Language-Guided One-Step Diffusion Model for Nighttime Flare Removal

Supplementary Material

Abstract of Supplementary Material

This appendix provides more implementation details (Sec. 7), more ablation experiments (Sec. 8), the training algorithm for stage 1 of LG-ODM (Sec. 9), additional details on instruction generation in the synthetic data pipeline (Sec. 10), more comparisons with existing datasets (Sec. 11), user study (Sec. 12), and qualitative on real-world scenes with different training datasets. (Sec. 13).

7. More Implementation Details

Training Flare-VLM. We fine-tune Qwen2.5-VL-7B [1] on our two constructed instruction datasets to obtain Flare-VLM. The model is trained in a multi-task manner: given a flare-corrupted image, it is supervised to produce a structured textual description of the observed flare; given a flare-free image and its depth map, it is supervised to generate an edit instruction specifying the flare to be added. We adopt LoRA-based fine-tuning with rank = 8, learning rate = 1×10^{-4}, and train for 500 epochs.

8. More Ablation Experiments

Setting of SADD hyperparameters. As shown in Tab. 5, the setting $(\gamma_f, \gamma_b) = (0.10, -0.30)$ yields higher PSNRin / PSNRout and SSIMin / SSIMout than $(0.05, -0.10)$, indicating that moderately stronger perturbation in flare regions combined with background suppression is beneficial for both in-mask restoration and overall image quality. In contrast, the more aggressive configuration $(0.20, -0.50)$ does not further improve PSNRin or SSIMin and leads to a clear drop in PSNRout and SSIMout. We therefore adopt $(\gamma_f, \gamma_b) = (0.10, -0.30)$ as the default trade-off.

Tab. 6 evaluates the effect of the timestep sequence length s in SADD on reconstruction quality. Increasing s from 2 to 4 improves the quality, indicating that a moderate increase in sequence length helps the student better align its denoising trajectory with the teacher. Further increasing to $s = 8$ yields no additional gains. Since the computational and memory costs grow approximately linearly with the sequence length, we adopt $s = 4$ as a balanced trade-off between accuracy and efficiency.

Effectiveness of Different Loss. As shown in Tab. 7, removing the global reconstruction loss $\mathcal{L}_{rec}$ leads to the most pronounced degradation in structural similarity and perceptual quality, indicating that this term plays a key role in enforcing overall reconstruction fidelity. The regularizer loss $\mathcal{L}_{diff}$ promotes alignment between the teacher and student distributions, driving predictions toward more realistic scenes. In contrast, removing the light source loss $\mathcal{L}_{light}$ mainly weakens the restoration of light source regions and thus yields only a mild but consistent drop in all quantitative metrics.

Table 5. Comparison of SADD hyperparameters γ_f and γ_b. "in" and "out" denote evaluation inside and outside the flare mask, respectively.

γ_f	γ_b	$PSNR_{in}\uparrow$	$PSNR_{out}\uparrow$	$SSIM_{in}\uparrow$	$SSIM_{out}\uparrow$
0.05	−0.10	31.620	30.951	0.959	0.939
0.10	−0.30	31.918	30.984	0.962	0.942
0.20	−0.50	31.893	30.880	0.960	0.938

Table 6. Comparison of the timestep sequence length s in SADD.

s	PSNR↑	SSIM↑	LPIPS↓	FID↓
2	27.922	0.907	0.0693	36.744
4	28.061	0.915	0.0625	34.206
8	28.062	0.915	0.0625	34.208

Table 7. Comparison of different losses.

Loss	PSNR↑	SSIM↑	LPIPS↓	FID↓
w/o $\mathcal{L}_{diff}$	27.670	0.908	0.0644	36.120
w/o $\mathcal{L}_{rec}$	27.146	0.893	0.0722	38.669
w/o $\mathcal{L}_{light}$	27.954	0.913	0.0641	34.850
Full	28.061	0.915	0.0625	34.206

Effectiveness of Different Prompting Strategies. We evaluated the impact of different prompting strategies, including no descriptions, generic captions, structured prompts generated by a generic VLM, and the flare-specific prompts introduced by our method, as shown in Tab. 8.

As requested, we also randomly remove semantic slots (Ours*), which leads to only minor degradation.

Table 8. Effectiveness of different prompting strategies.

Prompt Setting	None	Caption	Structure	Ours*	Ours
PSNR↑	27.595	27.621	27.633	27.973	28.061
SSIM↑	0.870	0.872	0.875	0.910	0.915

9. Algorithm of LG-ODM

The pseudo-code of our LG-ODM training algorithm is summarized as Algorithm 1.

Algorithm 1 Training scheme for Stage 1 of LG-ODM

Input: Training set $\mathcal{S}$, pretrained SD parameterized by ψ including VAE encoder E_ψ, latent diffusion network ϵ_ψ and VAE decoder D_ψ, flare description generator Flare-VLM, text inversion LLM, CLIPSeg, TextEncoder, light source mask $\mathbf{M}_{\text{Light}}$, fixed timestep T in the one-step denoiser, training iterations N, noise schedule $\{\alpha_t, \bar{\alpha}_t\}_{t=1}^{T}$, SADD hyperparameters $(\gamma_{\text{f}}, \gamma_{\text{b}}, s)$

Initialize G_θ parameterized by θ, including
  $\mathrm{E}_\theta \leftarrow \mathrm{E}_\psi$ with trainable LoRA
  $\epsilon_\theta \leftarrow \epsilon_\psi$ with trainable LoRA
  $\mathrm{D}_\theta \leftarrow \mathrm{D}_\psi$ with trainable LoRA
Initialize regularizer $\epsilon_\phi \leftarrow \epsilon_\psi$ with trainable LoRA
for $i \leftarrow 1$ to N **do**
  Sample $(\mathbf{I}_{LQ}, \mathbf{I}_{GT})$ from $\mathcal{S}$
  /* Language-guided conditioning */
  $\mathbf{P}^{\text{f}} \leftarrow$ Flare-VLM$(\mathbf{I}_{LQ})$
  $\mathbf{P}^{\text{c}} \leftarrow$ LLM$(\mathbf{P}^{\text{f}})$
  $\boldsymbol{c} \leftarrow$ TextEncoder$(\mathbf{P}^{\text{c}})$
  /* Network forward */
  $\hat{\mathbf{Z}}, \mathbf{Z}_0 \leftarrow \mathrm{E}_\theta(\mathbf{I}_{LQ}), \mathrm{E}_\theta(\mathbf{I}_{GT})$
  $\varepsilon \leftarrow \epsilon_\theta(\hat{\mathbf{Z}}, \boldsymbol{c}, \mathrm{T})$
  $\hat{\mathbf{Z}}_0 \leftarrow \big(\hat{\mathbf{Z}} - \sqrt{1-\bar{\alpha}_{\mathrm{T}}}\,\varepsilon\big)/\sqrt{\bar{\alpha}_{\mathrm{T}}}$
  $\hat{\mathbf{I}} \leftarrow \mathrm{D}_\theta(\hat{\mathbf{Z}}_0)$
  /* Reconstruction and light source losses */
  $\mathcal{L}_{\text{rec}} \leftarrow \text{LPIPS}(\hat{\mathbf{I}}, \mathbf{I}_{GT})$
  $\mathcal{L}_{\text{light}} \leftarrow \text{LPIPS}(\mathbf{M}_{\text{Light}} \odot \hat{\mathbf{I}}, \mathbf{M}_{\text{Light}} \odot \mathbf{I}_{GT})$
  $\nabla_\theta \mathcal{L}_{\text{data}} \leftarrow [\mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{light}}] \frac{\partial \hat{\mathbf{I}}}{\partial \theta}$
  /* Semantics-Aware Distribution Distillation */
  $\mathbf{M} \leftarrow \text{CLIPSeg}(\mathbf{I}_{LQ}, \mathbf{P}^{\text{f}})$
  $\sigma(\mathbf{M}) \leftarrow (1+\gamma_{\text{f}})\mathbf{M} + (1+\gamma_{\text{b}})(1-\mathbf{M})$
  Sample t from $\{1, \ldots, T\}$, $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$
  $\hat{\mathbf{Z}}_t \leftarrow \sqrt{\bar{\alpha}_t}\hat{\mathbf{Z}}_0 + \sqrt{1-\bar{\alpha}_t}\big(\sigma(\mathbf{M}) \odot \varepsilon\big)$
  $\mathbf{Z}_t \leftarrow \sqrt{\bar{\alpha}_t}\mathbf{Z}_0 + \sqrt{1-\bar{\alpha}_t}\big(\sigma(\mathbf{M}) \odot \varepsilon\big)$
  $\mathcal{T} \leftarrow \{t-s, \ldots, t\}$
  Compute $\nabla_\theta \mathcal{L}_{\text{SADD}}(\hat{\mathbf{Z}}_0, \mathbf{Z}_0, \boldsymbol{c})$ according to Eq. (6) in main paper
  /* Regularizer finetuning objective */
  Sample $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$, t from $\{1, \ldots, T\}$
  $\mathcal{L}_{\text{diff}} = \mathcal{L}_{\text{MSE}}\big(\epsilon_\phi\big(\alpha_t \hat{\mathbf{Z}}_0 + \beta_t \varepsilon;\, t, \boldsymbol{c}\big), \varepsilon\big)$.
  /* Network parameter update */
  Update θ with $(\mathcal{L}_{\text{data}} + \mathcal{L}_{\text{SADD}})$
  Update ϕ with $\mathcal{L}_{\text{diff}}$
end for
Output: Generator G_θ including encoder E_θ, one-step denoiser ϵ_θ, decoder D_ψ

10. Additional Details on Instruction Generation in the Synthetic Data Pipeline

We first detect and select physically valid light source regions in each flare-free image $\mathbf{I}_{\text{c}}$. For each light source, we obtain its semantic category, consistent with the light source type used in the main text, and its location in the image plane, corresponding to its position. The position can be represented by partitioning the image into a regular grid and assigning each source to a coarse region, such as upper-left or center, or by converting its normalized image coordinates into a short textual description. Based on this information, we instantiate the instruction template: `[Add <flare shape> around <light source type, position>.]`.

For the geometric prior, we introduce two intermediate expressions to clarify the underlying optical structure. The complex pupil function of the imaging system is written as:

$$P_\lambda(u,v) = A(u,v)\,\mathrm{e}^{i\,\phi_\lambda(u,v)}, \tag{14}$$

where $A(u,v)$ is the pupil amplitude at pupil-plane coordinates (u,v) and $\phi_\lambda(u,v)$ is the phase for wavelength λ. The phase induced by a point light source at three-dimensional position (x,y,z) is decomposed as

$$\phi_\lambda(x,y,z) = \phi_\lambda^S\big(x/z,\, y/z\big) + \phi_\lambda^{DF}(z), \tag{15}$$

where ϕ_λ^S depends on the viewing direction and $\phi_\lambda^{DF}(z)$ depends only on the depth z. As z increases, the defocus term $\phi_\lambda^{DF}(z)$ introduces an additional radius-dependent phase on the pupil, which weakens coherent addition on the sensor and spreads the energy over a larger area. This optical behavior leads to the depth-dependent PSF_λ and the flare intensity $F_c(s,t;z)$ for channel c derived in the main text, together with the inverse-square falloff $\mathrm{E} \propto 1/z^2$.

To generates physics-guided prompts, we replace the scene depth by the per-pixel depth $\mathbf{D}(s,t)$, and precompute the depth-dependent PSF_c on a discrete set of depths using the same optical model as above. At a fixed reference location in the PSF, we read off a scalar value and store it in a lookup table indexed by depth. Combined with the binary light source mask $\mathbf{M}(s,t)$, we then compute $\mathrm{I}_c(s,t)$ according to the expression provided in the main text, so that the supervised flare intensity is nonzero only where $\mathbf{M}(s,t) = 1$.

To convert continuous intensities into discrete levels that can be expressed in text, we aggregate the three channel-wise maps $\mathrm{I}_c(s,t)$ over $c \in \{R,G,B\}$ within the light source regions defined by $\mathbf{M}(s,t)$ and obtain a single scalar intensity map. For each image, this scalar map is normalized to the range $[0,1]$ based on the minimum and maximum values of light source pixels, thereby reducing the influence of global exposure differences. The normalized values are then thresholded into five ordered

Figure 10. Comparison of flare synthesis results from different methods with Real-world references. The top row illustrates streak flares, while the bottom row demonstrates diffused glow effects. Our instruction-driven method generates flare effects that closely match real-world optical phenomena, exhibiting superior realism and contextual integration compared to existing approaches [9, 48].

Table 9. Qwen2.5-VL perceptual evaluation results. Score 1 measures optical accuracy and positioning, while Score 2 assesses naturalness and scene integration. Average denotes the overall perceptual quality.

Metric	Score 1↑	Score 2↑	Average↑
Wu et al.	2.77	2.26	2.52
Flare7K++	2.92	2.14	2.53
Ours	3.84	3.15	3.50

bins. For each light source, we aggregate the quantized labels over the connected region of $\mathbf{M}(s,t)$ associated with that source, for example, by taking the mode, and obtain a single discrete level `<level>`. Finally, this level is inserted into the extended instruction template: `[Add <flare shape> around <light source type, position>. The flare intensity is <level>.]`.

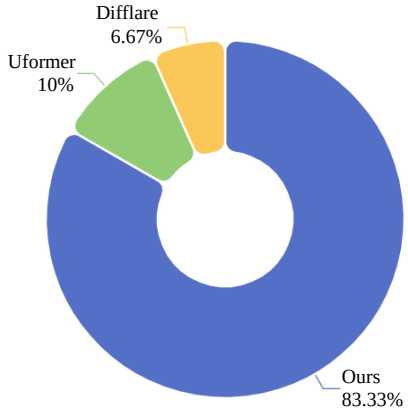


Figure 11. User preference pie chart. Our approach has garnered an 83.33% user satisfaction rating.

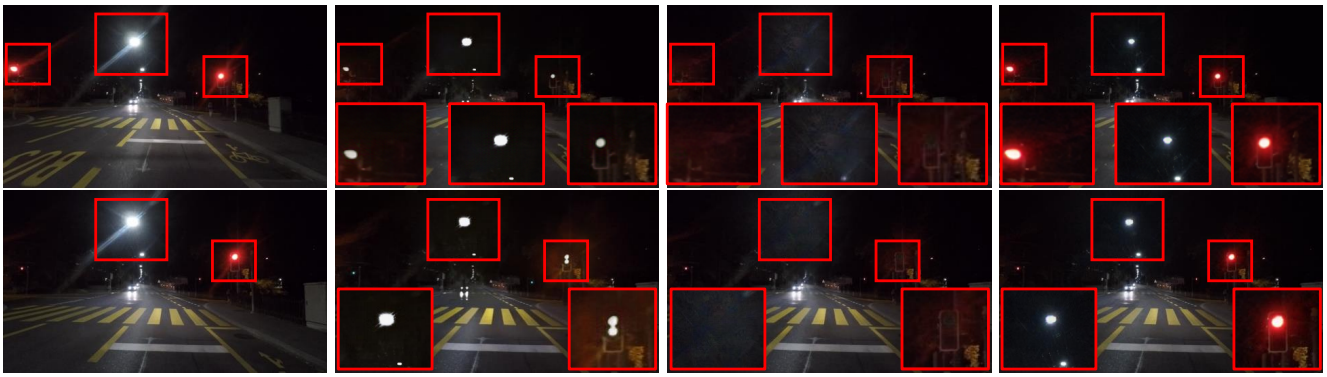
11. More Comparisons with Existing Datasets

Qualitative Comparison. In real-world nighttime scenes, flares are primarily categorized into streak flare and diffused glow. As shown in Fig. 10, existing methods exhibit apparent limitations in modeling these types of flares. Wu et al. [48] predominantly generate structured streak patterns but do not accurately capture the soft, volumetric scattering around light sources. In contrast, Flare7K++ [9] tends to produce extensive diffused glow, which introduces noticeable artifacts and reduces image fidelity. Moreover, neither baseline explicitly leverages the semantic and geometric information of the background image, leading to discrepancies between the synthesized flares and real degradation patterns. Our method addresses these issues by synthesizing both streak flare and diffused glow with more consistent internal structure and physically motivated behavior.

Evaluation Based on Vision-Language Model. We employ Qwen2.5-VL 7B [1] as an automatic perceptual evaluator of the synthesized images, following recent practices that use vision–language models as surrogate human raters. For each test sample, we provide the model with the flare-free reference image, the synthesized image, and a textual instruction, and ask it to assign two scores on a 1–5 scale, where larger values indicate better perceptual quality.par Score 1 (optical accuracy and positioning): evaluates whether the flare is correctly aligned with the light source, whether its brightness decays naturally from the center, and whether its shape, orientation, size, and intensity are consistent with basic physical principles.

Score 2 (naturalness and scene integration): evaluates whether the flare blends plausibly with the scene, whether the overall image appears realistic and coherent, whether unmodified regions remain intact, and whether the flare avoids an obviously artificial appearance.

As shown in Tab. 9, our method attains higher scores than existing baselines in both optical accuracy and scene integration, thereby improving the realism and physical consistency of the synthesized flares.



Input Wu et al. [48] Flare7k++ [9] Our data

Figure 12. Comparison of flare removal results on our DarkZurich test dataset using models trained on different datasets.

Input Wu et al. [48] Flare7k++ [9] Our data

Figure 13. Comparison of flare removal results on our ExDark test dataset using models trained on different datasets.

12. User Study

We conducted a user study to further assess the perceptual quality of flare removal. Our method was compared against two representative baselines of different model types, Uformer [45] and Difflare [57]. To obtain a diverse evaluation set, we selected nighttime images containing various flare patterns and intensities. For each test case, the ground-truth image and the three restored results were displayed simultaneously at the same resolution on the screen. The three methods were anonymized, and their spatial positions were randomly permuted for each trial to reduce positional and labeling bias. Participants were instructed to choose the restoration that best matched the ground truth in terms of flare suppression and preservation of scene details. A total of 30 participants took part in the study. As summarized in Fig. 11, our method receives 83.33% of the votes under this protocol, indicating that it is most frequently perceived as producing the closest reconstruction to the reference images.

13. Qualitative on Real-world Scenes with Different Training Datasets.

Fig. 12 and Fig. 13 show flare removal results on real nighttime scenes obtained with models trained on different flare removal datasets. Across both evaluation sets, the model trained on our dataset achieves more consistent suppression of streak flare and diffuse glow around strong light sources while preserving lane markings, object contours, and other scene structures. In contrast, models trained on Wu et al. [48] and Flare7K++ [9] often damage the light sources and surrounding scene structures.