ReflexSplit: Single Image Reflection Separation via Layer Fusion-Separation

Supplementary Material

Overview

This supplementary material provides comprehensive details to support the main paper. The document is organized as follows:

- **Section 1: Algorithm Description** – Complete algorithmic specification of the ReflexSplit training pipeline, including dual-branch feature extraction, hierarchical decoding with CrGF and LFSB, and curriculum learning strategy.
- **Section 2: Loss Functions Details** – Detailed formulation of all loss components: reconstruction losses, perceptual loss, exclusion loss, reconstruction consistency, and color consistency loss with weight configurations.
- **Section 3: Feature Separation Analysis with DSIT** – t-SNE and PCA analysis comparing progressive disentanglement strategy of ReflexSplit against immediate separation approach of DSIT, demonstrating more distributed feature representations and curriculum-based layer learning.
- **Section 4: Comparison with RDNet on Real-World Scenarios** – Performance comparison with RDNet on OpenRR-1K, showing ReflexSplit achieves improvements on 63% of test images. Discussion of complementary architectural philosophies between reversible designs and explicit layer separation approaches.
- **Section 5: Additional Visual Comparisons** – Extensive qualitative results on OpenRR-1K and SIR^2 datasets, demonstrating reflection suppression, color fidelity, and detail preservation compared to state-of-the-art methods.
- **Section 6: Failure Cases and Limitations** – Analysis of challenging scenarios where ReflexSplit struggles: complex outdoor lighting, specular reflections, and mixed indoor-outdoor scenes with extreme brightness differences.
- **Section 7: Network Architecture Details** – Complete architecture specification with layer-by-layer breakdown of dual-branch encoders, feature mixing, hierarchical decoder stages, and output generation modules.
- **Section 8: More Complexity Comparison** – We present a comprehensive efficiency analysis comparing ReflexSplit with existing methods, demonstrating its practical advantage as a single-stage solution with competitive computational cost.

Algorithm 1 ReflexSplit Training Algorithm

Require: Mixed image $\mathbf{I} \in \mathbb{R}^{H\times W\times 3}$, ground truth transmission $\mathbf{T}$ and reflection $\mathbf{R}$, training epoch e, warmup epoch E_{warmup}
Ensure: Predicted transmission $\hat{\mathbf{T}}$, reflection $\hat{\mathbf{R}}$, and residual $\hat{\mathbf{RR}}$

// Stage 1: Dual-Branch Feature Extraction
Extract global semantic priors via Swin Transformer:
$\{\mathbf{P}_2, \mathbf{P}_3, \mathbf{P}_4, \mathbf{P}_5\} \leftarrow \text{GFEB}(\mathbf{I})$
Extract local texture features via MuGI-CNN:
$\{\mathbf{E}_0, \mathbf{E}_1, \mathbf{E}_2, \mathbf{E}_3, \mathbf{E}_4, \mathbf{E}_5\} \leftarrow \text{LFEB}(\mathbf{I})$
// Stage 2: Hierarchical Decoding with CrGF and LFSB
for decoder level $\ell = 5$ to 0 **do**
  if $\ell \in \{4, 3, 2\}$ **then**
    // Cross-scale Gated Fusion (CrGF)
    $\mathbf{F}_\ell^{\text{raw}} \leftarrow \mathbf{F}_{\ell+1} + \mathbf{P}_\ell + \mathbf{E}_\ell$
    Compute bidirectional gating:
      $\mathbf{F}_\ell^{\text{main}} = \mathcal{G}_1(\mathbf{F}_\ell^{\text{raw}}) \odot \mathcal{G}_2(\mathbf{F}_{\ell+1})$
      $\mathbf{F}_\ell^{\text{aux}} = \mathcal{G}_1(\mathbf{F}_{\ell+1}) \odot \mathcal{G}_2(\mathbf{F}_\ell^{\text{raw}})$
    $\mathbf{F}_\ell^{\text{fused}} = w_\ell^{(1)}\phi_1(\mathbf{F}_\ell^{\text{main}}) + w_\ell^{(2)}\phi_2(\mathbf{F}_\ell^{\text{aux}})$
  else
    // Direct Aggregation
    $\mathbf{F}_\ell^{\text{fused}} \leftarrow \mathbf{F}_{\ell+1} + \mathbf{E}_\ell$
  end if
  // Layer Fusion-Separation Block (LFSB)
  /* Early Fusion: Bidirectional Projection */
  $\mathbf{F}_\ell^{t'} = \mathbf{W}^t[\mathbf{F}_\ell^t \| \mathbf{F}_\ell^r]$
  $\mathbf{F}_\ell^{r'} = \mathbf{W}^r[\mathbf{F}_\ell^r \| \mathbf{F}_\ell^t]$
  /* Differential Dual-Dimensional Attention */
  Compute self-attention (batch-wise):
    $\mathbf{A}_{\text{SA}}^t, \mathbf{A}_{\text{SA}}^r \leftarrow \text{SA}(\text{concat}([\mathbf{F}_\ell^{t'}, \mathbf{F}_\ell^{r'}], \text{dim} = 0))$
  Compute cross-attention (sequence-wise):
    $\mathbf{A}_{\text{CA}}^t, \mathbf{A}_{\text{CA}}^r \leftarrow \text{CA}(\text{concat}([\mathbf{F}_\ell^{t'}, \mathbf{F}_\ell^{r'}], \text{dim} = 1))$
  /* Differential Separation with Curriculum */
  Compute depth-dependent weight:
    $\lambda_\ell^{\text{init}} = 0.8 - 0.6\exp(-0.3\ell)$
  Compute epoch-wise warmup:
    $\lambda_{\text{diff}}(e) = \begin{cases} 0.1 + 0.9\frac{e}{E_{\text{warmup}}}, & e < E_{\text{warmup}} \\ 1.0, & e \geq E_{\text{warmup}} \end{cases}$
  $\lambda_\ell(e) \leftarrow \lambda_\ell^{\text{init}} \cdot \lambda_{\text{diff}}(e)$
  Apply differential operators:
    $\mathbf{A}_{\text{diff}}^t = (\mathbf{A}_{\text{SA}}^t + \mathbf{A}_{\text{CA}}^t) - \sigma(\lambda_\ell)(\mathbf{A}_{\text{SA}}^r + \mathbf{A}_{\text{CA}}^r)$
    $\mathbf{A}_{\text{diff}}^r = (\mathbf{A}_{\text{SA}}^r + \mathbf{A}_{\text{CA}}^r) - \sigma(\lambda_\ell)(\mathbf{A}_{\text{SA}}^t + \mathbf{A}_{\text{CA}}^t)$
  /* Late Fusion: FFN with Residual */
  $\mathbf{F}_{\ell+1}^t = \mathbf{F}_\ell^t + \text{FFN}(\mathbf{A}_{\text{diff}}^t)$
  $\mathbf{F}_{\ell+1}^r = \mathbf{F}_\ell^r + \text{FFN}(\mathbf{A}_{\text{diff}}^r)$
end for
// Stage 3: Output Generation
$\hat{\mathbf{T}}, \hat{\mathbf{R}} \leftarrow \text{Conv}_{3\times3}(\mathbf{F}_0^t), \text{Conv}_{3\times3}(\mathbf{F}_0^r)$
$\hat{\mathbf{RR}} \leftarrow \text{LRM}(\mathbf{F}_0^t + \mathbf{F}_0^r)$
// Loss Computation
$\mathcal{L}_{\text{total}} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{refl}}\mathcal{L}_{\text{refl}} + \lambda_{\text{vgg}}\mathcal{L}_{\text{vgg}}$
  $+\lambda_{\text{exclu}}\mathcal{L}_{\text{exclu}} + \lambda_{\text{recons}}\mathcal{L}_{\text{recons}} + \lambda_{\text{color}}\mathcal{L}_{\text{color}}$
return $\hat{\mathbf{T}}, \hat{\mathbf{R}}, \hat{\mathbf{RR}}$

1. Algorithm Description

We provide a detailed algorithmic description of ReflexSplit in Algorithm 1, which illustrates the complete training pipeline including dual-branch feature extraction, hierarchical decoding with CrGF and LFSB, and curriculum learning strategy.

2. Loss Functions Details

Let $\hat{\mathbf{T}}$, $\hat{\mathbf{R}}$, and $\hat{\mathbf{R}\mathbf{R}}$ denote the predicted transmission, reflection, and residual layers, respectively, with $\mathbf{T}$ and $\mathbf{R}$ as ground truth. Our training objective combines multiple complementary loss terms:

$$\begin{aligned}\mathcal{L}_{\text{total}} = \lambda_{\text{rec}}\,\mathcal{L}_{\text{rec}} + \lambda_{\text{refl}}\,\mathcal{L}_{\text{refl}} + \lambda_{\text{vgg}}\,\mathcal{L}_{\text{vgg}} \\ + \lambda_{\text{exclu}}\,\mathcal{L}_{\text{exclu}} + \lambda_{\text{recons}}\,\mathcal{L}_{\text{recons}} + \lambda_{\text{color}}\,\mathcal{L}_{\text{color}}.\end{aligned} \tag{1}$$

Reconstruction losses. For transmission layer reconstruction, we employ Charbonnier loss [1] to handle outliers robustly:

$$\mathcal{L}_{\text{rec}} = \sqrt{\|\hat{\mathbf{T}} - \mathbf{T}\|^2 + \epsilon^2}, \quad \epsilon = 10^{-3}. \tag{2}$$

For reflection layer supervision, we use ℓ_1 loss:

$$\mathcal{L}_{\text{refl}} = \|\hat{\mathbf{R}} - \mathbf{R}\|_1. \tag{3}$$

Perceptual loss. To preserve semantic content and texture details, we adopt VGG perceptual loss [3] using features extracted from layers $\{2, 7, 12, 21, 30\}$ of a pretrained VGG-19 network:

$$\mathcal{L}_{\text{vgg}} = \sum_{i \in \{2,7,12,21,30\}} w_i \cdot \|\phi_i(\hat{\mathbf{T}}) - \phi_i(\mathbf{T})\|_1, \tag{4}$$

where $\phi_i(\cdot)$ denotes the feature extractor at layer i and w_i are weights balancing contributions across layers.

Exclusion loss. Following [6], we enforce gradient orthogonality between transmission and reflection to encourage layer independence:

$$\mathcal{L}_{\text{exclu}} = \sum_{l=1}^{3} \left(\|\nabla_x \hat{\mathbf{T}} \odot \nabla_x \hat{\mathbf{R}}\|_1 + \|\nabla_y \hat{\mathbf{T}} \odot \nabla_y \hat{\mathbf{R}}\|_1 \right), \tag{5}$$

where ∇_x and ∇_y denote spatial gradients along horizontal and vertical directions, and $\odot$ represents element-wise multiplication. This term penalizes overlapping gradient patterns, thereby promoting structural separation between layers.

Reconstruction consistency. To ensure the decomposed layers reconstruct the input accurately, we enforce:

$$\mathcal{L}_{\text{recons}} = \|\hat{\mathbf{T}} + \hat{\mathbf{R}} + \hat{\mathbf{R}\mathbf{R}} - \mathbf{I}\|_1, \tag{6}$$

where $\mathbf{I}$ is the input mixed image. This constraint guarantees that no information is lost during decomposition.

Color consistency loss. To maintain color fidelity in the separated reflection layer, we introduce a color consistency term that matches color statistics between prediction and ground truth:

$$\mathcal{L}_{\text{color}} = \|\mu(\hat{\mathbf{R}}) - \mu(\mathbf{R})\|_1 + \|\sigma(\hat{\mathbf{R}}) - \sigma(\mathbf{R})\|_1, \tag{7}$$

Table 1. **Performance margin distribution on OpenRR-1K.** Comparison of ReflexSplit against RDNet [7] on both test and validation splits. $\Delta_{\text{PSNR}} = \text{PSNR}_{\text{Ours}} - \text{PSNR}_{\text{RDNet}}$.

Condition	Test Set (99 images)		Val Set (100 images)	
	Count	Percentage	Count	Percentage
$\Delta_{\text{PSNR}} > 0$ dB	63 / 99	63.64%	62 / 100	62.00%
$\Delta_{\text{PSNR}} > 3$ dB	20 / 99	20.20%	18 / 100	18.00%
$\Delta_{\text{PSNR}} > 5$ dB	8 / 99	8.08%	10 / 100	10.00%
$\Delta_{\text{PSNR}} > 7$ dB	5 / 99	5.05%	4 / 100	4.00%

where $\mu(\cdot)$ and $\sigma(\cdot)$ compute channel-wise mean and standard deviation.

Loss weights. We set the loss weights as: $\lambda_{\text{rec}} = 1.0$, $\lambda_{\text{refl}} = 0.5$, $\lambda_{\text{vgg}} = 0.1$, $\lambda_{\text{exclu}} = 1.0$, $\lambda_{\text{recons}} = 0.2$, and $\lambda_{\text{color}} = 0.1$.

3. Feature Separation Analysis: Comparison with DSIT

DSIT [2] represents a strong baseline with its dual-stream transformer architecture. However, we observe that different separation strategies may lead to distinct feature learning behaviors under complex real-world mixing conditions.

Figures 1 and 2 provide feature space analysis comparing ReflexSplit and DSIT on OpenRR-1K [5], revealing fundamental differences in separation strategies.

t-SNE feature distribution. Figure 1 visualizes transmission and reflection features across LFSB decoder levels. DSIT maintains strong separation throughout all levels, which represents an architectural philosophy that prioritizes immediate layer distinction. ReflexSplit demonstrates *progressive disentanglement*: features overlap at Level 4 to facilitate shared feature learning at deep layers, then gradually separate toward Level 0. This curriculum-based strategy $\mathbf{A}^t - \lambda(e)\mathbf{A}^r$ may enable learning shared features before specializing to layer-specific characteristics.

Principal component analysis. Figure 2 shows ReflexSplit features accumulate variance gradually, requiring more principal components to reach 95% threshold—indicating more distributed representations. DSIT exhibits steeper accumulation with variance concentrated in fewer components, suggesting more compact encoding. While both achieve efficient output encoding, ReflexSplit's intermediate LFSB features require more principal components to capture equivalent variance, which may facilitate progressive layer disentanglement.

These findings validate our differential attention design with curriculum learning, demonstrating that progressive separation represents a viable alternative to immediate separation approaches, particularly for complex real-world reflection removal scenarios.

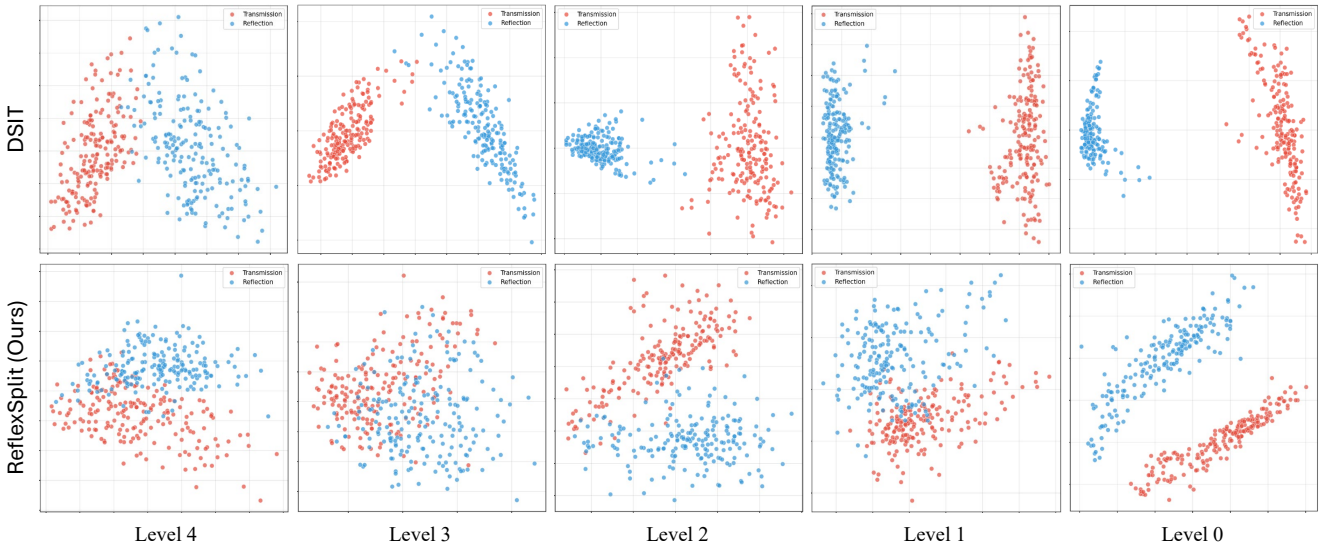

Figure 1. **t-SNE visualization reveals different separation strategies.** DSIT [2] (top) maintains strong separation across all levels, while ReflexSplit (bottom) demonstrates progressive disentanglement from overlap (Level 4) to clear separation (Level 0). Red/blue: transmission/reflection features.

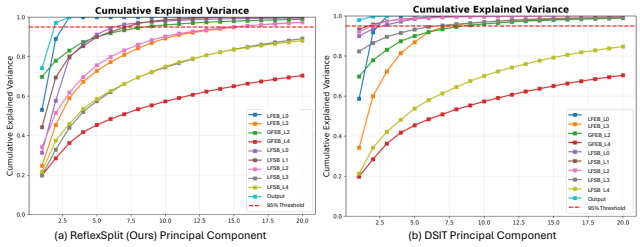

Figure 2. **Cumulative explained variance comparison.** (a) ReflexSplit shows gradual accumulation indicating richer representations. (b) DSIT exhibits steeper accumulation with variance concentrated in fewer components.

4. Comparison with RDNet on Real-World Scenarios

RDNet [7] represents a strong baseline with its reversible architecture that preserves information flow through the network. While implicit feature learning through reversible architectures shows strong performance across diverse scenarios, we observe that explicit layer separation constraints may provide complementary advantages in certain complex cases, particularly scenes with over-exposure, specular highlights, and spatially-varying attenuation.

Table 1 provides a detailed performance margin distribution comparing ReflexSplit and RDNet on OpenRR-1K [5]. ReflexSplit achieves improvements on 63% of test images, with notable gains (>3 dB) on 20% of cases. These improvements are particularly evident in challenging scenarios shown in Figure 3.

We hypothesize that these gains are related to our explicit differential attention mechanism $\mathbf{A}^t - \lambda \mathbf{A}^r$, which provides clear separation constraints at each decoder level. This architectural choice appears to offer advantages for maintaining layer distinction under complex nonlinear mixing, representing a complementary design philosophy to RDNet's reversible approach. Both strategies demonstrate merit, with performance differences dependent on specific scene characteristics.

5. Additional Visual Comparisons

We provide additional visual comparisons on OpenRR-1K [5] and SIR^2 [4] datasets in Figures 3–6.

As shown in Figure 3, ReflexSplit achieves cleaner transmission layers in challenging real-world scenarios. Figures 4 and 5 demonstrate consistent performance on synthetic benchmarks across different methods, each exhibiting different characteristics: DSRNet and DSIT show residual reflections in certain regions, MaxRF tends toward aggressive smoothing, while RDNet occasionally exhibits color shifts. Our method aims to balance these trade-offs through explicit layer separation constraints.

6. Failure Cases and Limitations

While ReflexSplit achieves state-of-the-art performance, we observe failure cases in challenging lighting conditions (Figures 7, 8):

Complex outdoor lighting. In outdoor scenes with strong sunlight, shadows, and varying illumination, the nonlinear mixing becomes highly spatially-varying. The separation

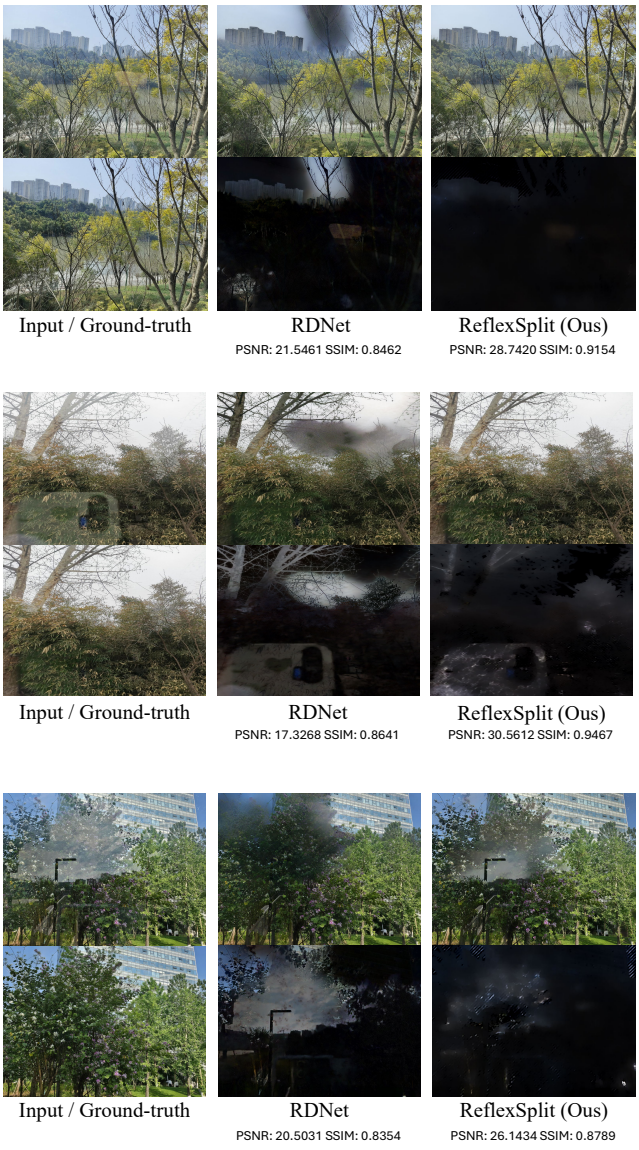

Figure 3. **Challenging real-world scenarios on OpenRR-1K [5].** Comparison of ReflexSplit with state-of-the-art RDNet [7] on diverse degradation types. Our differential attention mechanism provides explicit layer separation constraints, achieving natural color fidelity and sharp detail preservation in these challenging cases.

becomes particularly challenging when transmission and reflection layers experience different lighting conditions simultaneously, causing inconsistent separation across image regions.

Specular reflections. Strong specular highlights from mirror-like surfaces produce localized over-exposure that saturates both layers. In these regions, the separation becomes ill-posed as no texture information remains to distinguish transmission from reflection—a fundamental challenge for learning-based approaches.

Table 2. Efficiency comparison on 384×384 input resolution.

Method	Params (M)	FLOPs (G)	Time (ms)	FPS	Training Stage	Reflection Separation
DExNet	9.6	995.841	185.114	5.40	Single	✓
MaxRF	27.9	92.135	40.855	24.48	Multiple	×
DSIT	136	867.704	172.199	5.81	Single	✓
RDNet	266.4	1047.141	122.800	8.14	Multiple	✓
DAI	1738.5	7850.567	418.649	2.39	Multiple	×
ReflexSplit	174	969.050	211.349	4.73	Single	✓

Mixed indoor-outdoor scenes. When capturing indoor scenes through glass with outdoor backgrounds, the extreme brightness difference (often exceeding camera dynamic range) causes either indoor under-exposure or outdoor over-saturation. Reliable layer separation becomes difficult when one component is dominated by noise or clipping artifacts, representing a limitation shared by most decomposition methods.

7. Network Architecture Details

Table 3 details ReflexSplit's complete architecture across four stages: dual-branch feature extraction (GFEB, LFEB), feature mixing, hierarchical decoding with CrGF and LFSB, and output generation.

8. More Complexity Comparison

Table 2 compares the computational efficiency of ReflexSplit against existing methods on 384×384 input resolution. ReflexSplit achieves a favorable balance between model complexity and task capability. While DAI and RDNet require multiple training stages and incur substantially higher parameter counts (1738.5M and 266.4M, respectively), ReflexSplit completes training in a single stage with 174M parameters, simplifying the training pipeline without sacrificing reflection separation capability. Compared to DExNet and DSIT, which are also single-stage methods supporting reflection separation, ReflexSplit achieves competitive FLOPs (969.050G vs. 995.841G and 867.704G) while offering more comprehensive dual-branch feature modeling. Although MaxRF reports lower FLOPs (92.135G) and faster inference (24.48 FPS), it does not support direct reflection separation, limiting its applicability to this task. Overall, ReflexSplit represents a practical single-stage solution that targets reflection separation with reasonable computational overhead, making it suitable for deployment without the complexity of multi-stage training pipelines.

References

[1] Jonathan T. Barron. A general and adaptive robust loss function. In CVPR, 2019. 2

[2] Qiming Hu, Hainuo Wang, and Xiaojie Guo. Single image reflection separation via dual-stream interactive transformers. NeurIPS, 2024. 2 and 3

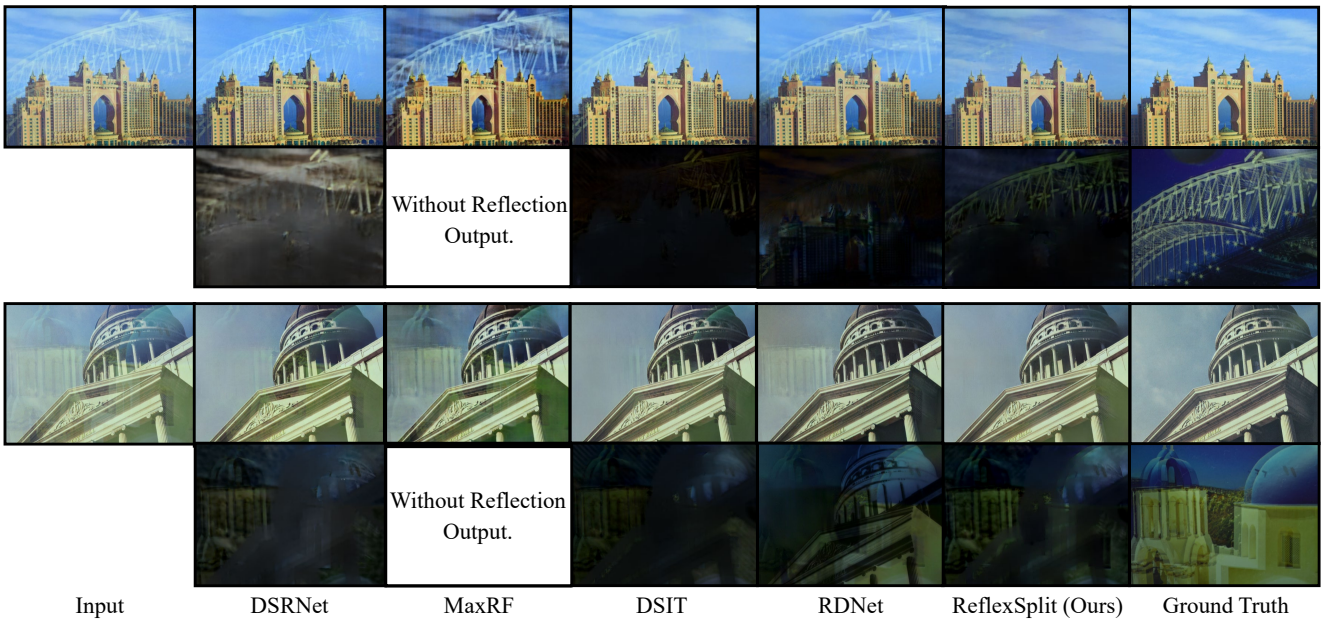

Figure 4. **Qualitative comparison on Postcard [4].** ReflexSplit achieves superior reflection suppression with faithful detail preservation.

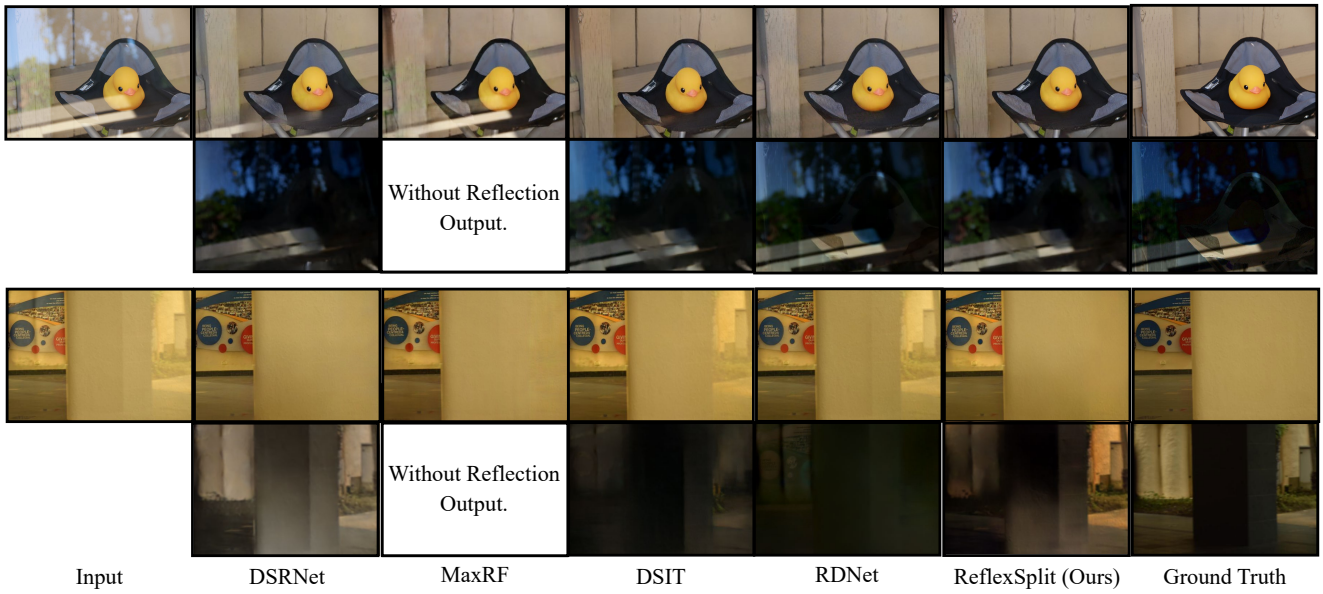

Figure 5. **Qualitative comparison on Wild and SolidObject [4].**

[3] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2015. 2

[4] Renjie Wan, Boxin Shi, Ling-Yu Duan, Ah-Hwee Tan, and Alex C Kot. Benchmarking single-image reflection removal algorithms. In ICCV, 2017. 3 and 5

[5] Kangning Yang, Ling Ouyang, Huiming Sun, Jie Cai, Lan Fu, Jiaming Ding, Chiu Man Ho, and Zibo Meng. Openrr-1k: A scalable dataset for real-world reflection removal, 2025. 2, 3, 4, and 6

[6] Xuaner Zhang, Ren Ng, and Qifeng Chen. Single image reflection separation with perceptual losses. In CVPR, 2018. 2

[7] Hao Zhao, Mingjia Li, Qiming Hu, and Xiaojie Guo. Reversible decoupling network for single image reflection removal. arXiv preprint arXiv:2410.08063, 2024. 2, 3, and 4

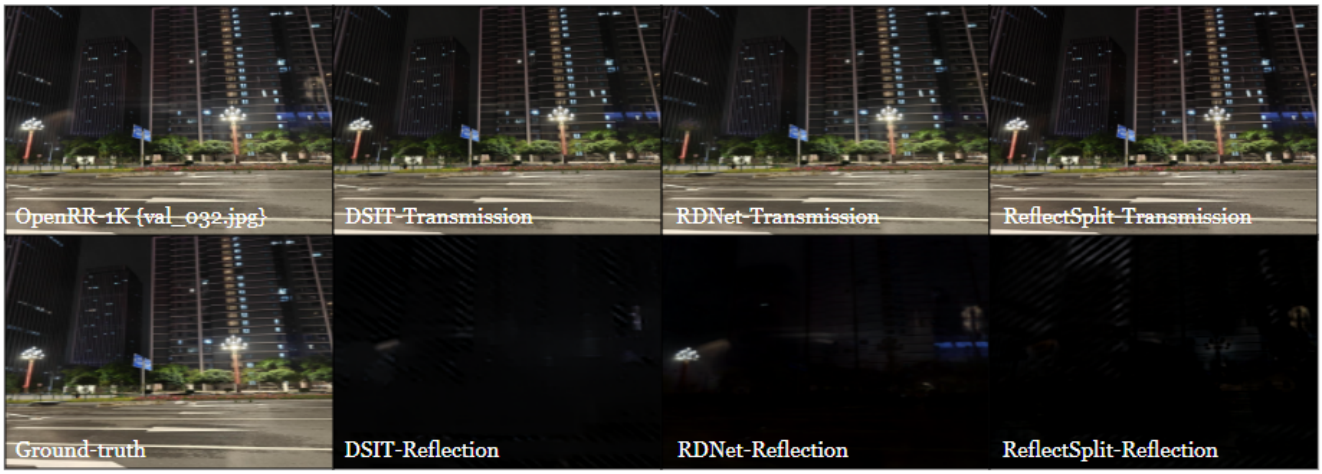

Figure 6. **Additional visualization on OpenRR-1K [5].**

Figure 7. **Failure cases on OpenRR-1K [5].**

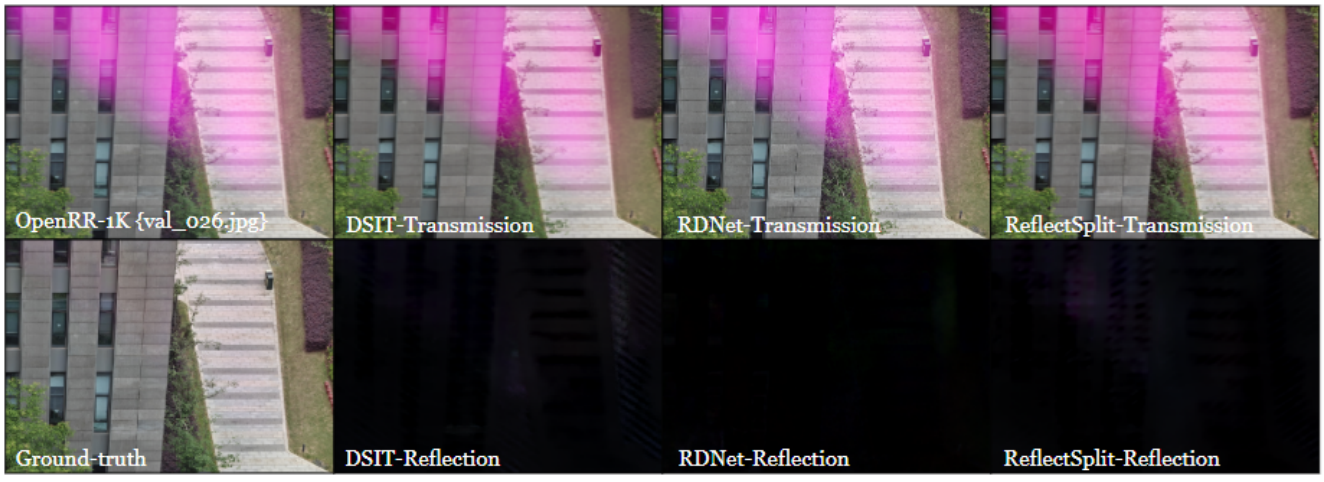

Figure 8. **Failure cases on OpenRR-1K [5].**

Table 3. **Architecture of ReflexSplit.** The model takes a 384×384 input image and processes it through dual-branch encoders (GFEB and LFEB) followed by hierarchical decoding with CrGF and LFSB.

Block Name	Output Size	Transmission Branch	Reflection Branch	Order
Stage 1: Dual-Branch Feature Extraction				
Global Feature Extractor Block (GFEB)				
Swin Transformer Stage 2	$96 \times 96 \times 192$	Shared ($\mathbf{P}_2$)		1
Swin Transformer Stage 3	$48 \times 48 \times 384$	Shared ($\mathbf{P}_3$)		2
Swin Transformer Stage 4	$24 \times 24 \times 768$	Shared ($\mathbf{P}_4$)		3
Swin Transformer Stage 5	$12 \times 12 \times 1536$	Shared ($\mathbf{P}_5$)		4
Local Feature Extractor Block (LFEB)				
Conv 3×3	$384 \times 384 \times 48$	$3 \to 48$ ($\mathbf{E}_0$)	$3 \to 48$ ($\mathbf{E}_0$)	1
MuGI Block (×2)	$384 \times 384 \times 48$	$48 \to 48$	$48 \to 48$	2
Conv 2×2, stride=2	$192 \times 192 \times 96$	$48 \to 96$ ($\mathbf{E}_1$)	$48 \to 96$ ($\mathbf{E}_1$)	3
MuGI Block (×2)	$192 \times 192 \times 96$	$96 \to 96$	$96 \to 96$	4
Conv 2×2, stride=2	$96 \times 96 \times 192$	$96 \to 192$ ($\mathbf{E}_2$)	$96 \to 192$ ($\mathbf{E}_2$)	5
MuGI Block (×2)	$96 \times 96 \times 192$	$192 \to 192$	$192 \to 192$	6
Conv 2×2, stride=2	$48 \times 48 \times 384$	$192 \to 384$ ($\mathbf{E}_3$)	$192 \to 384$ ($\mathbf{E}_3$)	7
MuGI Block (×2)	$48 \times 48 \times 384$	$384 \to 384$	$384 \to 384$	8
Conv 2×2, stride=2	$24 \times 24 \times 768$	$384 \to 768$ ($\mathbf{E}_4$)	$384 \to 768$ ($\mathbf{E}_4$)	9
MuGI Block (×2)	$24 \times 24 \times 768$	$768 \to 768$	$768 \to 768$	10
Conv 2×2, stride=2	$12 \times 12 \times 1536$	$768 \to 1536$ ($\mathbf{E}_5$)	$768 \to 1536$ ($\mathbf{E}_5$)	11
Stage 2: Feature Mixing				
Initial Feature Interaction				
PixelShuffle (×2)	$24 \times 24 \times 384$	$1536 \to 384$	$1536 \to 384$	5
LFSB (depth=5)	$24 \times 24 \times 384$	$384 \to 384$	$384 \to 384$	6
Conv 1×1	$24 \times 24 \times 768$	$384 \to 768$	$384 \to 768$	7
LFSB (depth=4)	$24 \times 24 \times 768$	$768 \to 768$	$768 \to 768$	8
Stage 3: Hierarchical Decoding with CrGF and LFSB				
Decoder Level 4				
CrGF	$24 \times 24 \times 768$	$\mathbf{P}_4 + \mathbf{E}_4 + \mathbf{F}_5$	$\mathbf{P}_4 + \mathbf{E}_4 + \mathbf{F}_5$	9
LFSB (×12, depth=4)	$24 \times 24 \times 768$	$768 \to 768$	$768 \to 768$	10
PixelShuffle (×2)	$48 \times 48 \times 192$	$768 \to 192$	$768 \to 192$	11
MuGI Block (×2)	$48 \times 48 \times 192$	$192 \to 192$	$192 \to 192$	12
Conv 1×1	$48 \times 48 \times 384$	$192 \to 384$	$192 \to 384$	13
Decoder Level 3				
CrGF	$48 \times 48 \times 384$	$\mathbf{P}_3 + \mathbf{E}_3 + \mathbf{F}_4$	$\mathbf{P}_3 + \mathbf{E}_3 + \mathbf{F}_4$	14
LFSB (depth=3)	$48 \times 48 \times 384$	$384 \to 384$	$384 \to 384$	15
LFSB (×8, depth=3)	$48 \times 48 \times 384$	$384 \to 384$	$384 \to 384$	16
PixelShuffle (×2)	$96 \times 96 \times 96$	$384 \to 96$	$384 \to 96$	17
MuGI Block (×2)	$96 \times 96 \times 96$	$96 \to 96$	$96 \to 96$	18
Conv 1×1	$96 \times 96 \times 192$	$96 \to 192$	$96 \to 192$	19
Decoder Level 2				
CrGF	$96 \times 96 \times 192$	$\mathbf{P}_2 + \mathbf{E}_2 + \mathbf{F}_3$	$\mathbf{P}_2 + \mathbf{E}_2 + \mathbf{F}_3$	20
LFSB (depth=2)	$96 \times 96 \times 192$	$192 \to 192$	$192 \to 192$	21
LFSB (×4, depth=2)	$96 \times 96 \times 192$	$192 \to 192$	$192 \to 192$	22
PixelShuffle (×2)	$192 \times 192 \times 48$	$192 \to 48$	$192 \to 48$	23
MuGI Block (×2)	$192 \times 192 \times 48$	$48 \to 48$	$48 \to 48$	24
Conv 1×1	$192 \times 192 \times 96$	$48 \to 96$	$48 \to 96$	25
Decoder Level 1				
Direct Aggregation	$192 \times 192 \times 96$	$\mathbf{E}_1 + \mathbf{F}_2$	$\mathbf{E}_1 + \mathbf{F}_2$	26
LFSB (×2, depth=1)	$192 \times 192 \times 96$	$96 \to 96$	$96 \to 96$	27
PixelShuffle (×2)	$384 \times 384 \times 24$	$96 \to 24$	$96 \to 24$	28
MuGI Block (×2)	$384 \times 384 \times 24$	$24 \to 24$	$24 \to 24$	29
Conv 1×1	$384 \times 384 \times 48$	$24 \to 48$	$24 \to 48$	30
Decoder Level 0				
Direct Aggregation	$384 \times 384 \times 48$	$\mathbf{E}_0 + \mathbf{F}_1$	$\mathbf{E}_0 + \mathbf{F}_1$	31
LFSB (×2, depth=0)	$384 \times 384 \times 48$	$48 \to 48$	$48 \to 48$	32
MuGI Block (×2)	$384 \times 384 \times 48$	$48 \to 48$	$48 \to 48$	33
Stage 4: Output Generation				
Conv 3×3	$384 \times 384 \times 3$	$48 \to 3$ ($\hat{\mathbf{T}}$)	$48 \to 3$ ($\hat{\mathbf{R}}$)	34
Learnable Residue Module (LRM)				
Feature Aggregation	$384 \times 384 \times 48$	$\mathbf{F}_0^t + \mathbf{F}_0^r$		35
SinBlock	$384 \times 384 \times 48$	$48 \to 48$		36
Conv 3×3 + Tanh	$384 \times 384 \times 3$	$48 \to 3$ ($\hat{\mathbf{R}}\hat{\mathbf{R}}$)		37