Locally Orderless Images for Optimization in Differentiable Rendering

Supplementary Material

1. Outline

We provide additional results and insights for our method. First, we visualize histogram contributions for an image in Figure 4. Through relevant experiments, we expand on the choice of parameters α, β and σ in Section 2. We analyze how kernel widths affect runtime in Section 3. We provide additional qualitative and quantitative comparisons along with corresponding error plots in Figures 7, 8 and 6. Implementation details are provided in Section 4. We also include reference code for a 2D task in `code.zip`, with the main implementation available in `model.py`.

2. Choosing kernel parameters

The parameter β defines the tonal scale, σ defines the image resolution, and α defines the spatial extent over which histograms are integrated. Since β implicitly defines the bin-width as $1/\beta$, it represents the effective tonal resolution. Intuitively, the number of bins limits the number of modes in the distribution. Since our goal is to preserve these modes at coarser scales, we choose β appropriately. In the limit as $\beta \to 0$, the *full* intensity distribution is maintained. For all of our experiments, we find that the maximum number of local histogram modes is 6, and choosing $\beta \geq 1/32$ (*i.e.* 32 bins) works reasonably well. The parameter α is critical for long-range motion of image features, as it defines the extent of gradient support. Intuitively, low α values can be used for more local alignment in image space, while high values enable longer-range alignment. For complex problems involving both local and global alignment, we use multiple scales.

Ablating σ and α Since β does not extend gradient support, using it alone without blurring histogram contributions via α is functionally equivalent to matching images at a stationary resolution. Figure 2 illustrates this with an optimization problem where using β in isolation leads to poor local minima.

Texture Recovery We recover the texture of a planar surface starting from a random initialization. Contrary to what one might expect, high β values do not degrade the quality of recovery on this task. The low tonal resolution resulting from high β values does not affect the resolution of the underlying texture. As shown in Figure **??**, the choice of β has a marginal effect on recovery quality. We demonstrate the advantages of using α space compared to σ space on this task in Figure 1.

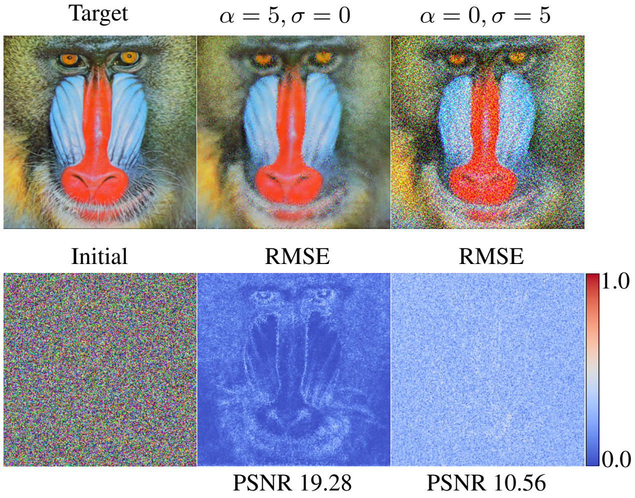

Figure 1. **Texture Recovery.** Starting from a random initialization, we recover a reference texture using $\beta = 1/8$. We compare our histogram-space blurring approach ($\alpha = 5, \sigma = 0$) with standard σ-space blurring ($\alpha = 0, \sigma = 5$). The σ-space matching yields suboptimal results since it only measures errors in the means of the distributions.

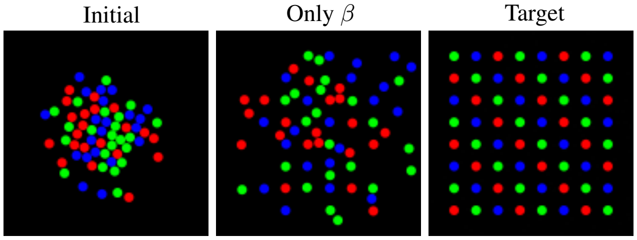

Figure 2. **Ablation on α, σ.** The parameter β alone does not influence the support of RGB gradients. When used in isolation without the α and σ spaces, the optimization converges to poor local minima. We demonstrate this by attempting to recover disk positions from random initializations using only β scale space.

3. Runtime

We measure the effect of the LOI parameters on wall-time for feedforward operations. Please refer to Figure 3.

4. Implementation Details

Differentiable Vectorization For this set of experiments, we use 128×128 images and recover the positions of n disks, making it an $n \times 2$ dimensional optimization problem. The disk positions are initialized by sampling from a normal distribution $\mathcal{N}(0, 1)$. We compute image gradients using diffvg [3] and implement our method in PyTorch. We use

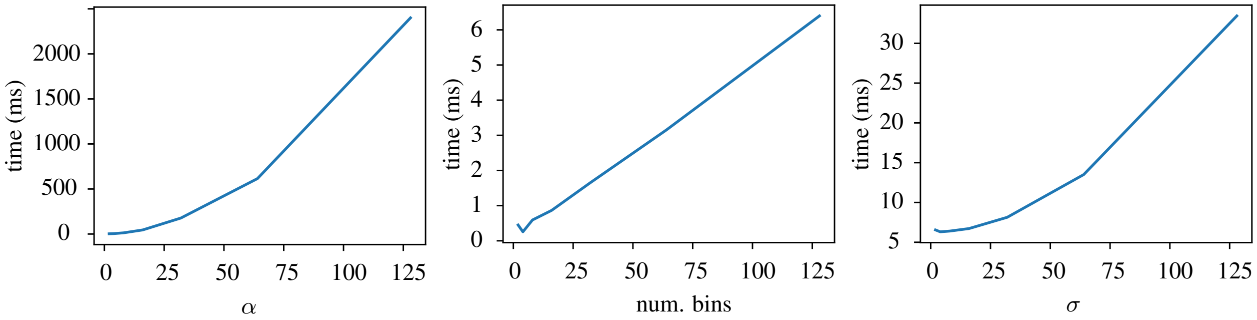

Figure 3. **Runtime Analysis.** We measure the effects of kernel-width parameters (α, β, and σ) on the total runtime of computing LOIs. The runtime increases linearly with kernel width, with histogram-space blurring (α kernels) being the slowest operation.

a learning rate of 0.1 and the Adam optimizer with default parameters. For the kernel parameters, we set $\beta = 0.125$, $\alpha = [1, 5, 15]$ and $\sigma = [1, 5, 15, 45]$. The parameter β also determines the histogram bin width, resulting in 8 bins for these experiments. Each disk has a width of 4 pixels. To prevent disk overlap, we resolve collisions at each optimization step. We provide error plots and visual comparisons for randomly selected runs in Figure 8.

Differentiable Path Tracing For path tracing experiments, we render images at 256×192 resolution using Mitsuba 3 [2] with 64 samples per pixel for computing gradients. We use the `prb_reparam` integrator [4] with a maximum path length of 4. For optimization, we use Adam with default parameters and a learning rate of 0.01 for all parameters, except for light position parameters, for which we use a learning rate of 1. We provide additional visual comparisons and convergence plots beyond those in the main manuscript in Figure 7.

Differentiable Rasterization We evaluate our method on the rasterization benchmark from [6]. All images are rendered at 128×128 resolution, using the same rendering and optimization parameters as [6]. Visual comparisons between our method and standard scale space matching are shown in Figure 6. We use five random initializations for each task and report average error metrics in Table 2 in the main manuscript.

References

[1] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt, and Joan M Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984. 4

[2] Wenzel Jakob, Sébastien Speierer, Nicolas Roussel, Merlin Nimier-David, Delio Vicini, Tizian Zeltner, Baptiste Nicolet, Miguel Crespo, Vincent Leroy, and Ziyi Zhang. Mitsuba 3 renderer, 2022. https://mitsuba-renderer.org. 2

[3] Tzu-Mao Li, Michal Lukáč, Gharbi Michaël, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15, 2020. 1

[4] Delio Vicini, Sébastien Speierer, and Wenzel Jakob. Path replay backpropagation: Differentiating light paths using constant memory and linear time. *Transactions on Graphics (Proceedings of SIGGRAPH)*, 40(4), 2021. 2

[5] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, pages 1398–1402. Ieee, 2003. 4

[6] Jiankai Xing, Fujun Luan, Ling-Qi Yan, Xuejun Hu, Houde Qian, and Kun Xu. Differentiable Rendering Using RGBXY Derivatives and Optimal Transport. *ACM Transactions on Graphics*, 41(6):1–13, 2022. 2, 3

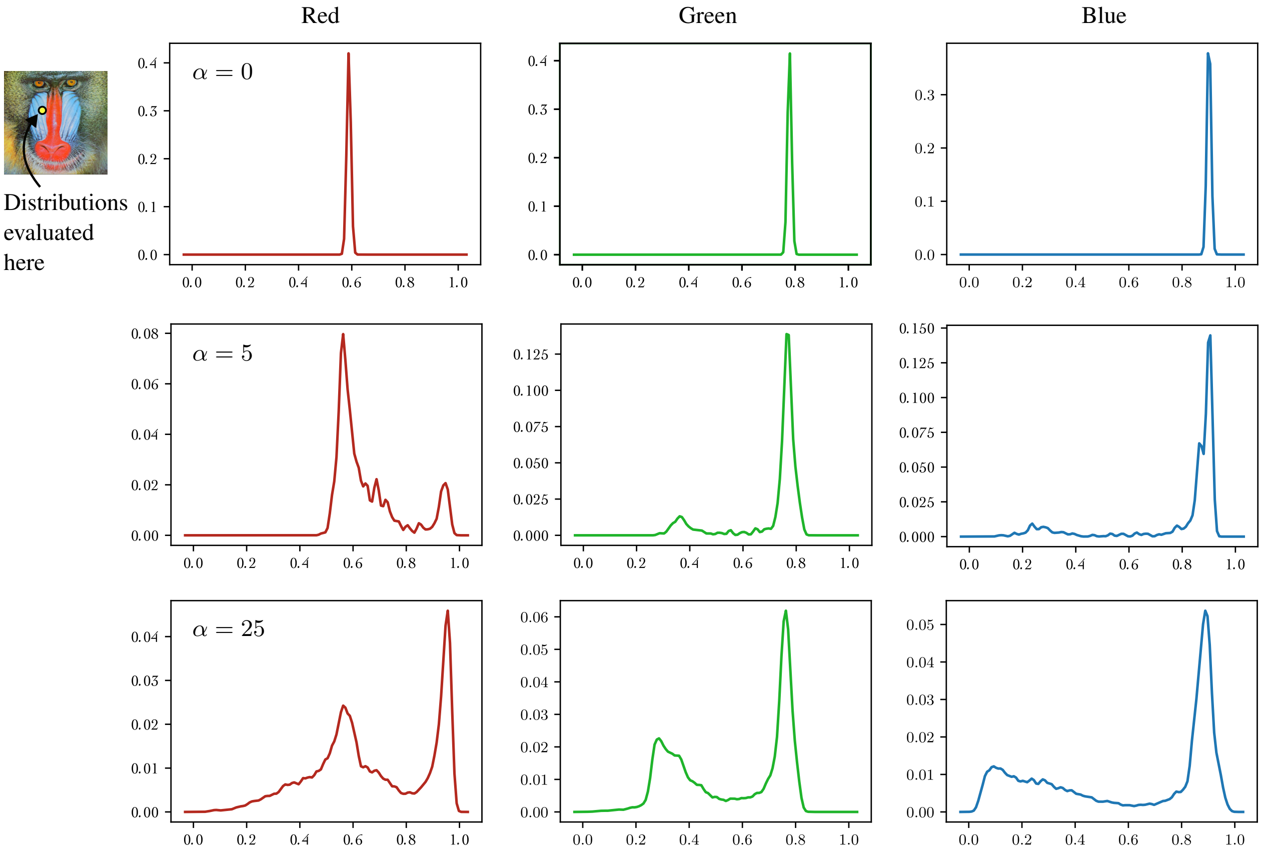

Figure 4. **Histogram visualization.** We visualize the histogram contributions across the three color channels at different α kernel widths. While standard Gaussian Pyramids only preserve the mean of each distribution, our method preserves their modes.

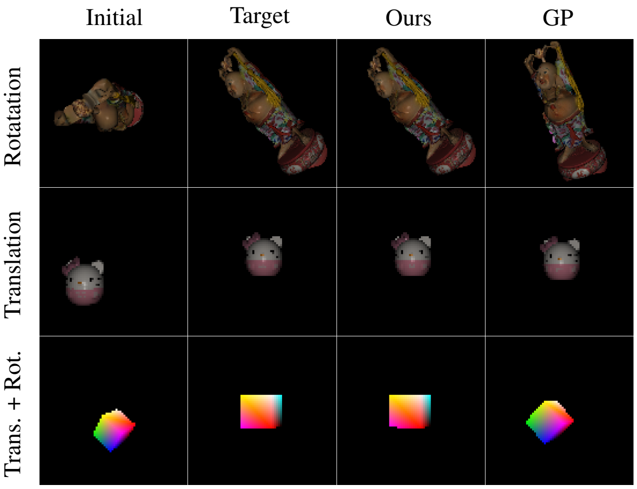

Figure 6. **Visual comparisons on rasterization benchmark.** We show visual comparisons on three tasks from the rasterization benchmark in [6].

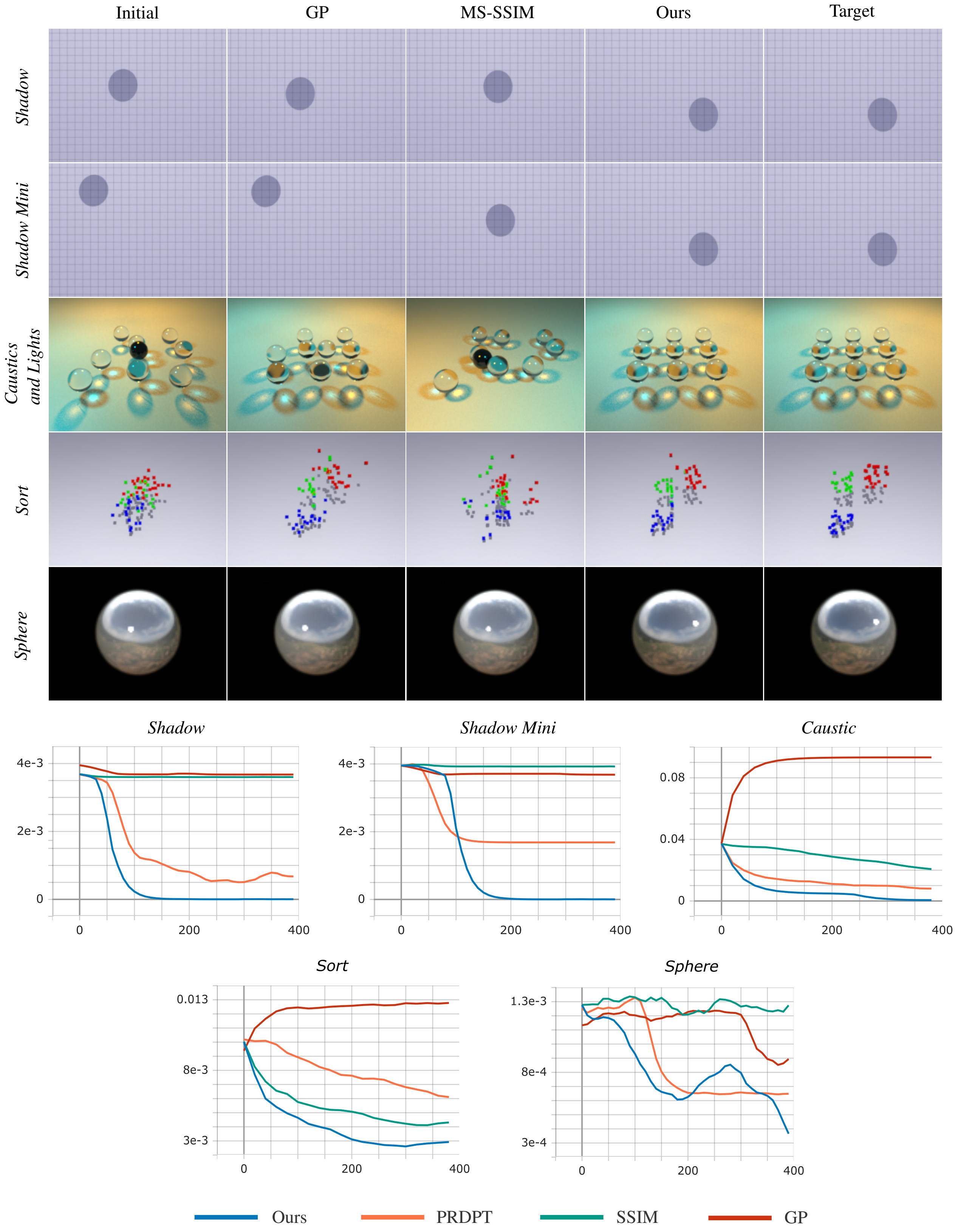

Figure 7. **More comparisons on Differentiable Path Tracing.** We show additional comparisons with GP [1] and MS-SSIM [5] along with Error plots (Image MSE *vs*. Iterations) for each scene in Figure 7 in the main manuscript.

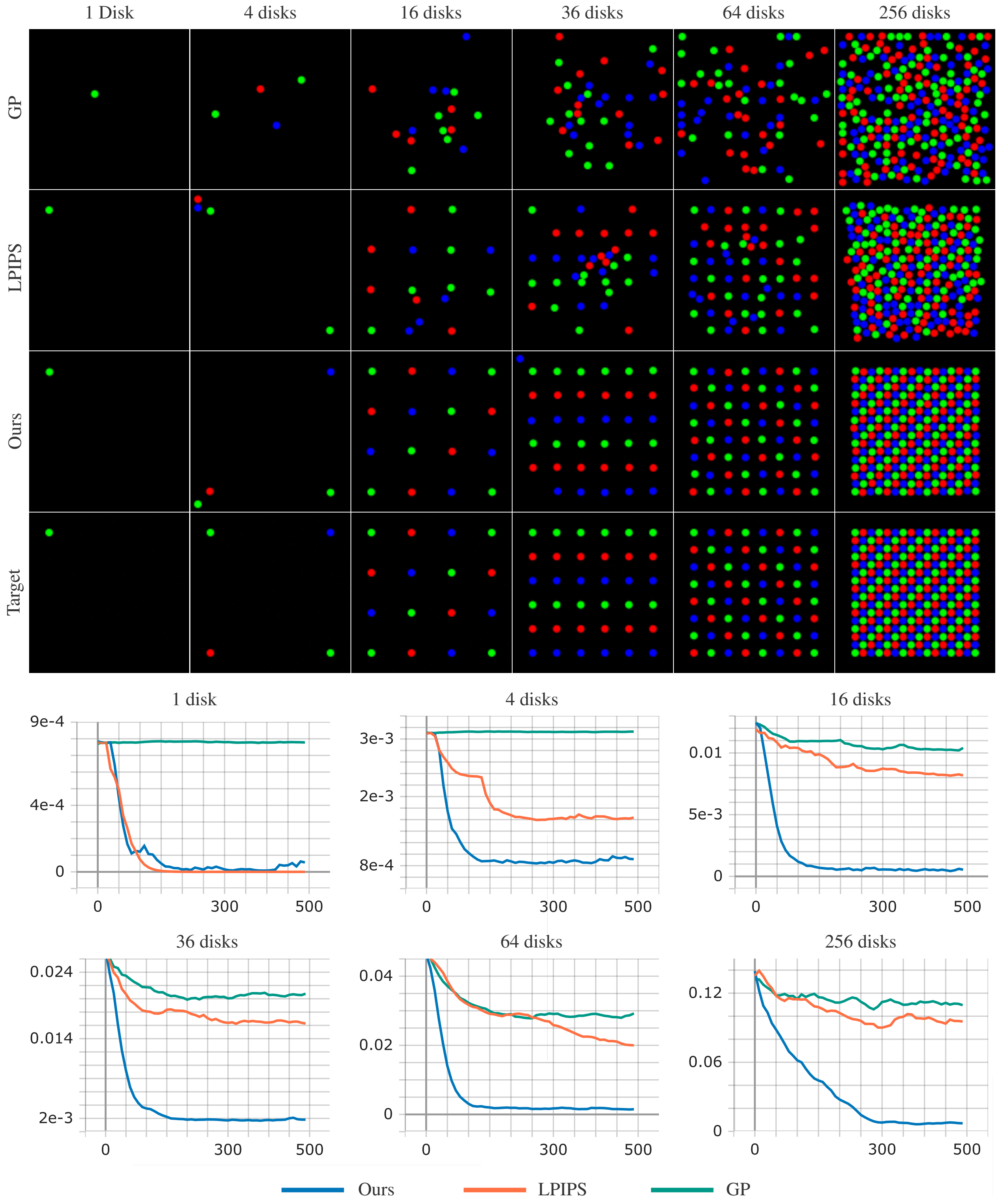

Figure 8. **2D Comparisons.** We show convergence plots (Image MSE *vs*. Iterations) and optimized results for the 2D benchmark task of optimizing disk positions (Table 1 and Figure 6 in the main manuscript).