# Unaligned Image Guided Denoising via Cross-modal Conditional Flow Matching

**Runmin Zhang**[1*] **Linshan Li**[1*] **Si-Yuan Cao**[1] **Zhu Yu**[1]
**Lingyu Zhu**[2] **Huaqi Zhang**[3] **Hui-Liang Shen**[1†]
[1]Zhejiang University [2]City University of Hong Kong [3]vivo Mobile Communication Co., Ltd.
{runmin_zhang, linshan_li, cao_siyuan, yu_zhu}@zju.edu.cn
lingyzhu-c@my.cityu.edu.hk, zhanghuaqi@vivo.com, shenhl@zju.edu.cn

## Abstract

This work presents UGD-FM, a novel unaligned image guided denoising framework based on cross-modal conditional flow matching. Unlike previous approaches that assume spatially aligned inputs or only handle small-baseline rectified pairs, we address a more challenging setting where severe target noise and large-range two-dimensional cross-modal misalignment coexist. Specifically, we formulate guided denoising as a progressive flow matching process, allowing image restoration and aligned guidance aggregation to mutually reinforce each other. To support efficient few-step inference, we introduce a stage-focused and inference-consistent training strategy that focuses on task-critical denoising stages, and mitigates the training-inference mismatch. We further propose adaptive sparse guidance propagation for efficient and reliable large-range matching, regularized by perceptual and geometric auxiliary losses. Experiments demonstrate that UGD-FM achieves state-of-the-art denoising performance with only one or two inference steps, and is compatible with modern single-image restoration backbones. Code is available at https://github.com/RM-Zhang/UGD-FM.

## 1 Introduction

Image denoising is a fundamental problem in low-level vision. Although modern restoration networks [47, 39, 46, 4, 48] have substantially improved denoising performance, faithfully recovering fine structures from heavily corrupted observations without over-smoothing remains challenging. To alleviate this limitation, recent works [6, 10, 30, 44, 36, 31, 51, 37] have leveraged auxiliary guidance images, such as near-infrared (NIR) images, which provide cleaner structural cues for denoising.

However, most guided denoising approaches [6, 30, 44, 36, 37] assume that the noisy RGB image and the guidance image are spatially aligned. This assumption is difficult to satisfy in practical mobile imaging systems, where cross-modal images are typically captured by different cameras [53, 33]. Recent approaches [31, 51] have extended guided denoising to cross-modal stereo pairs, but remain limited to horizontally rectified pairs with small baselines. In contrast, we study unaligned guided denoising under **more severe spatial misalignment** than the small-baseline rectified stereo setting considered in prior work. This expands the correspondence search from a one-dimensional horizontal search to a two-dimensional search space [54, 28, 50], making reliable cross-modal matching and guidance extraction significantly more difficult. This challenge is further amplified by severe noise in the target RGB image and inherent cross-modal appearance differences.

To address these challenges, we propose UGD-FM, a novel framework for **U**naligned image **G**uided **D**enoising via cross-modal conditional **F**low **M**atching. Specifically, we formulate guided denoising

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

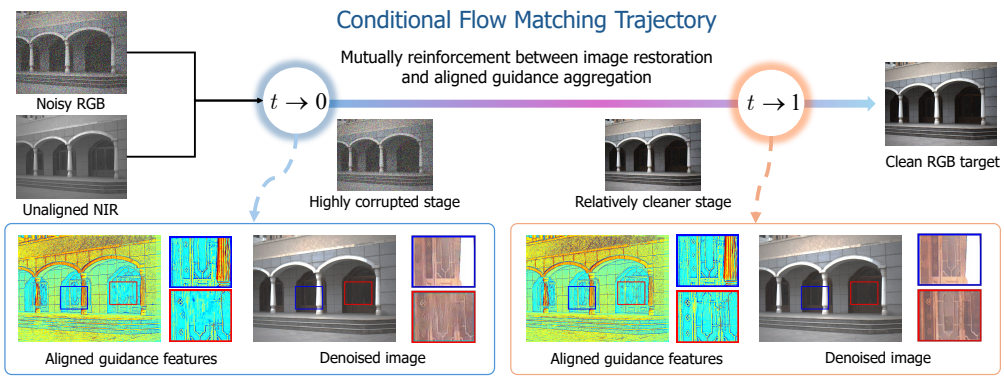


Figure 1: Illustration of the cross-modal conditional flow matching trajectory of UGD-FM. Given a noisy RGB image and an unaligned NIR guidance image, UGD-FM progressively restores the RGB state toward the clean target, forming a mutual reinforcing process between image restoration and guidance aggregation. For clearer visualization, the brightness of the zoom-in patches is enhanced.

as a conditional flow matching process that progressively transports the noisy RGB image distribution toward the clean RGB image distribution, conditioned on unaligned cross-modal guidance. As the transport process progresses, the target image becomes cleaner, making cross-modal correspondence easier to establish and enabling more effective use of the guidance information. As illustrated in Fig. 1, this creates a mutually reinforcing process between guidance extraction and image restoration, leading to more robust denoising under severe spatial misalignment and noise.

To enable efficient and effective few-step guided denoising, we further design a **stage-focused and inference-consistent training strategy**. Our key observation is that the denoising trajectory contains two critical stages, *i.e.*, a **highly corrupted stage** for coarse RGB structure recovery, and a **relatively cleaner stage** for reliable cross-modal matching and fine-detail restoration. Based on this observation, we place the inference timesteps around these stages, and align training with this schedule through stage-focused time sampling and inference-consistent state construction. With this design, UGD-FM delivers strong performance with only a single inference step, and achieves state-of-the-art results with two-step inference.

As for cross-modal matching and guidance extraction, a straightforward solution is to perform dense correspondence search over a large displacement range [31, 51]. However, such a dense matching strategy incurs prohibitive computational cost. To mitigate this issue, we introduce an **adaptive sparse guidance propagation** mechanism. Specifically, our method progressively focuses on a small set of high-confidence correspondence candidates, and propagates reliable guidance cues in a coarse-to-fine manner. To stabilize sparse guidance aggregation, we further introduce two auxiliary losses that provide perceptual and geometric regularization.

Importantly, we highlight that both our adaptive sparse guidance propagation mechanism and conditional flow matching formulation are compatible with a broad range of modern single-image restoration backbones, such as NAFNet [4], Uformer [39], and Restormer [46]. Our contributions are summarized as follows:

- We propose UGD-FM, a general cross-modal conditional flow matching framework for unaligned image guided denoising. UGD-FM establishes a mutually reinforcing process between image restoration and guidance aggregation, enabling robust restoration under severe noise and large-range spatial misalignment.
- We design a stage-focused and inference-consistent training strategy for few-step inference. It focuses inference and training on two critical denoising stages, and mitigates the training-inference mismatch, enabling effective one-step and two-step guided denoising.
- We introduce adaptive sparse guidance propagation for efficient large-range cross-modal matching and aligned guidance aggregation, regularized by perceptual and geometric auxiliary losses to stabilize sparse correspondence learning.
- Comprehensive experiments demonstrate that UGD-FM achieves state-of-the-art denoising performance, and is compatible with modern single-image restoration backbones.

## 2 Related Work

**Single Image Restoration.** Single image restoration aims to recover clean images from degraded observations. Recently, deep convolutional neural networks have become a dominant paradigm by learning mappings from degraded inputs to clean targets. Representative designs include residual learning [49], multi-stage refinement [47], and inter-/intra-block architectures [4]. Vision transformers and their variants [3, 19, 39, 46, 48] further improve restoration performance by modeling long-range context. Despite these advances, single image restoration remains fundamentally ill-posed under severe degradation, often leading to over-smoothed results and missing structural details.

**Guided Image Restoration.** Guided image restoration leverages guidance images from auxiliary modalities to provide complementary structural information, with NIR-guided RGB denoising being a representative application. Early methods [9, 18, 26] employ joint filtering to model structural relationships between target and guidance images. Further approaches improve denoising performance with convolutional sparse coding [6], ADMM-based optimization [44], frequency decomposition [30, 23, 37], and multi-modal inconsistency priors [10, 36, 43]. Nevertheless, most of these approaches assume that the target and guidance images are strictly aligned. This requirement usually depends on specialized imaging systems or careful calibration [8, 31, 12], making it difficult to satisfy in practical mobile imaging systems. Recent approaches [32, 42, 52, 5, 31, 51] start to relax the strict alignment assumption. However, they either depend on external optical flow networks for pre-alignment, or handle limited misalignment. Other approaches [20, 21, 13, 16, 17] handle infrared and visible image fusion, in which SFRF [17] introduces multi-scale iterative registration. In contrast, we address a more challenging setting, where severe target noise and large-range cross-modal displacement must be addressed jointly.

**Generative Models for Image Restoration.** Generative priors, particularly diffusion [5, 55, 34, 15] and flow matching [24, 40, 27] models, have demonstrated strong capabilities in image restoration by learning to transport a simple base distribution to the data manifold. Recent studies [2, 45, 7] further show that initializing from low-quality observations, rather than Gaussian noise, reduces the transport distance and improves structural fidelity by anchoring the generation process to the input.

## 3 Method

Given a noisy RGB image $\mathbf{I}_{\mathrm{N}} \in \mathbb{R}^{H\times W\times 3}$ and an unaligned NIR guidance image $\mathbf{I}_{\mathrm{G}} \in \mathbb{R}^{H\times W}$ captured by different cameras, our goal is to recover the clean RGB image $\mathbf{I}_{\mathrm{T}} \in \mathbb{R}^{H\times W\times 3}$. The spatial misalignment induces cross-modal displacements spanning a 2D range $[-D, D]^2$, requiring joint image restoration and correspondence estimation under severe noise.

To tackle this challenge, we propose UGD-FM, an unaligned image guided denoising framework based on cross-modal conditional flow matching. We first introduce the **conditional flow formulation** and **overall network architecture** in Sec. 3.1. We then describe the **stage-focused and inference-consistent training strategy** for effective few-step flow matching in Sec. 3.2, followed by **adaptive sparse guidance propagation** for efficient guidance aggregation in Sec. 3.3. Finally, we present the training losses in Sec. 3.4.

### 3.1 Overview

**Conditional Flow Formulation.** Previous guided denoising approaches [10, 44, 31, 36, 43] typically learn a direct mapping from the noisy target and guidance images to the clean target image. However, this paradigm is less effective under severe target noise and spatial misalignment, since reliable cross-modal correspondences must be estimated from a highly corrupted target state. To address this issue, we formulate unaligned guided denoising as a conditional flow matching process that progressively transports the noisy RGB distribution to the clean RGB distribution.

For each training triplet $(\mathbf{I}_{\mathrm{N}}, \mathbf{I}_{\mathrm{G}}, \mathbf{I}_{\mathrm{T}})$, we define a linear path [22, 7] that satisfies

$$\mathbf{z}_t = (1-t)\mathbf{I}_{\mathrm{N}} + t\mathbf{I}_{\mathrm{T}}, \quad t \in [0,1], \tag{1}$$

where $\mathbf{z}_0 = \mathbf{I}_{\mathrm{N}}$ and $\mathbf{z}_1 = \mathbf{I}_{\mathrm{T}}$. The corresponding target velocity is

$$\mathbf{u}_t = \frac{d\mathbf{z}_t}{dt} = \mathbf{I}_{\mathrm{T}} - \mathbf{I}_{\mathrm{N}}. \tag{2}$$

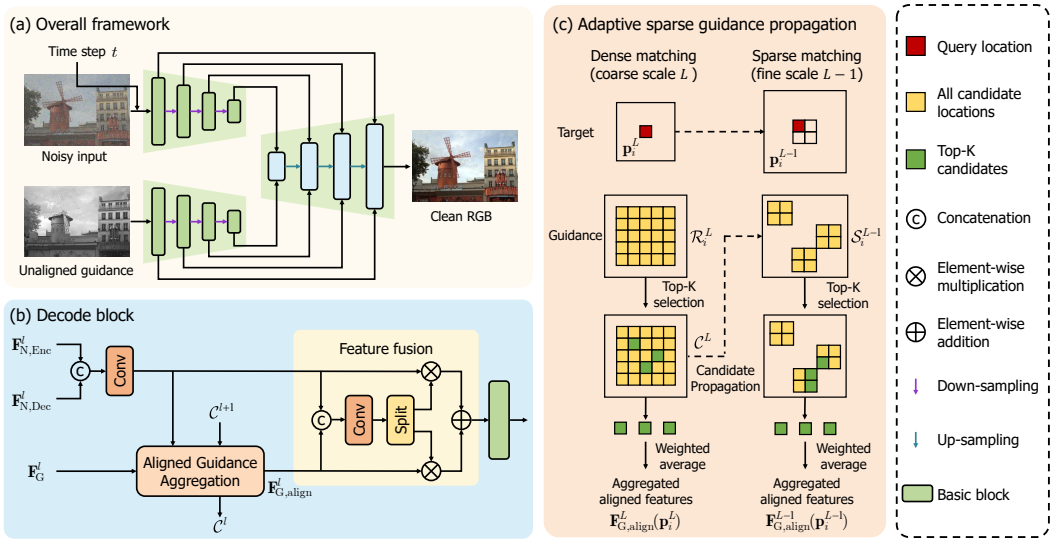


Figure 2: Network architecture of UGD-FM. (a) UGD-FM adopts a two-branch framework for unaligned RGB-NIR guided denoising. (b) The decoder block aggregates aligned guidance features and fuses them with target features. (c) Adaptive sparse guidance propagation transfers reliable coarse candidates to finer scales for efficient sparse matching and guidance aggregation.

We train a conditional velocity network $v_\theta$ to predict this velocity from the current state $\mathbf{z}_t$, the unaligned guidance $\mathbf{I}_\mathrm{G}$, and the time step $t$. The training objective is

$$\mathcal{L}_\mathrm{fm} = \mathbb{E}_{(\mathbf{I}_\mathrm{N},\mathbf{I}_\mathrm{G},\mathbf{I}_\mathrm{T}),t} \left\| v_\theta(\mathbf{z}_t, \mathbf{I}_\mathrm{G}, t) - (\mathbf{I}_\mathrm{T} - \mathbf{I}_\mathrm{N}) \right\|_1 . \tag{3}$$

During inference, we initialize $\mathbf{z}_0 = \mathbf{I}_\mathrm{N}$, and solve $d\mathbf{z}_t/dt = v_\theta(\mathbf{z}_t, \mathbf{I}_\mathrm{G}, t)$ with a few-step Euler solver. Unlike direct regression, this formulation provides a progressive denoising trajectory. Along this trajectory, the target state $\mathbf{z}_t$ is encouraged to become cleaner and more structurally reliable as $t$ increases, which facilitates cross-modal correspondence estimation. The aggregated guidance further improves velocity prediction, forming a mutually reinforcing process between guidance aggregation and image restoration.

**Overall Network Architecture.** As illustrated in Fig. 2(a), we implement the velocity network with a two-branch U-Net [29]. We first extract multi-scale RGB and NIR features from the current denoising state $\mathbf{z}_t$ and the guidance image $\mathbf{I}_\mathrm{G}$, respectively. The decoder then performs cross-modal guidance aggregation and feature fusion in a coarse-to-fine manner. As shown in Fig. 2(b), at each decoding scale, we first perform cross-modal guidance aggregation to obtain aligned NIR features, and then inject them into the target features through spatially variant feature fusion [51]. Across decoding scales, we further introduce adaptive sparse guidance propagation, as shown in Fig. 2(c), to avoid redundant large-range feature matching. Following feature alignment, a gated feature fusion module is added to adaptively adjust the role of the guiding information, preventing the injection of erroneous information in case of misalignment.

We highlight that both the conditional flow formulation and the guidance aggregation design are compatible with modern single-image restoration backbones, such as NAFNet [4], Uformer [39], and Restormer [46]. Thus, our UGD-FM provides a general framework for adapting existing single-image denoisers to unaligned cross-modal guided denoising.

### 3.2 Stage-Focused and Inference-Consistent Training

Although the conditional flow formulation improves over direct regression, directly applying standard flow matching is still suboptimal for efficient image restoration. Increasing solver steps introduces additional computation, and may accumulate prediction errors along solver-induced inference trajectories. Therefore, we aim to learn an effective and efficient few-step model, *e.g.*, with one-step or two-step inference, for high-quality guided denoising.

To this end, we revisit the denoising trajectory of unaligned guided denoising, and identify two task-critical stages. The **highly corrupted stage** with small $t$ focuses on coarse RGB structure recovery, while the **relatively cleaner stage** with larger $t$ enables more reliable cross-modal matching and fine-detail restoration. Based on this observation, we place the few-step inference timesteps around these two stages, such that the solver directly visits the states most relevant to guided denoising. We then align training with this stage-focused inference schedule at two levels, *i.e.*, stage-focused time sampling aligns the training timesteps with the inference anchors, while inference-consistent state construction aligns the training states with solver-induced inference states.

**Stage-Focused Time Sampling.** We first sample $t$ uniformly from $[0, 1]$ for stable velocity learning, and then switch to stage-focused sampling to match the few-step inference schedule. Specifically, we use two inference anchors $\tau_{\mathrm{hc}}$ and $\tau_{\mathrm{rc}}$ for the highly corrupted and relatively cleaner stages, respectively, where $\tau_{\mathrm{hc}} = 0$ and $0 < \tau_{\mathrm{rc}} < 1$. With training iteration $n$ and warm-up length $N_{\mathrm{warm}}$, the training time distribution is defined as

$$p_{\mathrm{train}}^{(n)} = \begin{cases} \mathcal{U}(0,1), & n < N_{\mathrm{warm}}, \\ p_{\mathrm{sf}}, & n \geq N_{\mathrm{warm}}, \end{cases} \tag{4}$$

where $\mathcal{U}(0, 1)$ denotes the uniform distribution over $[0, 1]$. The stage-focused distribution $p_{\mathrm{sf}}$ is defined as a mixture of two truncated Gaussian distributions as

$$p_{\mathrm{sf}} = \lambda_{\mathrm{hc}} q_{\mathrm{hc}} + \lambda_{\mathrm{rc}} q_{\mathrm{rc}}, \quad \lambda_{\mathrm{hc}} + \lambda_{\mathrm{rc}} = 1, \tag{5}$$

with

$$q_s = \mathcal{N}_{[0,1]}(\tau_s, \sigma_s^2), \quad s \in \{\mathrm{hc}, \mathrm{rc}\}. \tag{6}$$

Here, $\sigma_s$ controls the sampling concentration around the anchor, and $\mathcal{N}_{[0,1]}$ denotes a Gaussian distribution truncated and normalized on $[0, 1]$.

**Inference-Consistent State Construction.** Although stage-focused sampling aligns training and inference at the timestep level, the relatively cleaner stage still has a state mismatch. It is reached after the first solver update during inference, whereas the network is trained on the interpolation state $\mathbf{z}_t$ in Eq. 1. We therefore replace the interpolation state with a solver-induced state for samples from $q_{\mathrm{rc}}$. For $t \sim q_{\mathrm{rc}}$, we first predict the initial velocity as $\mathbf{v}_0 = v_\theta(\mathbf{I}_{\mathrm{N}}, \mathbf{I}_{\mathrm{G}}, 0)$. The inference-consistent training state is then constructed by a local Euler update as $\tilde{\mathbf{z}}_t = \mathbf{I}_{\mathrm{N}} + t\mathbf{v}_0$. Finally, the corresponding velocity target can be defined as the residual direction to the clean target as $\tilde{\mathbf{u}}_t = (\mathbf{I}_{\mathrm{T}} - \tilde{\mathbf{z}}_t)/(1 - t)$.

By combining stage-focused time sampling with inference-consistent state construction, UGD-FM better aligns training with few-step inference at both the timestep and state levels, enabling high-quality guided denoising with only one or two solver steps.

### 3.3 Adaptive Sparse Guidance Propagation

Large-range cross-modal matching is essential for unaligned guided denoising. A straightforward solution is to perform dense large-range matching [31, 51, 5] at every decoding scale. However, it is computationally expensive, especially at high resolutions. Moreover, effective guidance aggregation mainly depends on a sparse set of reliable correspondences rather than exhaustive dense relations. We therefore propose adaptive sparse guidance propagation, which discovers reliable candidates at a coarse scale and propagates them to finer scales for sparse refinement, as illustrated in Fig. 2(c).

At scale $l$, let $s_i^l(\mathbf{p})$ denote the similarity score between target location $\mathbf{p}_i^l$ and guidance location $\mathbf{p}$, and let $\mathbf{v}^l(\mathbf{p})$ denote the guidance value feature at $\mathbf{p}$. A larger $l$ indicates a coarser scale.

**Coarse Candidate Discovery.** At the coarsest scale $L$, we search within the range $\mathcal{R}_i^L = \{\mathbf{p} \mid \mathbf{p} - \mathbf{p}_i^L \in [-D/2^{L-1}, D/2^{L-1}]^2\}$ for each target location $\mathbf{p}_i^L$, and retain the top-$K$ guidance candidates as

$$\mathcal{C}_i^L = \left\{ (\mathbf{p}, \alpha_i^L(\mathbf{p})) \mid \mathbf{p} \in \mathrm{TopK}_{\mathbf{p} \in \mathcal{R}_i^L} \left( s_i^L(\mathbf{p}) \right) \right\}, \tag{7}$$

where $\alpha_i^L(\mathbf{p})$ is obtained by applying softmax over the selected top-$K$ candidates. The coordinates are propagated to finer scales, while the weights are used for aligned guidance aggregation. Thus, $\mathcal{C}_i^L$ preserves reliable large-displacement candidates, and discards most irrelevant regions.

**Sparse Candidate Propagation.** At a finer scale $l < L$, we refine the candidates propagated from the previous coarser scale. Let $\pi(i)$ denote the parent location of target location $i$ at scale $l + 1$. We

expand the coordinates in $\mathcal{C}_{\pi(i)}^{l+1}$ to their $2 \times 2$ child locations at scale $l$, forming a sparse search set as

$$\mathcal{S}_i^l = \mathrm{Child}\left(\{\mathbf{p} \mid (\mathbf{p}, \alpha) \in \mathcal{C}_{\pi(i)}^{l+1}\}\right), \tag{8}$$

where $\mathrm{Child}(\cdot)$ maps each coarse candidate coordinate to its $2 \times 2$ child locations at the current scale. We then select the top-$K$ candidates only within this sparse set as

$$\mathcal{C}_i^l = \left\{(\mathbf{p}, \alpha_i^l(\mathbf{p})) \mid \mathbf{p} \in \mathrm{TopK}_{\mathbf{p} \in \mathcal{S}_i^l}\left(s_i^l(\mathbf{p})\right)\right\}, \tag{9}$$

where $\alpha_i^l(\mathbf{p})$ is normalized over the selected top-$K$ candidates. The refined candidates are used for guidance aggregation at scale $l$, and further propagated to the next finer scale.

**Aligned Guidance Aggregation.** Given $\mathcal{C}_i^l$, we perform aligned guidance aggregation only over the selected sparse candidates as $\mathbf{F}_{\mathrm{G,align}}^l(\mathbf{p}_i^l) = \sum_{(\mathbf{p},\alpha_i^l(\mathbf{p})) \in \mathcal{C}_i^l} \alpha_i^l(\mathbf{p})\, \mathbf{v}^l(\mathbf{p})$. Following SGDFormer [51], we further fuse the aligned guidance and target features using spatially variant weights, which adaptively modulate each location and mitigate artifacts from imprecise matches. In this way, we avoid redundant dense matching, and aggregate only the most useful cross-modal cues for guided denoising.

### 3.4 Loss Functions

Although adaptive sparse guidance propagation avoids exhaustive dense matching at finer scales, learning reliable sparse candidates remains challenging. To regularize training, we introduce two auxiliary losses, namely **perceptual consistency loss** and **flow consistency loss**, which provide auxiliary perceptual regularization and explicit displacement supervision, respectively. These losses encourage the model to select reliable guidance information for restoration.

**Perceptual Consistency Loss.** To encourage feature-space compatibility between the aggregated NIR guidance and the RGB target, we apply the learned aggregation weights to the guidance image $\mathbf{I}_{\mathrm{G}}$ and obtain an aligned guidance image at scale $l$ as $\mathbf{I}_{\mathrm{G,align}}^l(\mathbf{p}_i^l) = \sum_{(\mathbf{p},\alpha_i^l(\mathbf{p})) \in \mathcal{C}_i^l} \alpha_i^l(\mathbf{p})\, \mathbf{I}_{\mathrm{G}}^l(\mathbf{p})$, where $\mathbf{I}_{\mathrm{G}}^l$ is the downsampled NIR image at scale $l$. We then compute the perceptual consistency loss between the aligned guidance image and the clean RGB target as

$$\mathcal{L}_{\mathrm{pc}} = \sum_l \mathcal{P}\left(\mathbf{I}_{\mathrm{G,align}}^l, \mathbf{I}_{\mathrm{T}}^l\right), \tag{10}$$

where $\mathcal{P}(\cdot,\cdot)$ denotes the LPIPS [11] loss, and $\mathbf{I}_{\mathrm{T}}^l$ is the clean RGB target downsampled to scale $l$.

**Flow Consistency Loss.** To further constrain sparse correspondence learning, we supervise the displacement induced by the selected candidates with pseudo flow annotations. The pseudo annotations are precomputed by an existing flow estimation network [54] between the clean RGB target $\mathbf{I}_{\mathrm{T}}$ and the guidance image $\mathbf{I}_{\mathrm{G}}$, and are used as geometric supervision during training. Specifically, we compute a soft displacement field from the aggregation weights as $\mathbf{f}^l(\mathbf{p}_i^l) = \sum_{(\mathbf{p},\alpha_i^l(\mathbf{p})) \in \mathcal{C}_i^l} \alpha_i^l(\mathbf{p})\left(\mathbf{p} - \mathbf{p}_i^l\right)$. Let $\mathbf{f}_{\mathrm{pseudo}}^l$ denote the pseudo displacement field downsampled to scale $l$. The flow consistency loss is defined as

$$\mathcal{L}_{\mathrm{fc}} = \sum_l \sum_i \left\|\mathbf{f}^l(\mathbf{p}_i^l) - \mathbf{f}_{\mathrm{pseudo}}^l(\mathbf{p}_i^l)\right\|_1. \tag{11}$$

This loss provides geometric supervision for sparse candidate selection, and further encourages reliable cross-modal guidance aggregation.

**Overall Objective.** The final training objective is

$$\mathcal{L}_{\mathrm{total}} = \mathcal{L}_{\mathrm{fm}} + \lambda_{\mathrm{pc}}\mathcal{L}_{\mathrm{pc}} + \lambda_{\mathrm{fc}}\mathcal{L}_{\mathrm{fc}}, \tag{12}$$

where $\mathcal{L}_{\mathrm{fm}}$ is the flow matching loss in Eq. 3, and $\lambda_{\mathrm{pc}}$ and $\lambda_{\mathrm{fc}}$ balance the auxiliary losses. We note that $\mathcal{L}_{\mathrm{pc}}$ and $\mathcal{L}_{\mathrm{fc}}$ are used only during training to provide perceptual regularization and geometric supervision, respectively, and introduce no additional inference cost.

## 4 Experiments

### 4.1 Experimental Setting

**Datasets.** We evaluate UGD-FM on FANVID [8], RGB-NIR Scene [1], and MS2 [33]. Dataset details are given in Appendix A.1.

Table 1: Quantitative comparison on the FANVID [8] dataset under Gaussian noise ($\sigma = 0.3$), Poisson-Gaussian noise ($\alpha = 0.02, \sigma = 0.2$), and Poisson-Gaussian noise ($\alpha = 0.03, \sigma = 0.3$). The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | $\sigma = 0.3$ | | | $\alpha = 0.02, \sigma = 0.2$ | | | $\alpha = 0.03, \sigma = 0.3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| MPRNet [47] | 29.48 | 0.8039 | 0.3480 | 30.82 | 0.8358 | 0.3040 | 29.25 | 0.7995 | 0.3520 |
| NAFNet [4] | 29.05 | 0.7932 | 0.3541 | 30.28 | 0.8239 | 0.3165 | 28.84 | 0.7891 | 0.3583 |
| Uformer [39] | 29.44 | 0.8037 | 0.3437 | 30.76 | 0.8358 | 0.3003 | 29.22 | 0.7995 | 0.3464 |
| Restormer [46] | 29.42 | 0.8024 | 0.3459 | 30.75 | 0.8339 | 0.3044 | 29.20 | 0.7981 | 0.3497 |
| Xformer [48] | 29.36 | 0.8008 | 0.3514 | 30.69 | 0.8332 | 0.3078 | 29.14 | 0.7967 | 0.3548 |
| CUNet [6] | 28.16 | 0.7618 | 0.4097 | 29.55 | 0.8023 | 0.3711 | 27.87 | 0.7537 | 0.4154 |
| DVN [10] | 29.27 | 0.7985 | 0.3592 | 30.60 | 0.8311 | 0.3152 | 29.04 | 0.7938 | 0.3631 |
| RFFNet [36] | 29.34 | 0.8013 | 0.3506 | 30.67 | 0.8332 | 0.3074 | 29.12 | 0.7968 | 0.3546 |
| NAID [43] | 29.21 | 0.7976 | 0.3504 | 30.49 | 0.8297 | 0.3098 | 28.99 | 0.7933 | 0.3540 |
| SANet [31] | 29.15 | 0.8072 | 0.3611 | 30.45 | 0.8388 | 0.3194 | 28.92 | 0.8029 | 0.3651 |
| UGD-FM (NAFNet, 1-step) | 30.34 | 0.8384 | 0.2313 | 31.55 | 0.8648 | 0.2015 | 30.12 | 0.8351 | 0.2351 |
| UGD-FM (NAFNet, 2-step) | 30.62 | 0.8486 | 0.2210 | 31.79 | 0.8722 | 0.1964 | 30.42 | 0.8458 | 0.2237 |
| UGD-FM (Uformer, 1-step) | 30.32 | 0.8342 | 0.2308 | 31.54 | 0.8652 | 0.2010 | 30.09 | 0.8343 | 0.2351 |
| UGD-FM (Uformer, 2-step) | 30.66 | 0.8499 | 0.2201 | 31.81 | 0.8732 | 0.1962 | 30.45 | 0.8467 | 0.2229 |
| UGD-FM (Restormer, 1-step) | 30.50 | 0.8436 | 0.2179 | 31.71 | 0.8694 | 0.1909 | 30.29 | 0.8401 | 0.2215 |
| UGD-FM (Restormer, 2-step) | **30.77** | **0.8535** | **0.2098** | **31.93** | **0.8764** | **0.1870** | **30.58** | **0.8506** | **0.2121** |

Table 2: Quantitative comparison on the RGB-NIR Scene dataset [1] under Gaussian noise ($\sigma = 0.3$), Poisson-Gaussian noise ($\alpha = 0.02, \sigma = 0.2$), and Poisson-Gaussian noise ($\alpha = 0.03, \sigma = 0.3$). The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | $\sigma = 0.3$ | | | $\alpha = 0.02, \sigma = 0.2$ | | | $\alpha = 0.03, \sigma = 0.3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| MPRNet [47] | 29.37 | 0.7887 | 0.3428 | 30.55 | 0.8189 | 0.3069 | 29.12 | 0.7831 | 0.3480 |
| NAFNet [4] | 29.07 | 0.7851 | 0.3409 | 30.23 | 0.8152 | 0.3068 | 28.81 | 0.7794 | 0.3466 |
| Uformer [39] | 29.35 | 0.7890 | 0.3370 | 30.54 | 0.8201 | 0.3010 | 29.09 | 0.7833 | 0.3407 |
| Restormer [46] | 29.38 | 0.7897 | 0.3365 | 30.56 | 0.8198 | 0.3014 | 29.12 | 0.7841 | 0.3417 |
| Xformer [48] | 29.46 | 0.7912 | 0.3349 | 30.65 | 0.8216 | 0.2978 | 29.19 | 0.7859 | 0.3394 |
| CUNet [6] | 27.37 | 0.7248 | 0.4484 | 28.70 | 0.7709 | 0.4046 | 27.00 | 0.7115 | 0.4612 |
| DVN [10] | 29.14 | 0.7831 | 0.3528 | 30.32 | 0.8142 | 0.3168 | 28.86 | 0.7771 | 0.3587 |
| RFFNet [36] | 29.06 | 0.7817 | 0.3540 | 30.25 | 0.8130 | 0.3175 | 28.79 | 0.7759 | 0.3603 |
| NAID [43] | 29.10 | 0.7856 | 0.3399 | 30.25 | 0.8155 | 0.3065 | 28.53 | 0.7801 | 0.3456 |
| SANet [31] | 28.23 | 0.7540 | 0.3850 | 29.27 | 0.7878 | 0.3477 | 28.01 | 0.7477 | 0.3917 |
| UGD-FM (NAFNet, 1-step) | 29.54 | 0.8077 | 0.2697 | 30.65 | 0.8346 | 0.2413 | 29.29 | 0.8024 | 0.2753 |
| UGD-FM (NAFNet, 2-step) | 29.77 | 0.8163 | 0.2600 | 30.83 | 0.8407 | 0.2363 | 29.54 | 0.8117 | 0.2642 |
| UGD-FM (Uformer, 1-step) | 29.56 | 0.8074 | 0.2706 | 30.70 | 0.8350 | 0.2408 | 29.29 | 0.8018 | 0.2768 |
| UGD-FM (Uformer, 2-step) | 29.84 | 0.8172 | 0.2598 | 30.94 | 0.8421 | 0.2351 | 29.60 | 0.8123 | 0.2636 |
| UGD-FM (Restormer, 1-step) | 29.62 | 0.8095 | 0.2599 | 30.77 | 0.8369 | 0.2322 | 29.35 | 0.8040 | 0.2651 |
| UGD-FM (Restormer, 2-step) | **29.87** | **0.8188** | **0.2551** | **30.98** | **0.8437** | **0.2315** | **29.63** | **0.8141** | **0.2585** |

**Implementation Details.** We implement UGD-FM with the basic blocks of NAFNet [4], Uformer [39], and Restormer [46], yielding three model variants. The displacement search range is $[-D, D]^2$ with radius $D = 80$. During training, degraded observations are synthesized by randomly adding Poisson-Gaussian noise to clean RGB images. Specifically, the Poisson noise parameter $\alpha$ is sampled from $[0, 0.03]$, and the Gaussian noise level $\sigma$ is sampled from $[0, 0.3]$. More implementation and training details are provided in Appendix A.2.

**Metrics.** We quantitatively evaluate the denoising performance using peak signal-to-noise ratio (PSNR), structural similarity (SSIM) [38], and learned perceptual image patch similarity (LPIPS) [11]. Higher PSNR and SSIM, and lower LPIPS, indicate better performance.

Table 3: Quantitative comparison on the MS2 [33] dataset under Gaussian noise ($\sigma = 0.3$), Poisson-Gaussian noise ($\alpha = 0.02, \sigma = 0.2$), and Poisson-Gaussian noise ($\alpha = 0.03, \sigma = 0.3$). The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | $\sigma = 0.3$ | | | $\alpha = 0.02, \sigma = 0.2$ | | | $\alpha = 0.03, \sigma = 0.3$ | | |
|---|---|---|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| MPRNet [47] | 30.89 | 0.8836 | 0.3201 | 32.53 | 0.9091 | 0.2785 | 30.57 | 0.8797 | 0.3253 |
| NAFNet [4] | 30.76 | 0.8838 | 0.3095 | 32.36 | 0.9087 | 0.2718 | 30.48 | 0.8803 | 0.3133 |
| Uformer [39] | 30.10 | 0.8837 | 0.3188 | 31.42 | 0.9095 | 0.2799 | 29.84 | 0.8799 | 0.3234 |
| Restormer [46] | 30.98 | 0.8857 | 0.3098 | 32.62 | 0.9106 | 0.2707 | 30.69 | 0.8821 | 0.3145 |
| Xformer [48] | 30.34 | 0.8623 | 0.2966 | 31.98 | 0.8930 | 0.2538 | 30.01 | 0.8573 | 0.3033 |
| CUNet [6] | 29.61 | 0.8542 | 0.3887 | 31.27 | 0.8877 | 0.3444 | 29.29 | 0.8487 | 0.3959 |
| DVN [10] | 31.78 | 0.9096 | 0.2552 | 33.29 | 0.9272 | 0.2283 | 31.64 | 0.9077 | 0.2574 |
| RFFNet [36] | 30.08 | 0.8685 | 0.3666 | 31.78 | 0.8971 | 0.3219 | 29.77 | 0.8643 | 0.3728 |
| NAID [43] | 30.33 | 0.8732 | 0.3362 | 31.91 | 0.9000 | 0.2965 | 30.04 | 0.8693 | 0.3412 |
| SANet [31] | 29.68 | 0.8466 | 0.3489 | 31.27 | 0.8806 | 0.3021 | 29.33 | 0.8402 | 0.3579 |
| UGD-FM (NAFNet, 1-step) | 32.42 | 0.9237 | 0.1752 | 33.79 | 0.9379 | 0.1550 | 32.14 | 0.9219 | 0.1779 |
| UGD-FM (NAFNet, 2-step) | 32.64 | 0.9285 | 0.1685 | 33.96 | 0.9409 | 0.1524 | 32.39 | 0.9270 | 0.1704 |
| UGD-FM (Uformer, 1-step) | 32.65 | 0.9282 | 0.1794 | 33.99 | 0.9411 | 0.1589 | 32.36 | 0.9261 | 0.1827 |
| UGD-FM (Uformer, 2-step) | 32.97 | 0.9330 | 0.1648 | 34.24 | 0.9441 | 0.1490 | 32.71 | 0.9313 | 0.1667 |
| UGD-FM (Restormer, 1-step) | 32.85 | 0.9303 | 0.1635 | 34.18 | 0.9425 | 0.1458 | 32.59 | 0.9285 | 0.1658 |
| UGD-FM (Restormer, 2-step) | **33.07** | **0.9343** | **0.1592** | **34.34** | **0.9452** | **0.1444** | **32.82** | **0.9329** | **0.1609** |

### 4.2 Comparisons with Existing Approaches

We compare UGD-FM with ten representative methods, including five single-image denoisers, *i.e.*, MPRNet [47], NAFNet [4], Uformer [39], Restormer [46], and Xformer [48], and five guided denoising methods, *i.e.*, CUNet [6], DVN [10], RFFNet [36], NAID [43], and SANet [31]. We evaluate UGD-FM with three backbone blocks, including NAFNet, Uformer, and Restormer. The denoising performance is evaluated under three noise settings: Gaussian noise with $\sigma = 0.3$, Poisson-Gaussian noise with $\alpha = 0.02, \sigma = 0.2$, and Poisson-Gaussian noise with $\alpha = 0.03, \sigma = 0.3$.

Tables 1, 2, and 3 report quantitative results on FANVID [8], RGB-NIR Scene [1], and MS2 [33], respectively. UGD-FM consistently outperforms both single-image and guided denoising baselines across all datasets and noise settings. The gains over single-image denoisers demonstrate the benefit of exploiting NIR guidance, while the improvements over guided denoising methods show the effectiveness of UGD-FM under cross-modal misalignment. Among different variants, two-step inference further improves over one-step inference, and UGD-FM with the Restormer block achieves the best overall performance. Fig. 3 presents qualitative comparisons of different approaches. Single-image denoising methods tend to oversmooth fine structures, while existing guided methods struggle to handle unaligned cross-modal inputs. In contrast, UGD-FM better exploits unaligned NIR guidance and recovers sharper textures and more faithful structural details.

**Diffusion and Flow Matching Baselines.** We further compare our UGD-FM with more diffusion and flow matching baselines like RDBM [35], 3M-TI [5] and RFfusion [40]. As shown in Table 4, UGD-FM outperforms these baselines in PSNR, SSIM, and LPIPS.

**RGB-Guided Thermal Image Super-Resolution.** We extend UGD-FM to RGB-guided thermal image super-resolution following the 3M-TI protocol [5], and list results in Table 5. UGD-FM achieves the best SSIM and MUSIQ, while obtaining competitive PSNR and perceptual quality, validating the generalizability and extensibility of our framework beyond guided denoising.

### 4.3 Ablation Study

We conduct ablation studies on FANVID [8] under Poisson-Gaussian noise with $\alpha = 0.03$ and $\sigma = 0.3$. We use UGD-FM with the Restormer block as the default setting, and list results in Table 6.

**Flow Matching Formulation and Training Strategy.** We first examine the effect of the flow formulation. As shown in Table 6a, compared with direct regression, base flow matching achieves comparable or slightly better one-step performance. More importantly, the flow formulation nat-

Table 4: Comparison with more diffusion and flow matching methods on FANVID [8] under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$). The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| RDBM [35] | 28.80 | 0.7828 | 0.3535 |
| 3M-TI [5] | 28.89 | 0.7872 | 0.3221 |
| RFfusion [40] | 28.53 | 0.7702 | 0.3251 |
| UGD-FM (NAFNet, 1-step) | 30.12 | 0.8351 | 0.2351 |
| UGD-FM (NAFNet, 2-step) | **30.42** | **0.8458** | **0.2237** |

Table 5: Comparison of RGB-guided thermal image super-resolution following the 3M-TI [5] protocol. The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ | MANIQA ↑ | MUSIQ ↑ |
|---|---|---|---|---|---|
| CoRPLE [14] | **30.47** | 0.8642 | 0.3206 | 0.2833 | 30.46 |
| SeeSR [41] | 29.41 | 0.8495 | 0.1828 | 0.4278 | 35.22 |
| 3M-TI [5] | 30.09 | 0.8610 | **0.1787** | **0.4443** | 36.66 |
| UGD-FM | 30.36 | **0.8710** | 0.2012 | 0.4008 | **36.94** |

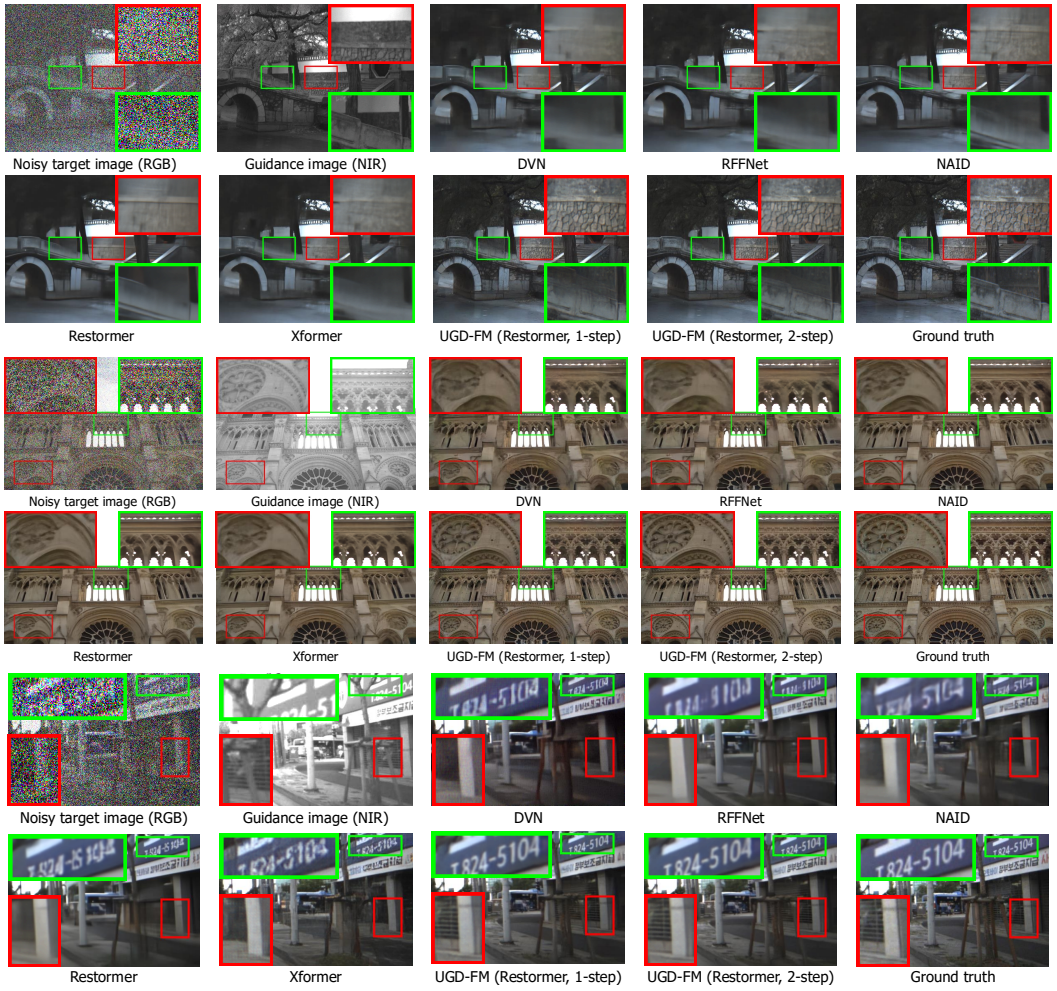


Figure 3: Qualitative comparison of image denoising results across different approaches.

urally supports multi-step inference, improving PSNR from 30.28 to 30.34. This demonstrates the benefit of modeling guided denoising as a progressive transport process. Cumulatively adding stage-focused sampling and inference-consistent states increases two-step PSNR to 30.42 and 30.58, respectively. Stage-focused sampling aligns the training-time distribution with the fixed inference anchors. Inference-consistent state construction further aligns the training and inference states by replacing the interpolation state with a solver-induced state.

Table 6: Ablation studies on FANVID [8] under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$).

(a) Flow matching formulation and training strategy.

| Setting | 1-step | | | 2-step | | |
|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| Direct regression | 30.27 | 0.8367 | 0.2376 | – | – | – |
| Standard FM | 30.28 | 0.8389 | 0.2357 | 30.34 | 0.8401 | 0.2312 |
| + Stage-focused sampling | **30.31** | 0.8396 | 0.2238 | 30.42 | 0.8421 | 0.2224 |
| + Inference-consistent state construction | 30.29 | **0.8401** | **0.2215** | **30.58** | **0.8506** | **0.2121** |

(b) Guidance aggregation.

| Setting | 1-step | | | 2-step | | |
|---|---|---|---|---|---|---|
| | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| No attention | 28.97 | 0.7920 | 0.3247 | 29.10 | 0.7976 | 0.3244 |
| Global attention | OOM | OOM | OOM | OOM | OOM | OOM |
| Window attention | 29.64 | 0.8188 | 0.2512 | 30.03 | 0.8370 | 0.2369 |
| Sparse propagation | **30.29** | **0.8401** | **0.2215** | **30.58** | **0.8506** | **0.2121** |

(c) Auxiliary losses.

| $\mathcal{L}_{\text{pc}}$ | $\mathcal{L}_{\text{fc}}$ | 1-step | | | 2-step | | |
|---|---|---|---|---|---|---|---|
| | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| ✗ | ✗ | 29.48 | 0.8062 | 0.2862 | 29.78 | 0.8213 | 0.2660 |
| ✗ | ✓ | 30.13 | 0.8355 | 0.2306 | 30.45 | 0.8465 | 0.2217 |
| ✓ | ✗ | 30.01 | 0.8324 | 0.2345 | 30.27 | 0.8445 | 0.2234 |
| ✓ | ✓ | **30.29** | **0.8401** | **0.2215** | **30.58** | **0.8506** | **0.2121** |

**Adaptive Sparse Guidance Propagation.** We evaluate the efficiency of the guidance aggregation design in Table 6b. Without attention, the model cannot handle spatial misalignment, leading to a large performance drop. Global attention leads to out-of-memory errors, showing that dense large-range matching is impractical at high resolutions. Window attention provides a non-local receptive field but is less flexible. In contrast, our sparse propagation avoids exhaustive matching while retaining reliable candidates for guidance aggregation, enabling efficient inference.

**Auxiliary Losses.** Table 6c shows complementary benefits from perceptual regularization and displacement supervision. Removing either $\mathcal{L}_{\text{pc}}$ or $\mathcal{L}_{\text{fc}}$ degrades the restoration quality, and removing both losses causes a clear performance drop, with the two-step PSNR decreasing from 30.58 to 29.78. This suggests that perceptual and geometric regularization are complementary and important for stabilizing sparse correspondence learning and improving aligned guidance aggregation.

## 5 Conclusions

We have presented UGD-FM, a general cross-modal conditional flow matching framework for unaligned RGB-NIR guided denoising. To support efficient few-step inference, we design a stage-focused and inference-consistent training strategy that aligns training with task-critical solver states. We also introduce adaptive sparse guidance propagation with perceptual and geometric auxiliary losses for efficient and reliable large-range cross-modal matching. Extensive experiments demonstrate that UGD-FM achieves state-of-the-art denoising performance with only one or two inference steps.

## Acknowledgments and Disclosure of Funding

This work was supported by the National Natural Science Foundation of China under Grant 62571478.

## References


[1] Matthew Brown and Sabine Süsstrunk. Multi-spectral SIFT for scene category recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 177–184, 2011.

[2] Clément Chadebec, Onur Tasar, Sanjeev Sreetharan, and Benjamin Aubin. LBM: Latent bridge matching for fast image-to-image translation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 29086–29098, 2025.

[3] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12299–12310, 2021.

[4] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33, 2022.

[5] Minchong Chen, Xiaoyun Yuan, Junzhe Wan, Jianing Zhang, and Jun Zhang. 3M-TI: High-quality mobile thermal imaging via calibration-free multi-camera cross-modal diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5659–5669, 2026.

[6] Xin Deng and Pier Luigi Dragotti. Deep convolutional neural network for multi-modal image restoration and fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(10):3333–3348, 2020.

[7] Tobias Fischer, Samuel Rota Bulò, Yung-Hsu Yang, Nikhil Keetha, Lorenzo Porzi, Norman Müller, Katja Schwarz, Jonathon Luiten, Marc Pollefeys, and Peter Kontschieder. FlowR: Flowing from sparse to dense 3D reconstructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 27702–27712, 2025.

[8] Yunyi Gao, Lin Gu, Qiankun Liu, and Ying Fu. Object-aware NIR-to-visible translation. In *European Conference on Computer Vision*, pages 93–109, 2024.

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, 2012.

[10] Shuangping Jin, Bingbing Yu, Minhao Jing, Yi Zhou, Jiajun Liang, and Renhe Ji. DarkVisionNet: Low-light imaging via RGB-NIR fusion with deep inconsistency prior. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 1104–1112, 2022.

[11] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711, 2016.

[12] Jinnyeong Kim and Seung-Hwan Baek. Pixel-aligned RGB-NIR stereo imaging and dataset for robot vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11482–11492, 2025.

[13] Xingyuan Li, Songcheng Du, Yang Zou, HaoYuan Xu, Zhiying Jiang, and Jinyuan Liu. UniFusion: A unified image fusion framework with robust representation and source-aware preservation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 33869–33880, 2026.

[14] Xingyuan Li, Jinyuan Liu, Zhixin Chen, Yang Zou, Long Ma, Xin Fan, and Risheng Liu. Contourlet residual for prompt learning enhanced infrared image super-resolution. In *European Conference on Computer Vision*, pages 270–288, 2024.

[15] Xingyuan Li, Zirui Wang, Yang Zou, Zhixin Chen, Jun Ma, Zhiying Jiang, Long Ma, and Jinyuan Liu. DifIISR: A diffusion model with gradient guidance for infrared image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7534–7544, 2025.

[16] Xingyuan Li, Haoyuan Xu, Shulin Li, Xiang Chen, Zhiying Jiang, and Jinyuan Liu. DR-Fusion: Drift-resilient temporally consistent infrared-visible video fusion. *arXiv preprint arXiv:2605.25775*, 2026.

[17] Xingyuan Li, Haoyuan Xu, Xingyue Zhu, Jun Ma, Yang Zou, Zhiying Jiang, and Jinyuan Liu. Uncertainty-aware spatial-frequency registration and fusion for infrared and visible images. *arXiv preprint arXiv:2605.13049*, 2026.

[18] Yijun Li, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Deep joint image filtering. In *European conference on computer vision*, pages 154–169, 2016.

[19] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. SwinIR: Image restoration using Swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021.

[20] Jinyuan Liu, Xingyuan Li, Zirui Wang, Zhiying Jiang, Wei Zhong, Wei Fan, and Bin Xu. PromptFusion: Harmonized semantic prompt learning for infrared and visible image fusion.

*IEEE/CAA Journal of Automatica Sinica*, 12(3):502–515, 2024.

[21] Jinyuan Liu, Bowei Zhang, Qingyun Mei, Xingyuan Li, Yang Zou, Zhiying Jiang, Long Ma, Risheng Liu, and Xin Fan. DCEvo: Discriminative cross-dimensional evolutionary learning for infrared and visible image fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2226–2235, 2025.

[22] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022.

[23] Xiongwei Liu, Zehua Sheng, and Hui-Liang Shen. Frequency-relevant residual learning for multi-modal image denoising. In *International Conference on Image Processing*, pages 86–90, 2022.

[24] Zhen Liu, Diedong Feng, Hai Jiang, Liaoyuan Zeng, Hao Wang, Chaoyu Feng, Lei Lei, Bing Zeng, and Shuaicheng Liu. RAW-Flow: Advancing RGB-to-RAW image reconstruction with deterministic latent flow matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 7431–7439, 2026.

[25] Lars Mescheder, Wei Dong, Shiwei Li, Xuyang Bai, Marcel Santos, Peiyun Hu, Bruno Lecouat, Mingmin Zhen, Amaël Delaunoy, Tian Fang, et al. Sharp monocular view synthesis in less than a second. In *International Conference on Learning Representations*, pages 24192–24230, 2026.

[26] Jinshan Pan, Jiangxin Dong, Jimmy S Ren, Liang Lin, Jinhui Tang, and Ming-Hsuan Yang. Spatially variant linear representation models for joint filtering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1702–1711, 2019.

[27] Haina Qin, Wenyang Luo, Libin Wang, Dandan Zheng, Jingdong Chen, Ming Yang, Bing Li, and Weiming Hu. Reversing flow for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7545–7558, 2025.

[28] Jiangwei Ren, Xingyu Jiang, Zizhuo Li, Dingkang Liang, Xin Zhou, and Xiang Bai. MINIMA: Modality invariant image matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 23059–23068, 2025.

[29] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015.

[30] Zehua Sheng, Xiongwei Liu, Si-Yuan Cao, Hui-Liang Shen, and Huaqi Zhang. Frequency-domain deep guided image denoising. *IEEE Transactions on Multimedia*, 25:6767–6781, 2022.

[31] Zehua Sheng, Zhu Yu, Xiongwei Liu, Si-Yuan Cao, Yuqi Liu, Hui-Liang Shen, and Huaqi Zhang. Structure aggregation for cross-spectral stereo image guided denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13997–14006, 2023.

[32] Takashi Shibata, Masayuki Tanaka, and Masatoshi Okutomi. Misalignment-robust joint filter for cross-modal image pairs. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3295–3304, 2017.

[33] Ukcheol Shin, Jinsun Park, and In So Kweon. Deep depth estimation from thermal image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1043–1053, 2023.

[34] Darshan Thaker, Abhishek Goyal, and René Vidal. Frequency-guided posterior sampling for diffusion-based image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12873–12882, 2025.

[35] Hebaixu Wang, Jing Zhang, Haoyang Chen, Haonan Guo, Di Wang, Jiayi Ma, and Bo Du. Residual diffusion bridge model for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8375–8386, 2026.

[36] Qiang Wang, Yuning Cui, Yawen Li, Yaping Ruan, Ben Zhu, and Wenqi Ren. RFFNet: Towards robust and flexible fusion for low-light image denoising. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 836–845, 2024.

[37] Yuchen Wang, Hongyuan Wang, Lizhi Wang, Xin Wang, Lin Zhu, Wanxuan Lu, and Hua Huang. Complementary advantages: Exploiting cross-field frequency correlation for NIR-assisted image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12679–12689, 2025.

[38] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[39] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general U-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022.

[40] Zirui Wang, Jiayi Zhang, Tianwei Guan, Yuhan Zhou, Xingyuan Li, Minjing Dong, and Jinyuan Liu. Efficient rectified flow for image fusion. *Advances in Neural Information Processing Systems*, 2025.

[41] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. SeeSR: Towards semantics-aware real-world image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 25456–25467, 2024.

[42] Jinhui Xiong, Jian Wang, Wolfgang Heidrich, and Shree Nayar. Seeing in extra darkness using a deep-red flash. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10000–10009, 2021.

[43] Rongjian Xu, Zhilu Zhang, Renlong Wu, and Wangmeng Zuo. NIR-assisted image denoising: A selective fusion approach and a real-world benchmark dataset. *IEEE Transactions on Multimedia*, 27:2543–2555, 2024.

[44] Shuang Xu, Jiangshe Zhang, Jialin Wang, Kai Sun, Chunxia Zhang, Junmin Liu, and Junying Hu. A model-driven network for guided image denoising. *Information Fusion*, 85:60–71, 2022.

[45] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. ResShift: Efficient diffusion model for image super-resolution by residual shifting. *Advances in neural information processing systems*, 36:13294–13307, 2023.

[46] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022.

[47] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021.

[48] Jiale Zhang, Yulun Zhang, Jinjin Gu, Jiahua Dong, Linghe Kong, and Xiaokang Yang. Xformer: Hybrid X-shaped transformer for image denoising. In *International Conference on Learning Representations*, 2024.

[49] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[50] Runmin Zhang, Jialiang Wang, Si-Yuan Cao, Zhu Yu, Junchen Yu, Guangyi Zhang, and Hui-Liang Shen. Rethinking unsupervised cross-modal flow estimation: Learning from decoupled optimization and consistency constraint. In *International Conference on Learning Representations*, 2026.

[51] Runmin Zhang, Zhu Yu, Zehua Sheng, Jiacheng Ying, Si-Yuan Cao, Shu-Jie Chen, Bailin Yang, Junwei Li, and Hui-Liang Shen. SGDFormer: One-stage transformer-based architecture for cross-spectral stereo image guided denoising. *Information Fusion*, 113:102603, 2025.

[52] Yingkai Zhang, Zeqiang Lai, Tao Zhang, Ying Fu, and Chenghu Zhou. Unaligned RGB guided hyperspectral image super-resolution with spatial-spectral concordance. *International Journal of Computer Vision*, 133(9):6590–6610, 2025.

[53] Tiancheng Zhi, Bernardo R Pires, Martial Hebert, and Srinivasa G Narasimhan. Deep material-aware cross-spectral stereo matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1916–1925, 2018.

[54] Shili Zhou, Weimin Tan, and Bo Yan. Promoting single-modal optical flow network for diverse cross-modal flow estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3562–3570, 2022.

[55] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1219–1229, 2023.

# Appendix

## A Additional Experimental Details

### A.1 Details of Datasets

We evaluate UGD-FM on the RGB-NIR Scene [1], FANVID [8], and MS2 [33] datasets. A statistical summary of these datasets is provided in Table 7. Specifically, FANVID contains 453 RGB images and 3624 RGB-NIR pairs with an original image resolution of $576 \times 768$ ($H \times W$), while RGB-NIR Scene provides 434 RGB images and 3472 RGB-NIR pairs with original resolutions ranging from $620 \times 502$ to $768 \times 1024$. MS2 consists of 1229 RGB-NIR pairs captured from real-world multi-camera imaging systems with an original resolution of $320 \times 1022$. We use 3008/616, 3112/360, and 981/248 image pairs for training/testing on FANVID, RGB-NIR Scene, and MS2, respectively.

RGB-NIR Scene and FANVID originally provide spatially aligned RGB-NIR image pairs, whereas MS2 is captured using real multi-camera imaging systems and naturally contains viewpoint discrepancies between modalities. Therefore, for RGB-NIR Scene and FANVID, we further construct misaligned RGB-NIR pairs through a controlled parallax simulation strategy.

Specifically, starting from the aligned RGB-NIR pairs, the RGB image is retained as the reference target view, while the NIR image is independently reconstructed into a 3D Gaussian representation using SHARP [25]. Based on the reconstructed representation, eight neighboring virtual camera viewpoints are sampled around the original camera pose to render new NIR views, as illustrated in Fig. 4. We place the virtual cameras on a circular trajectory whose radius is normalized so that the nominal translational parallax at a representative near-scene depth is approximately $x\%$ of the image diagonal. We use $x = 6$ for FANVID and $x = 8$ for RGB-NIR Scene. This simulation protocol is designed to approximate practical dual-camera acquisition settings, where RGB and NIR sensors observe the same scene from different viewpoints. The rendered NIR images are subsequently paired with the original RGB images to form misaligned RGB-NIR image pairs with realistic 3D parallax.

Table 7 summarizes 2D displacement magnitudes. For the synthetic FANVID and RGB-NIR Scene datasets, we obtain ground-truth flow from 3DGS rendering. For the real-world MS2 dataset, we estimate flow using CrossModalRAFT [54]. The P95 displacements are 45.25, 76.25, and 50.75 pixels, respectively.

Table 7: Dataset characteristics and statistics of 2D displacement magnitudes.

| Characteristic | FANVID [8] | RGB-NIR Scene [1] | MS2 [33] |
|---|---|---|---|
| # RGB images | 453 | 434 | 1229 |
| # RGB-NIR pairs | 3624 | 3472 | 1229 |
| Resolution ($H \times W$) | $576 \times 768$ | $620 \times 502$–$768 \times 1024$ | $320 \times 1022$ |
| # Training pairs | 3008 | 3112 | 981 |
| # Testing pairs | 616 | 360 | 248 |
| Mean displacement (px) | 22.10 | 36.70 | 29.64 |
| Std. displacement (px) | 13.46 | 23.16 | 12.28 |
| P75 displacement (px) | 32.25 | 54.25 | 36.25 |
| P90 displacement (px) | 41.25 | 69.25 | 43.75 |
| P95 displacement (px) | 45.25 | 76.25 | 50.75 |

### A.2 Implementation and Experimental Details

Different variants instantiate Uformer, Restormer, or NAFNet as backbone blocks while sharing the same overall architecture. The base channel width is set to $C = 32$, and only a single block is employed at each scale. The maximum network depth is set to $L = 4$. The Top-$K$ parameter is fixed to $K = 5$. Inference under the flow matching framework is performed using two steps at $t = \tau_{\mathrm{hc}}$ and $t = \tau_{\mathrm{rc}}$, where $\tau_{\mathrm{hc}} = 0$ and $\tau_{\mathrm{rc}} = 0.9$.

We optimize the network using the AdamW optimizer with a weight decay of $0.01$, an initial learning rate of $3 \times 10^{-4}$, and a OneCycle learning rate scheduler. Training is performed for $80$k iterations

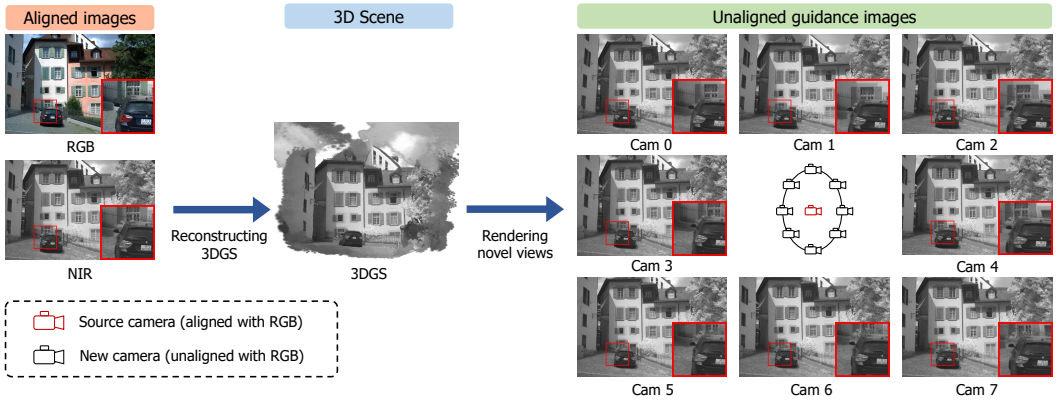


Figure 4: Visualization of the simulation process for constructing unaligned RGB-NIR datasets.

using a batch size of 8. The warm-up stage is conducted for $N_{\text{warm}} = 50\text{k}$ iterations. The auxiliary loss weights are set to $\lambda_{\text{pc}} = 0.05$ and $\lambda_{\text{fc}} = 0.05$, while the truncated Gaussian distribution weights are set to $\lambda_{\text{hc}} = 0.5$ and $\lambda_{\text{rc}} = 0.5$. In addition, the parameter $\sigma_s$ is set to $0.05$.

For fair comparison, all approaches are trained using the same training set, and the training iterations are also the same. All experiments are conducted on NVIDIA RTX 4090 GPUs.

# B More Quantitative and Qualitative Results

## B.1 Ablation on Hyperparameters

In Table 8, we provide detailed ablations on hyperparameters including the displacement search radius $D$, Top-$K$ candidate selection, inference time anchors $\tau_{\text{hc}}$ and $\tau_{\text{rc}}$, time sampling standard deviation $\sigma_s$, and warm-up length $N_{\text{warm}}$.

**Displacement Search Radius $D$.** We vary the displacement search radius from 32 to 96. A small search range may fail to cover the true displacement range, causing a performance drop, while increasing $D$ generally improves PSNR by providing a broader search range.

**Top-$K$ Candidate Selection.** We vary $K$ from 1 to 7. A small $K$ may discard plausible correspondences, causing a clear performance drop. Increasing $K$ improves performance, with only a small additional gain from 5 to 7. We use $K = 5$ as the default.

**Inference Time Anchors $\tau_{\text{hc}}$ and $\tau_{\text{rc}}$.** Moving $\tau_{\text{hc}}$ from 0 to 0.1 reduces performance, supporting the use of the initial noisy state as the highly corrupted-stage anchor. For $\tau_{\text{rc}}$, an early anchor provides limited separation between the highly corrupted and relatively cleaner stages, resulting in smaller two-step gains. When the anchor clearly separates these stages, as with $\tau_{\text{rc}} = 0.7$ or $0.9$, two-step inference yields clear improvements. Their comparable results indicate that the improvement does not require a precisely tuned anchor. We use $\tau_{\text{rc}} = 0.9$ as the default.

**Time Sampling Standard Deviation $\sigma_s$.** The parameter $\sigma_s$ controls the sampling concentration around the time anchors. Similar results at $0.025$ and $0.05$ show that the method does not require careful tuning within this range. However, an excessively large value such as $0.2$ disperses the training distribution away from the anchors and degrades performance.

**Warm-up Length $N_{\text{warm}}$.** We vary the warm-up length from $40\text{k}$ to $60\text{k}$ iterations. The results remain similar across these settings, indicating limited sensitivity to the warm-up length within the tested range. We use $N_{\text{warm}} = 50\text{k}$ as the default.

## B.2 Sensitivity to Flow-Label Quality

To evaluate the relationship between denoising performance and the quality of flow labels, we compare three types of supervision including synthetic ground-truth flow, estimated flow, and estimated flow with synthetic perturbations on the FANVID dataset. The ground-truth flow is derived from the

Table 8: Hyperparameter ablations on FANVID [8] under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$).

| Hyperparameter | Setting | 1-step | | | 2-step | | |
|---|---|---|---|---|---|---|---|
| | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| Search range $D$ | 32 | 30.19 | 0.8377 | 0.2241 | 30.42 | 0.8424 | 0.2179 |
| | 48 | 30.29 | 0.8396 | 0.2167 | 30.49 | 0.8471 | 0.2105 |
| | 64 | 30.32 | 0.8413 | 0.2179 | 30.56 | 0.8527 | 0.2097 |
| | 80 (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| | 96 | 30.31 | 0.8403 | 0.2221 | 30.61 | 0.8507 | 0.2128 |
| Top-$K$ selection | 1 | 29.89 | 0.8248 | 0.2449 | 30.19 | 0.8364 | 0.2345 |
| | 3 | 30.26 | 0.8382 | 0.2264 | 30.55 | 0.8492 | 0.2159 |
| | 5 (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| | 7 | 30.30 | 0.8417 | 0.2185 | 30.59 | 0.8520 | 0.2110 |
| Time anchor $\tau_{\mathrm{hc}}$ | 0 (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| | 0.1 | 29.85 | 0.8266 | 0.2509 | 30.21 | 0.8391 | 0.2358 |
| Time anchor $\tau_{\mathrm{rc}}$ | 0.3 | 30.34 | 0.8432 | 0.2235 | 30.39 | 0.8457 | 0.2216 |
| | 0.5 | 30.33 | 0.8412 | 0.2192 | 30.49 | 0.8459 | 0.2191 |
| | 0.7 | 30.33 | 0.8409 | 0.2237 | 30.56 | 0.8495 | 0.2133 |
| | 0.9 (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| Sampling std. $\sigma_s$ | 0.025 | 30.33 | 0.8414 | 0.2220 | 30.62 | 0.8524 | 0.2139 |
| | 0.05 (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| | 0.2 | 30.27 | 0.8384 | 0.2257 | 30.46 | 0.8453 | 0.2192 |
| Warm-up $N_{\mathrm{warm}}$ | 40k | 30.31 | 0.8406 | 0.2211 | 30.56 | 0.8512 | 0.2110 |
| | 50k (default) | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| | 60k | 30.29 | 0.8392 | 0.2224 | 30.55 | 0.8497 | 0.2123 |

Table 9: Effect of flow-label quality on FANVID [8] under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$).

| Flow label | Added mean EPE | 1-step | | | 2-step | | |
|---|---|---|---|---|---|---|---|
| | | PSNR ↑ | SSIM ↑ | LPIPS ↓ | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
| Ground-truth flow | – | 30.31 | 0.8422 | 0.2199 | 30.61 | 0.8513 | 0.2085 |
| Estimated flow (default) | – | 30.29 | 0.8401 | 0.2215 | 30.58 | 0.8506 | 0.2121 |
| Perturbed estimated flow | 5 px | 30.27 | 0.8393 | 0.2232 | 30.56 | 0.8497 | 0.2140 |
| | 10 px | 30.25 | 0.8382 | 0.2249 | 30.53 | 0.8487 | 0.2164 |
| | 15 px | 30.25 | 0.8383 | 0.2261 | 30.53 | 0.8486 | 0.2173 |

known geometry and camera transformations used in 3DGS rendering. For the perturbed flow, we apply Gaussian blur and additive Gaussian noise to the estimated flow, increasing its mean endpoint error (EPE) by up to 15 pixels. As shown in Table 9, replacing estimated flow with ground-truth flow improves the 2-step results by only 0.03 dB PSNR, 0.0007 SSIM, and 0.0036 LPIPS. Conversely, increasing the mean EPE by 15 pixels decreases performance by only 0.05 dB PSNR, 0.0020 SSIM, and 0.0052 LPIPS. These results indicate that denoising performance is not highly sensitive to flow-label accuracy. Flow supervision mainly serves as an auxiliary signal for learning sparse correspondence attention, rather than being a key determinant of restoration quality.

### B.3 Number of Inference Steps

Table 10 compares standard FM with 1, 2, 4, and 8 solver steps and UGD-FM with one-step or two-step inference. Standard FM yields consistent gains from 1 to 4 solver steps. However, at 8 steps, PSNR and SSIM drop, while LPIPS improves, indicating that additional steps do not consistently improve restoration quality. We attribute the fidelity degradation to error accumulation during iterative

Table 10: Effect of the number of inference steps on FANVID [8] under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$).

| Method | Steps | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|---|
| Standard FM | 1 | 30.28 | 0.8389 | 0.2357 |
| | 2 | 30.34 | 0.8401 | 0.2312 |
| | 4 | 30.36 | 0.8422 | 0.2291 |
| | 8 | 30.02 | 0.8340 | 0.2265 |
| UGD-FM | 1 | 30.29 | 0.8401 | 0.2215 |
| | 2 | 30.58 | 0.8506 | 0.2121 |

Table 11: Comparison under different misalignment levels on FANVID [8] with Poisson–Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$).

| Misalignment | Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|---|
| Aligned | Restormer [46] | 29.20 | 0.7981 | 0.3497 |
| | NAID [43] | 31.93 | 0.8961 | 0.1616 |
| | UGD-FM (NAFNet, 1-step) | 31.75 | 0.8898 | 0.1771 |
| | UGD-FM (NAFNet, 2-step) | 32.06 | 0.8984 | 0.1535 |
| $x = 2$ | NAID [43] | 29.22 | 0.8071 | 0.2919 |
| | UGD-FM (NAFNet, 1-step) | 30.50 | 0.8478 | 0.2129 |
| | UGD-FM (NAFNet, 2-step) | 30.86 | 0.8597 | 0.1988 |
| $x = 4$ | NAID [43] | 29.05 | 0.7970 | 0.3324 |
| | UGD-FM (NAFNet, 1-step) | 30.29 | 0.8409 | 0.2250 |
| | UGD-FM (NAFNet, 2-step) | 30.62 | 0.8523 | 0.2119 |
| $x = 6$ | NAID [43] | 28.99 | 0.7933 | 0.3540 |
| | UGD-FM (NAFNet, 1-step) | 30.12 | 0.8351 | 0.2351 |
| | UGD-FM (NAFNet, 2-step) | 30.42 | 0.8458 | 0.2237 |
| $x = 8$ | NAID [43] | 28.97 | 0.7922 | 0.3673 |
| | UGD-FM (NAFNet, 1-step) | 29.97 | 0.8297 | 0.2448 |
| | UGD-FM (NAFNet, 2-step) | 30.24 | 0.8394 | 0.2352 |

forward-Euler updates, which may cause global blur and detail loss. In contrast, UGD-FM employs stage-focused and inference-consistent training for efficient inference, achieving SOTA performance with only 2 steps.

### B.4 Experiments on Misalignment Magnitudes

We vary the misalignment magnitudes on FANVID from aligned guidance to $x = 2, 4, 6, 8$, using the same virtual-camera displacement definition as in Appendix A.1. Larger $x$ indicates stronger parallax. As shown in Table 11, guided denoising methods like NAID [43] perform well only with aligned guidance or mild misalignment. In contrast, our UGD-FM maintains strong performance as the misalignment increases, demonstrating its robustness to unaligned guidance.

### B.5 More Evaluation on Real-World Datasets

We conduct additional experiments on RNS [12], which is a real-world dataset captured using an RGB-NIR dual-camera system. We use the original images without epipolar rectification, with RGB images from the right camera as targets and NIR images from the left camera as guidance. Table 12 reports results under Poisson-Gaussian noise ($\alpha = 0.03$, $\sigma = 0.3$). UGD-FM achieves the best performance, demonstrating its generalizability to real-world data.

Table 12: Quantitative comparison on the RNS [12] dataset under Poisson-Gaussian noise ($\alpha = 0.03, \sigma = 0.3$). The **best** and second-best results are highlighted in bold and underlined, respectively.

| Method | PSNR ↑ | SSIM ↑ | LPIPS ↓ |
|---|---|---|---|
| Restormer [46] | 28.20 | 0.7761 | 0.3793 |
| RDBM [35] | 28.38 | 0.7678 | 0.3377 |
| NAID [43] | 28.56 | 0.7997 | 0.3137 |
| SGDFormer [51] | 28.39 | 0.7863 | 0.3307 |
| RFfusion [40] | 27.76 | 0.7626 | 0.3144 |
| 3M-TI [5] | 28.62 | 0.8011 | 0.3024 |
| UGD-FM (NAFNet, 1-step) | 28.96 | 0.8110 | 0.2505 |
| UGD-FM (NAFNet, 2-step) | **29.10** | **0.8177** | **0.2280** |

Table 13: Inference efficiency at input resolution $768 \times 1024$ on an NVIDIA RTX 4090 GPU.

| Method | Time (ms) | Peak memory (GiB) |
|---|---|---|
| MPRNet [47] | 437 | 6.3 |
| NAFNet [4] | 76 | 0.9 |
| Uformer [39] | 171 | 4.7 |
| Restormer [46] | 676 | 6.7 |
| Xformer [48] | 1390 | 7.8 |
| RDBM [35] | 3431 | 9.3 |
| CUNet [6] | 162 | 1.9 |
| DVN [10] | 159 | 3.9 |
| RFFNet [36] | 164 | 11.4 |
| NAID [43] | 120 | 1.5 |
| SANet [31] | 194 | 4.4 |
| RFfusion [40] | 841 | 20.7 |
| 3M-TI [5] | 654 | 9.7 |
| UGD-FM (NAFNet, 1-step) | 250 | 5.1 |
| UGD-FM (NAFNet, 2-step) | 487 | 5.1 |
| UGD-FM (Uformer, 1-step) | 441 | 5.0 |
| UGD-FM (Uformer, 2-step) | 829 | 5.0 |
| UGD-FM (Restormer, 1-step) | 368 | 5.0 |
| UGD-FM (Restormer, 2-step) | 694 | 5.0 |

### B.6 Inference Efficiency

Table 13 reports inference time and peak memory usage at an input resolution of $768 \times 1024$ on a single NVIDIA RTX 4090 GPU. UGD-FM achieves state-of-the-art performance without incurring prohibitive memory or inference-time overhead.

### B.7 Visualization of the Mutually Reinforcing Process

Figure 5 demonstrates the mutually reinforcing process between image restoration and guidance aggregation. The visualized features correspond to the guidance features of the finest level of decoding blocks. As the flow matching trajectory evolves, the guidance features become more structured and coherent.

### B.8 More Qualitative Comparisons

We provide more qualitative comparisons of different approaches in Fig. 6. Single image denoising approaches generally cannot preserve structures and textures well. Among all guided denoising methods, our UGD-FM produces denoised images with more salient structures and details.

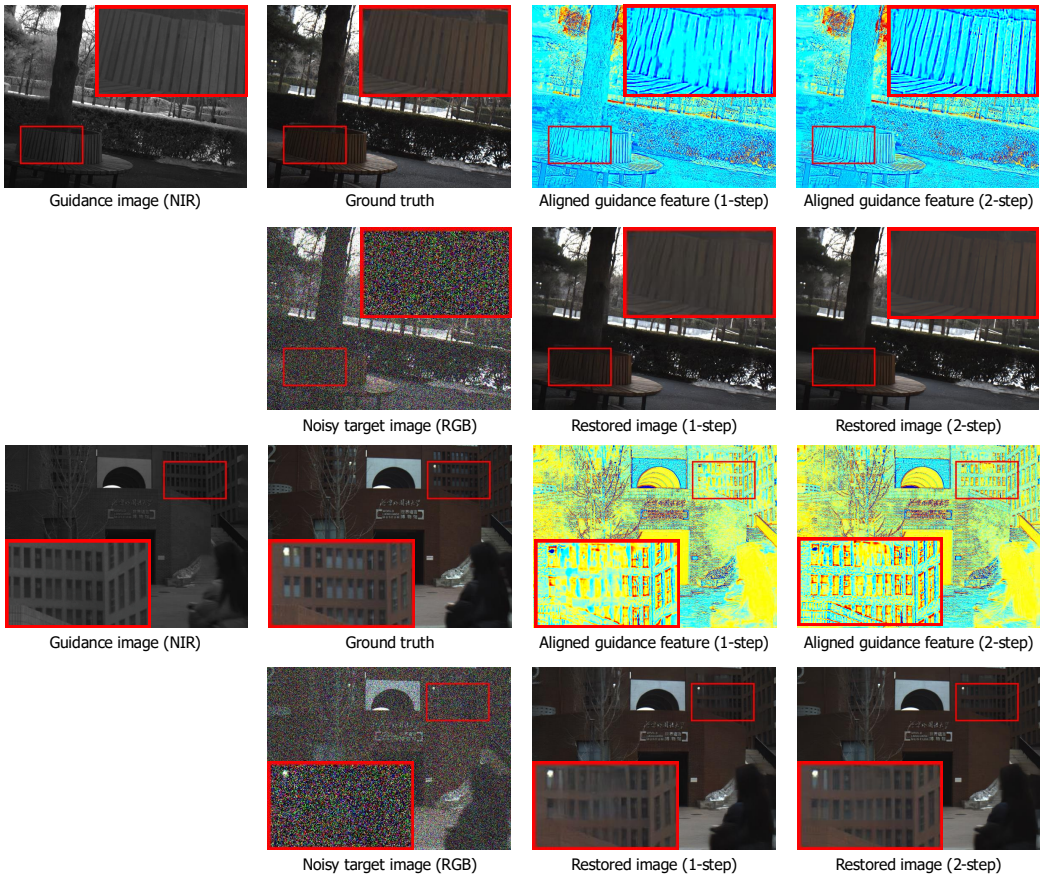


Figure 5: Visualization of the mutually reinforcing process between image restoration and guidance aggregation.

## C Limitations

UGD-FM remains challenged by severe 3D parallax, where occlusions and incomplete cross-view observations can make local guidance ambiguous or unreliable, as shown in Fig. 7. Joint restoration and cross-modal alignment also introduce computational overhead. Despite these limitations, we believe UGD-FM provides a practical solution for unaligned guided image denoising.

## D Broader Impacts

UGD-FM can benefit low-light photography through image restoration using cross-modal guidance. However, its use in unauthorized surveillance may raise privacy concerns, and restored images may contain plausible but inaccurate details. Deployment in safety-critical applications therefore requires careful validation and further assessment of uncertainty and reliability.




Figure 6: More qualitative comparison of image denoising results across different approaches.

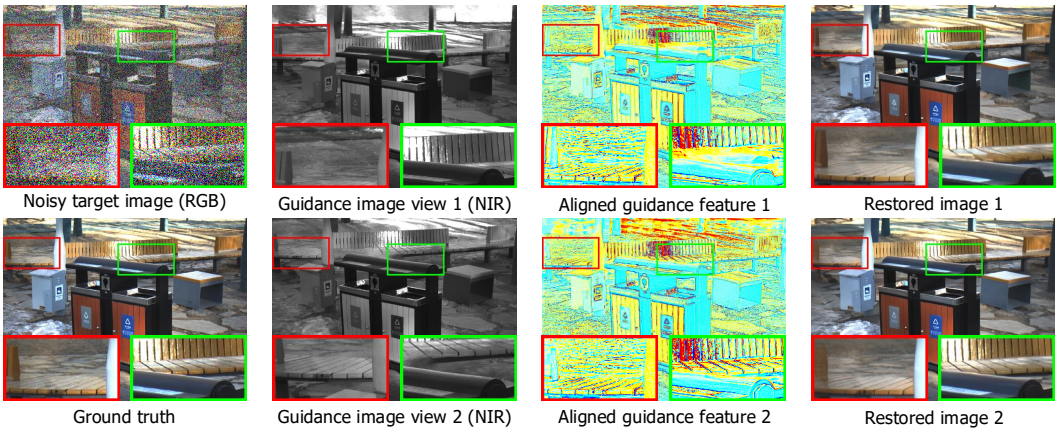


Figure 7: Impact of parallax-induced occlusion on guided denoising. The top and bottom rows present visualizations of the same sample under different parallax conditions. Red and green boxes highlight how occlusion affects feature alignment and reconstruction quality.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction are restricted to the proposed method, its target setting, and the evaluated experimental scenarios. These claims are supported by the method design and empirical results reported in the paper.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We have discussed the limitations in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [N/A]

Justification: The paper does not include formal theoretical claims or proofs. The proposed UGD-FM framework is evaluated empirically through quantitative comparisons, qualitative results, and ablation studies.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide the information needed to reproduce the main experimental results, including datasets, evaluation protocols, metrics, baselines, implementation details, and training settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Although the code is not released at submission time, the paper provides detailed descriptions of the model architecture, training objectives, datasets, and evaluation protocols. We plan to release the code and reproduction instructions after publication.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: We detail the training and testing aspects required to understand and reproduce the results, including data partitioning, hyperparameters, optimizers, training plans, and evaluation protocols. Explanations can be found in the main text or supplementary materials.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: We do not report error bars or formal statistical significance tests because repeated full training runs are computationally prohibitive for the proposed model. The main conclusions are supported by extensive evaluations on multiple datasets and standard benchmarks, using deterministic test protocols, multiple metrics, qualitative comparisons, and ablation studies.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provide the computational resources required for reproducing the experiments, including GPU type, memory and number of devices. Additional implementation and execution details are provided in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics and confirm that the research conforms to its requirements. The work uses publicly available datasets under standard research protocols, preserves submission anonymity, and does not involve human-subject studies, private data, or harmful applications.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss the potential positive and negative societal impacts of our work, including privacy concerns related to enhanced low-light imaging and the risks of generative hallucinations.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: This article does not disclose high-risk assets such as scraped datasets, private videos, large-scale generative models, or models intended for harmful use. UGD-FM is an unaligned guided denoising framework that has been evaluated on public datasets, and we discuss potential privacy and misuse issues in the **Broader Impacts** section.

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We properly cite the original papers and sources for the existing datasets, codebases, models, and evaluation tools used in this work. We follow their licenses and terms of use, and provide the relevant asset information in the paper and supplementary material.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [N/A]

Justification: This paper does not introduce any new assets such as new datasets or benchmarks. All experiments are conducted on existing datasets (FANVID, RGB-NIR Scene, and MS2).

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing experiments or research with human subjects. We do not recruit participants, conduct user studies, or collect new annotations from crowd workers; the experiments are conducted on existing public datasets.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects. We do not recruit participants, conduct user studies, or collect new human-subject data; the experiments are conducted on existing public datasets, so IRB approval or equivalent review is not applicable.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core methodology of UGD-FM does not include LLM as an important, original, or non-standard component. LLM is not used for method design, experimental conduct, or the derivation of scientific conclusions.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.