Reflection Separation from a Single Image via Joint Latent Diffusion

Supplementary Material

1. Overview

In this supplementary material, we provide additional results to complement the main manuscript. We begin with supplementary details on our method, including the algorithmic procedure of our inference framework in Sec. 2 and detailed formulation and discussion of the loss functions used during training in Sec. 3. Next, we present quantitative results for reflection and transmission layer evaluations across all datasets, using official pre-trained weights in Tab. 1 and Tab. 2. We then conduct extensive ablation studies across all datasets in Sec. 4. Additionally, we present an insightful experiment further illustrating the robustness and versatility of latent optimization in Sec. 5. Finally, we showcase visual examples demonstrating our method's effectiveness in various scenarios in Sec. 6.

2. Inference

Algorithm 1 details the inference procedure, including disjoint sampling and latent optimization.

Algorithm 1 Reflection Disjoint Sampling Strategy

Require: Noise prediction network ϵ_θ, Encoder $\mathcal{E}$, Composition model $\mathcal{C}$

$z_i^{\mathcal{T}}, z_i^{\mathcal{R}} \sim \mathcal{N}(0,1), \quad z^{\mathcal{I}} \leftarrow \mathcal{E}(\mathcal{I})$
for $t = N, \ldots, 1$ **do**
  $\epsilon^{\mathcal{T}} \leftarrow \epsilon_\theta(z^{\mathcal{I}}, z_i^{\mathcal{T}}, t, c^{\mathcal{T}}), \quad \epsilon^{\mathcal{R}} \leftarrow \epsilon_\theta(z^{\mathcal{I}}, z_i^{\mathcal{R}}, t, c^{\mathcal{R}})$
  if $t \mod 5$ **then**
    for $k = 1$ **to** 4 **do**
    $\hat{z}_0^{\mathcal{T}} \leftarrow \frac{1}{\sqrt{\alpha_i}}\left(z_i^{\mathcal{T}} + (1-\bar{\alpha}_i)\epsilon^{\mathcal{T}}\right)$
    $\hat{z}_0^{\mathcal{R}} \leftarrow \frac{1}{\sqrt{\alpha_i}}\left(z_i^{\mathcal{R}} + (1-\bar{\alpha}_i)\epsilon^{\mathcal{R}}\right)$
    $L \leftarrow \| z^{\mathcal{I}} - \mathcal{C}(\hat{z}_0^{\mathcal{T}}, \hat{z}_0^{\mathcal{R}}) \|_2^2$
    $z_i^{\mathcal{T}} \leftarrow z_i^{\mathcal{T}} - \gamma_i \| z_i^{\mathcal{T}} \| \nabla_{z_i^{\mathcal{T}}} L, \quad z_i^{\mathcal{R}} \leftarrow z_i^{\mathcal{R}} - \gamma_i \| z_i^{\mathcal{R}} \| \nabla_{z_i^{\mathcal{R}}} L$
    end for
  end if
  $\hat{\epsilon}^{\mathcal{T}} \leftarrow \epsilon^{\mathcal{T}} + w(\epsilon^{\mathcal{T}} - \epsilon^{\mathcal{R}}), \quad \hat{\epsilon}^{\mathcal{R}} \leftarrow \epsilon^{\mathcal{R}} + w(\epsilon^{\mathcal{R}} - \epsilon^{\mathcal{T}})$
  $z_{t-1}^{\mathcal{T}} \leftarrow \frac{1}{\sqrt{\alpha_i}}\left(z_i^{\mathcal{T}} - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}}\hat{\epsilon}^{\mathcal{T}}\right)$
  $z_{t-1}^{\mathcal{R}} \leftarrow \frac{1}{\sqrt{\alpha_i}}\left(z_i^{\mathcal{R}} - \frac{1-\alpha_i}{\sqrt{1-\bar{\alpha}_i}}\hat{\epsilon}^{\mathcal{R}}\right)$
end for

3. Loss Functions

This section outlines the diffusion loss functions in our two-stage training framework. In the first stage, a combined loss for reflection and transmission layers serves as the optimization objective for the diffusion U-Net.

The diffusion model learns to invert the noising process using a noise-prediction network $\epsilon\theta(\cdot)$ by minimizing the following objective:

$$\mathbb{E}_{\epsilon\sim\mathcal{N}(0,1),\, t,\, \mathcal{I},\, \mathcal{T},\, \mathcal{R}}\Big[\big\|\epsilon_t^{\mathcal{T}} - \epsilon_\theta\big(z_t^{\mathcal{T}},\, z^{\mathcal{I}},\, t,\, c^{\mathcal{T}}\big)\big\|_2^2 + \big\|\epsilon_t^{\mathcal{R}} - \epsilon_\theta\big(z_t^{\mathcal{R}},\, z^{\mathcal{I}},\, t,\, c^{\mathcal{R}}\big)\big\|_2^2\Big],$$

where c denotes the language prompt specifying the target layer, and $\epsilon_t^{\mathcal{T}}$ and $\epsilon_t^{\mathcal{R}}$ are sampled independently. For real training data, where ground truth for the reflection layer is unavailable, the loss is computed only for the transmission layer.

In the second stage, we optimize the Fidelity-Guided Feature Modulation (FGFM) module using a combination of L2 and LPIPS losses, formulated as:

$$\mathcal{L} = \beta_1 \left\|\hat{\mathcal{T}} - \mathcal{T}\right\|_2^2 + \beta_2 \sum_i w_i \left\|\phi_i(\hat{\mathcal{T}}) - \phi_i(\mathcal{T})\right\|_2^2, \quad (1)$$

where $\mathcal{T}$ denotes the decoded transmission, and $\hat{\mathcal{T}}$ represents the ground truth transmission. We set $\beta_1 = 1$ and $\beta_2 = 0.1$ based on empirical results.

4. Ablation Study

In the paper, we evaluate the contribution of each module: Cross-layer Self-Attention (C), Latent Optimization (O), and Disjoint Sampling (D) on the SIR^2 dataset. This section extends the ablation studies to all other datasets, including Real20 (Tab. 3) and the Nature (Tab. 4) dataset. The results consistently demonstrate that each component progressively enhances performance, with the full model (C+O+D) achieving the best quantitative metrics.

5. An experiment with Latent Optimization

We provide an illustrative example to further validate the effectiveness of latent optimization guided by the proposed composition model. Specifically, we apply a text-to-image model that has not been fine-tuned for layer separation, integrating it with our proposed latent optimization. As shown in Fig. 4, even with an empty prompt, the model achieves reasonable results solely under the guidance of the composition model. This result underscores the potential of composition-guided separation.

6. Visual Results

In this section, we present additional visual results (Fig. 1, Fig. 2, Fig. 3). Compared to other methods, our approach produces cleaner and more accurate transmission

Table 1. **Quantitative comparison of the transmission layer across different real-world datasets (Real20 [6], Nature [3], SIR^2 [5]) using official pre-trained weights.**

Dataset (size)	Metric	Method							
		YTMT	RobustSIRR	DSRNet	RRW	DSIT	RDNet	ControlNet	Ours
Real20 (20)	PSNR↑	23.03	22.75	23.34	21.52	24.57	25.03	18.68	25.32
	SSIM↑	0.800	0.785	0.788	0.764	0.813	0.827	0.645	0.850
	LPIPS↓	0.176	0.205	0.178	0.222	0.157	0.142	0.312	0.107
	DISTS↓	0.124	0.139	0.121	0.139	0.111	0.104	0.216	0.089
Nature (20)	PSNR↑	20.77	20.94	24.86	25.78	26.25	25.82	19.92	26.71
	SSIM↑	0.772	0.759	0.818	0.829	0.829	0.828	0.721	0.837
	LPIPS↓	0.185	0.245	0.144	0.122	0.156	0.131	0.242	0.080
	DISTS↓	0.117	0.144	0.093	0.086	0.096	0.084	0.168	0.064
SIR^2 (454)	PSNR↑	23.73	22.31	25.58	25.33	26.39	26.40	20.65	25.35
	SSIM↑	0.889	0.861	0.911	0.900	0.916	0.915	0.812	0.911
	LPIPS↓	0.124	0.188	0.103	0.127	0.105	0.095	0.174	0.075
	DISTS↓	0.087	0.120	0.077	0.087	0.076	0.071	0.138	0.065

Table 2. **Quantitative comparison of the reflection layer on SIR^2 [5] using official pre-trained weights.** Best and second best results are highlighted.

Metric	YTMT	DSRNet	DSIT	RDNet	Ours
↑ PSNR	16.10	17.65	18.34	17.43	21.14
↑ SSIM	0.114	0.344	0.461	0.302	0.681
↓ LPIPS	0.652	0.565	0.510	0.545	0.373
↓ DISTS	0.654	0.446	0.367	0.353	0.275

Table 3. **Ablation study on the Real20 dataset.**

C	O	D	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
			24.26	0.766	0.193	0.144
✓			24.23	0.771	0.178	0.139
✓	✓		24.36	0.778	0.174	0.135
✓	✓	✓	**25.32**	**0.850**	**0.107**	**0.089**

Table 4. **Ablation study on the Nature dataset.**

C	O	D	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
			25.61	0.769	0.181	0.129
✓			25.28	0.792	0.150	0.109
✓	✓		25.64	0.799	0.150	0.109
✓	✓	✓	**26.71**	**0.837**	**0.080**	**0.064**

(background) layers, effectively removing reflection artifacts while simultaneously recovering clearer and more meaningful reflection details. These results highlight significant improvements in challenging real-world scenarios.

7. Implementation Details.

Our training datasets consist of both synthetic and real-world images. We use the dataset from Setting 2 in [1] and apply the same data pre-processing methods. The training process is divided into two stages. In the first stage, we fully fine-tune the diffusion model initialized from Stable Diffusion v2 [4], using the Adam optimizer [2]. In the second stage, we train FGFM on output latents from the fine-tuned diffusion U-Net. To accelerate training, we limit this stage to 10 diffusion steps and apply FGFM modulation exclusively to the transmission layer since reflection layers typically contain limited high-frequency input information. For training the composition network, we utilize only synthetic data. The learning rate for all training is set to 3×10^{-5}, and the models are trained with an effective batch size of 32 on a single NVIDIA GeForce A6000 GPU. During inference, we generate results using a 50-step diffusion process.

To match the ground truth resolution, we downsample our results using area interpolation. In the official implementations of the compared methods, the ground-truth images are adjusted to the corresponding output size. To ensure a reasonable evaluation and a fair comparison, we upsample their outputs to the ground-truth resolution using bicubic interpolation when sizes do not match.

8. Importance of pre-trained weights.

Table 5. **Effect of pre-trained weights on transmission and reflection.**

	PSNR ↑	SSIM ↑	LPIPS ↓	DISTS ↓
Train from scratch (T)	21.31	0.763	0.191	0.171
w/ pretrained weights (T)	**24.67**	**0.858**	**0.120**	**0.094**
Train from scratch (R)	19.34	0.595	0.543	0.365
w/ pretrained weights (R)	**20.87**	**0.659**	**0.381**	**0.285**

We conduct an experiment to investigate the importance of pre-trained weights. For ease of comparison, we report results only on the SIR^2 dataset, without latent optimization or disjoint sampling. As shown in table 5, not using pre-trained weights results in significantly degraded perfor-

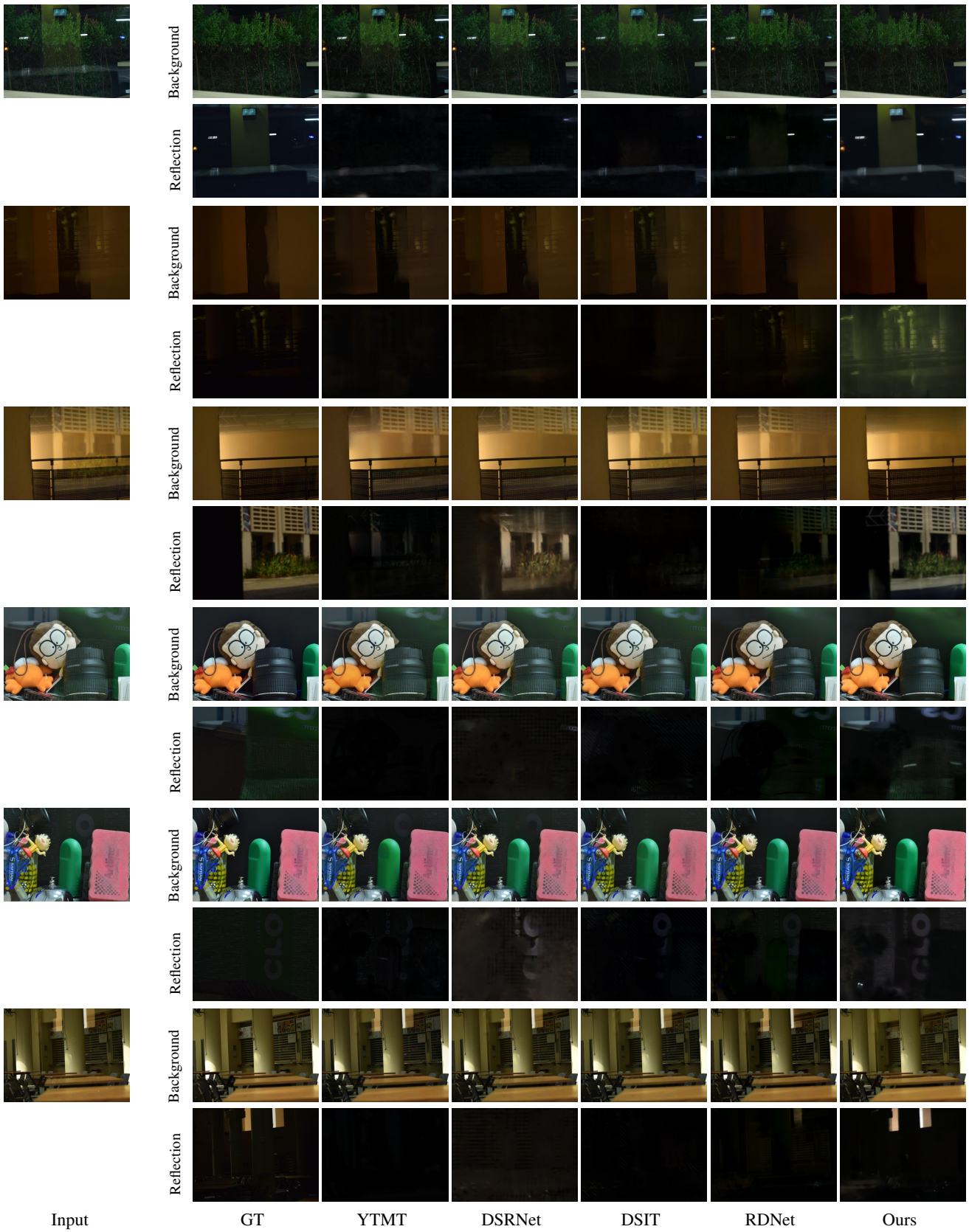


Figure 1. **Qualitative comparison of our method with state-of-the-art approaches on background-reflection separation using real-world images.**

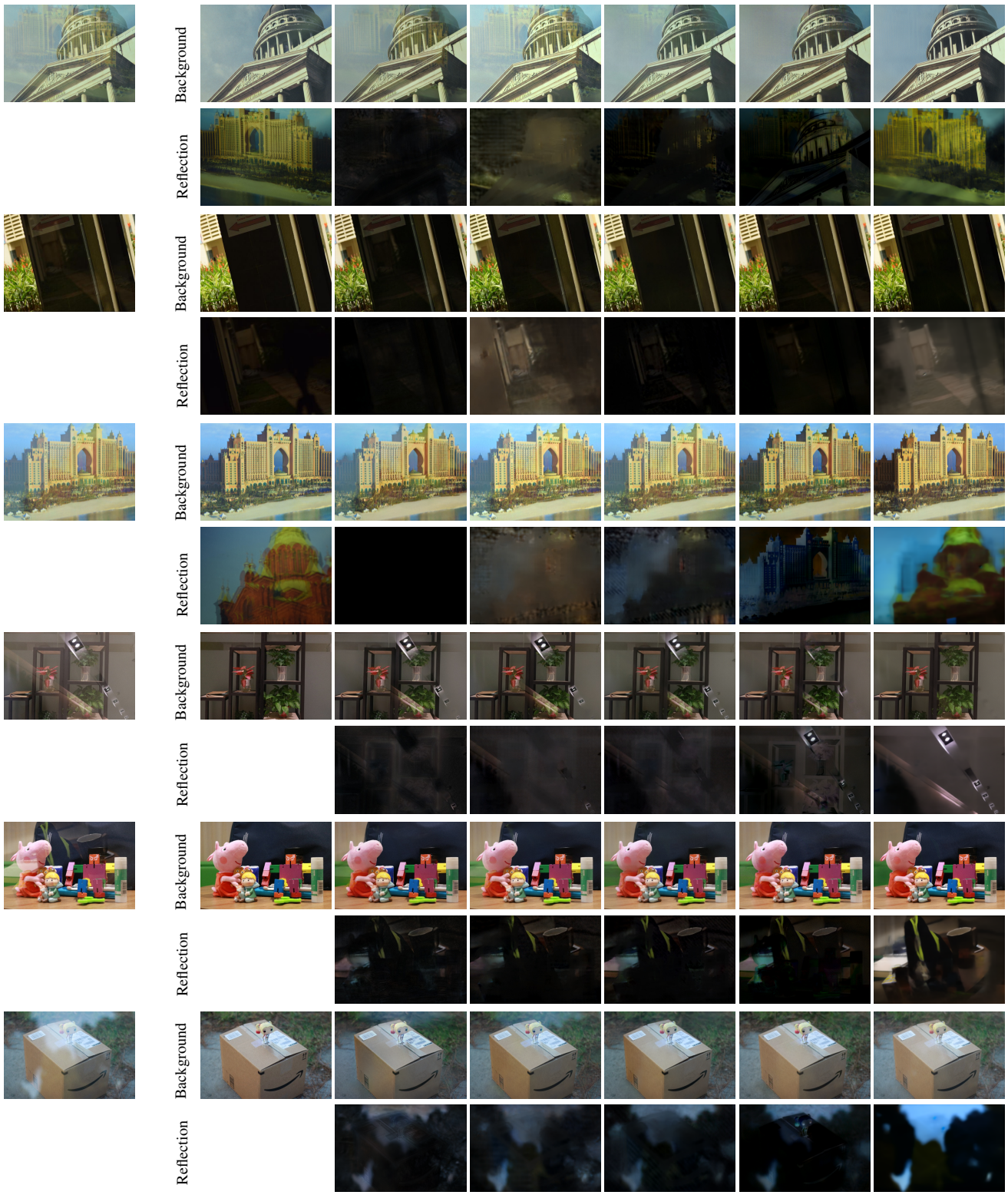


Figure 2. **Qualitative comparison of our method with state-of-the-art approaches on background-reflection separation using real-world images.**

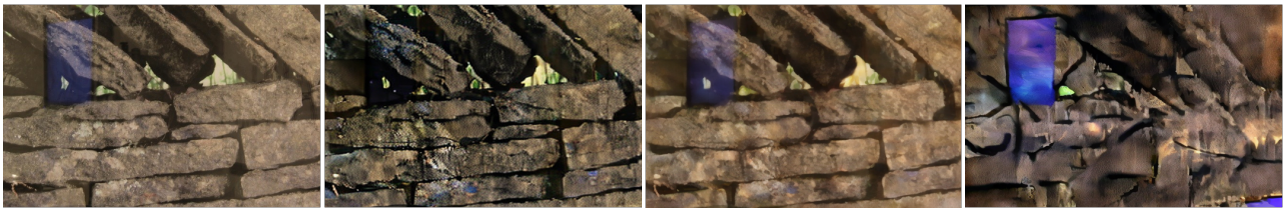
Figure 3. **Qualitative comparison of our method with state-of-the-art approaches on background-reflection separation using real-world images.**

mance. We attribute this to two main reasons. First, without pre-trained weights, the model converges substantially more slowly. Second, because the reflection layer lacks supervision in real-world data and its signal is typically weak in the mixed images, the absence of a generative prior hampers the model's ability to reconstruct a high-quality reflection layer.

References

[1] Qiming Hu and Xiaojie Guo. Single image reflection separation via component synergy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13138–13147, 2023. 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*,



Input Layer 1 Layer 2 Composition

Figure 4. **Layer separation by composition.**

2014. 2

[3] Chao Li, Yixiao Yang, Kun He, Stephen Lin, and John E Hopcroft. Single image reflection removal through cascaded refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3565–3574, 2020. 2

[4] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 2

[5] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3922–3930, 2017. 2

[6] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4786–4794, 2018. 2