

CAST-LUT: Tokenizer-Guided HSV Look-Up Tables for Purple Flare Removal

Pu Wang¹, Shuning Sun², Jialang Lu³, Chen Wu⁴, Zhihua Zhang¹, Youshan Zhang⁵, Chenggang Shan⁶, Dianjie Lu⁷, Guijuan Zhang⁷, Zhuoran Zheng⁸

¹ Shandong University

² University of the Chinese Academy of Sciences

³ Hubei University

⁴ University of Science and Technology of China

⁵ Yeshiva University

⁶ Zaozhuang University

⁷ Shandong Normal University

⁸ Qilu University of Technology

wangou@mail.sdu.edu.cn, sunshuning23@mailsucas.ac.cn, lujialang@stu.hubu.edu.cn, wuchen5X@mail.ustc.edu.cn, zhangzhihua@sdu.edu.cn, yz945@cornell.edu, shanchenggang@uzz.edu.cn, {ludianjie, zhangguijuan}@sdnu.edu.cn, zhengzr@njust.edu.cn

Abstract

Purple flare, a diffuse chromatic aberration artifact commonly found around highlight areas, severely degrades the tone transition and color of the image. Existing traditional methods are based on hand-crafted features, which lack flexibility and rely entirely on fixed priors, while the scarcity of paired training data critically hampers deep learning. To address this issue, we propose a novel network built upon decoupled HSV Look-Up Tables (LUTs). The method aims to simplify color correction by adjusting the Hue (H), Saturation (S), and Value (V) components independently. This approach resolves the inherent color coupling problems in traditional methods. Our model adopts a two-stage architecture: First, a Chroma-Aware Spectral Tokenizer (CAST) converts the input image from RGB space to HSV space and independently encodes the Hue (H) and Value (V) channels into a set of semantic tokens describing the Purple flare status; second, the HSV-LUT module takes these tokens as input and dynamically generates independent correction curves (1D-LUTs) for the three channels H, S, and V. To effectively train and validate our model, we built the first large-scale purple flare dataset with diverse scenes. We also proposed new metrics and a loss function specifically designed for this task. Extensive experiments demonstrate that our model not only significantly outperforms existing methods in visual effects but also achieves state-of-the-art performance on all quantitative metrics.

Code —

<https://github.com/Pu-Wang-alt/Reduce-Purple-Flare/>

Datasets —

https://huggingface.co/datasets/PuWang0/purple_flare

Formal version —

Introduction

In complex lighting environments, images are often affected by various color artifacts (Yu and Zhu 2017; Abbasi et al.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

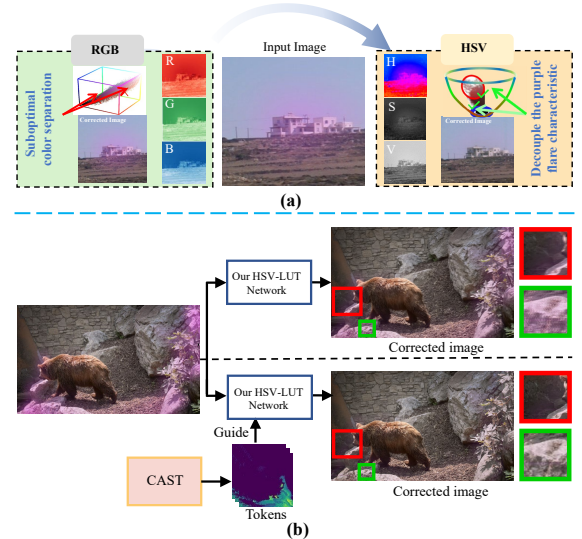


Figure 1: **Color space analysis and CAST-guided correction comparison.** (a) RGB vs. HSV: HSV decouples purple flare characteristics more clearly. (b) CAST-guided effect: The CAST module significantly improves performance.

2024), with the purple flare being one of the most common. Unlike general image degradations such as noise or blur, purple flare presents a unique challenge due to its diffuse nature, color specificity, and strong spatial correlation with highlight regions, severely impacting the image’s color fidelity (Min et al. 2025; Chen et al. 2024). This issue is especially prominent in mobile photography, consumer-grade cameras, and low-cost optical systems, primarily caused by factors such as internal lens reflections, sensor saturation, and chromatic dispersion (Spencer et al. 1995).

Recently, image enhancement techniques have made significant progress in areas such as deraining (Liu et al. 2025), dehazing (Sabitha and Eluri 2024; Kumari, Kumar, and Teja 2024; Agrawal et al. 2024), and color restoration (Chen et al.

2021; Liang et al. 2021). However, the purple flare has not been systematically studied yet. Early attempts relied on traditional methods utilizing decoupled color spaces like HSV or Lab (Lee, Kim, and Park 2011; Kim and Park 2010; Ware et al. 2018). These methods are constrained by their reliance on hand-crafted, fixed thresholds, which causes them to lack flexibility and generalization capability in complex scenes, limiting their practical application. To overcome the limitations of traditional methods, researchers have turned to deep learning. Wu et al. (2021) pioneered the application of end-to-end CNNs to general flare removal by constructing a large-scale semi-synthetic dataset. To further bridge the domain gap between synthetic data and real-world flares, Deng et al. (2024) proposed a knowledge-driven hybrid approach that utilizes a flare-level estimator as prior knowledge to guide the restoration network. While these methods represent the state-of-the-art, they are essentially pixel-to-pixel restoration networks operating in the coupled RGB color space, lacking precise control over color decoupling for specific chromatic artifacts. Among them, driven by the need for efficiency, LUT-based methods have emerged as a prominent choice for lightweight architectures. However, existing LUT methods rely on the RGB space and are not sensitive to capturing the purple flare of local region transformation (see Fig. 1(a)). We observed that the HSV space can amplify the characteristic of local purple flare. Still, the non-linear transformation from RGB to HSV can amplify local color space noise, creating new artifacts in the final result (Gevers et al. 2012). To leverage the benefits of the decoupled HSV color space without incurring the instability of traditional conversion, we propose CAST-LUT. This method employs a Chroma-Aware Spectral Tokenizer (CAST) to generate stable, high-level semantic representations of the purple flare. Fig. 1(b) illustrates the critical guiding role of the CAST module. These semantic tokens then guide the generation of independent 1D-LUTs for the H, S, and V channels.

Additionally, we constructed the Purple Flare Synthesis Dataset (PFSD), comprising 4,987 training pairs, 608 validation pairs, and 618 testing pairs. To simulate optical lens characteristics, we identify highlight regions at object contour edges as candidates, apply a Gaussian blur, and fuse purple flare into raw images. This is the first large-scale paired image collection for this task. Due to the localized nature of purple flare, averaging errors across the entire image can mask residual artifacts in small regions, making traditional global metrics ineffective for accurate evaluation. We therefore introduce new metrics: using a purple flare mask, we separate flare-region PSNR (PSNR-F) and non-flare-region PSNR (PSNR-NF) to assess flare restoration and clean-region fidelity, respectively. For color correction, we propose Hue Alignment Error (HAE), which quantifies hue differences in flare regions to evaluate color recovery.

- We propose CAST-LUT, which avoids traditional RGB-to-HSV noise amplification. Its Chroma-Aware Spectral Tokenizer (CAST) diagnoses purple flare into semantic tokens that guide an adaptive 1D-LUT for precise correction of each HSV channel.
- To address the data scarcity for the purple flare removal

task, we construct a large-scale, high-quality purple flare dataset containing diverse scenes and devices, providing a fair benchmark for training and evaluation.

- We introduce a new loss function and metrics for accurate evaluation. PSNR-F/NF assesses flare removal and detail preservation, respectively, while our proposed Hue Alignment Error (HAE) metric measures color fidelity.

Related Work

Chromatic Aberration and Flare Removal. Traditional methods note limitations in processing purple flare directly in sRGB space, as intense luminance variations interfere with color assessment. Researchers have explored other color spaces (Malik and Karthik 2018). Chung et al. (2009) identified chromatic fringes by comparing R, G, B channel intensities but faced false positives due to ambiguous purple definitions. To better separate chrominance from luminance, subsequent works shifted to other color spaces. Kim et al. (2010) used CIExy for luminance normalization, while Ju et al. leveraged YCbCr’s chrominance channels (Cb-Cr) to analyze color deviations independently (Ju and Park 2013). However, a common bottleneck is their reliance on static, experimentally determined absolute thresholds to filter high-contrast or near-saturated areas. Given varied purple flare characteristics across cameras and scenes, fixed thresholds lack universality and robustness. Considering degraded image representation reconstruction ability, we propose using a word segmenter to aid in detail reconstruction.

Discrete Semantic Representation with Vector Quantization. Vector Quantization (VQ) is widely used to generate semantic tokens, a key component of modern perceptual systems. Its core principle maps high-dimensional feature vectors to a finite set of learned codebook embeddings. This technique gained prominence with VQ-VAE (Van Den Oord, Vinyals et al. 2017), which established a discrete latent space in an autoencoder architecture and integrated a discrete latent bottleneck into such frameworks. A discrete latent space offers significant advantages: it enables learning rich, semantically meaningful representations, where each codebook entry captures distinct perceptual patterns analogous to vocabulary words (Schwettmann et al. 2021). Additionally, discretization improves generative model stability and efficiency by mitigating issues like posterior collapse, a challenge in traditional VAEs (Xue, Ding, and Lu 2019). The discretization of visual data into tokens has enabled powerful architectures like Transformers (Han et al. 2022) and inspired token or prompt-guided approaches for other vision tasks, such as few-shot learning (Li et al. 2024, 2025).

LUT-based Color Image Enhancement. Recently, the LUT paradigm has shown significant potential across image restoration tasks, evolving into complex task-specific architectures. For color image denoising, where high color fidelity is critical, early LUT methods were limited by independent color channel processing. Therefore, DnLUT (Yang et al. 2025) introduced a Pairwise Channel Mixer, which explicitly models inter-channel correlations by processing RGB channel pairs in parallel, boosting color denoising performance. For complex mixed-degradation scenarios

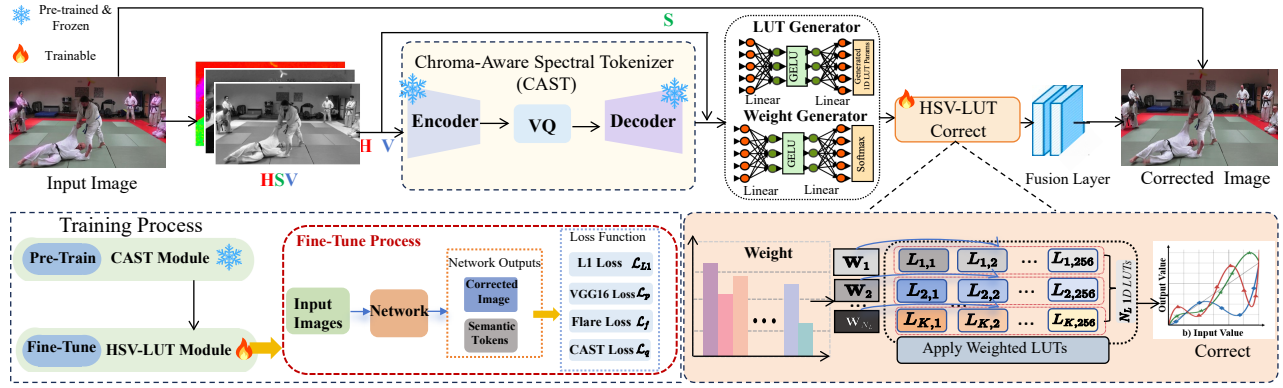


Figure 2: Overall architecture of the decoupled CAST-LUT network. The CAST module first extracts semantic tokens from the input image’s H and V channels via an Encoder-VQ pipeline. These tokens then guide the dynamic generation and weighted fusion of multiple, decoupled 1D-LUTs for the H, S, and V channels to obtain the final corrected image.

(e.g., low-light enhancement), the LUT framework has advanced further. DPLUT (Lin et al. 2025) proposed a two-stage approach, using a Light-Adjustment LUT and Noise-Suppression LUT to decouple brightening and denoising sub-tasks. However, despite SOTA performance in their domains, these advanced LUT methods are not designed for purple flare removal. Their model priors are inconsistent with the unique characteristics and spatial distribution of flare, and they are confined to operating in the RGB space.

Methodology

Our proposed purple flare removal network, named **Decoupled HSV-LUT Network**, adopts a novel two-stage “perceive-then-correct” paradigm. An overview of our framework is illustrated in Figure 2.

Chroma-Aware Spectral Tokenizer (CAST)

The CAST module acts as the framework’s perception engine, tasked with diagnosing purple flare features by translating complex visual artifacts into discrete semantic tokens.

Recognizing that purple flare is characterized by abnormal hue in highlight regions, we adopt a targeted approach. The **H** channel (chromatic signature) and **V** channel (highlight spatial context) undergo $4\times$ downsampling via a shared-weight CNN encoder, which processes them independently to extract hierarchical feature maps $\mathbf{F}_H, \mathbf{F}_V \in \mathbb{R}^{8 \times 256 \times H' \times W'}$. The **S** channel undergoes no encoding processing and remains in its original state to preserve the image’s inherent color richness. The encoder extracts hierarchical feature maps for each channel:

$$\mathbf{F}_H = \text{Encoder}(\mathbf{H}), \quad \mathbf{F}_V = \text{Encoder}(\mathbf{V}). \quad (1)$$

To bridge low-level visual features and high-level semantic understanding, these continuous feature maps $\mathbf{F}_H$ and $\mathbf{F}_V$ are discretized into semantic tokens via a Vector Quantization (VQ) module, where specific tokens learn to represent recurring purple flare characteristics (e.g., “high-intensity purple on edges”). Specifically, the VQ module maps each

feature vector to the nearest codebook entry, thereby effectively categorizing visual patterns and generating discrete token indices and quantized representations:

$$\mathbf{T}_H, \mathbf{F}_{\text{quant}}^H = \text{VQ}(\mathbf{F}_H), \quad \mathbf{T}_V, \mathbf{F}_{\text{quant}}^V = \text{VQ}(\mathbf{F}_V). \quad (2)$$

Next, a decoder network reconstructs the H and V channels from the quantized features.

$$\hat{\mathbf{H}} = \text{Decoder}(\mathbf{F}_{\text{quant}}^H), \quad \hat{\mathbf{V}} = \text{Decoder}(\mathbf{F}_{\text{quant}}^V). \quad (3)$$

The original **S** channel is combined with $\hat{\mathbf{H}}$ and $\hat{\mathbf{V}}$ to form a reconstructed HSV image, which is then converted back to RGB to obtain the initial reconstructed image $\mathbf{I}_{\text{recon}}^{\text{RGB}}$. The training of CAST is supervised by the reconstruction error between $\mathbf{I}_{\text{recon}}^{\text{RGB}}$ and $\mathbf{I}^{\text{RGB}}$, ensuring that quantized features preserve key information from the input image. In the subsequent color correction stage, we treat the discrete token indices $\mathbf{T}_{\text{combined}} = [\mathbf{T}_H, \mathbf{T}_V]$ as high-level semantic representations of purple flare characteristics and feed them into the LUT module.

HSV-LUT Color Correction

The color correction step of our method is executed by the HSV-based decoupled 1D-LUT module. Guided by the semantic tokens $\mathbf{T}_{\text{combined}}$ generated by CAST, this module performs color correction on the reconstructed image $\mathbf{I}_{\text{recon}}^{\text{RGB}}$ while leveraging residual information from the original image $\mathbf{I}^{\text{RGB}}$ to preserve high-frequency details.

Token-Guided LUT Generation. First, the semantic tokens $\mathbf{T}_{\text{combined}}$ are converted into a global feature vector $\mathbf{f}_{\text{token}}$ containing purple flare characteristics through embedding and aggregation operations, which drives two parallel sub-networks: the LUT generator and the weight generator.

$$\mathbf{f}_{\text{token}} = \text{Aggregate}(\text{Embed}(\mathbf{T}_{\text{combined}})). \quad (4)$$

(1) **LUT Generator:** This network processes $\mathbf{f}_{\text{token}}$ using an MLP to generate and reshape parameters $\mathbf{P}$ for N_L groups of LUTs. Each group contains three independent 1D-LUTs tailored to the H, S, and V channels, respectively. This decoupled design enables targeted adjustments for each color attribute.

$$\mathbf{P} = \text{Reshape}(\text{MLP}(\mathbf{f}_{\text{token}})). \quad (5)$$

(2) **Weight Generator.** Parallel to the LUT generator, the weight generator processes $\mathbf{f}_{\text{token}}$ using an MLP and normalizes via Softmax to predict dynamic fusion weights $\mathbf{W} \in \mathbb{R}^{N_L}$. These weights determine the contribution of the N_L LUT groups.

$$\mathbf{W} = \text{Softmax}(\text{MLP}(\mathbf{f}_{\text{token}})). \quad (6)$$

Decoupled HSV Correction via 1D-LUTs. First, the reconstructed image $\mathbf{I}_{\text{recon}}^{\text{RGB}}$ generated by CAST is converted to the HSV color space and decomposed into its three channels $[\mathbf{H}_{\text{in}}, \mathbf{S}_{\text{in}}, \mathbf{V}_{\text{in}}]$.

Our correction process operates on each channel independently within the HSV color space. For each LUT set $i = 1, \dots, N_L$, its corresponding three 1D-LUTs ($\text{LUT}_{\mathbf{H},i}, \text{LUT}_{\mathbf{S},i}, \text{LUT}_{\mathbf{V},i}$) are applied to the three input channels $[\mathbf{H}_{\text{in}}, \mathbf{S}_{\text{in}}, \mathbf{V}_{\text{in}}]$ respectively. These results are then fused into the final corrected channels $\mathbf{C}_{\text{final}}$ via weighted averaging with the predicted weight vector $\mathbf{W}$. This process can be described by the following equation:

$$\mathbf{C}_{\text{final}} = \sum_{i=1}^{N_L} \mathbf{W}_i \cdot \text{ApplyLUT}(\mathbf{C}_{\text{in}}, \text{LUT}_{\mathbf{C},i}), \quad (7)$$

where $\mathbf{C}_{\text{in}} \in \{\mathbf{H}_{\text{in}}, \mathbf{S}_{\text{in}}, \mathbf{V}_{\text{in}}\}$. Finally, the final corrected channels $\mathbf{C}_{\text{final}}$ are re-combined and converted back to the RGB color space, yielding the LUT-corrected image $\mathbf{I}_{\text{fused}}^{\text{RGB}}$.

Residual Branch and Final Fusion. To preserve details, a residual branch processes the original input image $\mathbf{I}^{\text{RGB}}$ to extract complementary high-frequency features, denoted as $\mathbf{I}_{\text{residual}}$. The color-corrected image from the main path, $\mathbf{I}_{\text{fused}}^{\text{RGB}}$, is then concatenated with these residual features to seamlessly integrate the information from both paths. A global skip connection adds the original image $\mathbf{I}^{\text{RGB}}$ to this fused result, producing the final corrected image $\mathbf{I}_{\text{output}}$.

$$\mathbf{I}_{\text{output}} = \text{Fusion}(\text{Concat}(\mathbf{I}_{\text{fused}}^{\text{RGB}}, \mathbf{I}_{\text{residual}})) + \mathbf{I}^{\text{RGB}}. \quad (8)$$

This architecture stabilizes training and focuses the network on learning necessary corrections to the residual instead of reconstructing the entire image from scratch. This approach stabilizes training and allows the two branches to specialize: the main branch focuses on complex color correction, while the residual branch ensures high-fidelity detail preservation.

Loss Function

Our model is trained using a composite loss function $\mathcal{L}_{\text{total}}$:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{L1} + \lambda_p \mathcal{L}_p + \lambda_f \mathcal{L}_f + \lambda_q \mathcal{L}_q, \quad (9)$$

where $\lambda_1, \lambda_p, \lambda_f, \lambda_q$ balance contributions from pixel accuracy, perceptual similarity, artifact suppression, and representation stability. $\mathcal{L}_{L1} = \|\mathbf{I}_{\text{output}} - \mathbf{I}_{\text{GT}}\|_1$ ensures pixel-level fidelity, while $\mathcal{L}_p = \|\phi(\mathbf{I}_{\text{output}}) - \phi(\mathbf{I}_{\text{GT}})\|_1$ uses VGG-16 features (Simonyan and Zisserman 2014) to enhance perceptual quality. Finally, $\mathcal{L}_q = \|\text{sg}(\mathbf{F}) - \mathbf{e}\|_2^2$ regularizes the codebook by penalizing the distance between the encoder’s output feature $\mathbf{F}$, and its nearest codebook vector $\mathbf{e}$, ensuring stable token representations.

Purple Flare Suppression Loss $\mathcal{L}_f$. We have introduced a purple flare suppression loss specifically targeting the purple

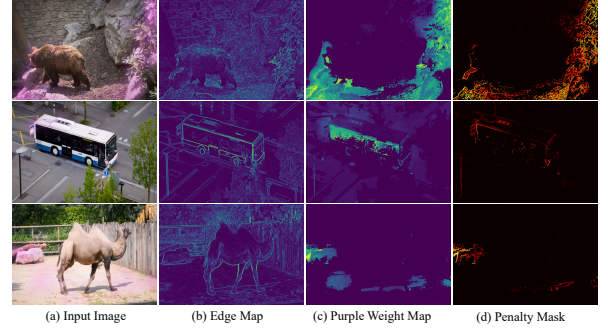


Figure 3: The generation process of the penalty mask $\mathbf{M}$ used in our flare suppression loss $\mathcal{L}_f$. The final mask (d) is obtained by element-wise multiplication of the edge map (b) and the purple weight map (c), which can precisely locate flare artifacts and apply targeted loss.

flare. Standard L1 or L2 losses treat errors across all pixels equally, failing to focus on addressing specific color artifacts in targeted regions. To tackle this, we introduce a weighted loss term that amplifies errors in purple flare regions through a penalty mask $\mathbf{M}$, as shown in Figure 3. This mask is obtained by element-wise multiplication of two components:

$$\mathbf{M} = \mathbf{M}_{\text{flare}} \odot \mathbf{M}_{\text{edge}}, \quad (10)$$

where $\odot$ denotes element-wise multiplication, $\mathbf{M}_{\text{flare}}$ is a color feature map used to identify pixels in the image with purple hue and high saturation, $\mathbf{M}_{\text{edge}}$ is an edge map generated by edge detection operators such as Sobel. In this way, the loss is significantly amplified only in regions that simultaneously satisfy the two conditions of “purple color” and “being located at edges”. The final suppression loss is defined as:

$$\mathcal{L}_f = \|\mathbf{M} \odot (\mathbf{I}_{\text{output}} - \mathbf{I}_{\text{GT}})\|_1. \quad (11)$$

Experiments

Experimental Settings

Datasets and Baselines. We conduct experiments on our new Purple Flare Synthesis Dataset (PFSD), created by applying a parametric synthesis pipeline to high-resolution frames from the DAVIS dataset. Our dataset consists of 4,987 pairs for training, 608 pairs for validation, and 618 pairs for testing. Further details on the dataset construction and the source code can be found in the supplementary material. For a comprehensive evaluation, we compare our method against state-of-the-art (SOTA) baselines from three distinct categories: classical, LUT-based, and scene-specific restoration methods.

Metrics. To comprehensively evaluate our method, we report standard metrics: PSNR and SSIM for image fidelity, LPIPS for perceptual quality, and ΔE in the CIELAB space for color difference and efficiency measures: model parameters, FLOPs, and runtime. However, since global metrics fail to simultaneously evaluate targeted artifact removal and the preservation of clean regions, we propose a more fine-grained protocol. We introduce two specialized metrics:

Category	Method	Venue & Year	Global Quality Metrics				Artifact Specific Metrics			Efficiency Metrics (256 ²)	
			PSNR $\uparrow$	SSIM $\uparrow$	LPIPS $\downarrow$	ΔE $\downarrow$	PSNR-F $\uparrow$	PSNR-NF $\uparrow$	HAE $\downarrow$	FLOPs (G) $\downarrow$	Runtime (ms) $\downarrow$
LUT-based	3DLUT (Zeng et al. 2020)	TPAMI'20	30.34	0.96	0.06	4.17	23.87	30.34	19.12	26.34	9.36
	SR-LUT (Jo and Kim 2021)	CVPR'21	29.76	0.85	0.12	5.66	21.35	29.58	10.21	25.39	8.25
	SPF-LUT (Li, Li, and Xiong 2024)	CVPR'24	32.19	0.94	0.09	5.18	21.93	33.19	7.57	34.38	9.12
	NILUT (Conde et al. 2024)	AAAI'24	32.31	0.95	0.07	4.50	24.86	32.93	4.87	30.89	7.12
	DnLUT (Yang et al. 2025)	CVPR'25	30.52	0.88	0.10	5.17	23.31	30.25	9.51	25.16	8.71
	CAST-LUT (Ours)	-	34.96	0.99	0.03	2.71	30.74	34.96	4.10	23.32	6.09
Classical	CBM3D (Dabov et al. 2007)	TIP'07	30.57	0.90	0.22	4.73	27.49	31.57	18.53	-	43.11
	MC-WNNM (Xu et al. 2017)	ICCV'17	27.98	0.80	0.20	9.53	26.23	28.00	18.01	-	121.23
Scene Specific Restoration	Zero-DCE (Guo et al. 2020)	CVPR'20	14.92	0.84	0.13	19.59	13.34	14.92	40.16	34.79	10.73
	RUAS (Liu et al. 2021)	CVPR'21	19.96	0.68	0.35	31.30	8.73	9.96	28.49	43.84	11.34
	LightenDiffusion (Jiang et al. 2024)	ECCV'24	30.59	0.97	0.08	5.01	24.66	30.59	27.25	27.01	10.25
	BPAM (Lou et al. 2025)	ICCV'25	29.93	0.95	0.12	5.10	27.26	28.93	24.75	15.21	13.04
	HVI-CIDNet (Yan et al. 2025)	CVPR'25	32.17	0.98	0.04	2.92	26.10	33.17	17.39	27.50	14.94

Table 1: Quantitative comparison with state-of-the-art methods on our purple flare dataset. $\uparrow$ indicates higher is better, and $\downarrow$ indicates lower is better.



Figure 4: Qualitative comparison with state-of-the-art methods on our PFSD dataset.

Flare Region PSNR (**PSNR-F**) to measure removal effectiveness and Non-Flare Region PSNR (**PSNR-NF**) to quantify detail preservation in unaffected areas.

Furthermore, to specifically assess the accuracy of color

restoration, we propose the Hue Alignment Error (**HAE**). This perceptually-driven metric calculates the saturation-weighted circular hue difference within the flare region, reflecting that the human eye is more sensitive to hue errors in



Figure 5: Visualization of learned semantic tokens from the CAST module.

vibrant colors.

$$\text{HAE} = \frac{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{M}_{\text{flare}}} \Delta H(\mathbf{x}, \mathbf{y}) \cdot \mathbf{S}_{GT}(\mathbf{x}, \mathbf{y})}{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{M}_{\text{flare}}} \mathbf{S}_{GT}(\mathbf{x}, \mathbf{y}) + \epsilon}, \quad (12)$$

where ΔH is the circular hue difference. Together, PSNR-F/NF and HAE provide a comprehensive framework for evaluating this localized removal task.

Implementation Details. The model was implemented in PyTorch and trained on an NVIDIA RTX 4090 (32G) via a two-stage strategy: a pre-trained CAST module (4096-entry codebook, 128-dim embedding) was loaded and frozen, then the main network trained for 100 epochs with AdamW (batch size 8, initial lr 1×10^{-4} decayed by cosine annealing). Inputs were resized to 256×256 with random horizontal flips and color jittering. Composite loss weights were set to 1.0, 0.1, 2.0, and 0.1 for $\lambda_1, \lambda_p, \lambda_f, \lambda_q$.

Comparison with State-of-the-Art Methods

Quantitative results. In the Table. 1, comprehensive quantitative comparisons on the PFSD test set demonstrate that our method achieves an HAE score of 4.10, the lowest among all methods, largely confirming its state-of-the-art performance in preserving chromatic fidelity. While general restoration methods like HVI-CIDNet perform well on global metrics such as ΔE , they struggle with targeted chromatic correction. CAST-LUT consistently outperforms all baselines across standard and artifact-specific metrics, validating its overall restoration quality and perceptual similarity to the ground truth, while its lightweight design ensures efficient runtime for real-time use on resource-constrained devices.

Qualitative results. Fig. 4 shows the visual comparisons on two challenging scenes from our test set. The first example (top row, a cyclist near grass) presents a common failure case where methods might misidentify and desaturate the naturally green-colored grass. As seen in the magnified patches, methods like d) and l) incorrectly alter the color of the grass. In contrast, our method (m) precisely removes the purple flare from the high-contrast areas while perfectly preserving the color fidelity of the grass. The second example (bottom row, a person in a glider) shows a sky region severely

ID	Model Configuration	PSNR $\uparrow$	PSNR-F $\uparrow$	PSNR-NF $\uparrow$	HAE $\downarrow$
<i>(1) Analysis of Core Correction Space & Mechanism</i>					
M1	w/o HSV (uses RGB 1D-LUTs)	28.04	24.15	29.81	8.81
M2	w/o Decoupled LUTs (uses 3D RGB-LUT)	30.05	25.66	30.95	7.79
<i>(2) Analysis of Guidance Module</i>					
M3	w/o CAST (simple CNN encoder)	31.54	29.01	29.42	6.5
M4	w/o VQ in CAST (continuous features)	29.50	28.45	29.49	6.7
<i>(3) Analysis of Architecture</i>					
M5	w/o Residual Branch	25.82	26.95	27.37	5.3
Full	CAST-LUT (Ours)	34.96	30.74	34.35	4.1

Table 2: Ablation study of the CAST-LUT framework.

corrupted by purple flare. While most competing methods either fail to completely remove the purple cast (e.g., g, h) or introduce unnatural color shifts (e.g., d, i), our method successfully restores the natural sky color and cloud details without any visible artifacts, demonstrating the effectiveness of our perception-correction paradigm.

Ablation study

To validate our design choices and quantify each key component’s contribution in CAST-LUT, we conducted comprehensive ablation experiments: starting with the full model, we systematically removed or replaced individual modules and analyzed performance impacts via our metrics. Results in Table 2 confirm the efficacy of each proposed component. **Effectiveness of the Decoupled HSV Space.** We validate our core mechanism by testing variants that operate in RGB space. As shown in Tab. 2, removing decoupled HSV correction and using RGB 1D-LUTs (M1) causes a catastrophic drop in performance, with PSNR-F plummeting from 30.74 to 24.15 and HAE worsening from 4.1 to 8.81. This highlights the severe limitations of color-coupled spaces. Similarly, using a standard 3D RGB-LUT (M2) is also significantly inferior, confirming the superiority of our decoupled 1D-LUT design for precise, targeted correction.

Effectiveness of the CAST Guidance. To show our semantic guidance’s importance, we replaced CAST with a simple CNN encoder (M3) and disabled its VQ step (M4). M3 drops PSNR-F to 29.01, M4 further to 28.45. This confirms CAST’s discrete semantic tokens offer more robust, effective

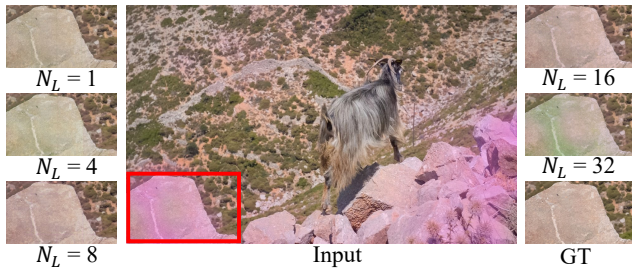


Figure 6: Visual ablation for the number of fused LUTs.

tive guidance than simple low-level features. Figure 5 intuitively demonstrates the semantic vocabulary learned by CAST using two key examples, comparing the full model (use VQ Tokens) with a baseline model that directly uses continuous features for guidance (w/o VQ Tokens). We feature blue sky as it is a common background for flares, and its spectral proximity to purple severely tests the model’s color stability. Concurrently, by visualizing a token for a purple object, we demonstrate CAST’s ability to distinguish a legitimate object from a purple flare based on contextual features rather than color alone. This content-aware discrimination is precisely what enables our model to eliminate artifacts while preserving the color fidelity of all objects in the scene.

Effectiveness of Architectural Choices and the number of fused LUTs (N_L). Final experiments validate our key architectural and loss function choices. First, we analyze our multi-LUT fusion mechanism by ablating the fused LUT count N_L . Results in Table 3 and Figure 6 confirm performance peaks at $N_L = 16$, achieving the best PSNR with effective artifact suppression. Other components are also critical: removing the residual branch in M5 severely degrades non-flare region detail preservation.

Analysis of Key Loss Weights. As presented in Table 4 and Figure 7, our ablation study on the loss weights, λ_f and λ_p , clearly demonstrates the necessity of both terms. Removing the flare suppression loss ($\mathcal{L}_f$) causes the HAE metric to worsen dramatically from 4.1 to 7.2, while removing the perceptual loss ($\mathcal{L}_p$) results in a significantly poorer LPIPS score of 0.061. Our final balanced configuration achieves the best overall performance by effectively navigating the trade-off between pixel-wise accuracy and perceptual quality.

Discuss

To verify practical applicability, we deployed the CAST-LUT model on mobile devices, achieving real-time performance. Leveraging adaptive 1D HSV LUTs, our approach is far more efficient than methods relying on large 3D LUTs or complex CNNs, while preserving fine details in unaffected areas through pixel-wise correction. Specifically, tests on the

Variant	$N_L = 1$	$N_L = 8$	$N_L = 16$ (Ours)	$N_L = 32$
PSNR $\uparrow$	30.11	31.35	34.96	32.43
HAE $\downarrow$	5.1	4.4	4.1	5.8

Table 3: Ablation on the number of fused LUTs (N_L).

ID	Variant	λ_f	λ_p	PSNR $\uparrow$	HAE $\downarrow$	LPIPS $\downarrow$
I	w/o $\mathcal{L}_f$	0.0	0.1	32.04	7.2	0.058
II	Low λ_f	0.5	0.1	32.21	5.8	0.054
III	High λ_f	5.0	0.1	32.32	5.9	0.052
IV	w/o $\mathcal{L}_p$	2.0	0.0	32.38	4.2	0.061
V	High λ_p	2.0	0.5	32.15	5.6	0.049
Ours (Balanced)		2.0	0.1	34.96	4.1	0.03

Table 4: Ablation study on the weights of the Purple Flare Suppression Loss (λ_f) and the Perceptual Loss (λ_p).

iPhone 14 recorded an 82ms execution time for 4K processing, demonstrating efficient hardware-software compatibility. Fig. 8 shows the mobile application’s interface and some failure and successful cases in real world. Supplementary material includes leading methods’ real-world results for broader comparison.

Conclusion

We propose CAST-LUT, a purple flare removal framework operating in decoupled HSV space to address traditional methods’ poor generalization and RGB-to-HSV conversion noise. Its core is CAST, which encodes purple flare artifacts into stable high-level semantic tokens. This avoids direct conversion noise while guiding the generation of independent 1D-LUTs for precise color correction. Extensive experiments on our large-scale PFSD dataset, using metrics like PSNR-F/NF and HAE, confirm its superior artifact removal and detail preservation performance.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Project (No. 62172265) and Shandong Provincial Natural Science Foundation (ZR2025MS1025, ZR2025MS1036).

References

Abbasi, M.; Váz, P.; Silva, J.; and Martins, P. 2024. Enhancing visual perception in immersive vr and ar environments:

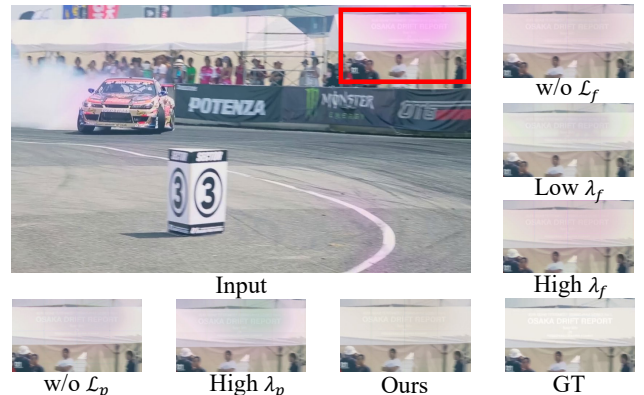


Figure 7: Visual ablation experiment of key loss terms.

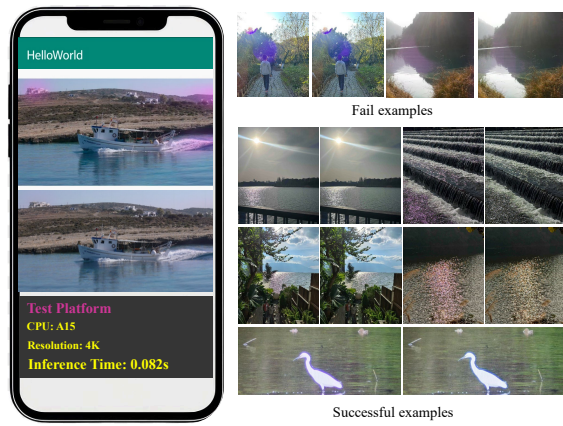


Figure 8: Sample image purple flare removing application on mobile devices based on the proposed CAST-LUT and fail and successful examples in real world.

Ai-driven color and clarity adjustments under dynamic lighting conditions. *Technologies*, 12(11): 216.

Agrawal, N. K.; Sharma, V.; Singh, P.; Sachi, S.; Jain, A.; and Alam, M. M. 2024. Fog Restoration in Hazy Images using Deep Transfer Learning. In *2023 International Conference on Smart Devices (ICSD)*, 1–5. IEEE.

Chen, C.; Mo, J.; Hou, J.; Wu, H.; Liao, L.; Sun, W.; Yan, Q.; and Lin, W. 2024. Topiq: A top-down approach from semantics to distortions for image quality assessment. *IEEE Transactions on Image Processing*, 33: 2404–2418.

Chen, H.; Wang, Y.; Guo, T.; Xu, C.; Deng, Y.; Liu, Z.; Ma, S.; Xu, C.; Xu, C.; and Gao, W. 2021. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12299–12310.

Chung, S.-W.; Kim, B.-K.; and Song, W.-J. 2009. Detecting and eliminating chromatic aberration in digital images. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, 3905–3908. IEEE.

Conde, M. V.; Vazquez-Corral, J.; Brown, M. S.; and Timofte, R. 2024. Nilut: Conditional neural implicit 3d lookup tables for image enhancement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 1371–1379.

Dabov, K.; Foi, A.; Katkovnik, V.; and Egiazarian, K. 2007. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *2007 IEEE international conference on image processing*, volume 1, 1–313. IEEE.

Deng, H.; Li, L.; Zhang, F.; Li, Z.; Xu, B.; Lu, Q.; Gao, C.; and Sang, N. 2024. Towards Blind Flare Removal Using Knowledge-driven Flare-level Estimator. *IEEE Transactions on Image Processing*.

Gevers, T.; Gijsenij, A.; Van de Weijer, J.; and Geusebroek, J.-M. 2012. *Color in computer vision: fundamentals and applications*. John Wiley & Sons.

Guo, C.; Li, C.; Guo, J.; Loy, C. C.; Hou, J.; Kwong, S.; and Cong, R. 2020. Zero-reference deep curve estima-

tion for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1780–1789.

Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. 2022. A survey on vision transformer. *IEEE transactions on pattern analysis and machine intelligence*, 45(1): 87–110.

Jiang, H.; Luo, A.; Liu, X.; Han, S.; and Liu, S. 2024. LighTenDiffusion: Unsupervised Low-Light Image Enhancement with Latent-Retinex Diffusion Models. In *European Conference on Computer Vision*.

Jo, Y.; and Kim, S. J. 2021. Practical single-image super-resolution using look-up table. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 691–700.

Ju, H. J.; and Park, R.-H. 2013. Colour fringe detection and correction in YCbCr colour space. *IET Image Processing*, 7(4): 300–309.

Kim, B.-K.; and Park, R.-H. 2010. Detection and correction of purple fringing using color desaturation in the xy chromaticity diagram and the gradient information. *Image and Vision Computing*, 28(6): 952–964.

Kumari, A.; Kumar, A. P.; and Teja, B. T. 2024. Real-Time Dehazing and Defogging: A Comprehensive Analysis for Single Image Haze and Fog Removal. In *2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS)*, 1–5. IEEE.

Lee, D.-K.; Kim, B.-K.; and Park, R.-H. 2011. Purple fringing correction using colorization in Yxy color space. In *2011 IEEE International Conference on Consumer Electronics (ICCE)*, 477–478. IEEE.

Li, W.; Wang, Q.; Meng, X.; Wu, Z.; and Yin, Y. 2025. VT-FSL: Bridging Vision and Text with LLMs for Few-Shot Learning. *arXiv preprint arXiv:2509.25033*.

Li, W.; Wang, Q.; Zhao, P.; and Yin, Y. 2024. KNN Transformer with Pyramid Prompts for Few-Shot Learning. In *Proceedings of the 32nd ACM International Conference on Multimedia*, 1082–1091.

Li, Y.; Li, J.; and Xiong, Z. 2024. Look-up table compression for efficient image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26016–26025.

Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; and Timofte, R. 2021. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844.

Lin, Y.; Fu, Z.; Wen, K.; Ye, T.; Chen, S.; Meng, G.; Wang, Y.; Kong, C.; Huang, Y.; Tu, X.; et al. 2025. DPLUT: Unsupervised Low-light Image Enhancement with Lookup Tables and Diffusion Priors. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 5316–5324.

Liu, R.; Ma, L.; Zhang, J.; Fan, X.; and Luo, Z. 2021. Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10561–10570.

- Liu, S.; Chen, X.; Chen, H.; Xu, Q.; and Li, M. 2025. DeRainGS: Gaussian Splatting for Enhanced Scene Reconstruction in Rainy Environments. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 5558–5566.
- Lou, J.; Zhao, X.; Shi, K.; and Gu, S. 2025. Learning Pixel-adaptive Multi-layer Perceptrons for Real-time Image Enhancement. *arXiv preprint arXiv:2507.12135*.
- Malik, P.; and Karthik, K. 2018. Iterative content adaptable purple fringe detection. *Signal, Image and Video Processing*, 12(1): 181–188.
- Min, X.; Gao, Y.; Cao, Y.; Zhai, G.; Zhang, W.; Sun, H.; and Chen, C. W. 2025. Exploring rich subjective quality information for image quality assessment in the wild. *IEEE Transactions on Circuits and Systems for Video Technology*.
- Sabitha, C.; and Eluri, S. 2024. Restoration of dehaze and defog image using novel cross entropy-based deep learning neural network. *Multimedia Tools and Applications*, 83(20): 58573–58606.
- Schwettmann, S.; Hernandez, E.; Bau, D.; Klein, S.; Andreas, J.; and Torralba, A. 2021. Toward a visual concept vocabulary for gan latent space. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 6804–6812.
- Simonyan, K.; and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- Spencer, G.; Shirley, P.; Zimmerman, K.; and Greenberg, D. P. 1995. Physically-based glare effects for digital images. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 325–334.
- Van Den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems*, 30.
- Ware, C.; Turton, T. L.; Bujack, R.; Samsel, F.; Shrivastava, P.; and Rogers, D. H. 2018. Measuring and modeling the feature detection threshold functions of colormaps. *IEEE transactions on visualization and computer graphics*, 25(9): 2777–2790.
- Wu, Y.; He, Q.; Xue, T.; Garg, R.; Chen, J.; Veeraraghavan, A.; and Barron, J. T. 2021. How to train neural networks for flare removal. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2239–2247.
- Xu, J.; Zhang, L.; Zhang, D.; and Feng, X. 2017. Multi-channel weighted nuclear norm minimization for real color image denoising. In *Proceedings of the IEEE international conference on computer vision*, 1096–1104.
- Xue, Y.; Ding, M.; and Lu, X. 2019. Supervised vector quantized variational autoencoder for learning interpretable global representations. *arXiv preprint arXiv:1909.11124*.
- Yan, Q.; Feng, Y.; Zhang, C.; Pang, G.; Shi, K.; Wu, P.; Dong, W.; Sun, J.; and Zhang, Y. 2025. Hvi: A new color space for low-light image enhancement. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5678–5687.
- Yang, S.; Huang, B.; Zhang, Y.; Yu, D.; Yang, Y.; and Wong, N. 2025. DnLUT: Ultra-Efficient Color Image Denoising via Channel-Aware Lookup Tables. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 7582–7591.
- Yu, S.-Y.; and Zhu, H. 2017. Low-illumination image enhancement algorithm based on a physical lighting model. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(1): 28–37.
- Zeng, H.; Cai, J.; Li, L.; Cao, Z.; and Zhang, L. 2020. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4): 2058–2073.

Supplementary Material

This supplementary material provides additional details to support our paper. Section A elaborates on the network architecture, including the specific implementations of the LUTGenerator, and WeightGenerator. Section B describes our synthetic dataset (PFSD), which covers the parameterized flare synthesis process and key hyperparameters. Section C presents further ablation analysis, offering additional studies on the impact of VQ codebook size and encoder/decoder depth on model performance and efficiency. Section D showcases extensive qualitative results, with visual comparisons against state-of-the-art methods on complex scenes, night scenes from the synthetic PFSD dataset, and various real-world images.

A. HSV-LUT Color Correction Module and Implementation

A.1. LUTGenerator Architecture.

The LUTGenerator is implemented as a Multi-Layer Perceptron (MLP) that maps the input feature vector $\mathbf{f}_{\text{token}}$ to a flat vector containing all necessary parameters for the 1D-LUTs. Its architecture is as follows:

Layer 1. A linear layer that maps the input dimension from `hidden_dim` to `hidden_dim`.

Activation. A GELU activation function (`nn.GELU`).

Layer 2 (Output). A final linear layer that maps from `hidden_dim` to the total number of LUT parameters. The output dimension is calculated as:

$$\mathbf{D}_{\text{out}} = N_L \times \mathbf{C}_{\text{HSV}} \times \mathbf{S}_{\text{LUT}}, \quad (13)$$

where N_L is the number of LUT sets, $\mathbf{C}_{\text{HSV}} = 3$ for the H, S, and V channels, and the size of each individual 1D-LUT is determined by $\mathbf{S}_{\text{LUT}}$ control points.

The resulting flat vector of dimension D_{out} is then reshaped into a structured tensor $\mathbf{P} \in \mathbb{R}^{B \times N_L \times \mathbf{C}_{\text{HSV}} \times \mathbf{S}_{\text{LUT}}}$, where B is the batch size.

A.2. WeightGenerator Architecture.

Running in parallel to the LUTGenerator, the WeightGenerator is another MLP designed to predict the fusion weights for each of the N_L LUT sets. Its architecture is as follows:

Layer 1. A linear layer that maps the input dimension from `hidden_dim` to a smaller intermediate dimension `hidden_dim` / 4. Given the input feature vector $\mathbf{f}_{\text{token}}$, this operation produces a pre-activation vector $\mathbf{h}'$:

$$\mathbf{h}' = \mathbf{f}_{\text{token}} W_1^T + b_1 \quad (14)$$

Activation. A GELU activation function (`nn.GELU`) is applied to the pre-activation vector $\mathbf{h}'$ to produce the activated intermediate representation $\mathbf{h}$:

$$\mathbf{h} = \text{GELU}(\mathbf{h}') \quad (15)$$

Layer 2 (Output). A final linear layer maps the intermediate representation $\mathbf{h}$ to a vector of N_L raw scores (logits), denoted as $\mathbf{z}$:

$$\mathbf{z} = \mathbf{h} W_2^T + b_2 \quad (16)$$

A Softmax function is subsequently applied to these logits to ensure the final weights $\mathbf{W}$ sum to one, forming a valid probability distribution:

$$\mathbf{W}_i = \text{Softmax}(\mathbf{z})_i = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{N_L} \exp(\mathbf{z}_j)} \quad (17)$$

B. Purple Flare Synthesis Dataset (PFSD)

Our Purple Flare Synthesis Dataset (PFSD) is the first large-scale, paired dataset constructed for this task. We first select diverse, clean image frames from the high-resolution video dataset DAVIS as our ground truth, denoted as $\mathbf{I}_{\text{GT}}$. We then generate purple flare input images $\mathbf{I}_{\text{flare}}$ using a physically motivated function, designed to mimic the characteristic appearance of purple flare in high-luminance regions and at object contour edges. This process is governed by a set of key hyperparameters, including the highlight percentile ρ_h , gradient threshold τ_g , edge width w_{edge} , blend strength α_s , and radial gamma γ . The specific generation process is as follows:

Dynamic Highlight Detection To identify potential flare regions, we first calculate a dynamic highlight threshold, τ_h , defined as the ρ_h -th percentile of the grayscale input image $\mathbf{I}_{\text{gray}}$. We then generate a binary highlight mask $\mathbf{M}_{\text{bright}}$ as follows:

$$\mathbf{M}_{\text{bright}} = (\mathbf{I}_{\text{gray}} > \tau_h). \quad (18)$$

Edge Candidate Identification We use a Sobel operator to compute the gradient magnitude G of the grayscale image. An edge mask $\mathbf{M}_{\text{edge}}$, is formed by thresholding the gradient magnitude with τ_g :

$$G = \sqrt{(\text{Sobel}_x * \mathbf{I}_{\text{gray}})^2 + (\text{Sobel}_y * \mathbf{I}_{\text{gray}})^2}, \quad (19)$$

where $\mathbf{M}_{\text{edge}} = (G > \tau_g)$.

Flare Mask Generation The initial region for the purple flare is identified as the intersection of highlight and edge regions, creating a candidate mask $\hat{\mathbf{M}}_{\text{flare}}$:

$$\hat{\mathbf{M}}_{\text{flare}} = \mathbf{M}_{\text{bright}} \cap \mathbf{M}_{\text{edge}}. \quad (20)$$

Flare Simulation To simulate the width and diffusion properties of flares, we expand the candidate mask $\hat{\mathbf{M}}_{\text{flare}}$ by applying morphological dilation with a structural element K_e , whose size is determined by the hyperparameter w_{edge} .

$$\mathbf{M}_{\text{dilated}} = \hat{\mathbf{M}}_{\text{flare}} \oplus K_e, \quad (21)$$

where $\oplus$ denotes morphological dilation.

$\mathbf{M}_{\text{dilated}}$ is convolved with a Gaussian kernel G_σ to create soft edges, generating the final flare band mask.

$$\mathbf{M}_{\text{flare}} = \mathbf{M}_{\text{dilated}} * G_\sigma, \quad (22)$$

where the standard deviation σ is proportional to w_{edge} .

Table 5: Key parameters for the synthetic purple flare generation pipeline. Default values are those used to generate our dataset.

Parameter	Description	Value
highlight_pct	Percentile for the dynamic highlight threshold.	99.0
grad_thresh	Gradient magnitude threshold for the Sobel edge detector.	25
edge_width	Width (in pixels) of the flare band.	80
strength	Global blending strength of the purple overlay.	0.7
gamma	Exponent for the radial mask to intensify the corner effect.	2.2

Spatially-Varying Alpha Mask To model the influence of lens geometry, where flares typically intensify toward the corners, we generate a radial attenuation mask $\mathbf{R}$.

$$\mathbf{R}(x, y) = \left(\frac{\text{dist}((x, y), \text{center})}{\text{dist}_{\max}} \right)^\gamma, \quad (23)$$

This mask $\mathbf{R}$ is combined with the normalized flare band $\mathbf{M}_{\text{flare}}$ and the blend strength α_s to create the final spatially-varying alpha mask α :

$$\alpha = \frac{\mathbf{M}_{\text{flare}}}{\max(\mathbf{M}_{\text{flare}})} \cdot \mathbf{R} \cdot \alpha_s. \quad (24)$$

Alpha Blending The final purple flare image $\mathbf{I}_{\text{flare}}$ is generated by alpha blending the ground truth image $\mathbf{I}_{\text{GT}}$ with a predefined purple color C_p using the generated alpha mask.

$$\mathbf{I}_{\text{flare}} = \mathbf{I}_{\text{GT}} \cdot (1 - \alpha) + C_p \cdot \alpha \quad (25)$$

The key parameters controlling the flare’s appearance are detailed in Table 5. The entire synthesis process is detailed in Algorithm 1. After generation, the dataset is divided into training, validation, and test sets. To prevent data leakage where frames from the same video sequence might appear in both training and evaluation splits, the division is performed at the scene level. The list of all scenes is randomly shuffled, and entire scenes are then assigned to the training, test, and validation sets, with proportions of 80%, 10%, and 10%, respectively.

Finally, the candidate mask $\hat{\mathbf{M}}_{\text{flare}}$ is directly used as the ground-truth mask $\mathbf{M}_{\text{GT}}$ for metric calculation. We select this explicit mask instead of using the diffuse final flare band mask $\mathbf{M}_{\text{flare}}$ for visual blending. This is because $\hat{\mathbf{M}}_{\text{flare}}$ enables accurate measurement of the model’s ability to restore core-damaged regions, avoiding penalties for failing to precisely reconstruct the feathered halo details of simulated flares. Let $\mathbf{I}_{\text{out}}$ be the model’s output image; the PSNR calculations for the flare region (PSNR-f) and non-flare region (PSNR-nf) are as follows:

$$\text{PSNR-F} = 10 \log_{10} \left(\frac{\text{MAX}_I^2 \cdot \sum \mathbf{M}_{\text{GT}}}{\sum (\mathbf{I}_{\text{out}} - \mathbf{I}_{\text{GT}})^2 \odot \mathbf{M}_{\text{GT}}} \right), \quad (26)$$

$$\text{PSNR-NF} = 10 \log_{10} \left(\frac{\text{MAX}_I^2 \cdot \sum (1 - \mathbf{M}_{\text{GT}})}{\sum (\mathbf{I}_{\text{out}} - \mathbf{I}_{\text{GT}})^2 \odot (1 - \mathbf{M}_{\text{GT}})} \right), \quad (27)$$

where $\odot$ is element-wise multiplication.

To specifically evaluate the chromatic accuracy of the restoration, we designed the Hue Alignment Error (HAE),

a metric more perceptually relevant than standard signal-based errors like PSNR. HAE is defined as the saturation-weighted average of the circular hue difference between the model’s output $\mathbf{I}_{\text{out}}$ and the ground-truth $\mathbf{I}_{\text{GT}}$, calculated exclusively within the regions of the original input image $\mathbf{I}_{\text{in}}$ affected by purple flare. A lower HAE score signifies a more accurate color restoration.

Algorithm 1: Parametric Purple Flare Synthesis

Input: Ground truth image $\mathbf{I}_{\text{GT}}$.

Parameters: Highlight percentile ρ_h , gradient threshold τ_g , edge width w_{edge} , blend strength α_s , radial gamma γ .

Output: Purple flare image $\mathbf{I}_{\text{flare}}$ and ground-truth mask $\mathbf{M}_{\text{GT}}$.

$\mathbf{I}_{\text{gray}} \leftarrow \text{Grayscale}(\mathbf{I}_{\text{GT}})$

$\tau_h \leftarrow \text{Percentile}(\mathbf{I}_{\text{gray}}, \rho_h)$

$\mathbf{M}_{\text{bright}} \leftarrow (\mathbf{I}_{\text{gray}} > \tau_h)$

if $\text{sum}(\mathbf{M}_{\text{bright}}) = 0$ **then return** None, None

end if

$G \leftarrow \text{SobelMagnitude}(\mathbf{I}_{\text{gray}})$

$\mathbf{M}_{\text{edge}} \leftarrow (G > \tau_g)$

$\hat{\mathbf{M}}_{\text{flare}} \leftarrow \mathbf{M}_{\text{bright}} \cap \mathbf{M}_{\text{edge}}$

if $\text{sum}(\hat{\mathbf{M}}_{\text{flare}}) = 0$ **then return** None, None

end if

$\mathbf{M}_{\text{GT}} \leftarrow \hat{\mathbf{M}}_{\text{flare}}$

$K_e \leftarrow \text{GetStructuringElement}(\text{shape}=\text{ELLIPSE}, \text{size} = w_{\text{edge}})$

$\mathbf{M}_{\text{dilated}} \leftarrow \text{Dilate}(\hat{\mathbf{M}}_{\text{flare}}, K_e)$

$\mathbf{M}_{\text{flare}} \leftarrow \text{GaussianBlur}(\mathbf{M}_{\text{dilated}}, \sigma = 0.6 \cdot w_{\text{edge}})$

$R \leftarrow \text{RadialFalloffMask}(\text{shape of } \mathbf{I}_{\text{GT}}, \gamma)$

$\alpha \leftarrow \frac{\mathbf{M}_{\text{flare}}}{\max(\mathbf{M}_{\text{flare}})} \cdot \mathbf{R} \cdot \alpha_s$

Let C_p be the target purple color (e.g., [255, 100, 255] in BGR)

$\mathbf{I}_{\text{flare}} \leftarrow \mathbf{I}_{\text{GT}} \cdot (1 - \alpha) + C_p \cdot \alpha$

return $\text{clip}(\mathbf{I}_{\text{flare}}, 0, 255), \mathbf{M}_{\text{GT}}$

A binary flare mask $\mathbf{M}_{\text{flare}}$ is generated from the input image to precisely identify the location of purple halos. This mask is used to locate pixels that satisfy two conditions: being on high-contrast edges detected by the Sobel filter, while exhibiting hues in the purple spectrum (e.g., 260°-340°) and significant saturation in the HSV color space. The specific calculation process is as follows: first, both the output image and the ground-truth image are converted to the HSV color space to separate hue, saturation, and brightness infor-

mation. Then, we extract the hue channel $\mathbf{H}_{\text{out}}$ of the output image, as well as the hue $\mathbf{H}_{\text{GT}}$ and saturation $\mathbf{S}_{\text{GT}}$ channels of the ground-truth image. Within the region defined by the mask, the circular hue difference is calculated pixel by pixel.

$$\Delta H = \min(|\mathbf{H}_{\text{out}} - \mathbf{H}_{\text{GT}}|, 360 - |\mathbf{H}_{\text{out}} - \mathbf{H}_{\text{GT}}|), \quad (28)$$

which correctly measures the shortest distance on the 360-degree hue circle. To align with human perception, where hue errors are more visible in vibrant colors, each ΔH value is then weighted by the saturation of the corresponding pixel in the ground-truth image, $\mathbf{S}_{\text{GT}}$. The final HAE score is the average of these saturation-weighted errors, defined as:

$$\text{HAE} = \frac{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathbf{M}_{\text{flare}}} \Delta H(\mathbf{x}, \mathbf{y}) \cdot \mathbf{S}_{\text{GT}}(\mathbf{x}, \mathbf{y})}{\sum_{(\mathbf{x}, \mathbf{y}) \in \mathbf{M}_{\text{flare}}} \mathbf{S}_{\text{GT}}(\mathbf{x}, \mathbf{y}) + \epsilon}, \quad (29)$$

where ϵ is a small constant to prevent division by zero.

C. Additional Ablation Analysis

C.1. Impact of Codebook Size in the VQ Module

The VQ layer is the core of the CAST module, and its codebook size directly affects the expressive power of the semantic tokens. In Table 6 and Figure 6, we study the impact of different codebook sizes on model performance. The results show that a codebook that is too small (e.g., 1024) cannot learn sufficiently rich semantic patterns, leading to a performance drop. Conversely, an overly large codebook (e.g., 8192) increases model parameters and the risk of overfitting without bringing significant performance gains. A codebook size of 4096 achieves the best balance between performance and model complexity.

Table 6: Visualization of Ablation study on different codebook sizes.

Codebook Size	FLOPs (G) ↓	PSNR ↑	HAE ↓
1024	19.45	33.52	6.83
2048	21.69	34.15	5.71
4096 (Ours)	23.32	34.96	4.10
8192	26.05	35.03	4.15

C.2. Impact of Encoder/Decoder Depth

The depth of the encoder and decoder in the CAST module determines the hierarchy of feature extraction. The results show in Table 7 and visualized in Figure 10. The experiments prove that a network that is too shallow (2 layers) cannot extract robust high-level features, resulting in poor performance. While deepening the network to 6 layers brings slight performance improvements, it also significantly increases the computational cost. Therefore, a 4-layer depth is the optimal configuration we chose, striking a good balance between feature extraction capability and model efficiency.

D. More Qualitative Results

In this section, we present extensive qualitative results to intuitively demonstrate the robustness and superiority of our

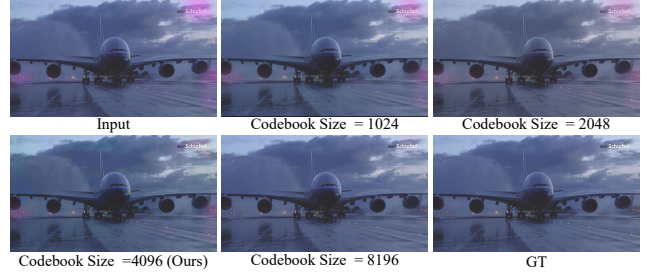


Figure 9: Ablation study on different codebook sizes.

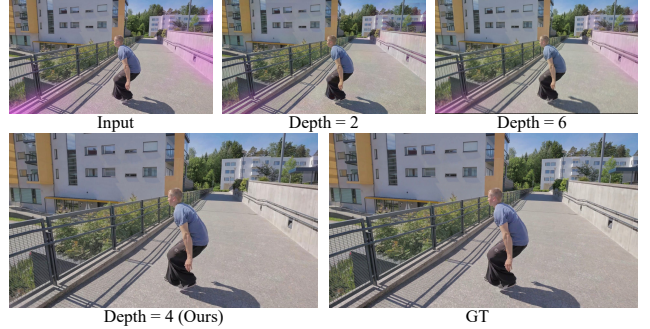


Figure 10: Visualization of the Ablation Study on Encoder/Decoder Depth.

Table 7: Ablation study on encoder/decoder depth.

Depth (Conv Layers)	PSNR ↑	HAE ↓
2	32.89	5.31
4 (Ours)	34.96	4.10
6	34.91	4.19

proposed method. The comparisons are divided into two categories: results from complex daylight scenes and challenging night scenes in the PFSD dataset, as well as results from real-world images used to test generalization ability.

D.1. Additional Qualitative Comparisons on the PFSD Dataset

To further demonstrate the comprehensive performance of the proposed method, additional comparisons with state-of-the-art approaches were conducted across various scenarios in the PFSD test set. These scenarios range from daylight scenes with severe backlighting and complex textures (Figs. 11 and 12) to challenging night scenes (Figs. 13).

The results highlight the superiority of our method. It not only removes purple flares more effectively but also excels in preserving details and maintaining color fidelity in non-flare regions. In night scene scenarios, while other methods often leave residual artifacts or introduce unnatural color changes, the proposed method successfully corrects the images.

D.2. Qualitative Comparisons on Real-World Images

To verify the practical applicability and generalization ability of the model, we tested it on real-world images not included in the PFSD dataset. As shown in Figs. 14, 15, and 16, the comparisons demonstrate that the proposed model effectively generalizes to real-world artifacts different from synthetic data, successfully removing purple flares while preserving the authenticity of the original photos. However, we note that in some rare and extreme cases, such as when the flare’s spectrum significantly deviates from typical purple flares (atypical colors, see the second and third examples in Fig. 16) or when the flare is particularly large (see the first example in Fig. 14), our model may fail to achieve perfect removal.

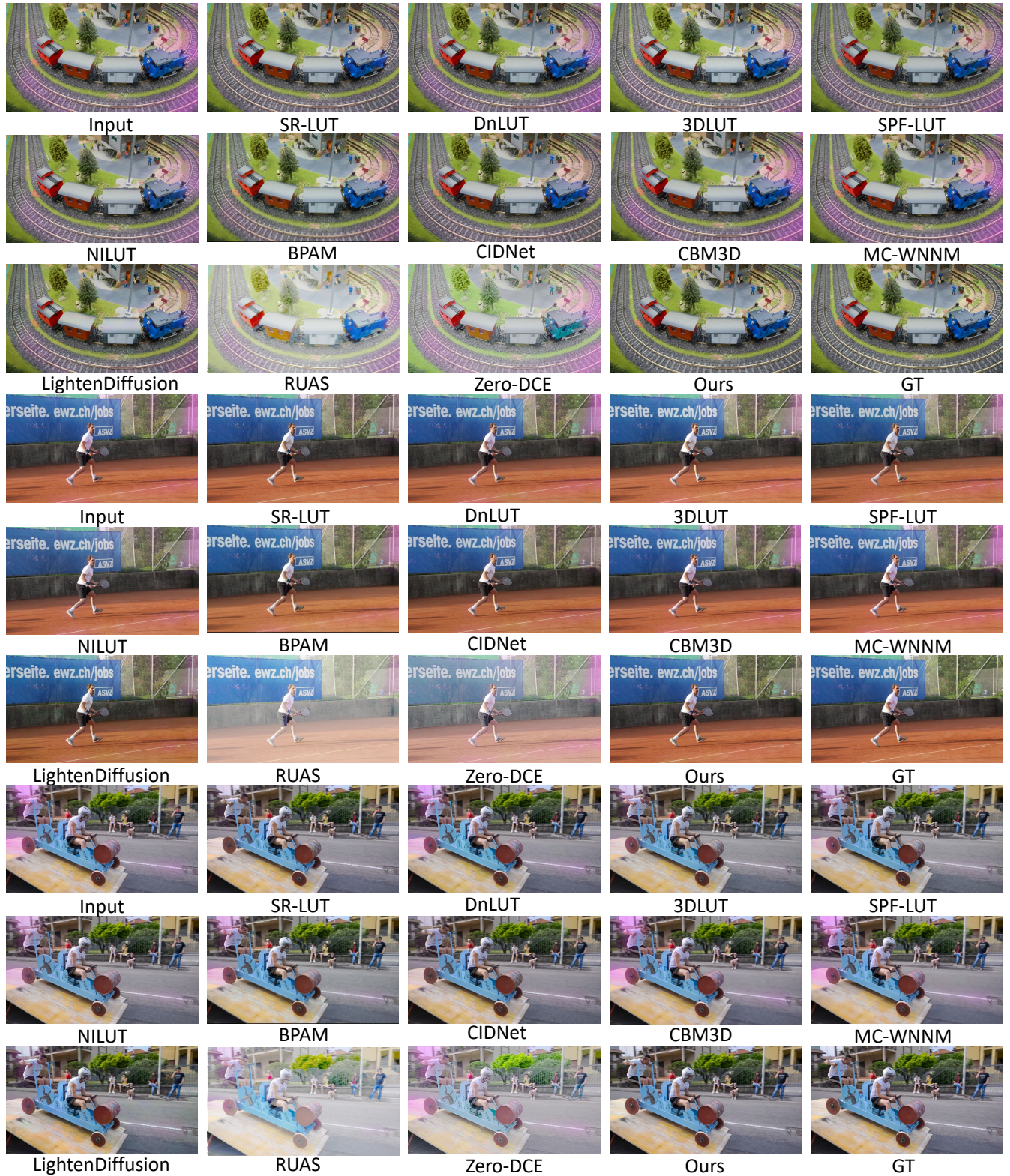


Figure 11: Visual comparison of scenes with complex textures from the PFSD dataset.. While removing purple flares, our method accurately preserves the original textures and gloss.

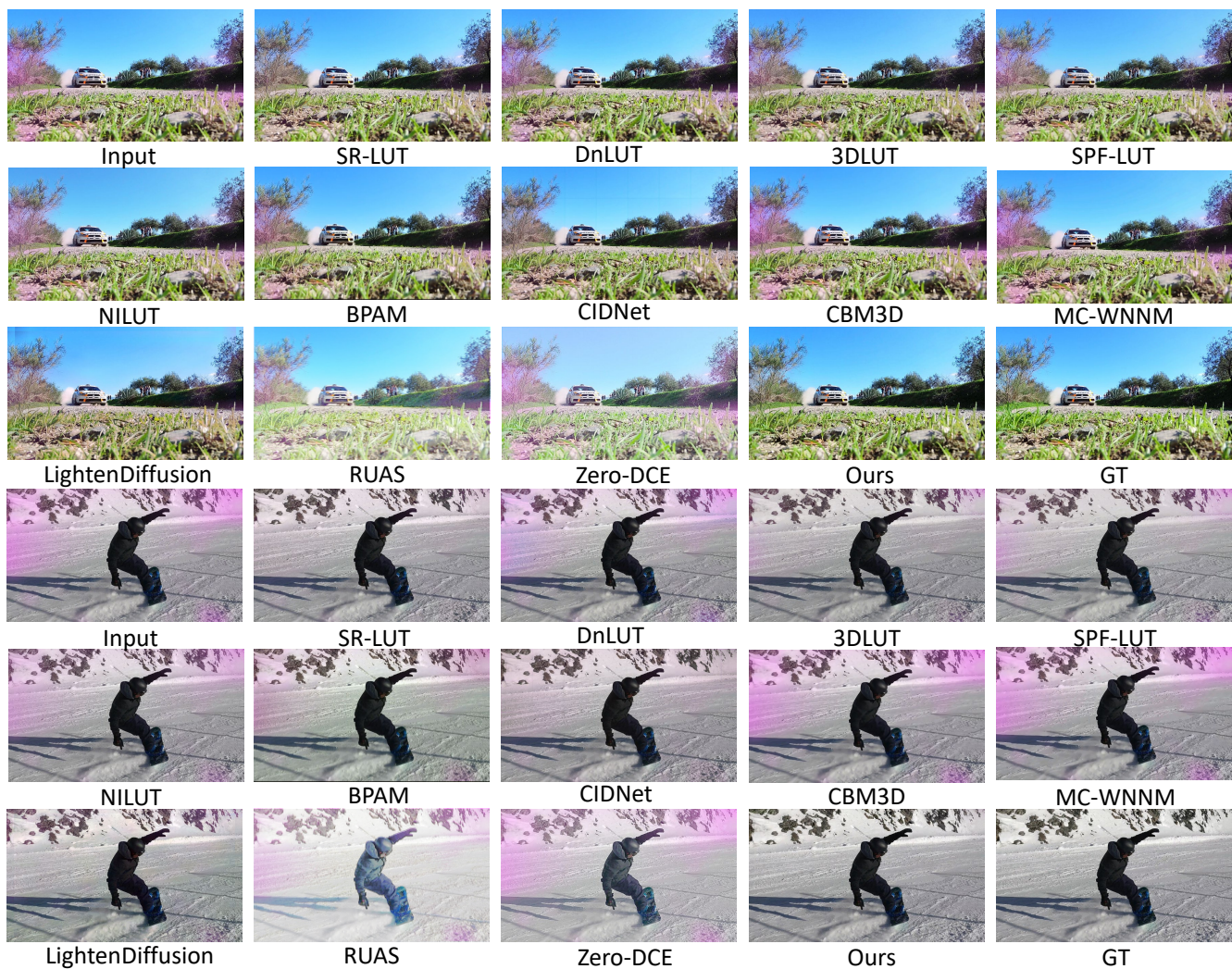


Figure 12: Visual comparison on backlit scenes from the PFSD dataset.. Our method excels at handling the severe purple flare at the boundary between the sky and trees, restoring a natural color transition.



Figure 13: Visual comparison of all methods on a challenging night scene from the PFSD dataset. Our method effectively removes the purple flare around the lights while preserving details in the dark areas.

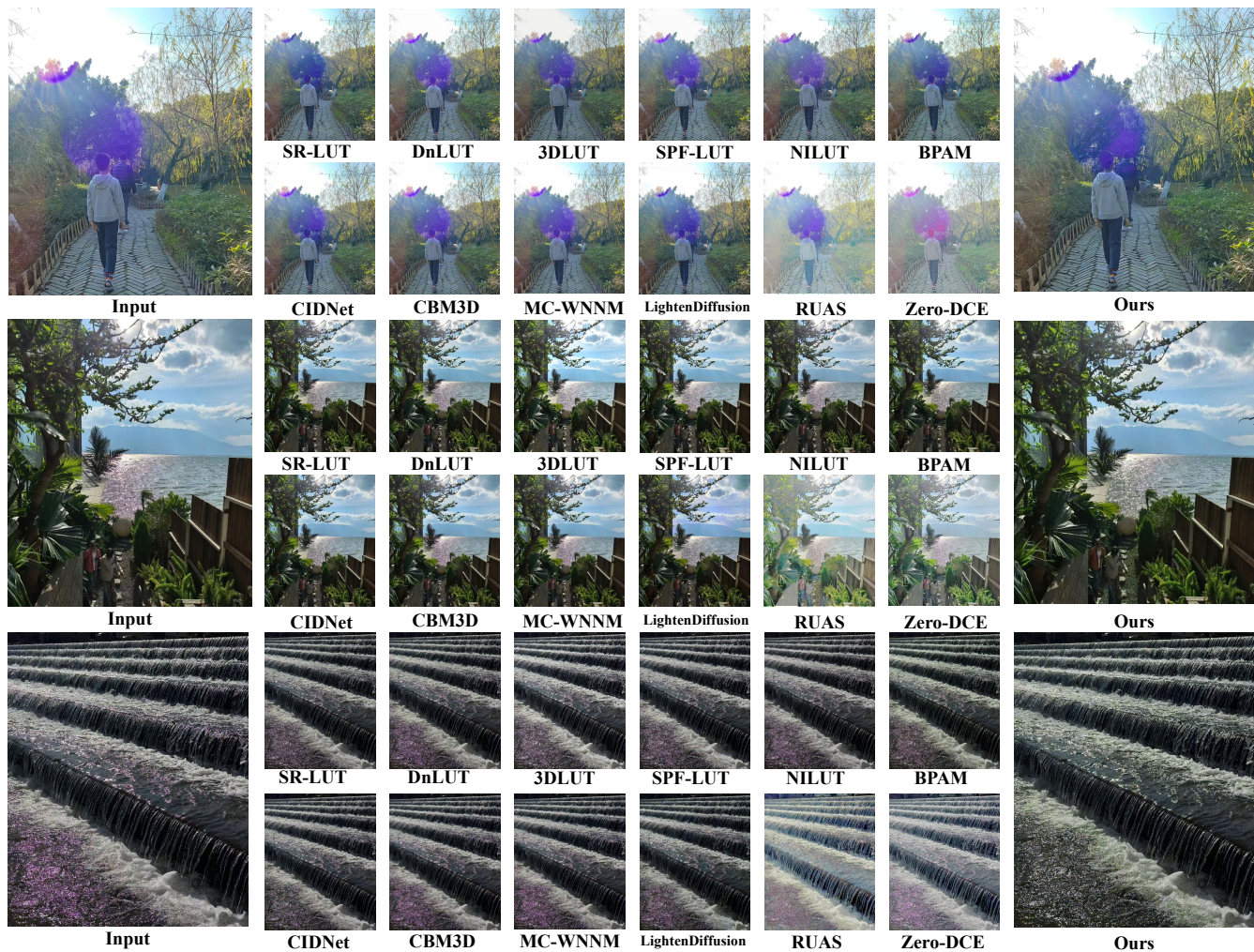


Figure 14: Visual comparison of all methods on a real-world photograph with purple flare.

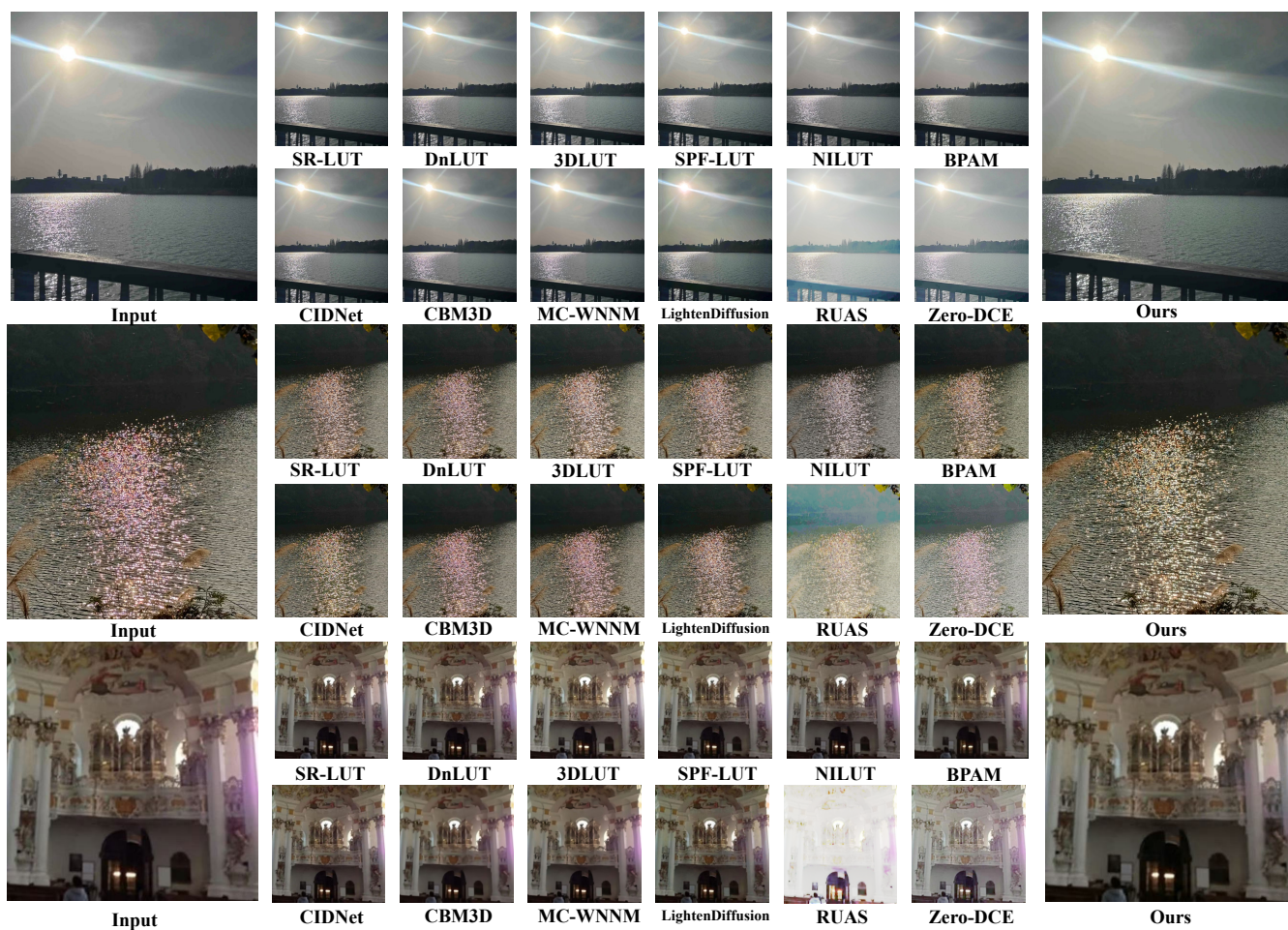


Figure 15: Visual comparison of all methods on a real-world photograph with purple flare.

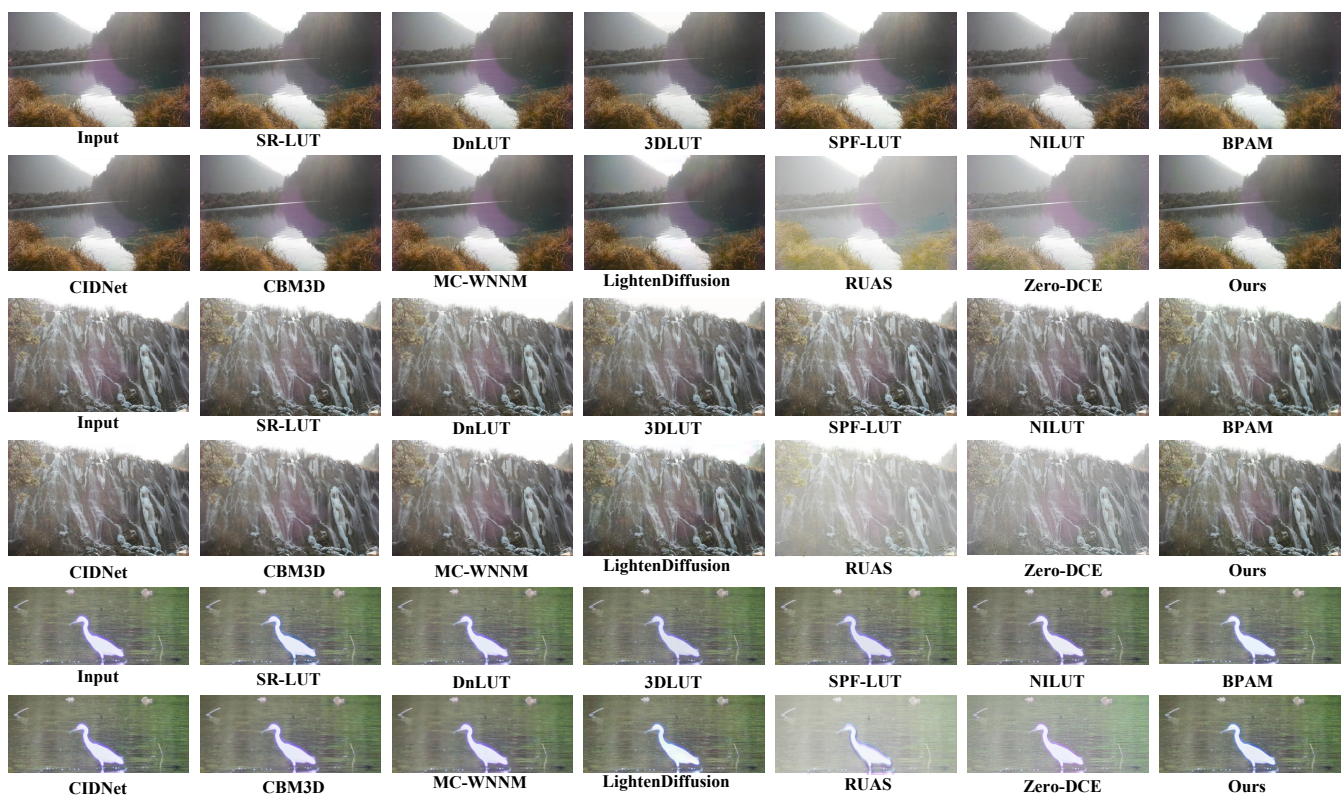


Figure 16: Visual comparison of all methods on a real-world photograph with purple flare.