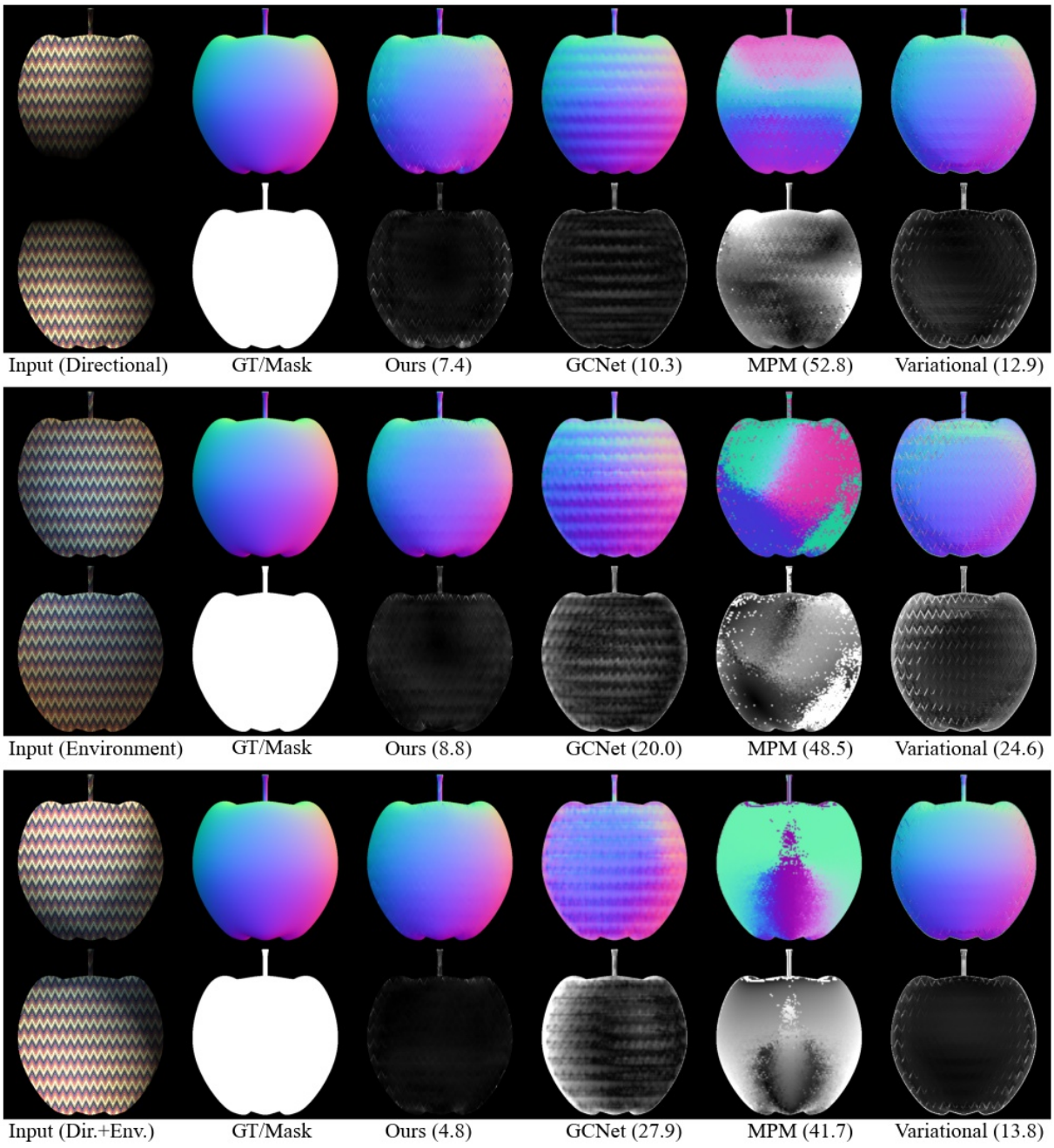

Figure 5. Results on object ID 3 (apple, fabric-86 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

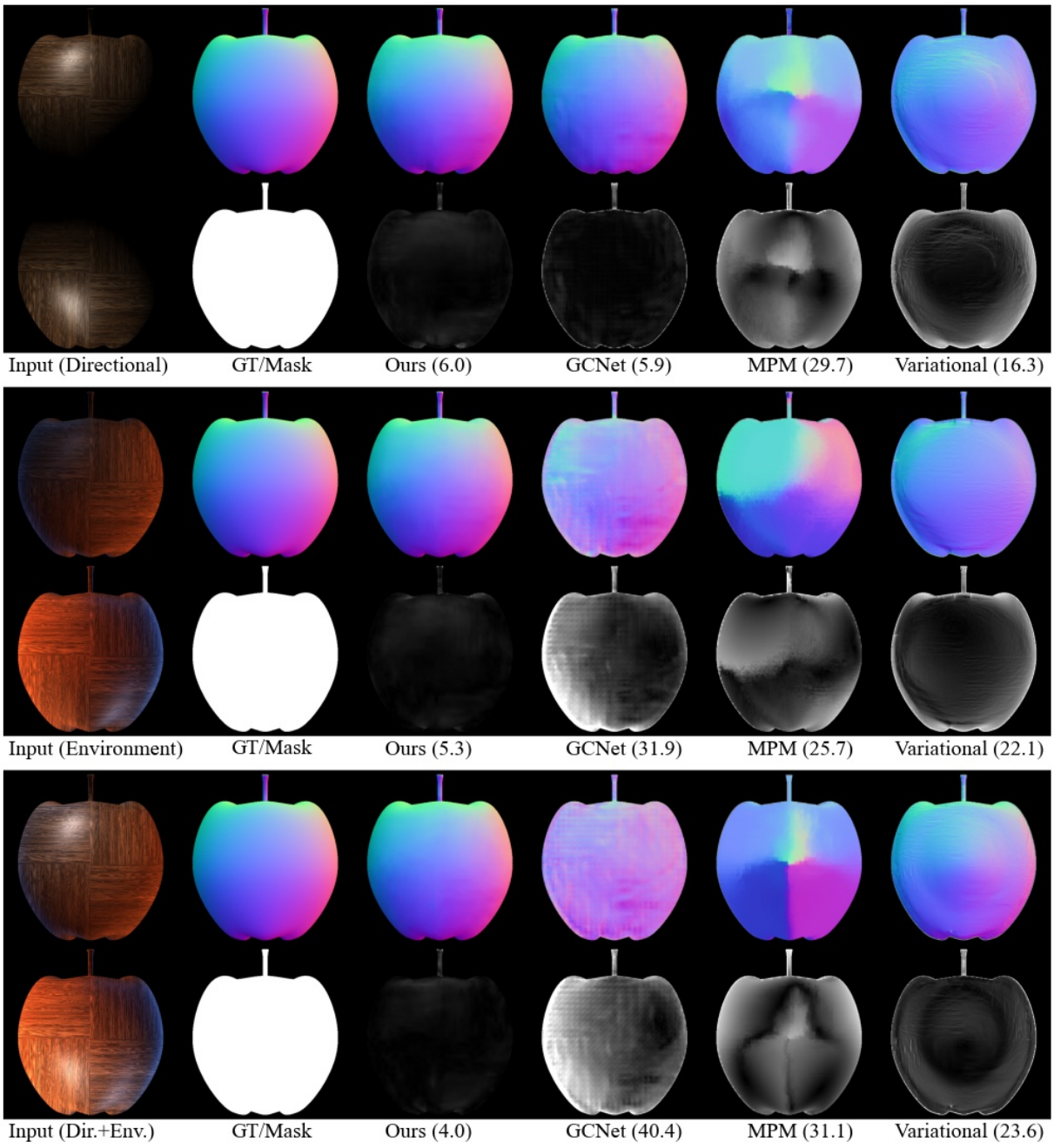

Figure 6. Results on object ID 4 (apple, square-pattern-parquet-1 [Wood]). MAEs (in degrees) are shown next to the name of the method.

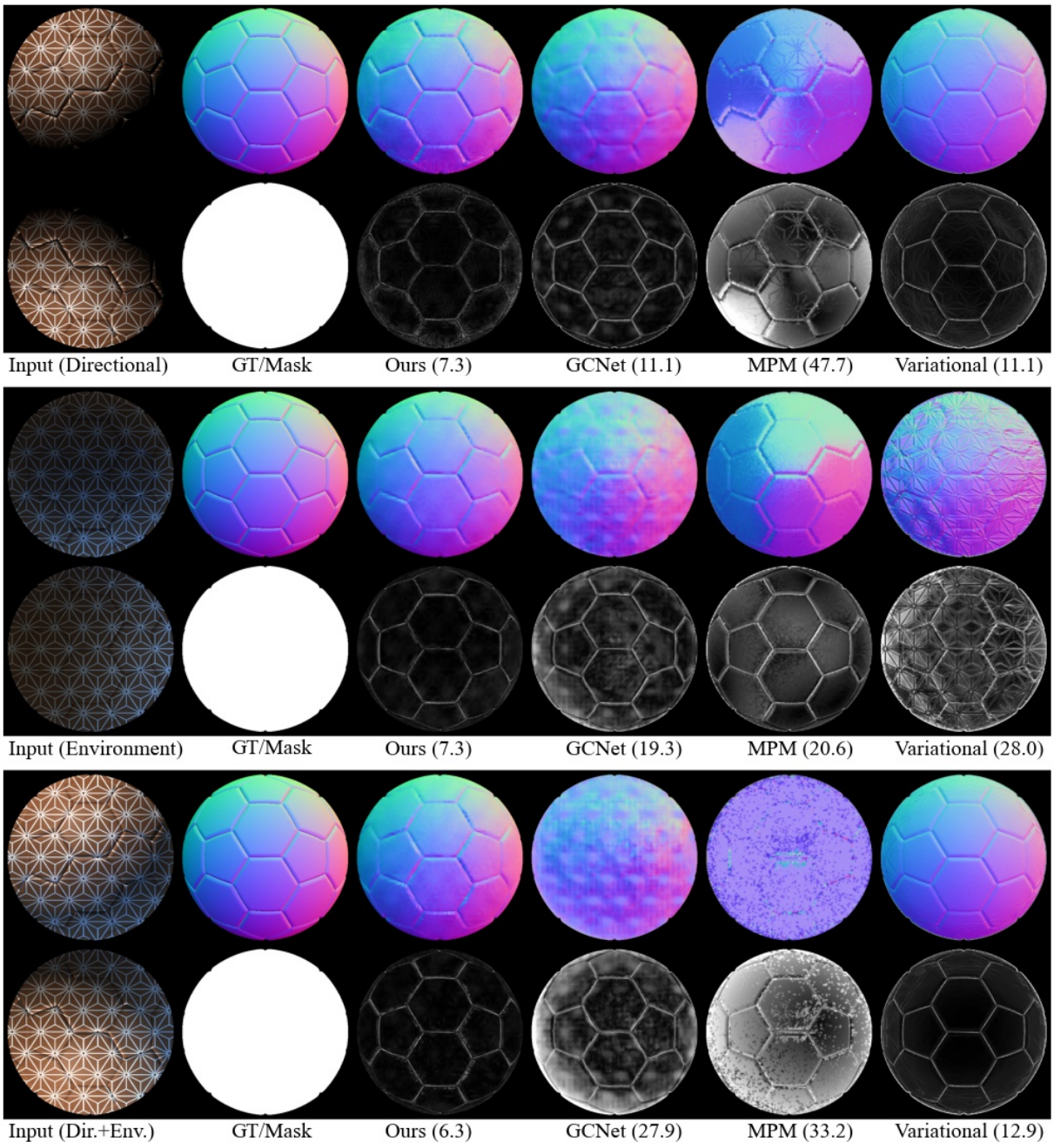

Figure 7. Results on object ID 5 (ball, fabric-95 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

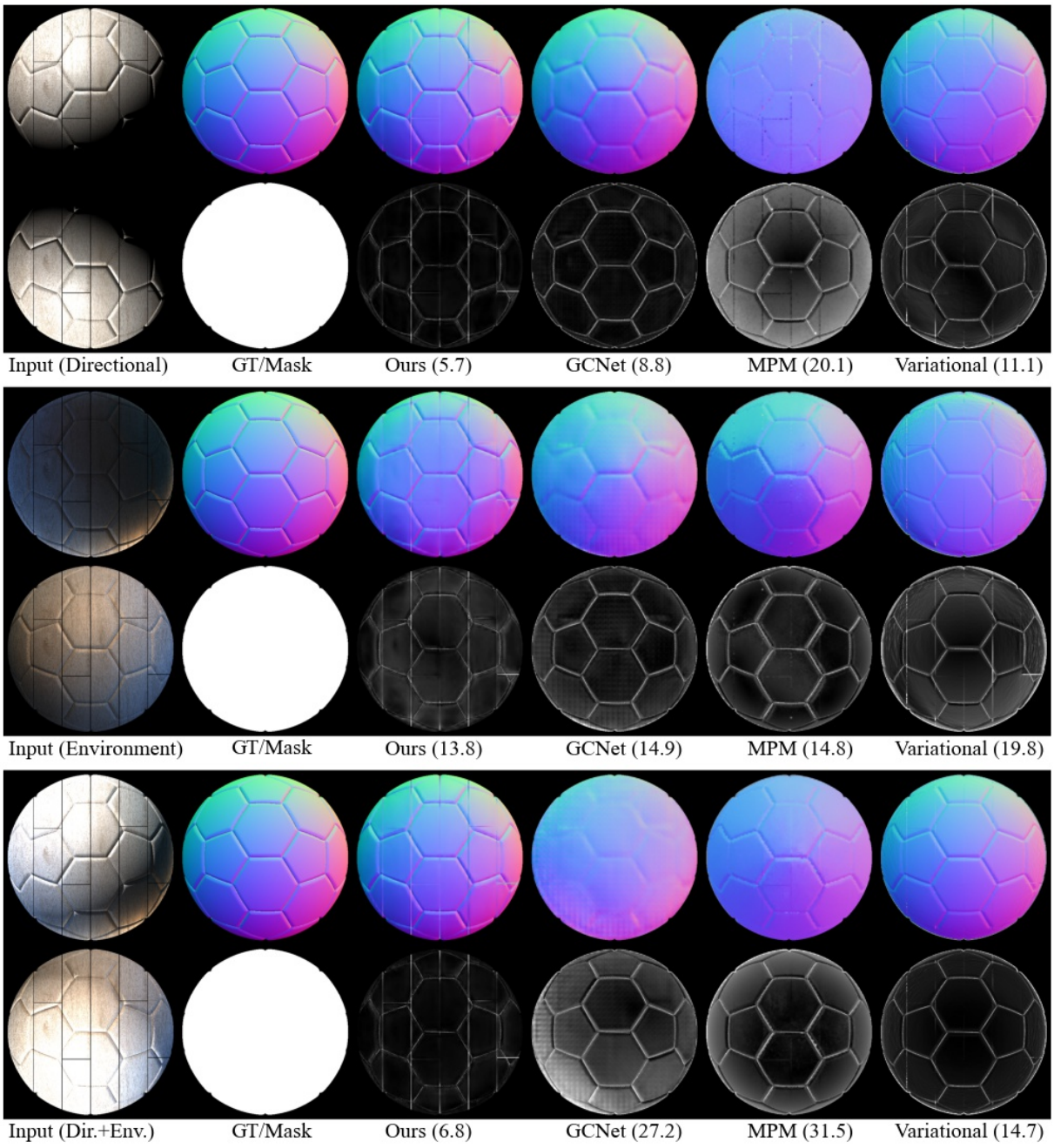

Figure 8. Results on object ID 6 (ball, wood-parquet-17 [Wood]). MAEs (in degrees) are shown next to the name of the method.

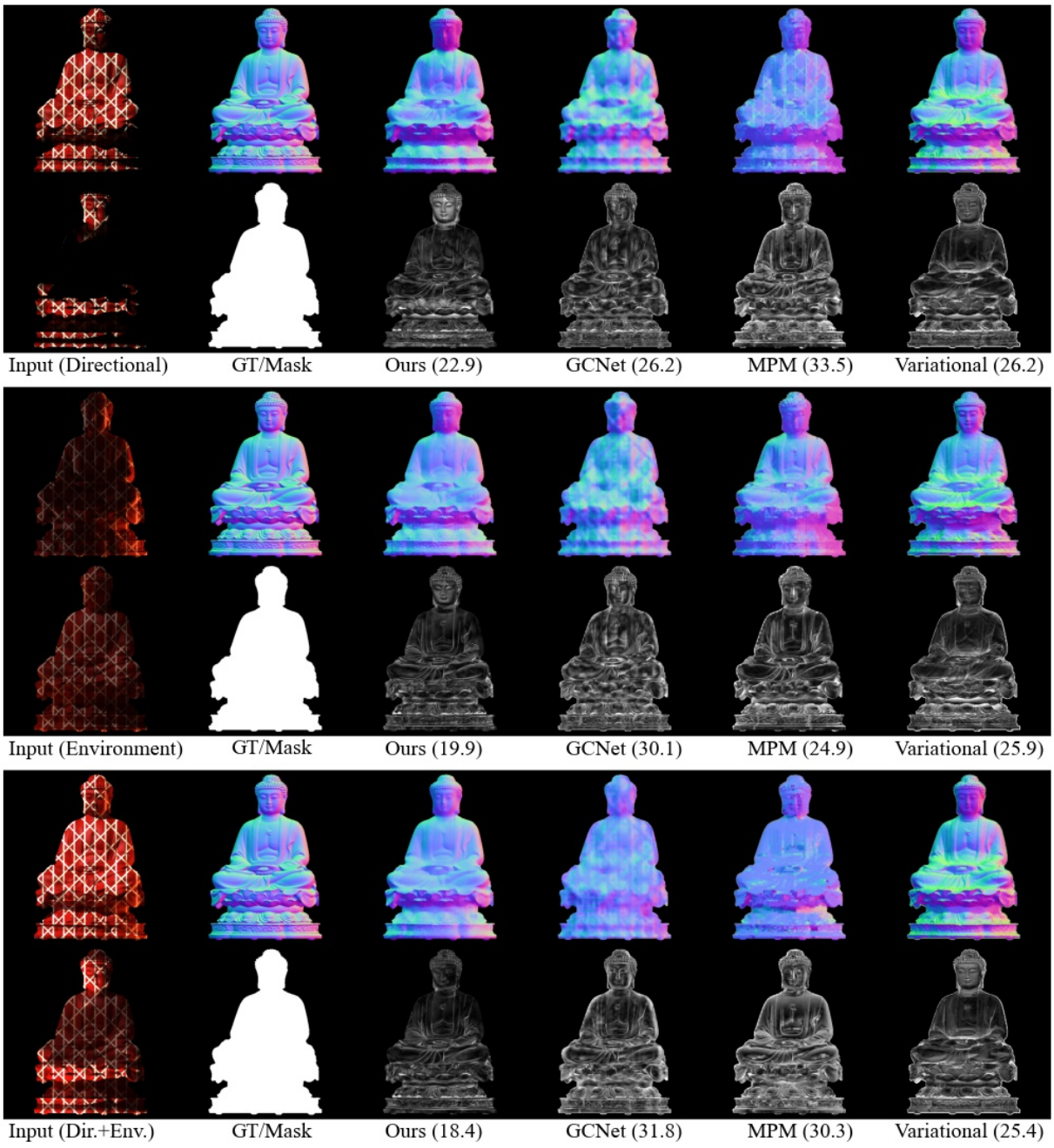

Figure 9. Results on object ID 7 (buddha, fabric-94 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

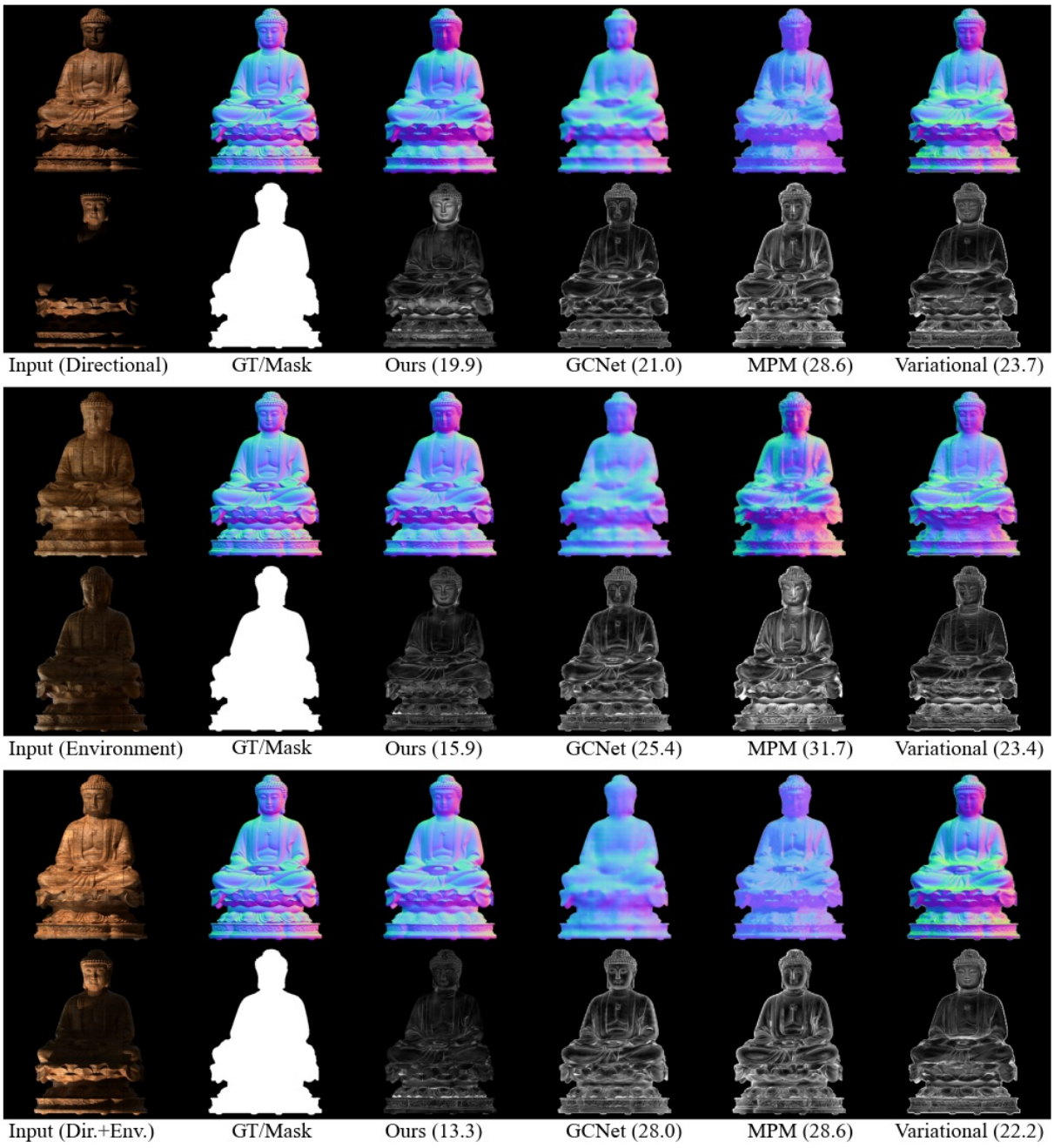

Figure 10. Results on object ID 8 (buddha, wood-parquet-57 [Wood]). MAEs (in degrees) are shown next to the name of the method.

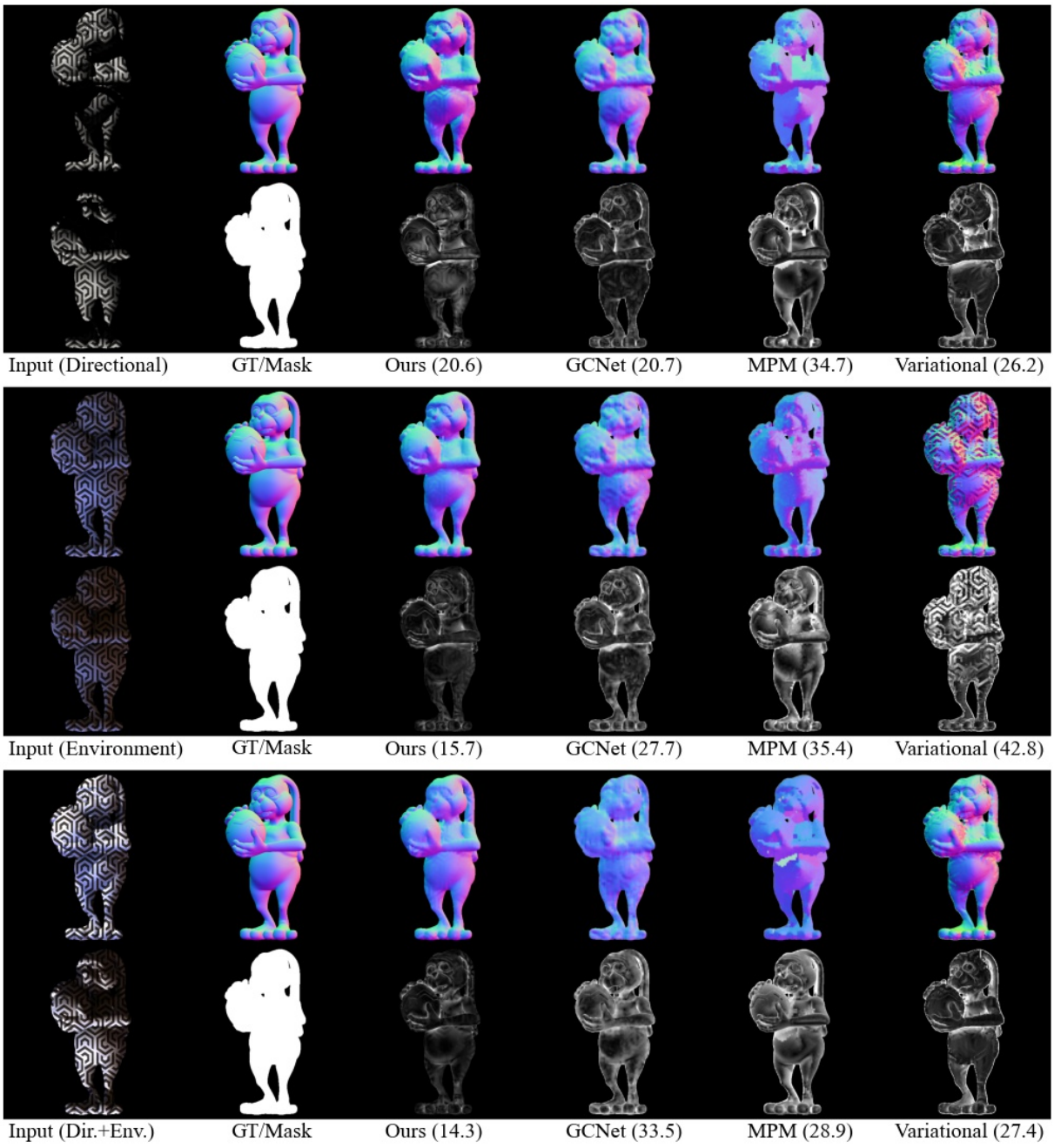

Figure 11. Results on object ID 9 (bunnydoll, fabric-96 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

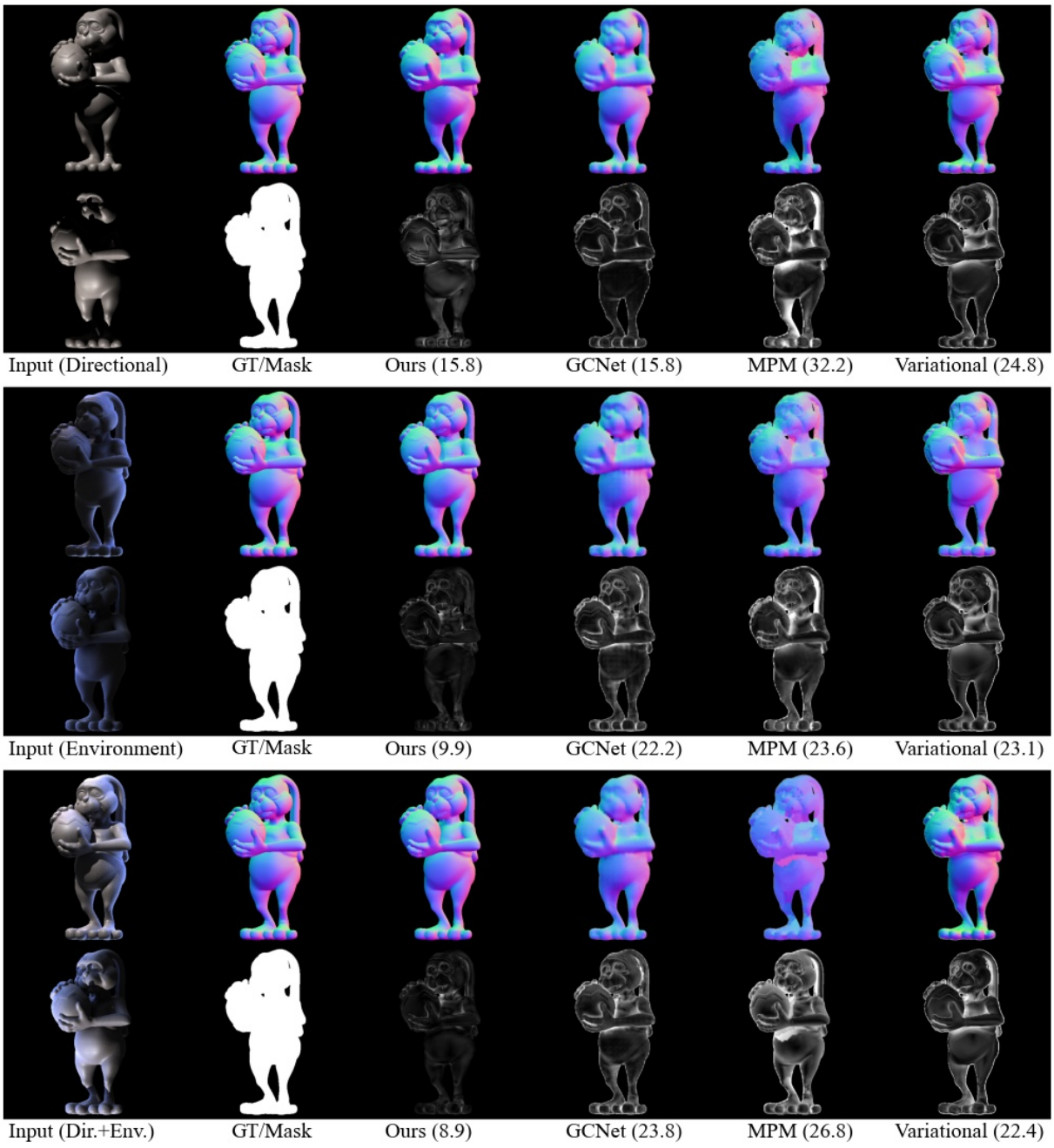

Figure 12. Results on object ID 10 (bunnydoll, wood-parquet-59 [Wood]). MAEs (in degrees) are shown next to the name of the method.

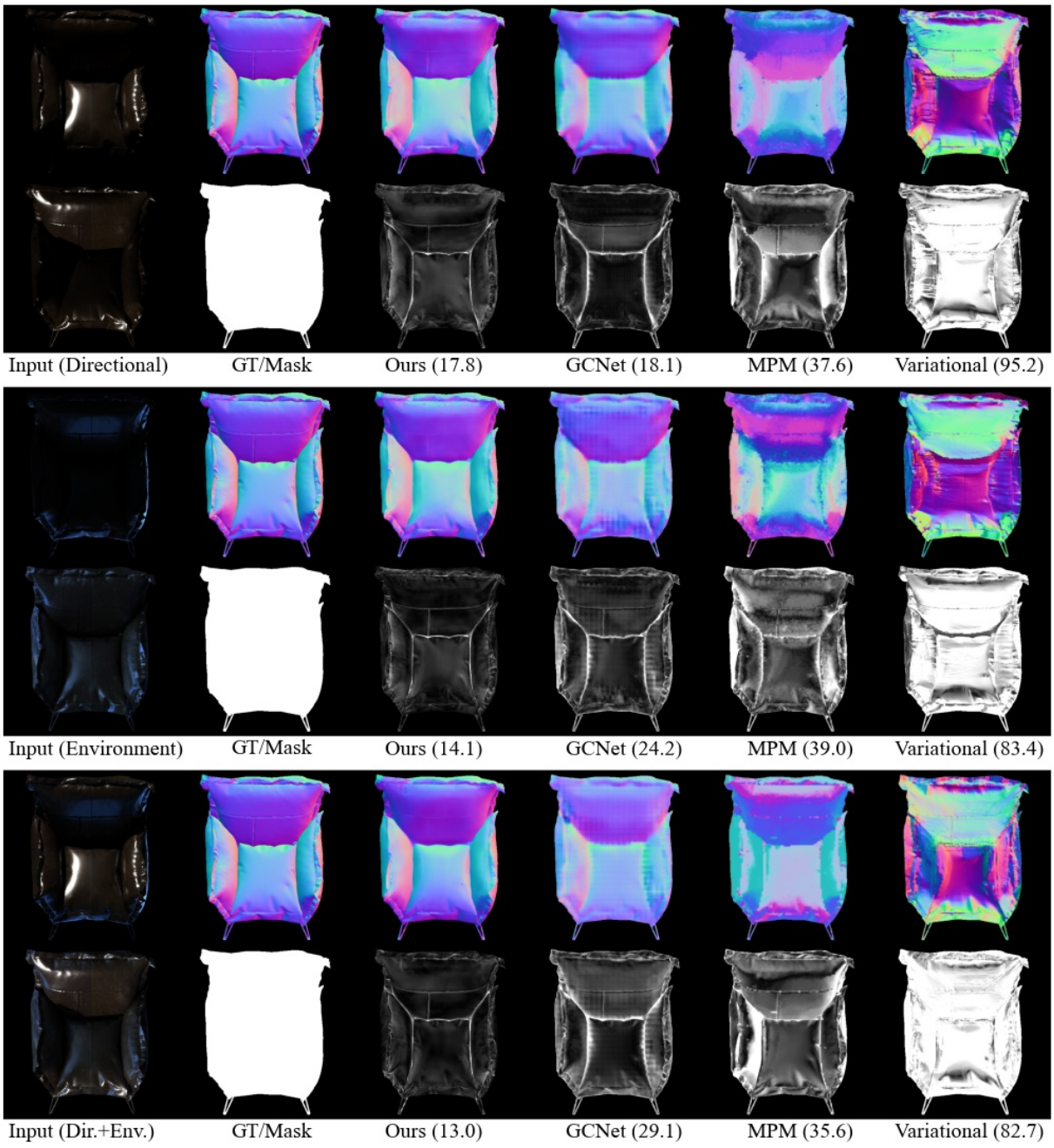

Figure 13. Results on object ID 11 (chair, brown-tiling [Floor]). MAEs (in degrees) are shown next to the name of the method.

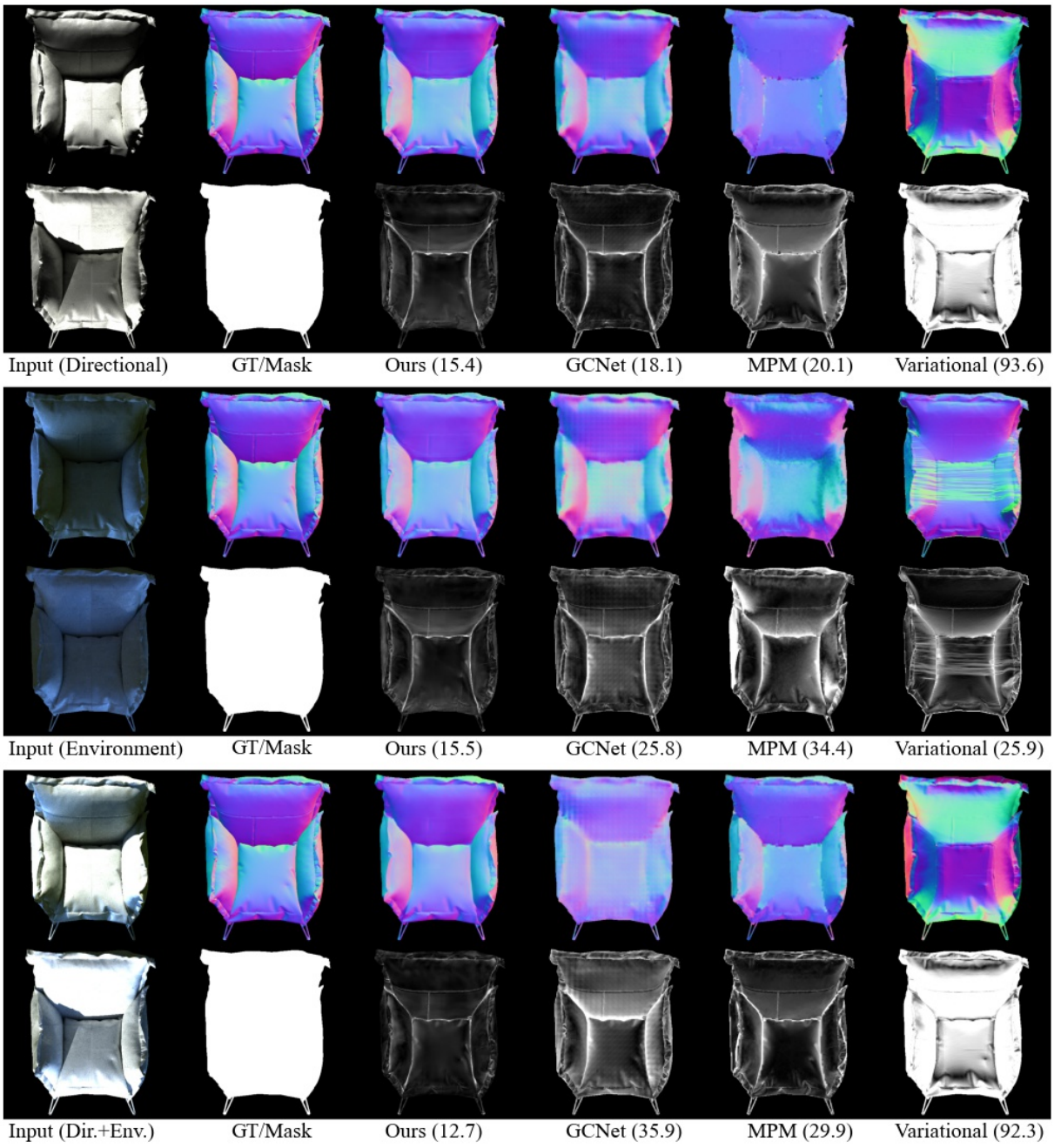

Figure 14. Results on object ID 12 (chair, wood-parquet-59 [Concrete]). MAEs (in degrees) are shown next to the name of the method.

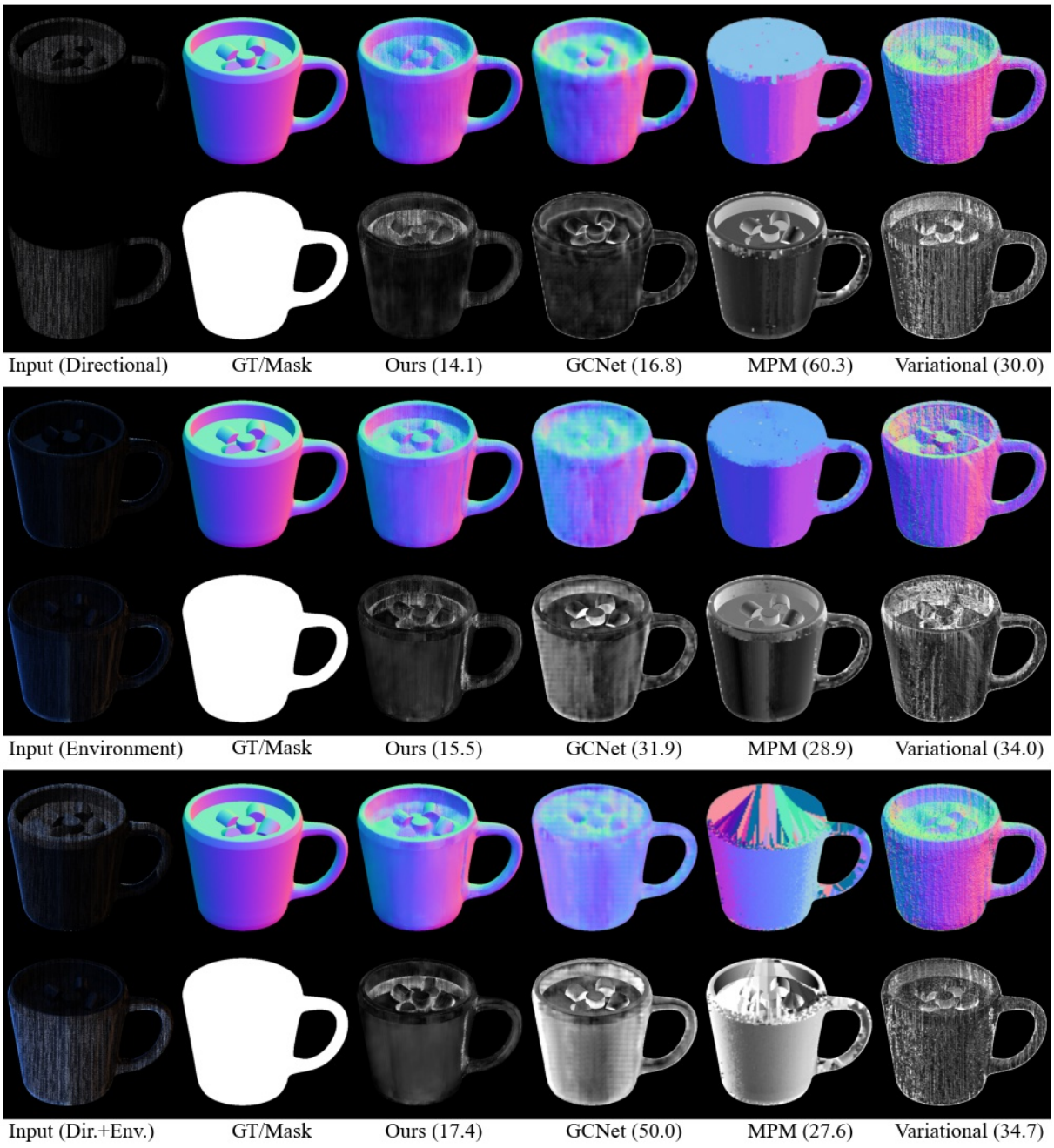

Figure 15. Results on object ID 13 (chococup, carpet-floor [Floor]). MAEs (in degrees) are shown next to the name of the method.

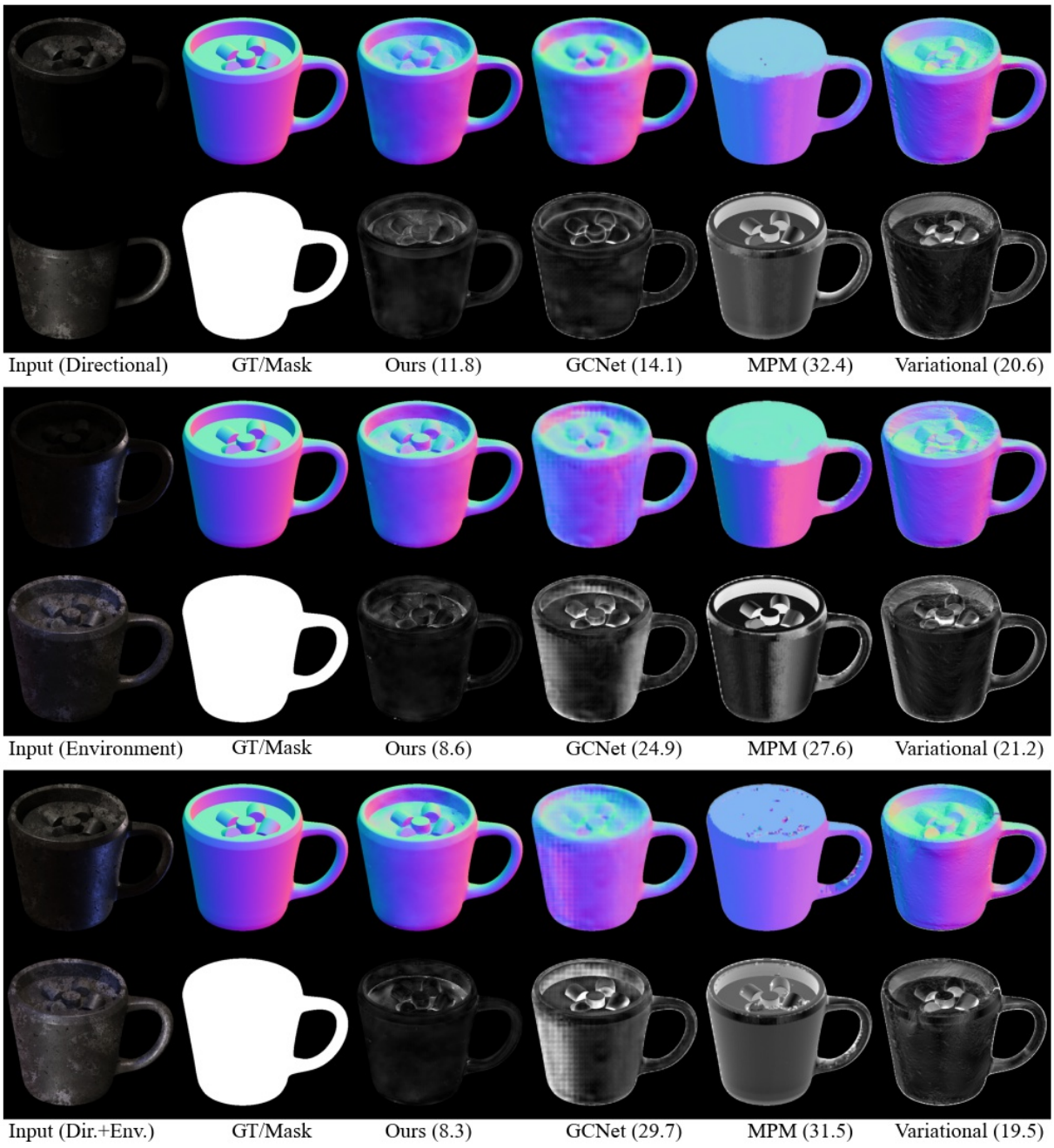

Figure 16. Results on object ID 14 (chococup, concrete-49 [Concrete]). MAEs (in degrees) are shown next to the name of the method.

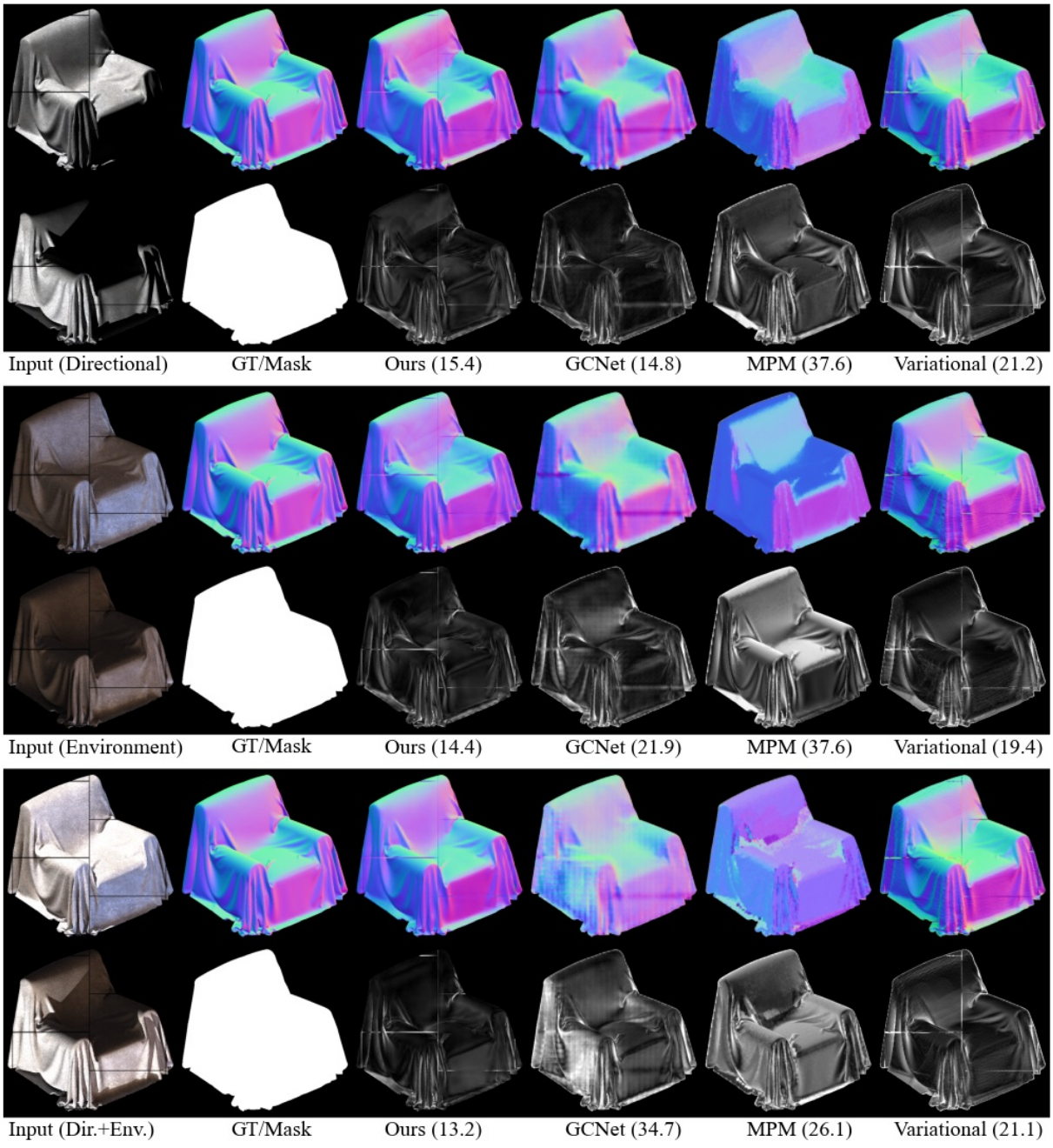

Figure 17. Results on object ID 15 (cloth-chair, concrete-tiling-6 [Concrete]). MAEs (in degrees) are shown next to the name of the method.

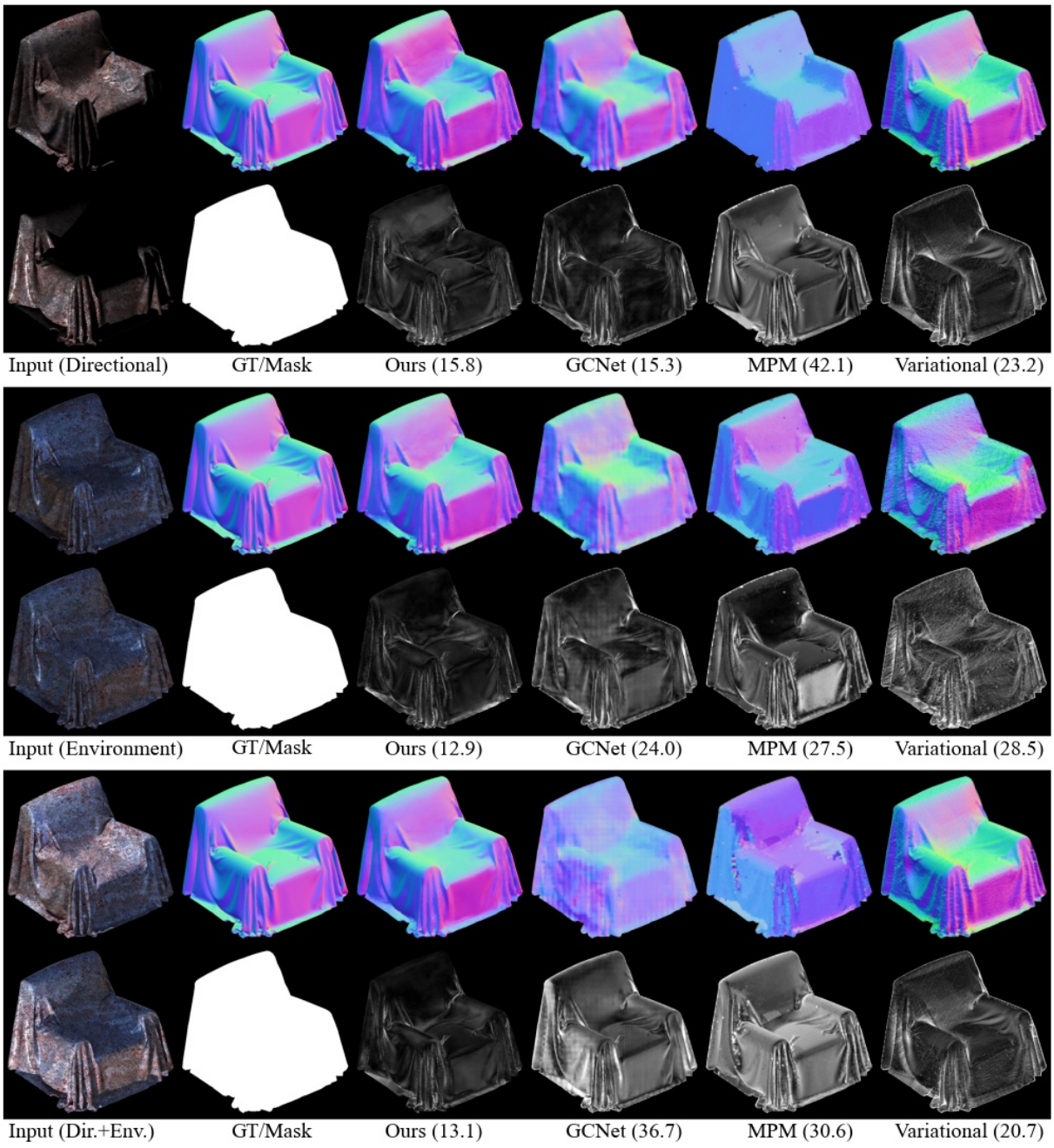

Figure 18. Results on object ID 16 (cloth-chair, copper-red-stone [Floor]). MAEs (in degrees) are shown next to the name of the method.

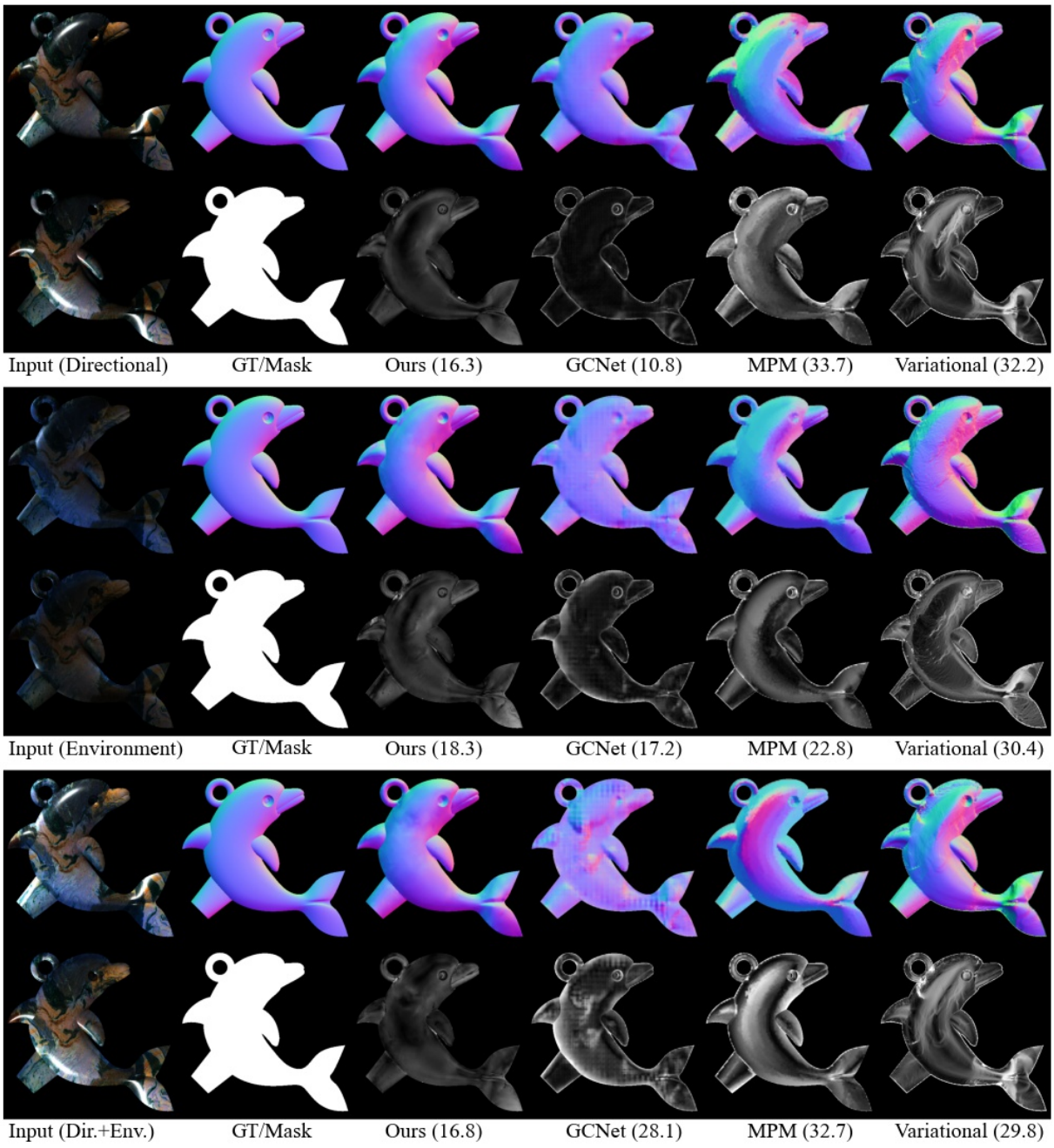

Figure 19. Results on object ID 17 (dolphin, explosion-blue-1 [Floor]). MAEs (in degrees) are shown next to the name of the method.

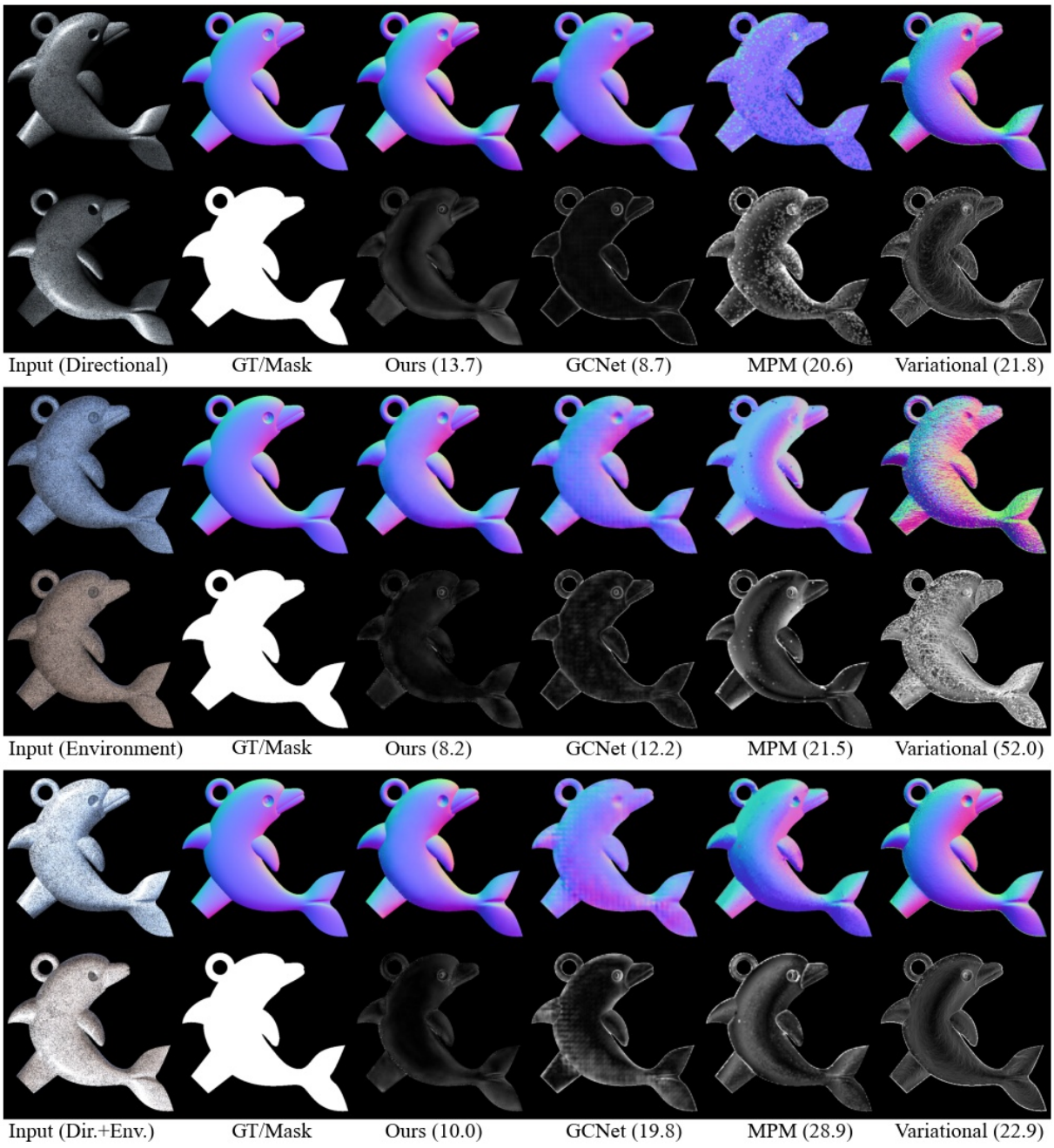

Figure 20. Results on object ID 18 (dolphin, seamless-concrete [Concrete]). MAEs (in degrees) are shown next to the name of the method.

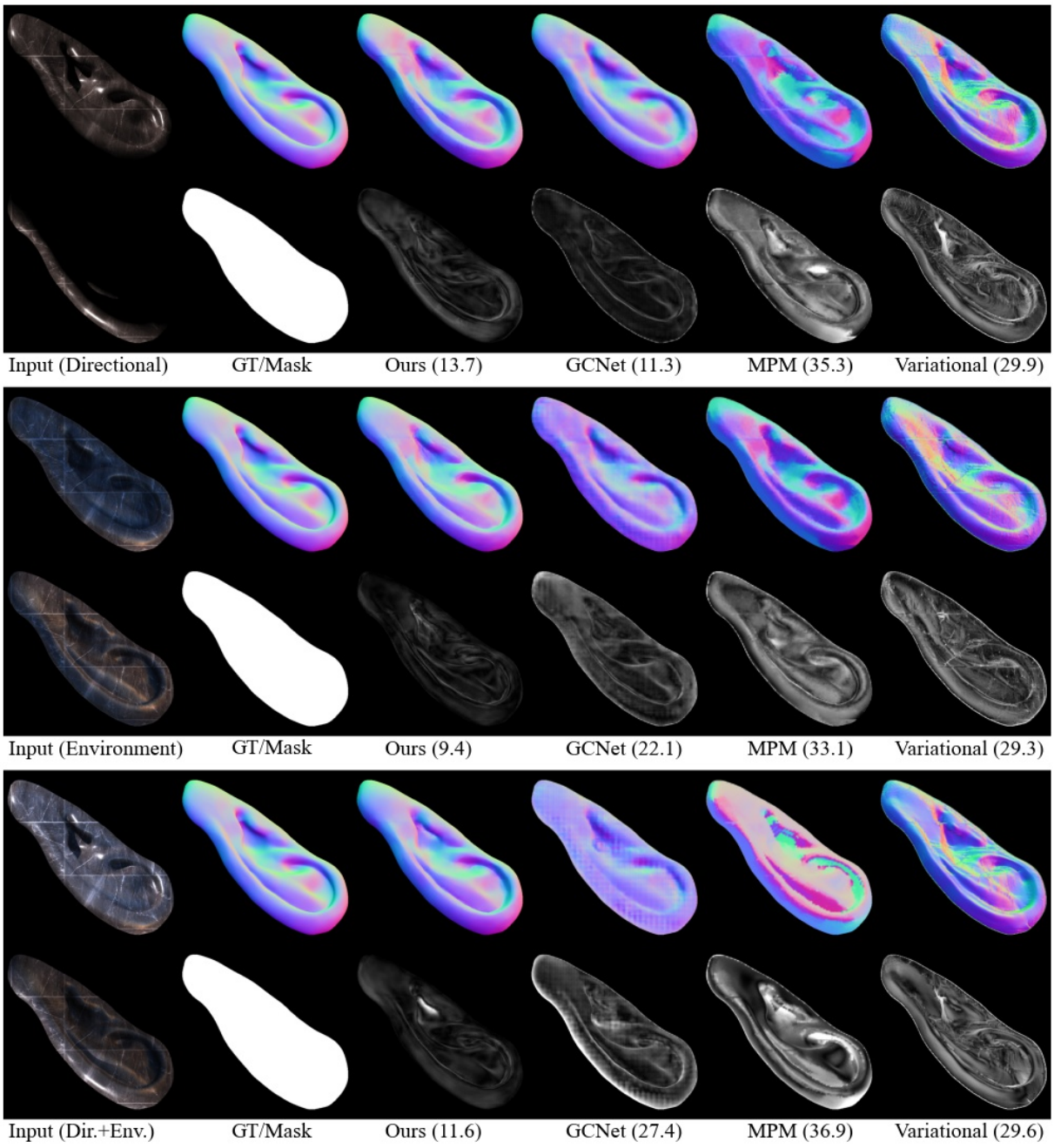

Figure 21. Results on object ID 19 (ear, tiling-42 [Floor]). MAEs (in degrees) are shown next to the name of the method.

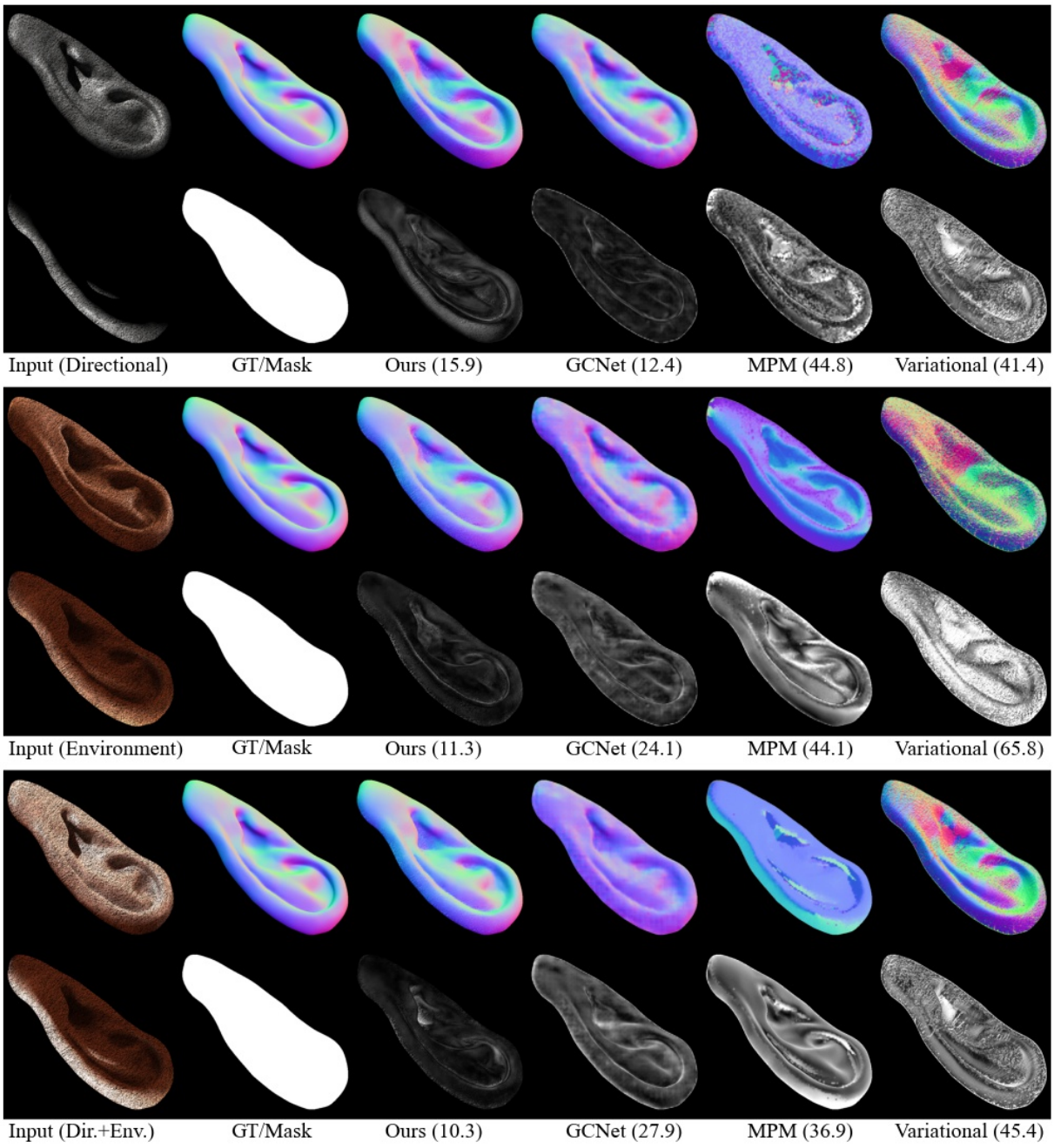

Figure 22. Results on object ID 20 (ear, white-concrete-46 [Concrete]). MAEs (in degrees) are shown next to the name of the method.

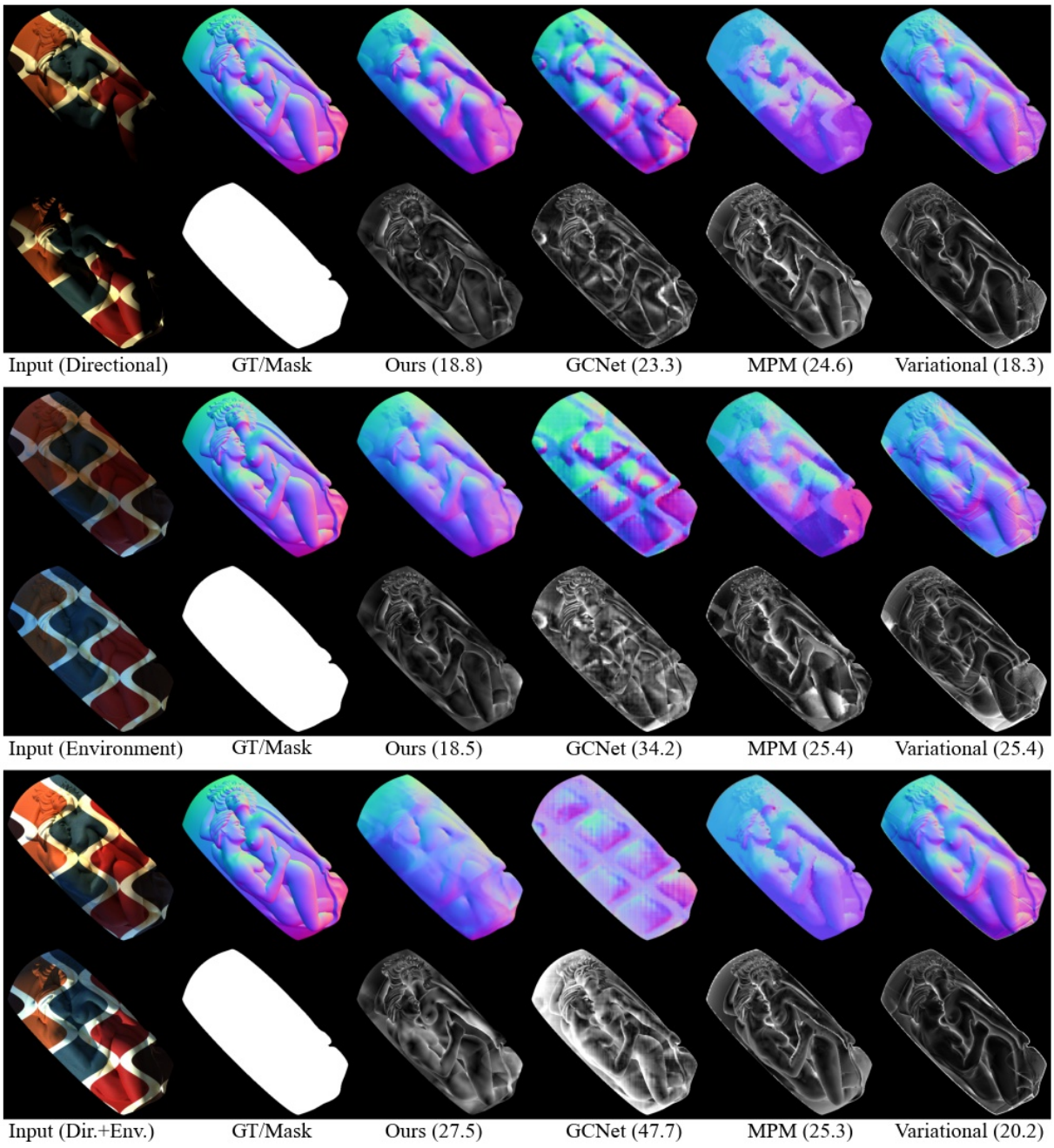

Figure 23. Results on object ID 21 (eden, fabric-85 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

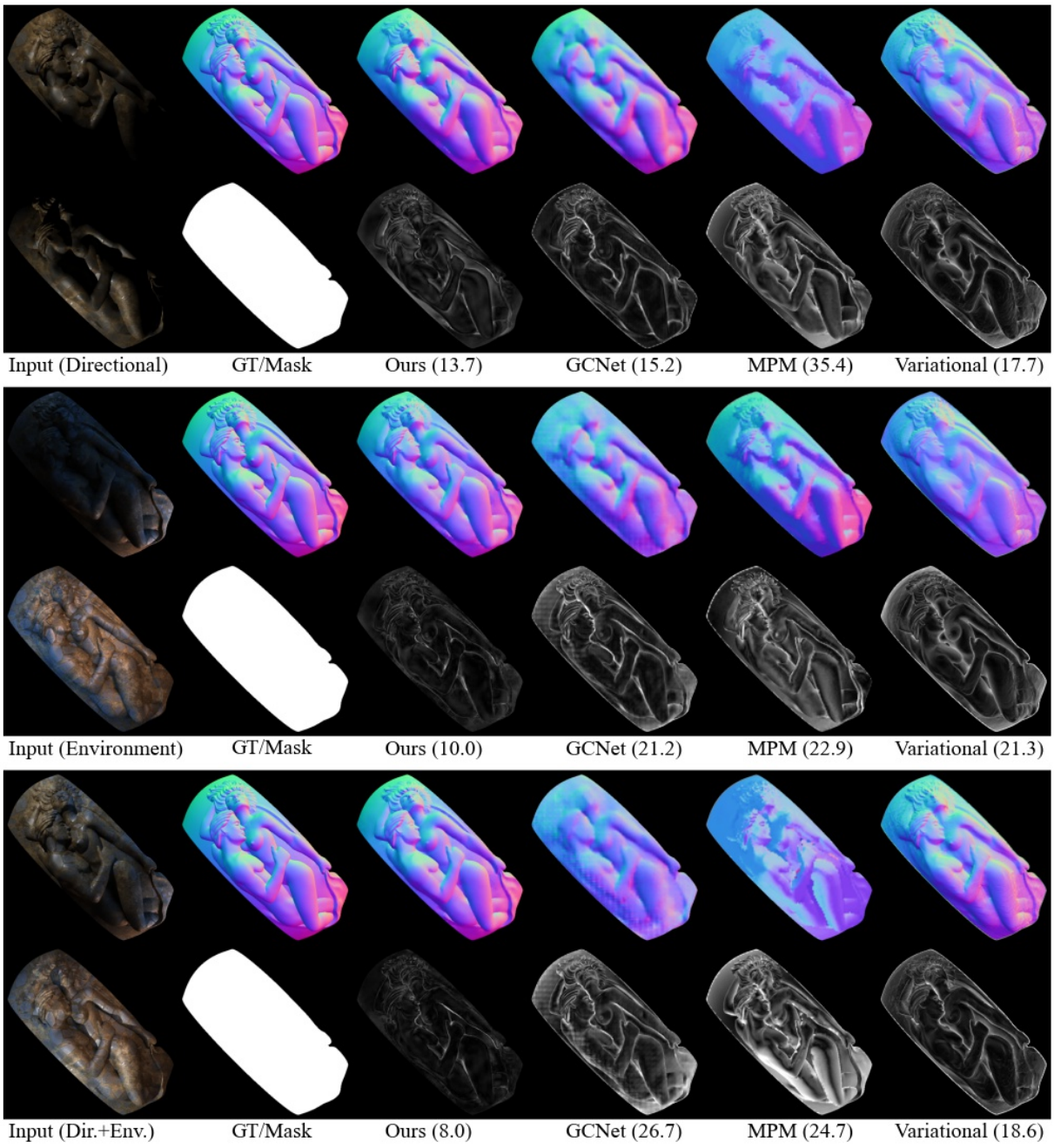

Figure 24. Results on object ID 22 (eden, ground-12 [Ground]). MAEs (in degrees) are shown next to the name of the method.

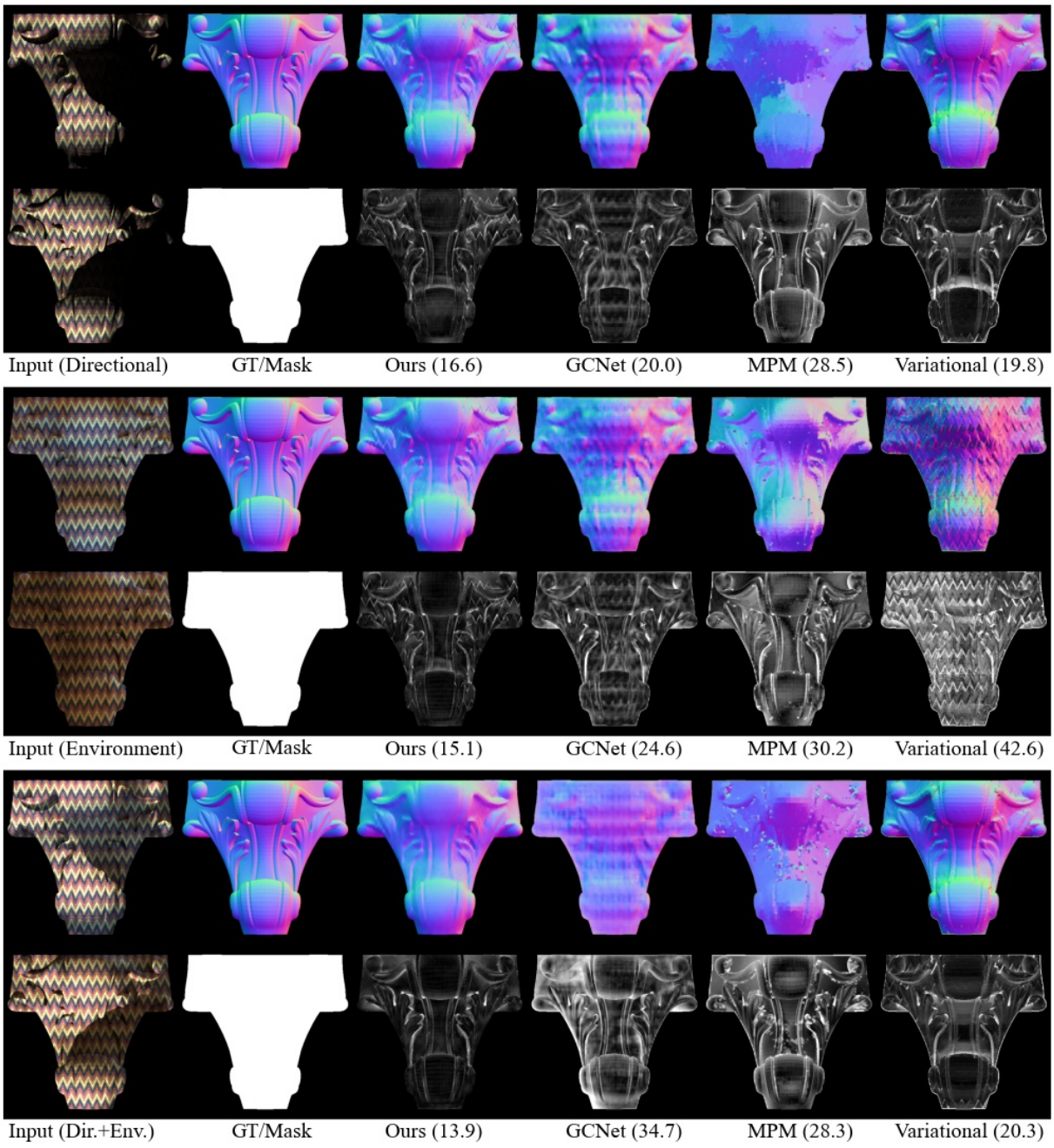

Figure 25. Results on object ID 23 (furniture-leg, fabric-86 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

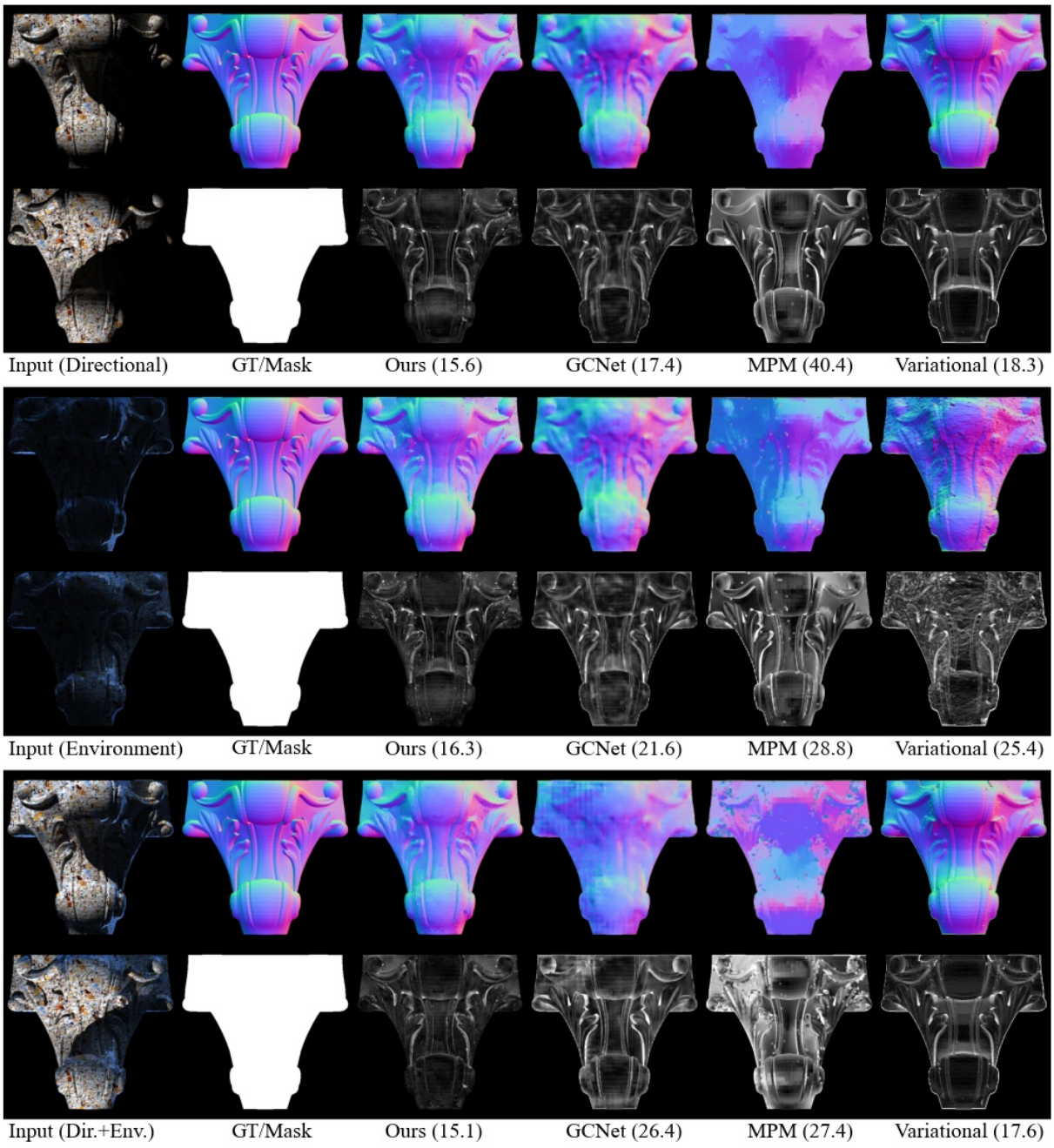

Figure 26. Results on object ID 24 (furniture-leg, sand-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.

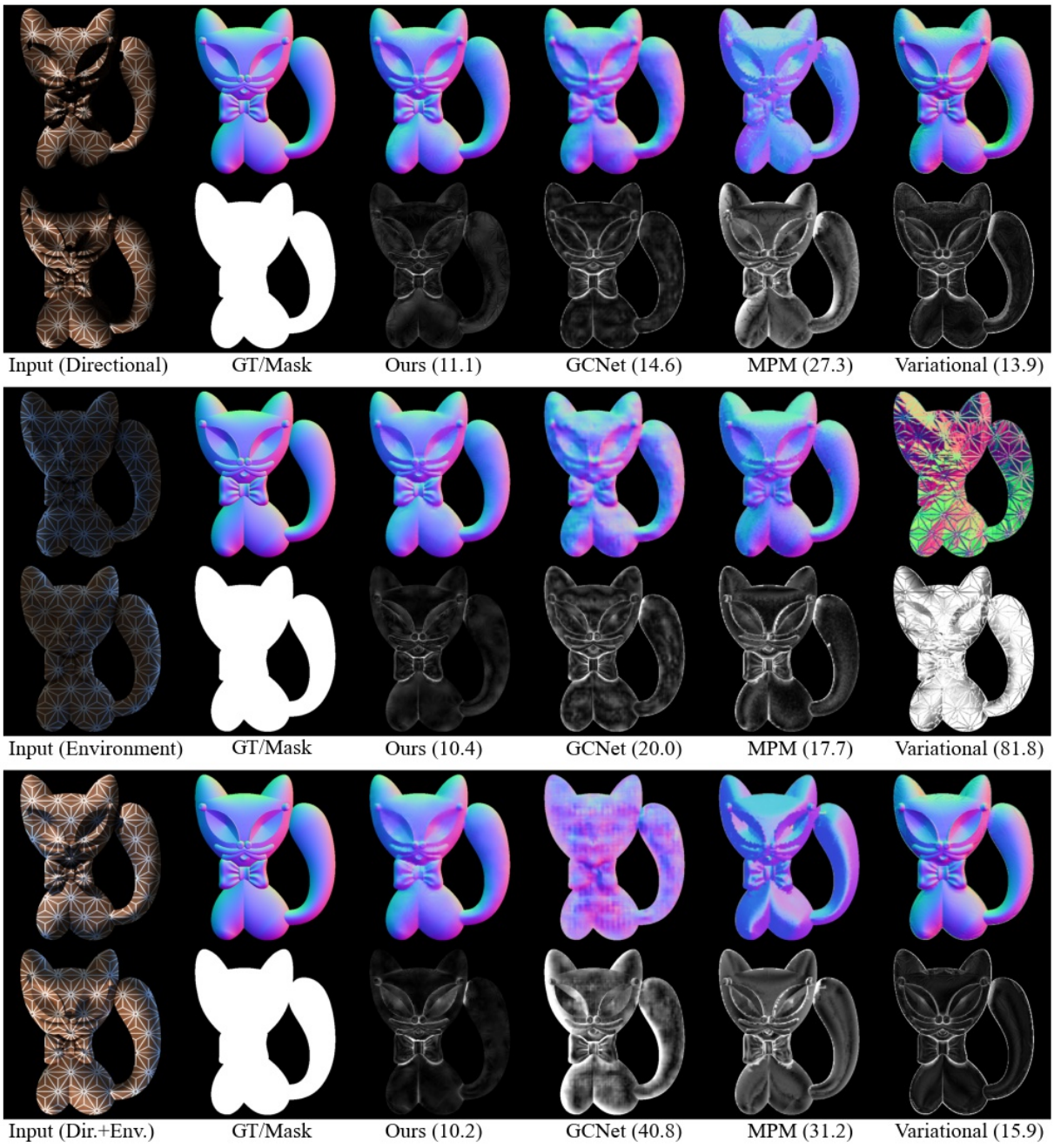

Figure 27. Results on object ID 25 (kitty, fabric-95 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

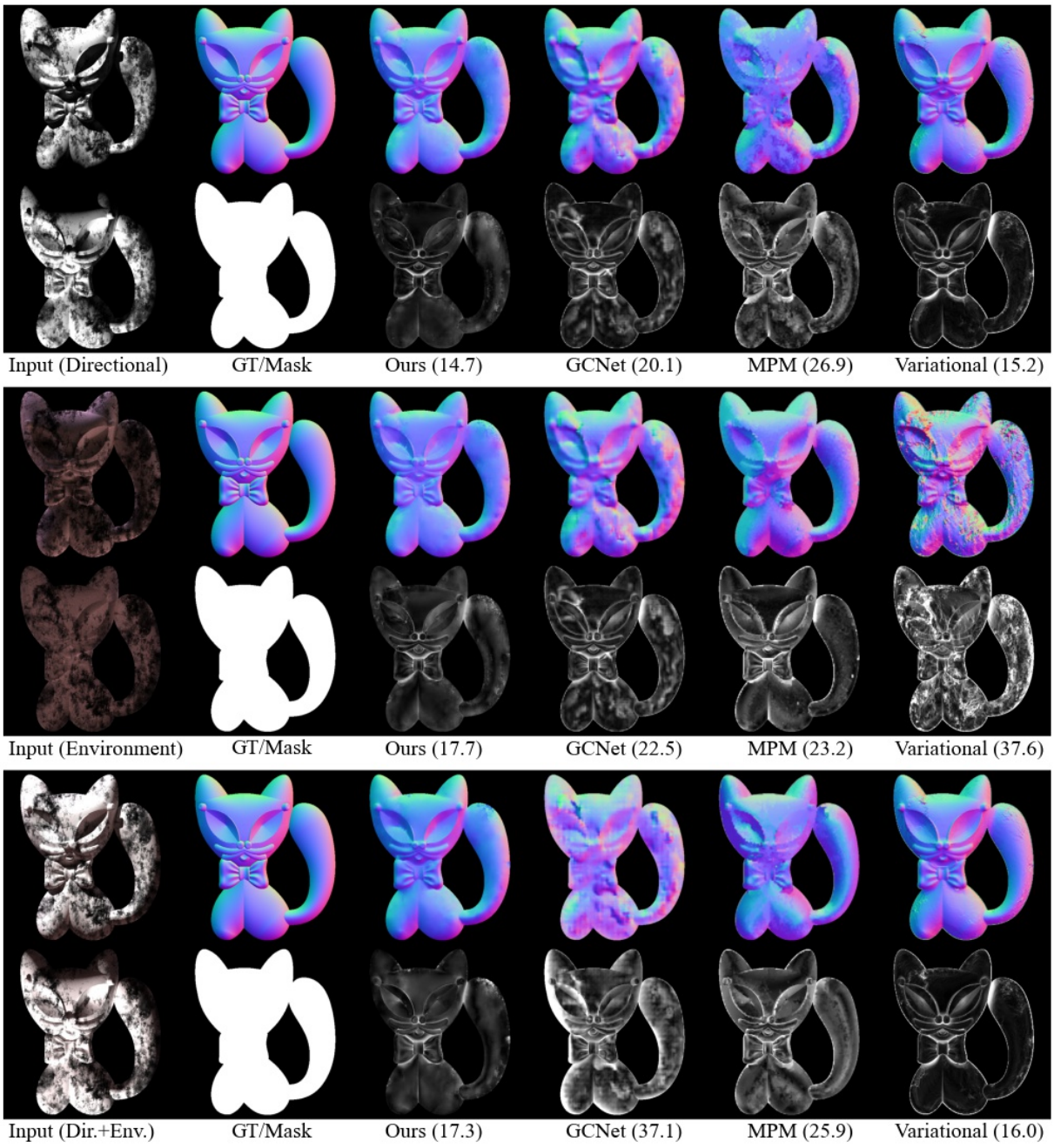

Figure 28. Results on object ID 26 (kitty, snow-ground-2 [Ground]). MAEs (in degrees) are shown next to the name of the method.

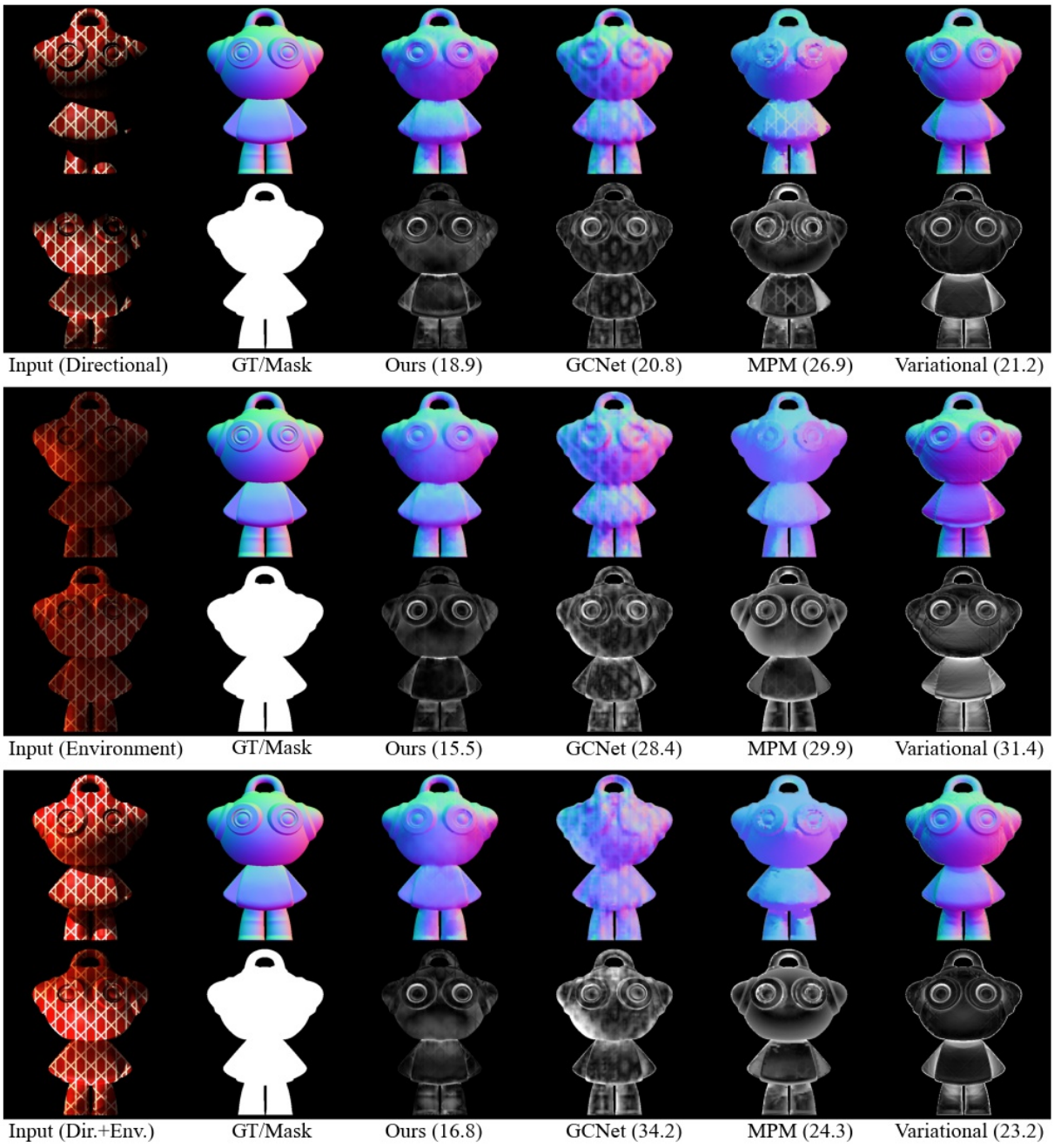

Figure 29. Results on object ID 27 (little-doll, fabric-94 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

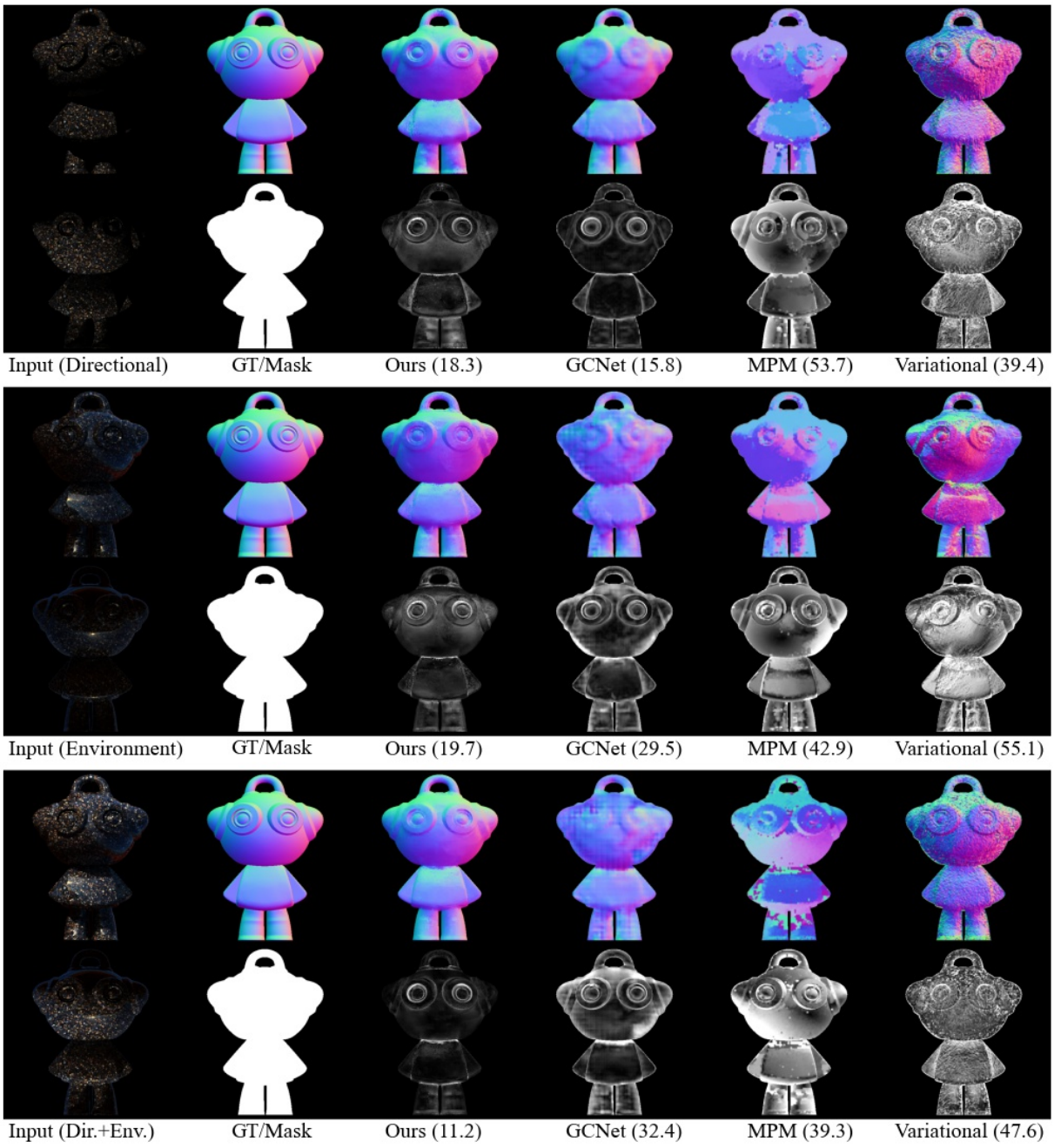

Figure 30. Results on object ID 28 (little-doll, pebble-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.

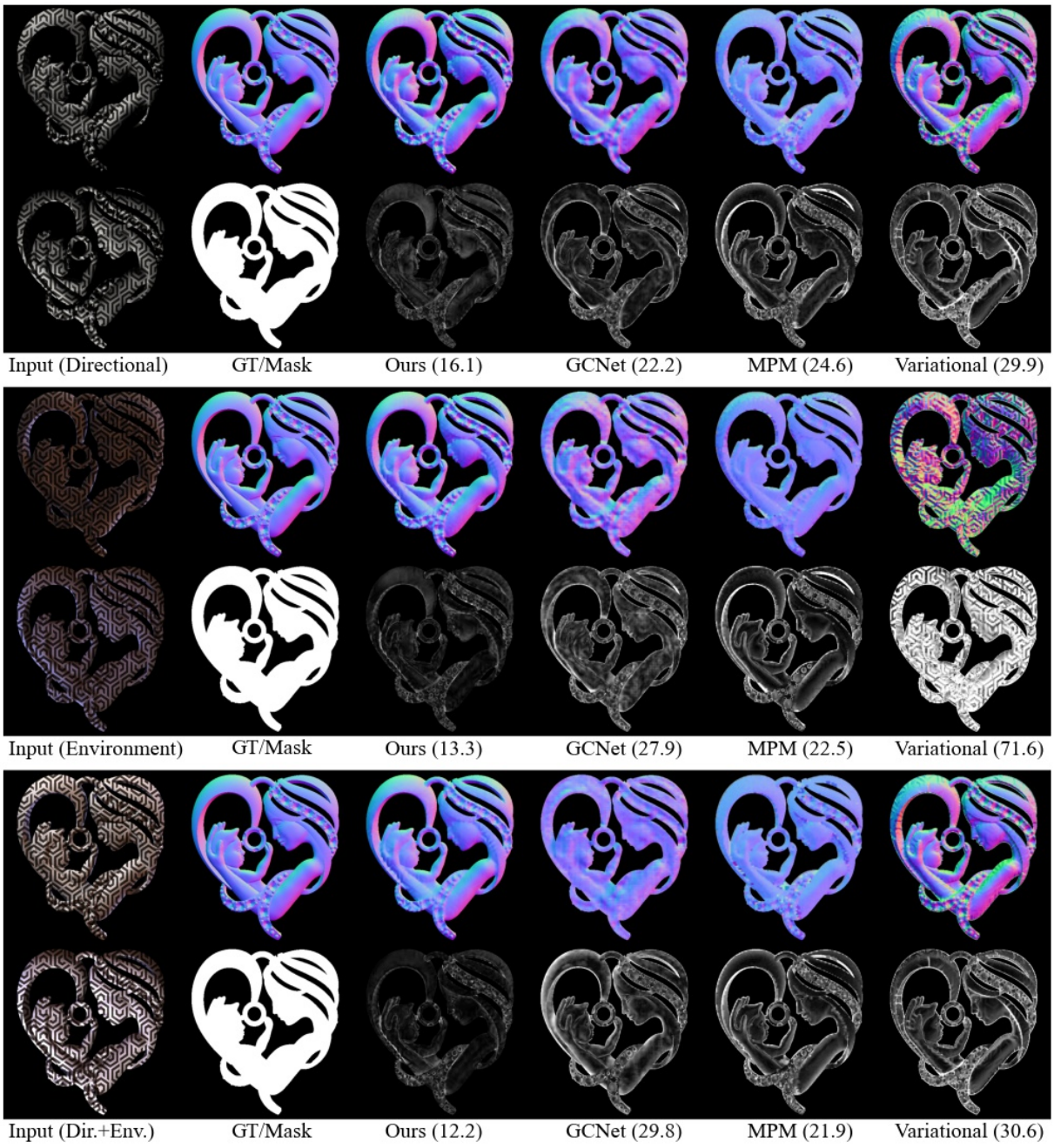

Figure 31. Results on object ID 29 (mother, fabric-96 [Fabric]). MAEs (in degrees) are shown next to the name of the method.

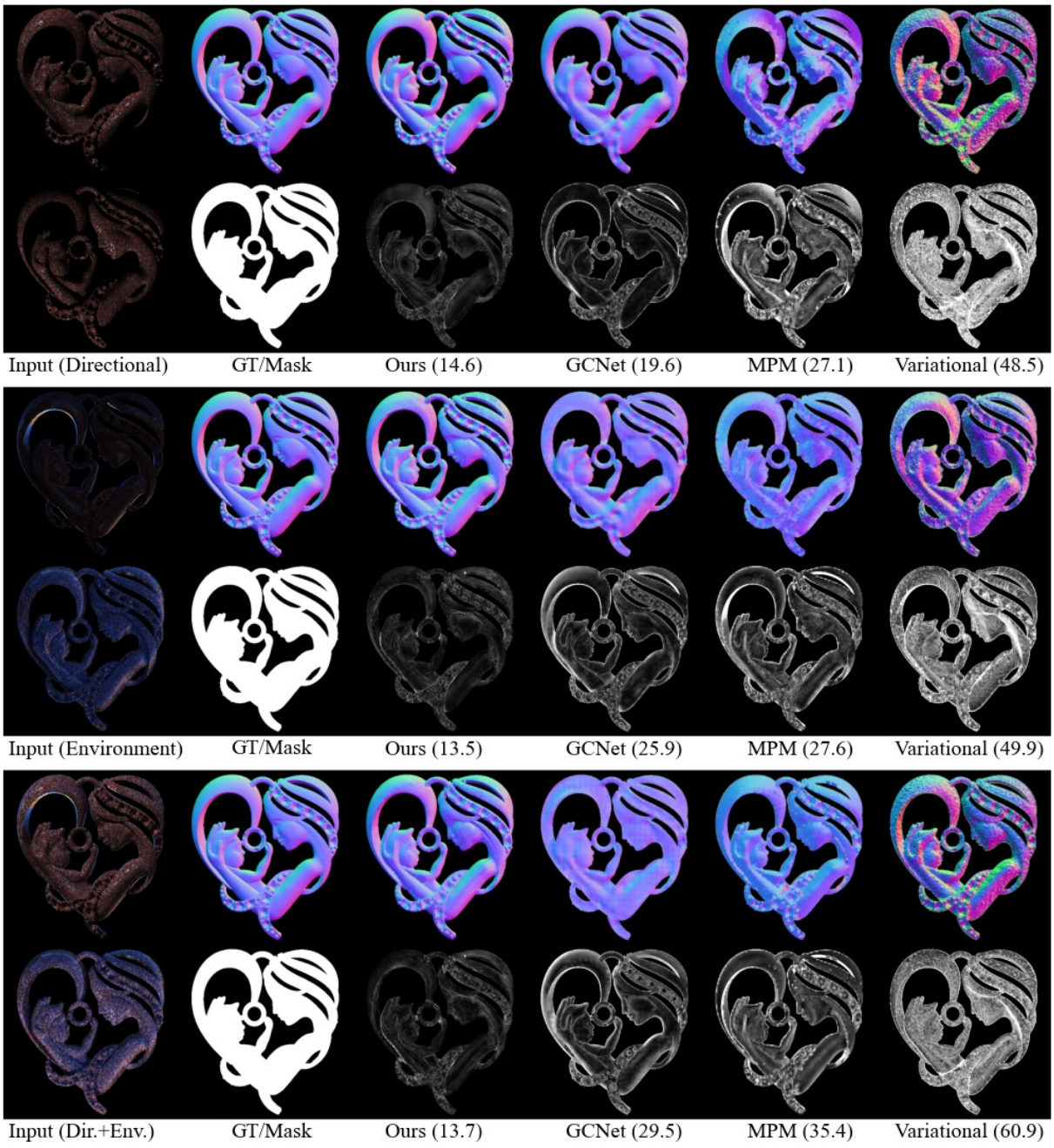

Figure 32. Results on object ID 30 (mother, pebble-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.

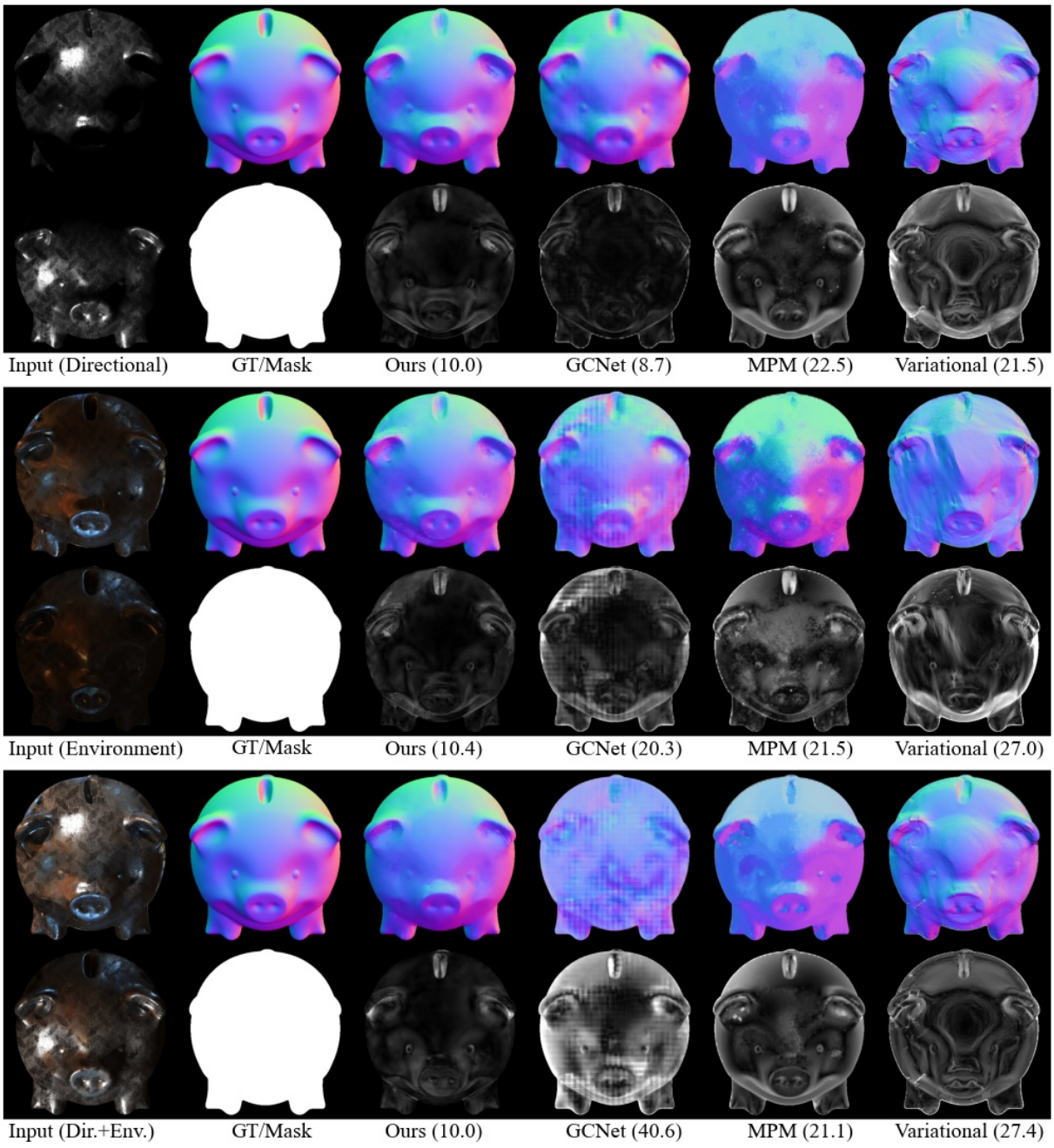

Figure 33. Results on object ID 31 (pig, black-metal-2 [Metal]). MAEs (in degrees) are shown next to the name of the method.

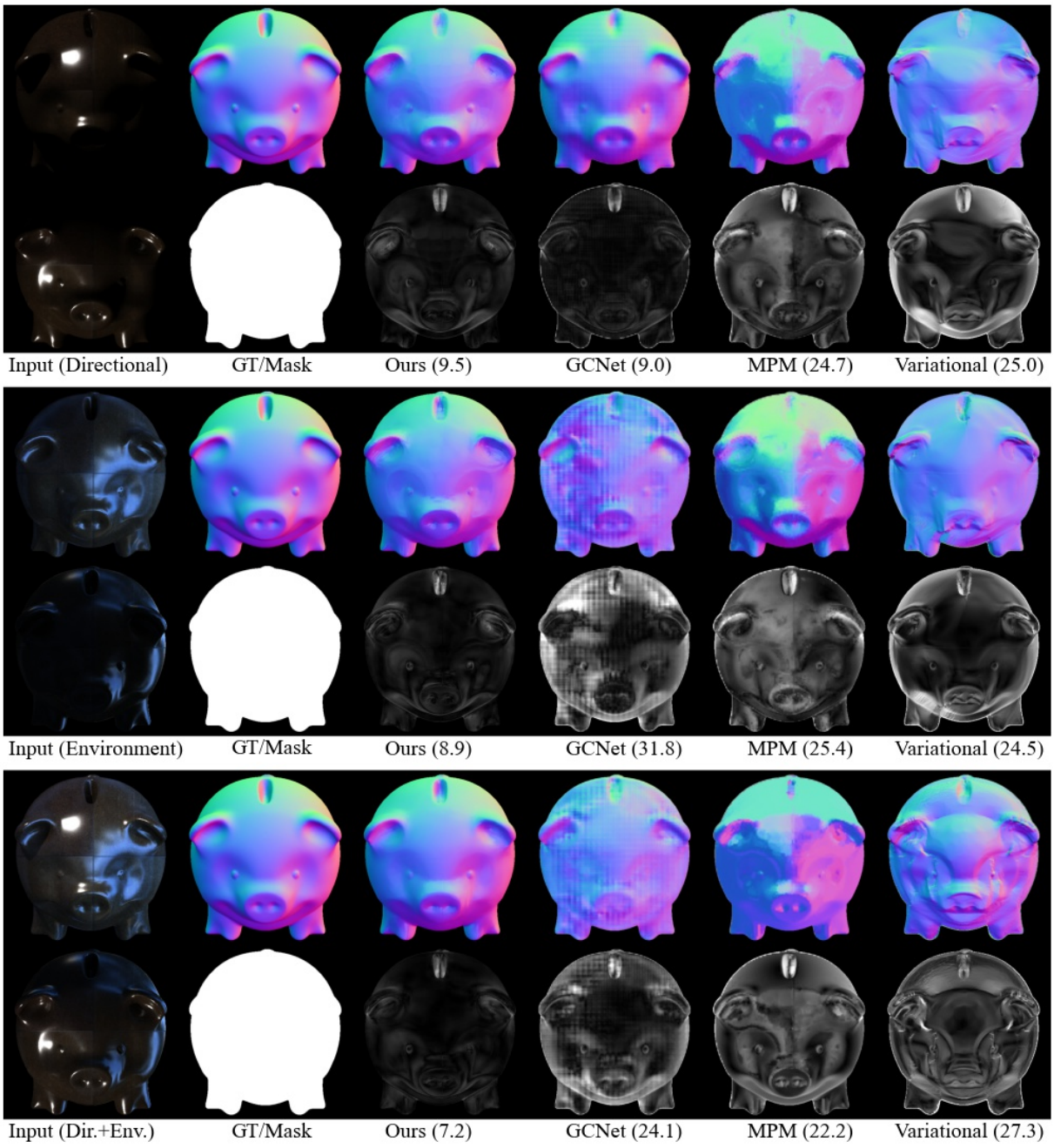

Figure 34. Results on object ID 32 (pig, brown-tiling [Floor]). MAEs (in degrees) are shown next to the name of the method.

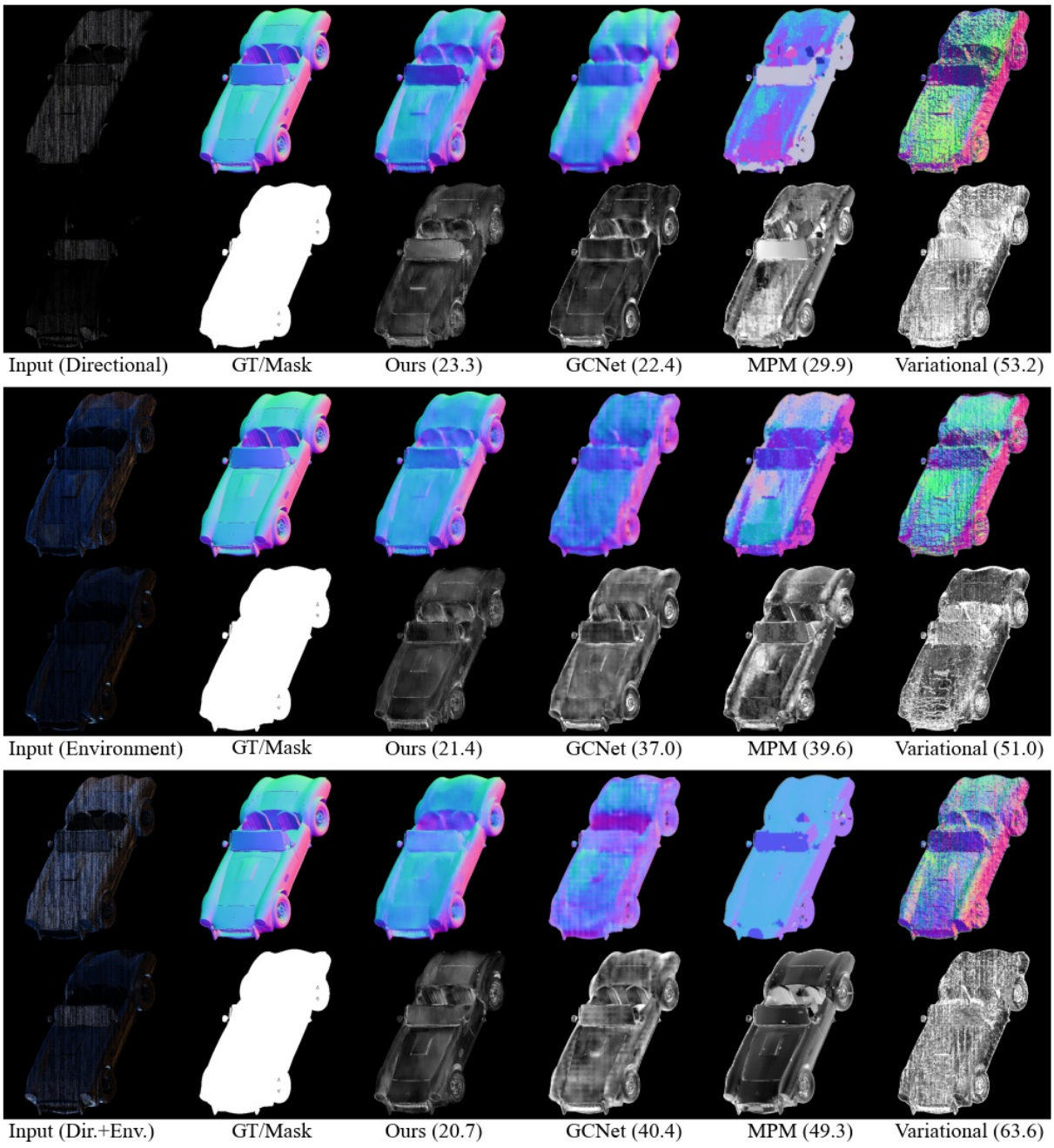

Figure 35. Results on object ID 33 (shelby, carpet-floor [Floor]). MAEs (in degrees) are shown next to the name of the method.

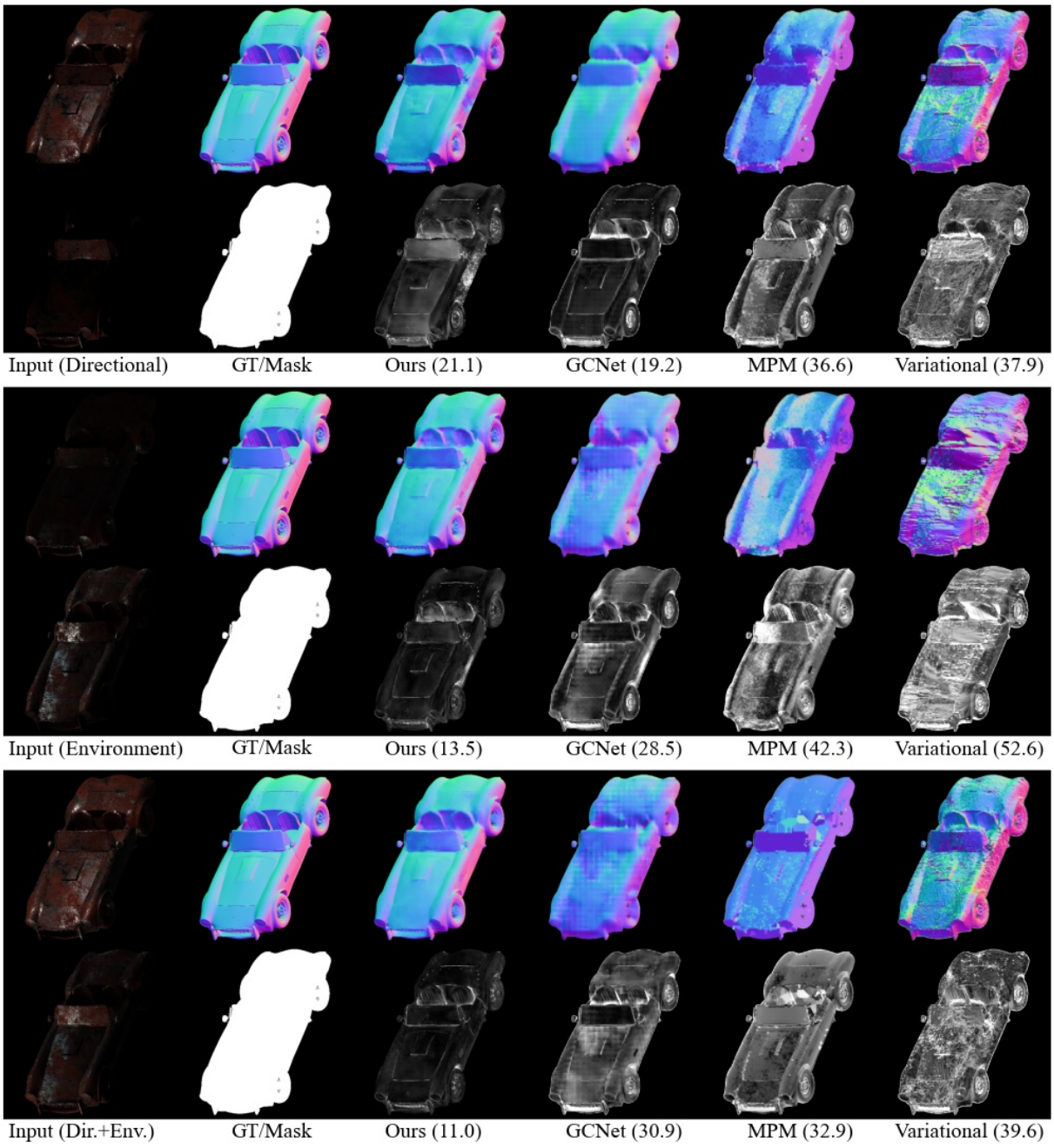

Figure 36. Results on object ID 34 (shelby, metal-frame-3 [Metal]). MAEs (in degrees) are shown next to the name of the method.

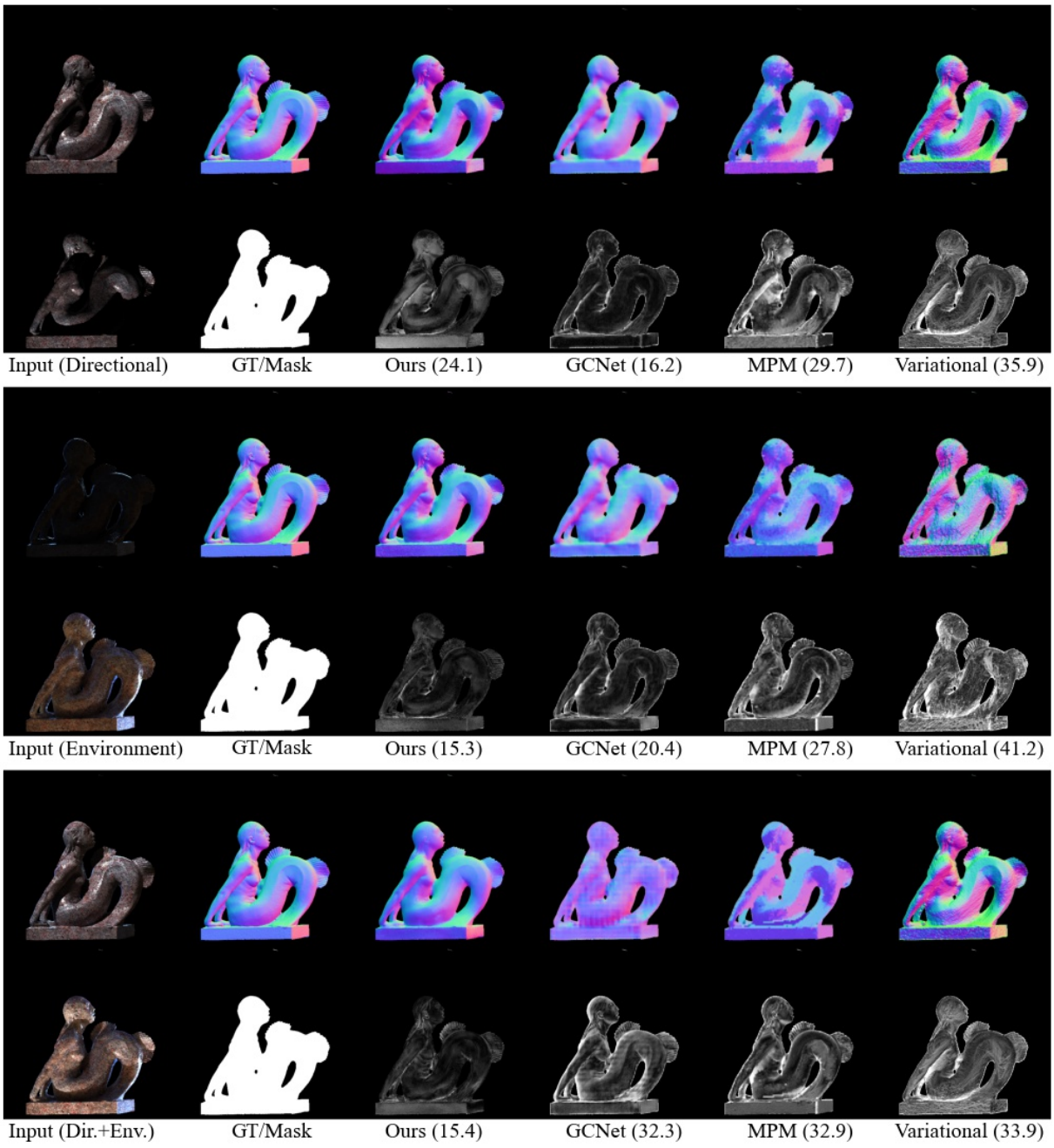

Figure 37. Results on object ID 35 (silane, copper-red-stone [Floor]). MAEs (in degrees) are shown next to the name of the method.

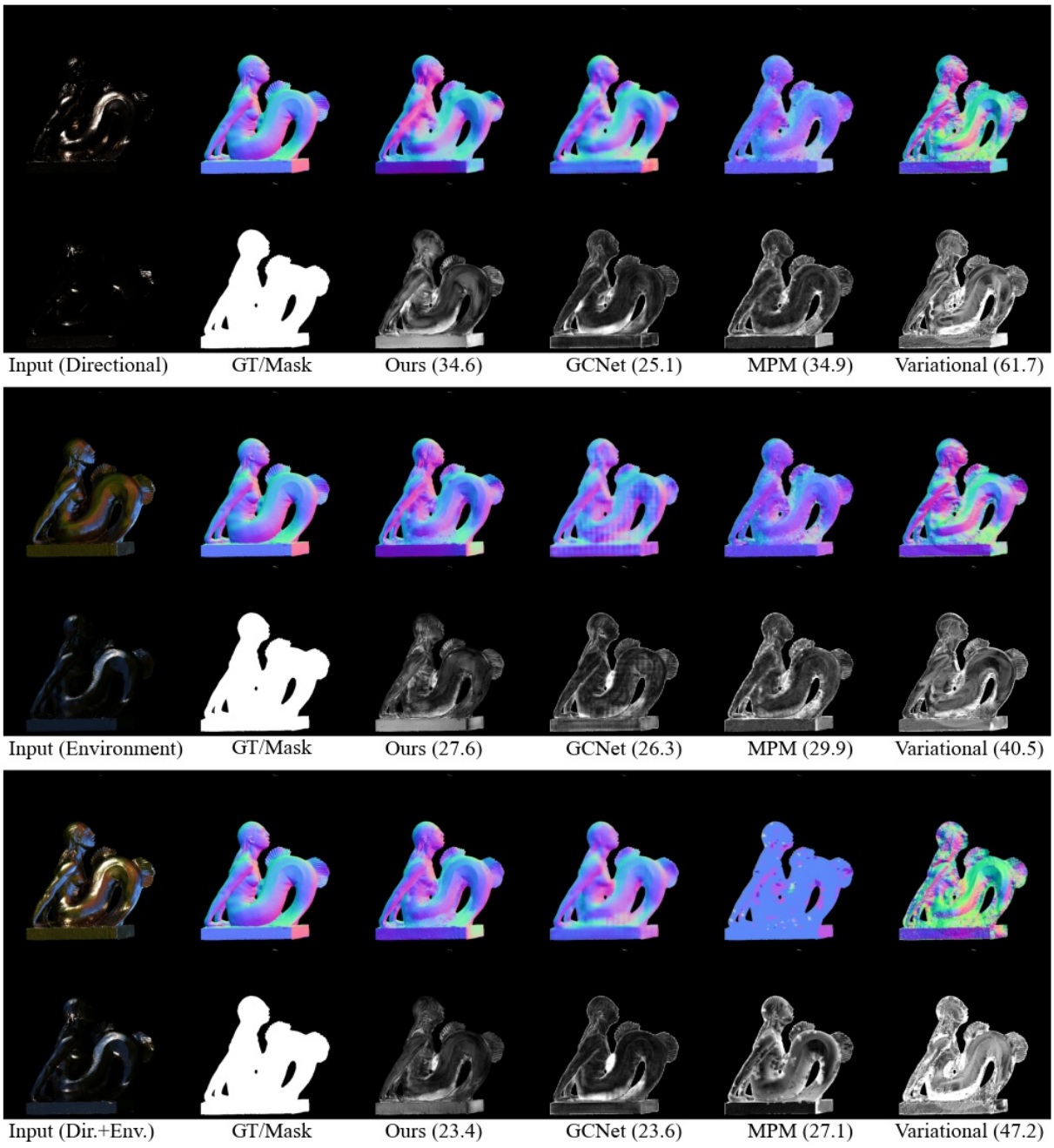

Figure 38. Results on object ID 36 (silane, old-bronze [Metal]). MAEs (in degrees) are shown next to the name of the method.

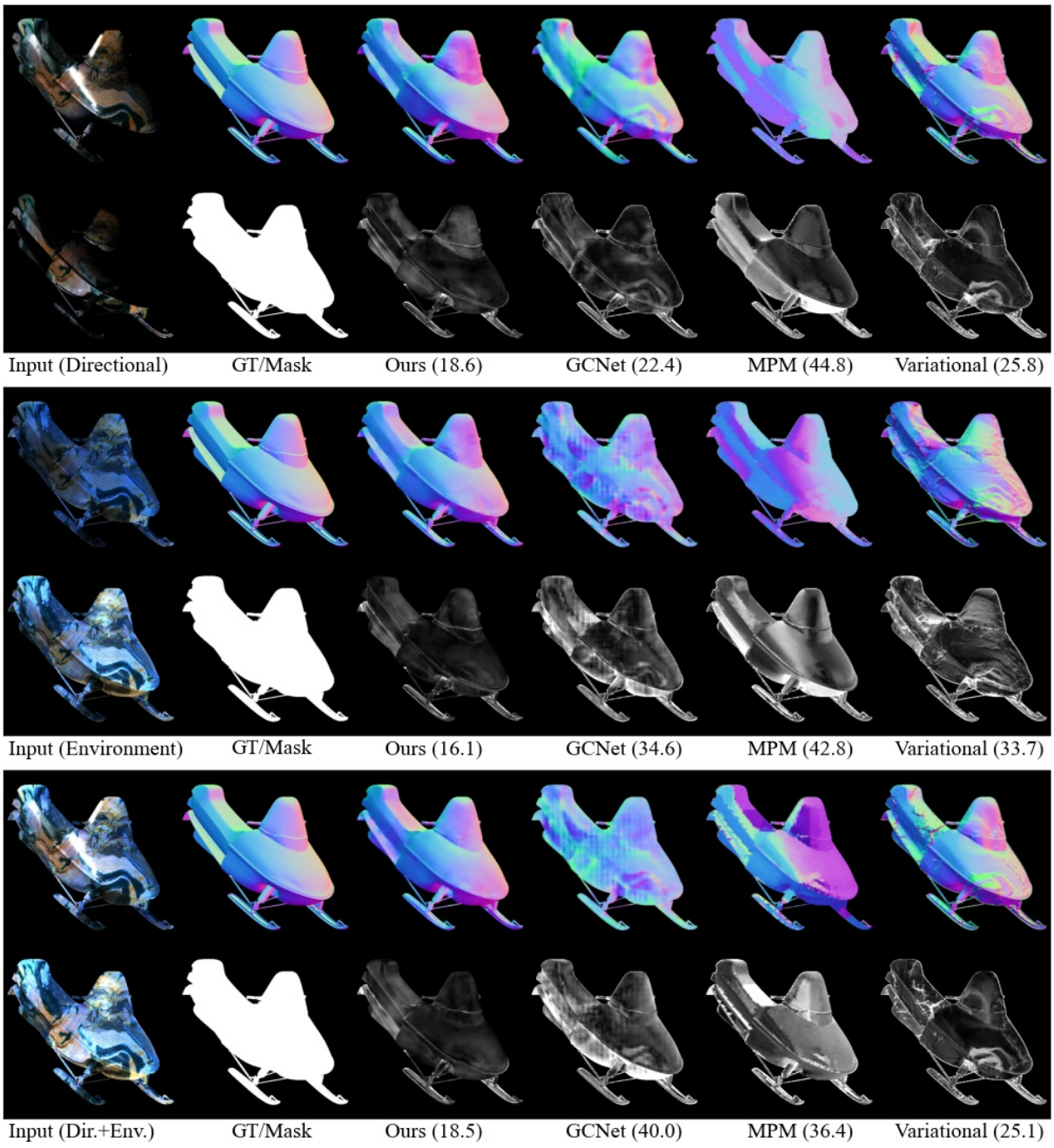

Figure 39. Results on object ID 37 (snowmobile, explosion-blue [Floor]). MAEs (in degrees) are shown next to the name of the method.

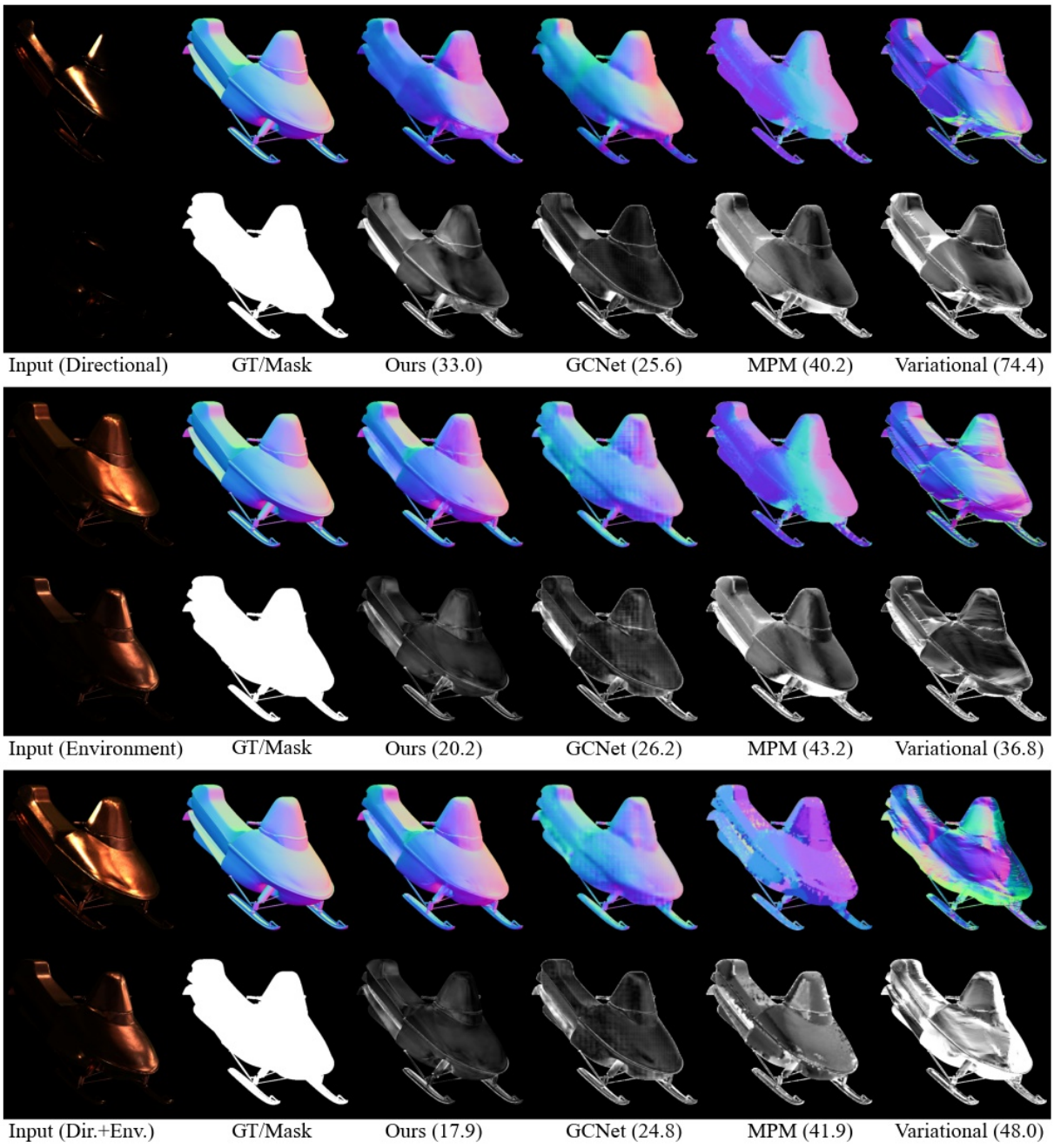

Figure 40. Results on object ID 38 (snowmobile, old-copper [Metal]). MAEs (in degrees) are shown next to the name of the method.

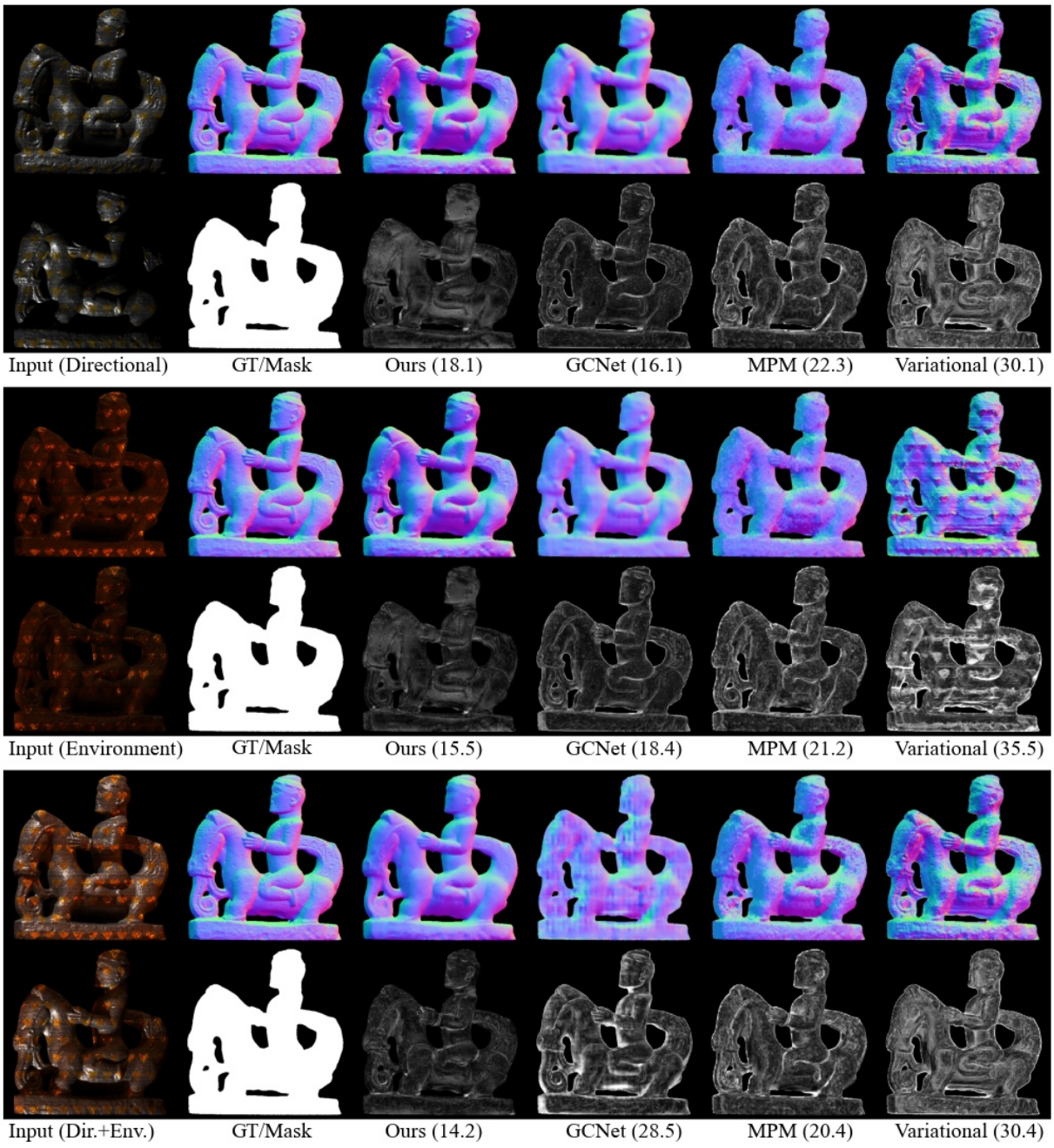

Figure 41. Results on object ID 39 (statue-2, metal-plate [Metal]). MAEs (in degrees) are shown next to the name of the method.

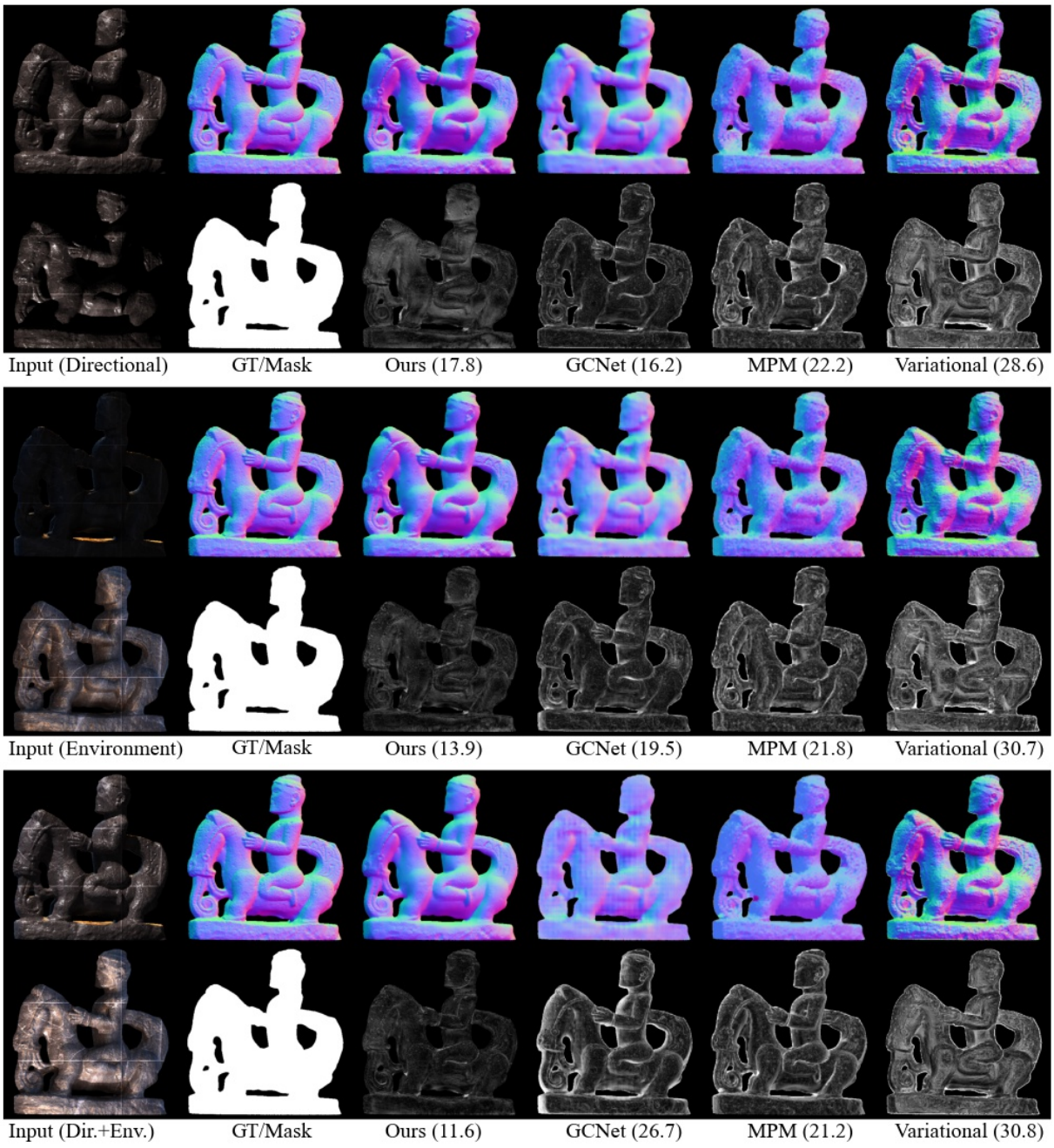

Figure 42. Results on object ID 40 (statue-2, tiling-42 [Floor]). MAEs (in degrees) are shown next to the name of the method.

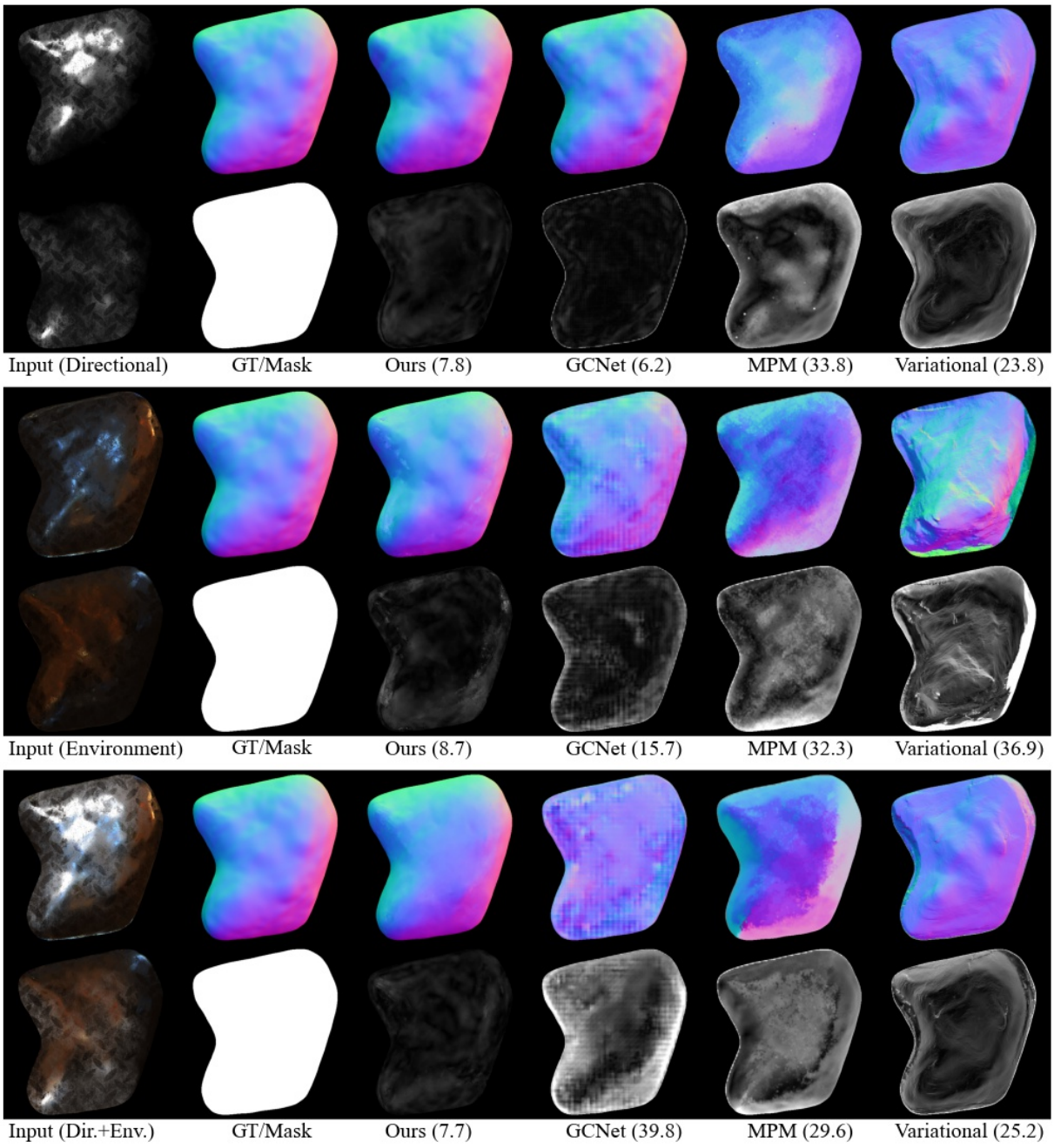

Figure 43. Results on object ID 41 (stone, black-metal-2 [Metal]). MAEs (in degrees) are shown next to the name of the method.

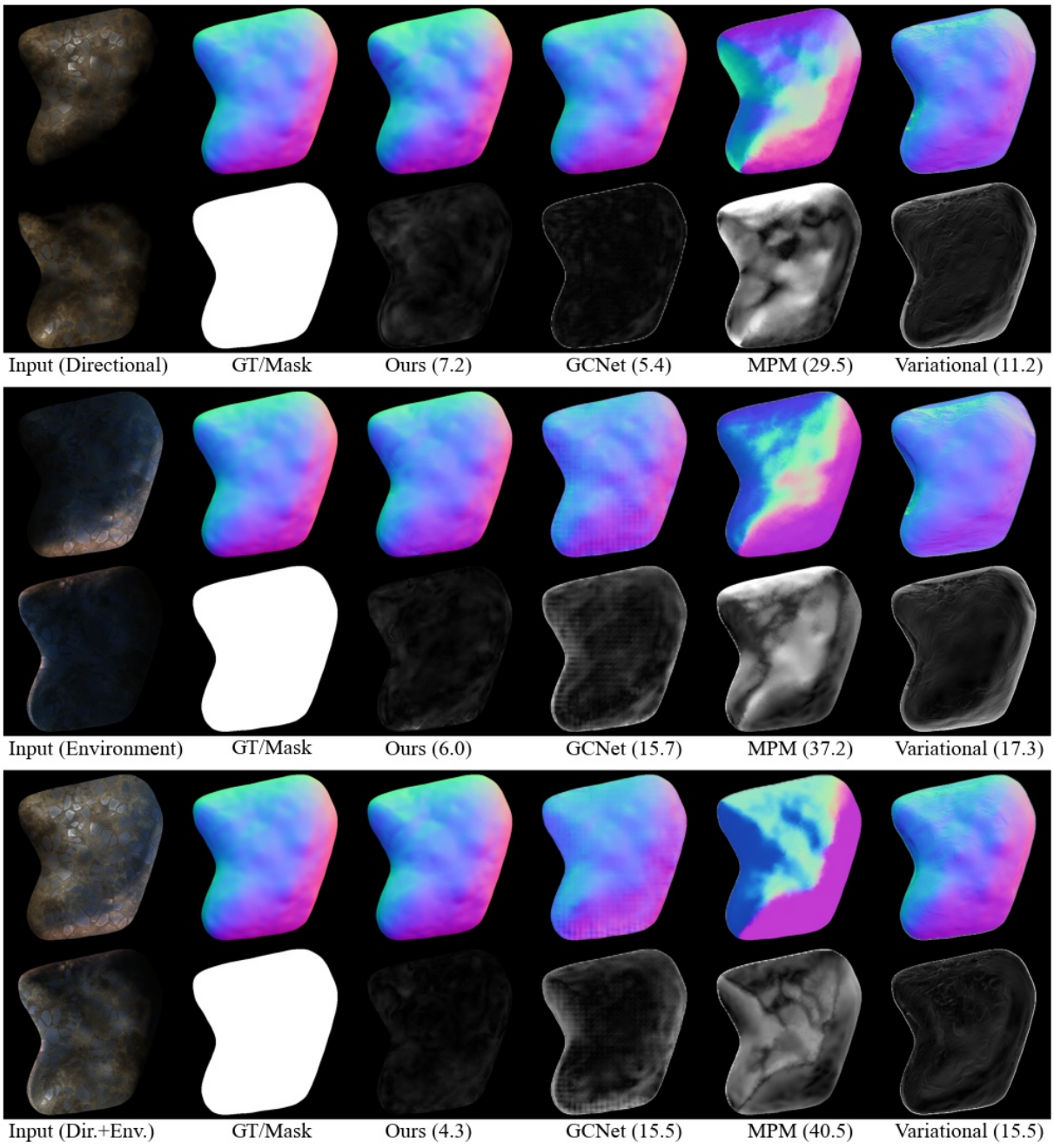

Figure 44. Results on object ID 42 (stone, ground-12 [Ground]). MAEs (in degrees) are shown next to the name of the method.

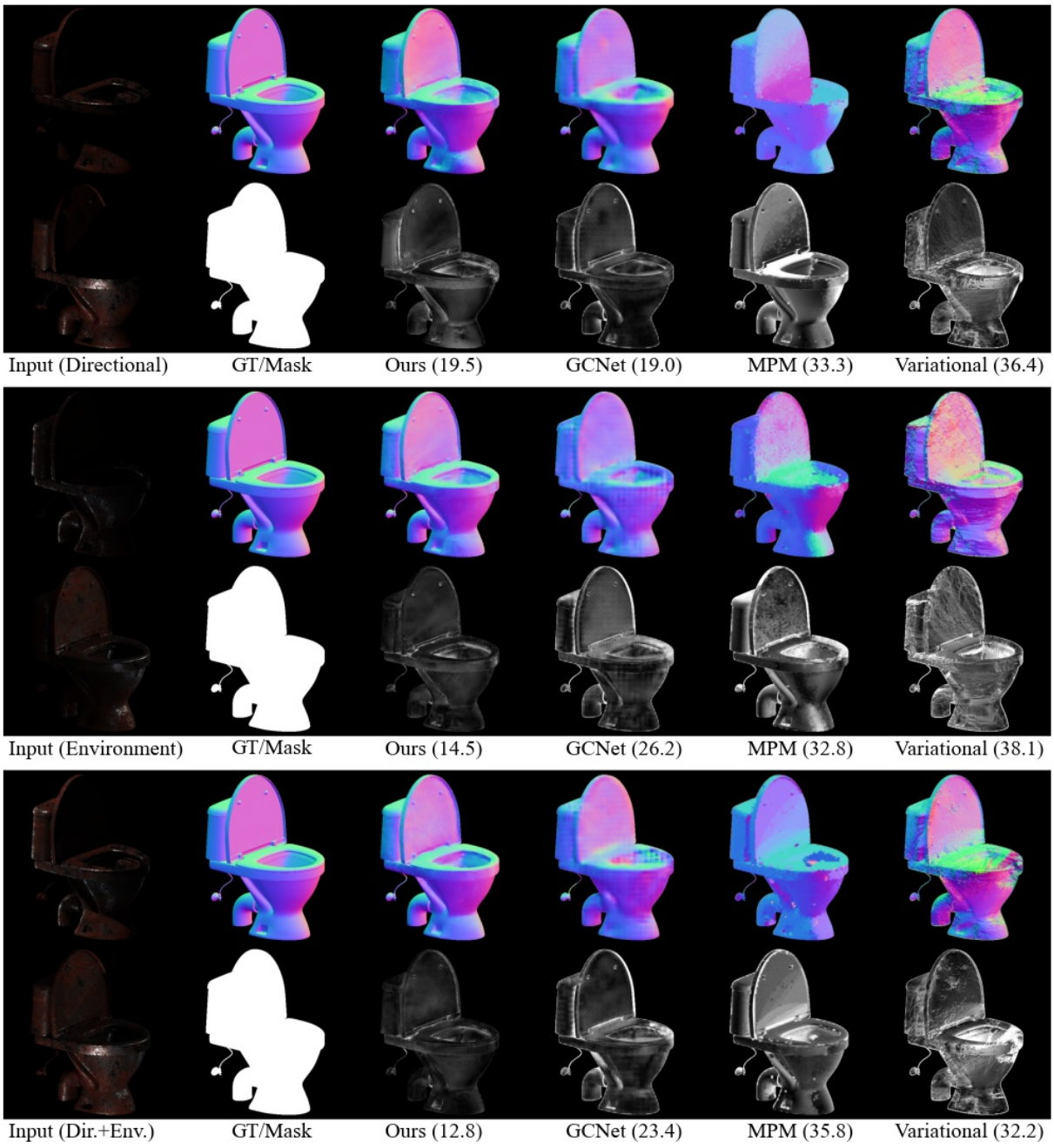

Figure 45. Results on object ID 43 (toilet, metal-frame [Metal]). MAEs (in degrees) are shown next to the name of the method.

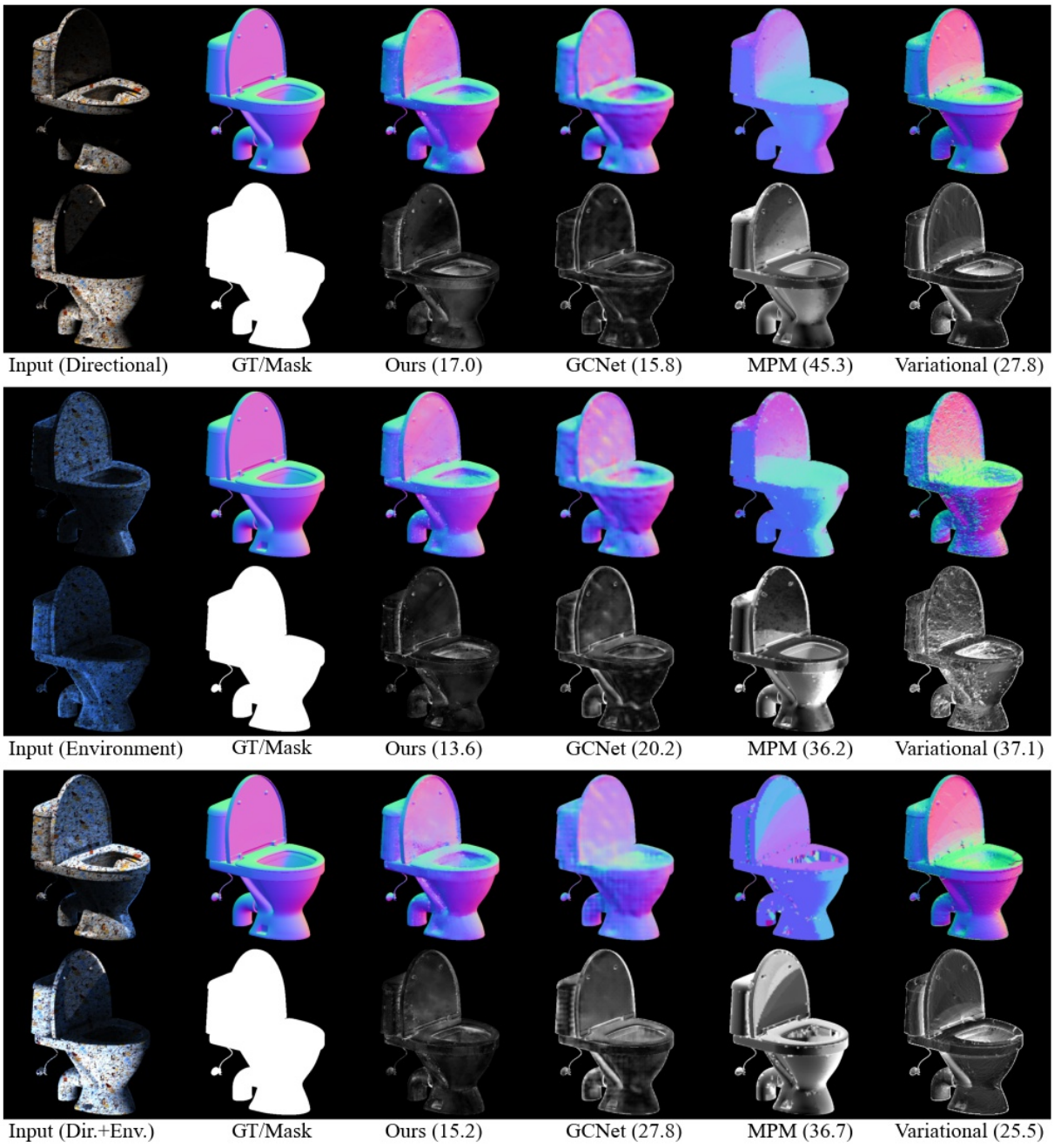

Figure 46. Results on object ID 44 (toilet, sand-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.

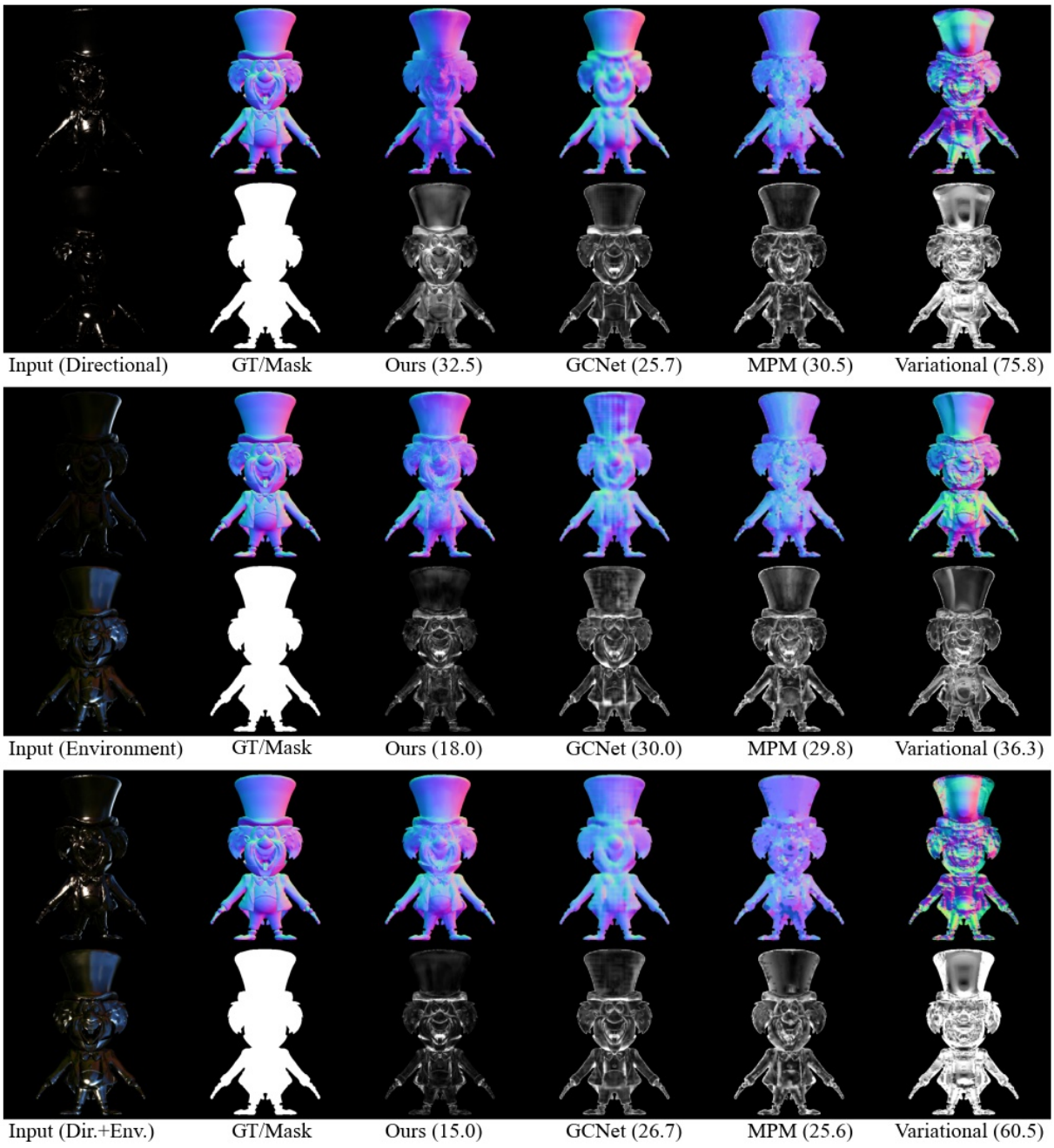

Figure 47. Results on object ID 45 (uncle, old-bronze [Metal]). MAEs (in degrees) are shown next to the name of the method.

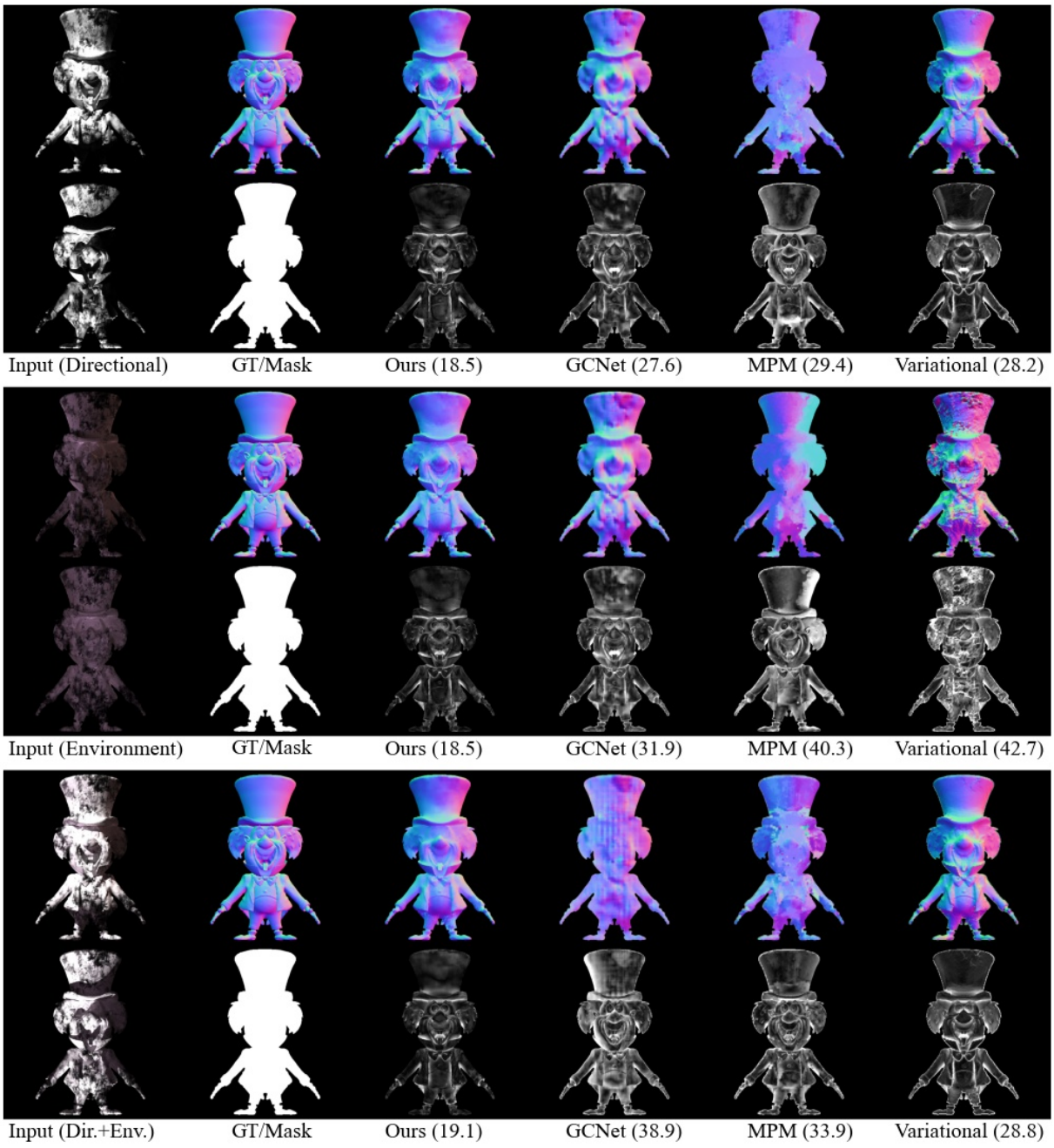

Figure 48. Results on object ID 46 (uncle, snow-ground [Ground]). MAEs (in degrees) are shown next to the name of the method.

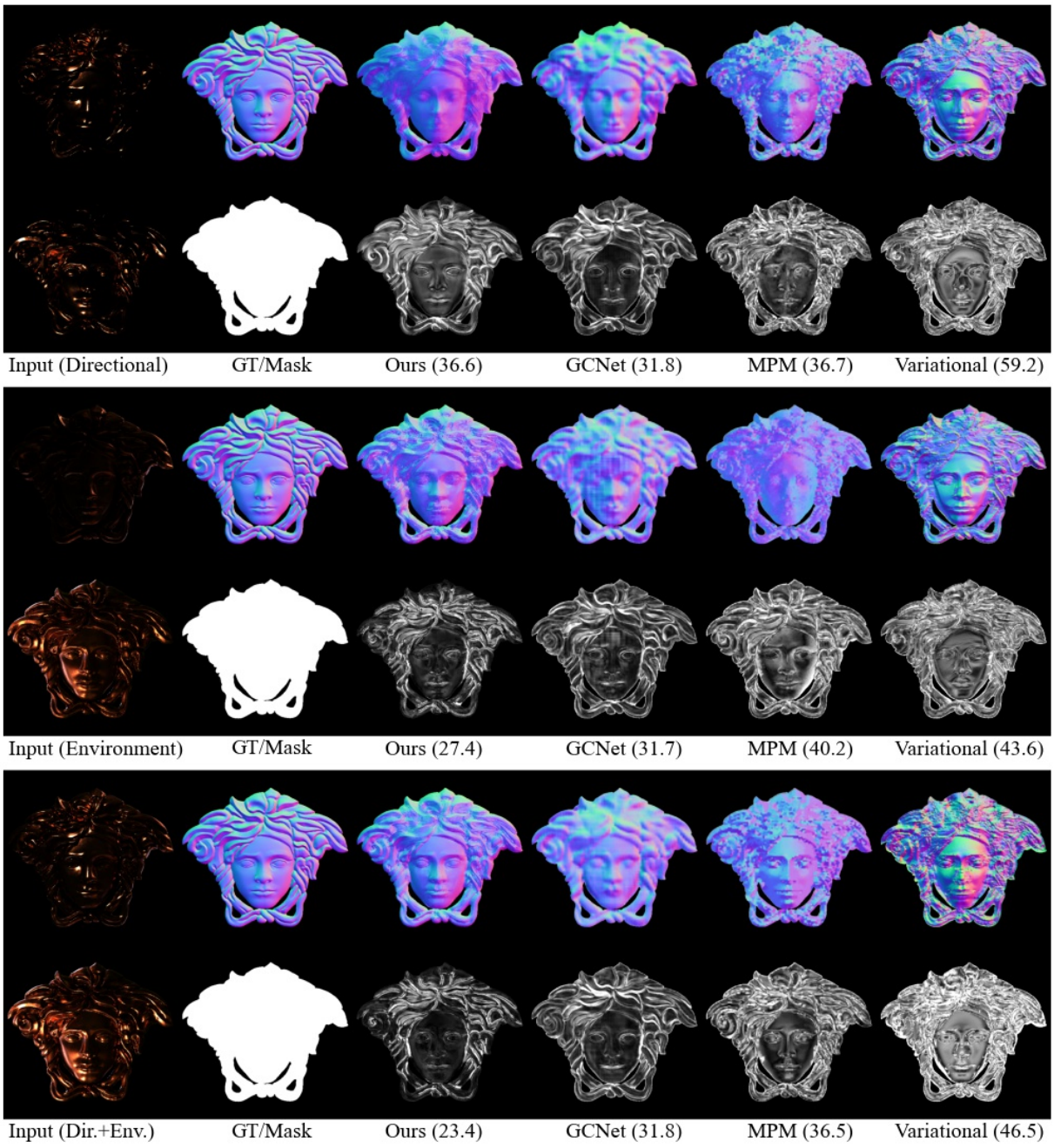

Figure 49. Results on object ID 47 (versace, old-copper [Metal]). MAEs (in degrees) are shown next to the name of the method.

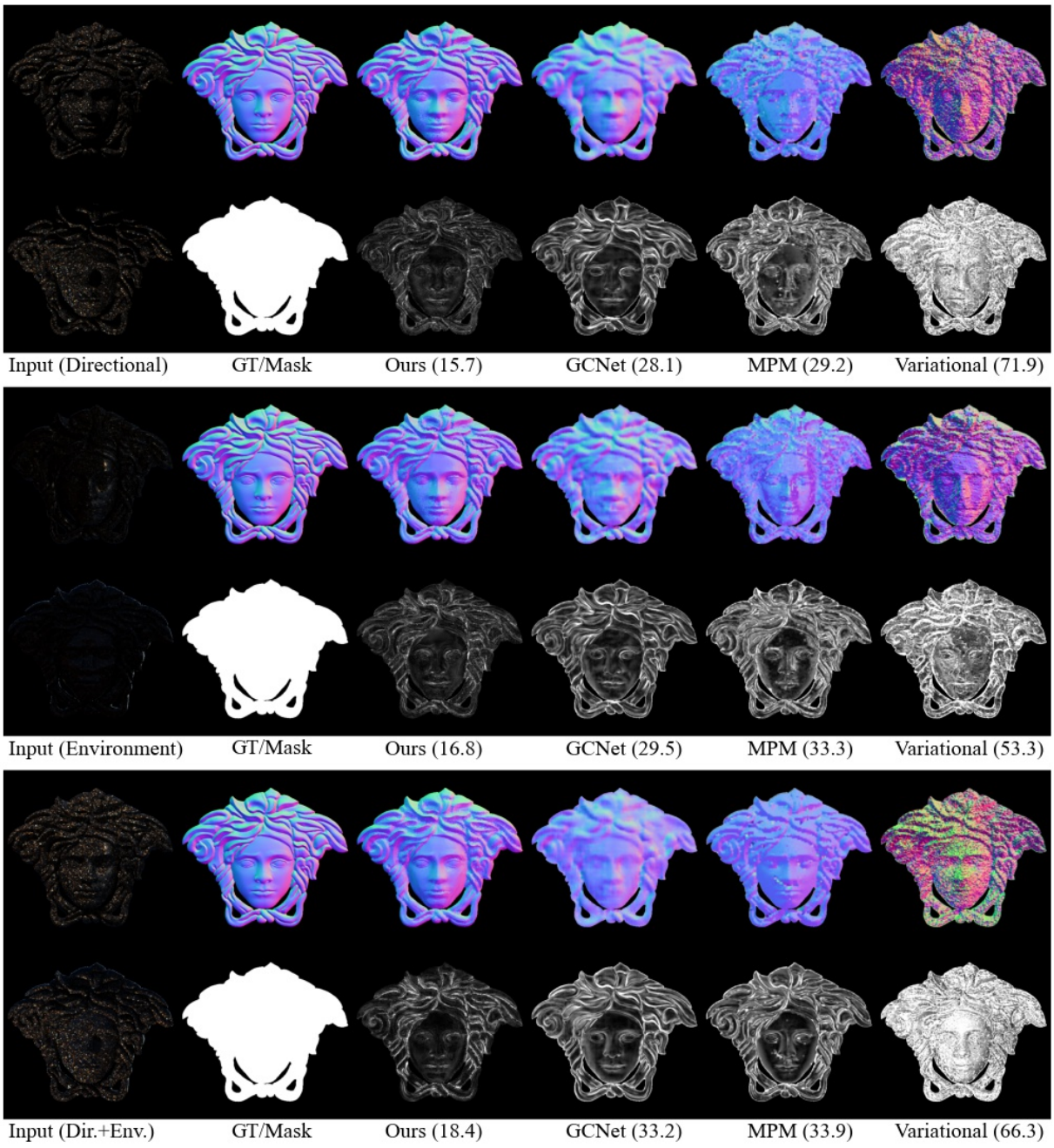

Figure 50. Results on object ID 48 (versace, pebble-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.

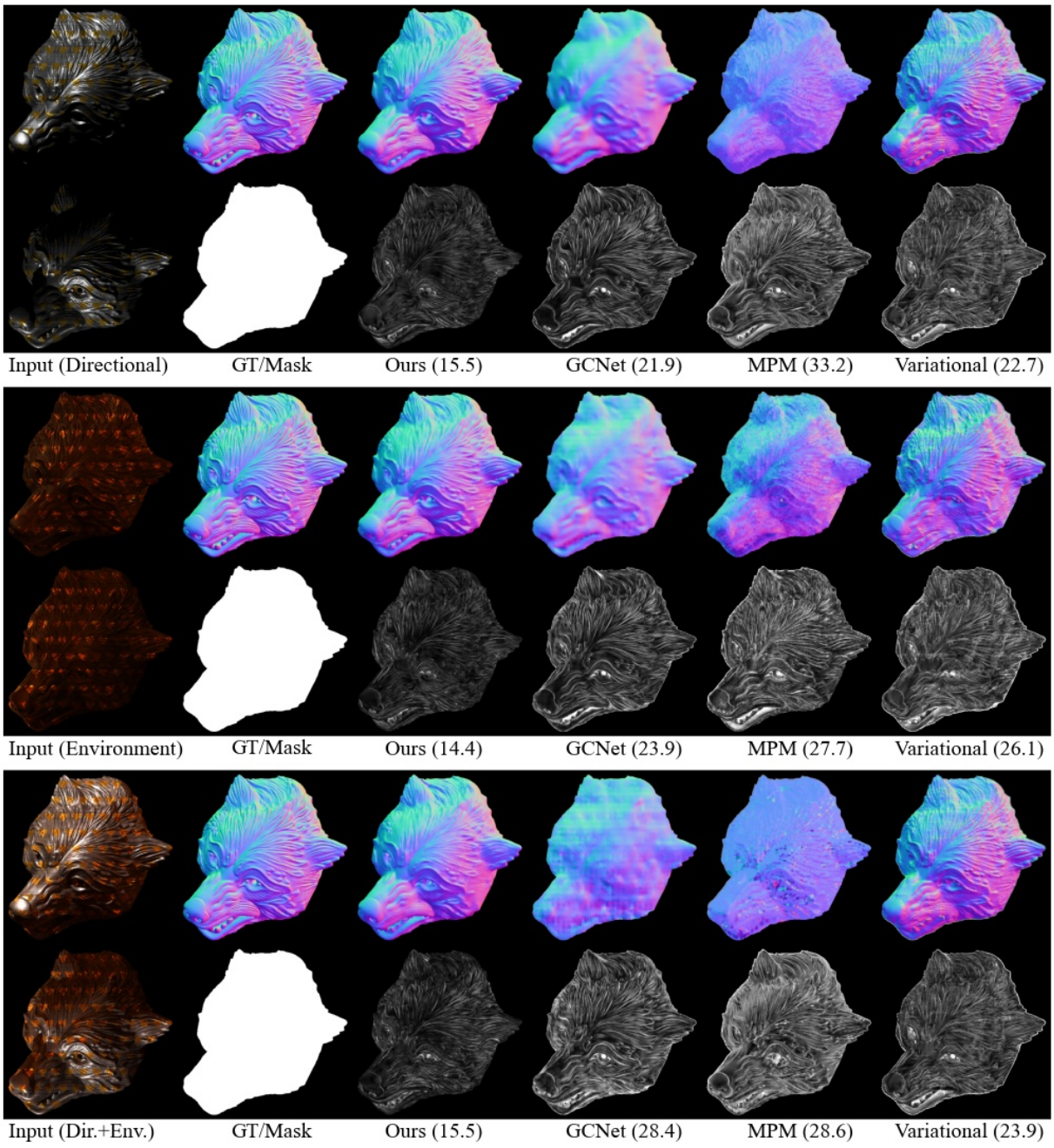

Figure 51. Results on object ID 49 (wolf, metal-plate [Metal]). MAEs (in degrees) are shown next to the name of the method.

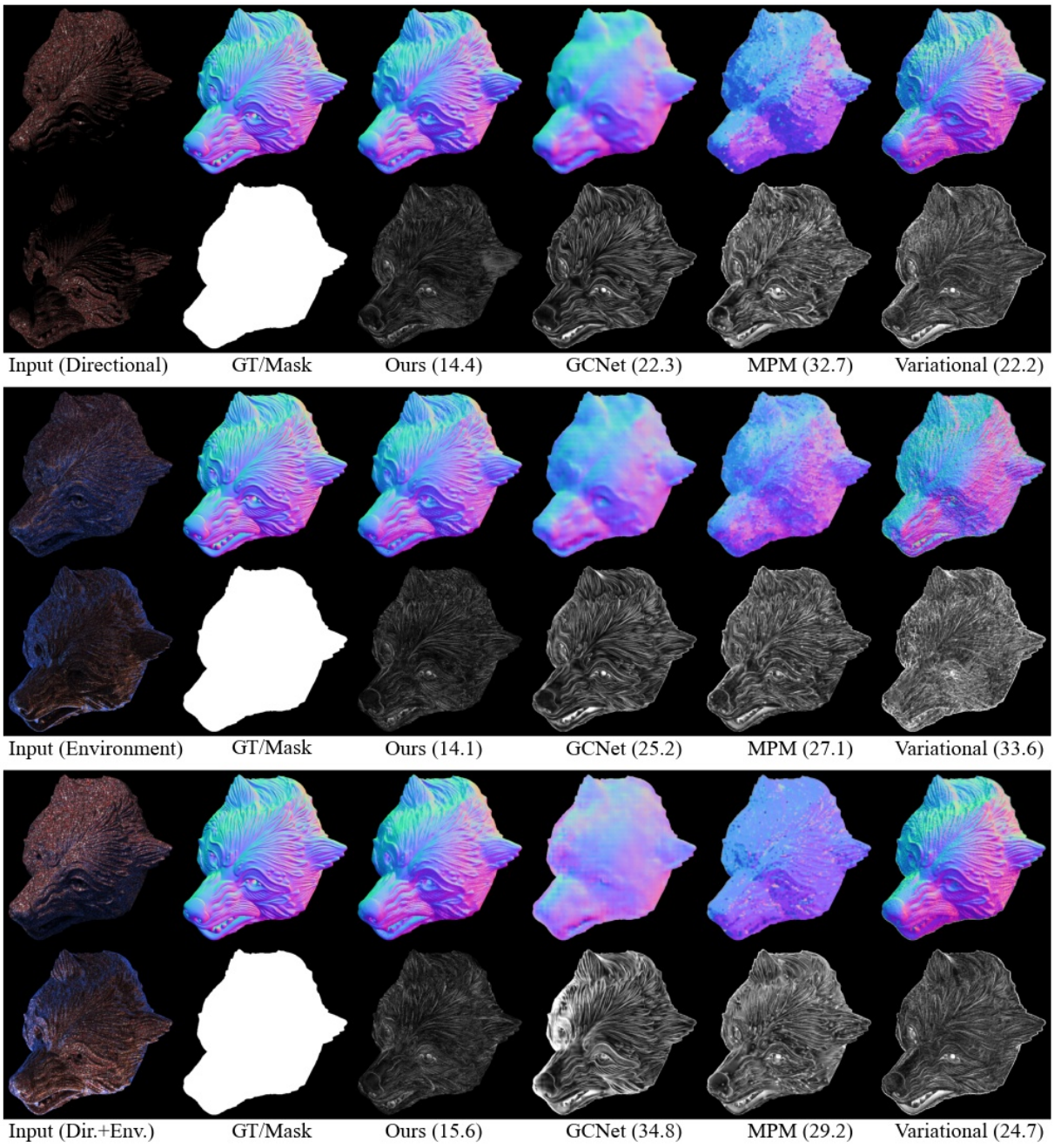

Figure 52. Results on object ID 50 (wolf, pebble-stone [Ground]). MAEs (in degrees) are shown next to the name of the method.