Improving Image-to-Image Translation via a Rectified Flow Reformulation

Satoshi Iizuka*, Shun Okamoto*, and Kazuhiro Fukui

University of Tsukuba, Japan

Abstract. In this work, we propose *Image-to-Image Rectified Flow Reformulation (I2I-RFR)*, a practical plug-in reformulation that recasts standard I2I regression networks as continuous-time transport models. While pixel-wise I2I regression is simple, stable, and easy to adapt across tasks, it often over-smooths ill-posed and multi-modal targets, whereas generative alternatives often require additional components, task-specific tuning, and more complex training and inference pipelines. Our method augments the backbone input by channel-wise concatenation with a noise-corrupted version of the ground-truth target and optimizes a simple t-reweighted pixel loss. This objective admits a rectified-flow interpretation via an induced velocity field, enabling ODE-based progressive refinement at inference time while largely preserving the standard supervised training pipeline. In most cases, adopting I2I-RFR requires only expanding the input channels, and inference can be performed with a few explicit solver steps (e.g., 3 steps) without distillation. Extensive experiments across multiple image-to-image translation and video restoration tasks show broad perceptual improvements, with possible perception–distortion trade-offs in highly pixel-aligned settings such as super-resolution.

Keywords: Image-to-image translation · Rectified flow · Regression reformulation · ODE-based refinement

1 Introduction

Image-to-Image (I2I) translation is a fundamental problem in computer vision, covering super-resolution, restoration, low-light enhancement, and many other tasks. A dominant and practical paradigm formulates I2I translation as supervised regression: given an input image x, a neural network predicts the target image y and is trained with pixel-wise losses such as ℓ_1 or mean squared error. This formulation is stable, easy to customize, and has benefited from a rich ecosystem of task-specific architectures, including U-Nets [33], residual designs [9], attention modules [40, 42], and multi-scale processing [20]. However, many I2I problems are intrinsically ill-posed: severe degradations, missing regions, or strong information loss often make the conditional distribution $p(y \mid x)$

* Equal contribution

Fig. 1: Comparison between conventional ℓ_1 regression and our ℓ_1 Rectified Flow Reformulation (RFR). Both use the same image-to-image network; the only difference is the number of input channels. RFR reduces regression artifacts and improves perceptual fidelity.

multi-modal. In such cases, pixel-wise regression tends to collapse toward a single point estimate, e.g., a conditional mean or median, yielding over-smoothed results, detail loss, or task-dependent artifacts, as shown in Fig. 1.

Generative objectives provide one way to address this limitation. Adversarial training can improve realism by adding a discriminator [8], but introduces extra networks, sensitive hyperparameters, and potential training instability. Diffusion models have also achieved strong image quality [10] and have been adapted to conditional I2I tasks such as super-resolution [35], general image-to-image translation [34], and inpainting [24]. Despite their ability to model multi-modal targets, diffusion-based I2I pipelines usually require diffusion-specific parameterizations, noise schedules, conditioning mechanisms, and iterative sampling with many denoising steps [10, 38]. Latent or pretrained variants further introduce autoencoders [14, 32] or large external generative priors, which can complicate direct reuse of task-specialized paired I2I backbones. This leaves a practical gap between simple regression models that are easy to train and adapt, and generative models that better address multi-modality but often require substantial additional design and computation.

In this work, we bridge this gap through the lens of Rectified Flow (RF), a continuous-time transport framework that learns a velocity field and generates samples by solving an ordinary differential equation (ODE) [22,23]. RF offers an appealing alternative to diffusion because its sampling dynamics are formulated as explicit ODE trajectories, enabling deterministic integration and controllable speed–quality trade-offs. However, existing RF formulations are mainly developed for unconditional or broadly conditioned generation, such as text-to-image synthesis [6], and typically rely on generic time-conditioned generative backbones. In contrast, paired I2I restoration has strong pixel-aligned conditioning and a long history of task-specialized regression architectures. Our key question is therefore: *can we obtain the progressive refinement benefits of continuous-time transport while preserving the simplicity, stability, and inductive biases of existing I2I regressors?*

We answer this question with *Image-to-Image Rectified Flow Reformulation (I2I-RFR)*, a plug-in training and inference scheme that converts standard I2I regression networks into RF-consistent ODE refiners. Given an input image x and target image y, we form a noisy target state $y_t = (1-t)y + t\varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I)$ and $t \in (0, 1]$, and feed the concatenation $[x; y_t]$ to an existing I2I backbone. The

network predicts the clean target y and is trained with a t-reweighted pixel loss. This objective induces the velocity field $v_\theta(x, y_t, t) = (y_t - f_\theta([x; y_t]))/t$, making the training objective equivalent to RF velocity regression while preserving the native clean-target prediction interface of standard I2I backbones. This induced-velocity parameterization is the key distinction of I2I-RFR and enables existing direct-regression architectures to be used as RF-consistent ODE refiners with minimal architectural changes.

At inference time, we solve the induced ODE by explicit Euler integration. Because the condition x provides strong spatial guidance, the state y_t acts as a refinement variable rather than a fully unconstrained generative latent, and a small number of steps is often sufficient in practice. I2I-RFR does not require a GAN discriminator, a latent autoencoder, a pretrained generative prior, explicit velocity/noise prediction, or a redesigned time-conditioned backbone; in most cases, applying it only requires expanding the input channels. We validate I2I-RFR across image and video restoration tasks, where it provides broad perceptual improvements and favorable perception–distortion trade-offs while reusing existing supervised I2I architectures.

2 Related Work

2.1 Supervised Image-to-Image Translation

Many problems in computer vision can be cast as image-to-image (I2I) translation, including super-resolution [5, 35], restoration [4, 26], low-light enhancement [7, 48], underwater image enhancement [39, 47], and video restoration [13, 25]. A large body of I2I research is developed under paired supervision and is often built on regression-style training with pixel-level objectives. This setting has produced a rich ecosystem of backbone architectures and modules that are tailored to specific degradations and application domains. Classic encoder–decoder designs such as U-Net [33] and residual learning [9] remain widely used as building blocks, while recent task-oriented backbones achieve strong performance by incorporating specialized attention, multi-scale processing, or frequency-domain priors. For example, SwinIR [18] targets image super-resolution with Transformer components, and Restormer [45] develops an efficient attention architecture for general image restoration. In domain-specific settings, task-tailored designs are often crucial; LEDNet [48] and DarkIR [7] are representative examples for low-light enhancement and low-light deblurring. These specialized I2I architectures constitute valuable engineering assets, motivating approaches that can improve perceptual quality while preserving the standard supervised regression interface and reusing existing backbones.

2.2 Perceptual and Adversarial Objectives

To enhance perceptual fidelity, prior work frequently augments pixel losses with feature-space objectives. Perceptual losses compare deep features extracted from

pretrained recognition networks such as VGG [37], and have been popularized for image translation tasks, including style transfer and super-resolution [12]. Learned perceptual metrics such as LPIPS [46] further motivate feature-based losses that correlate better with human judgments than pixel distances. While effective, perceptual losses often only partially mitigate multi-modality and may be insufficient under severe degradations or strong information loss. They also increase training cost due to feature extraction and require task-dependent choices of the feature network, layers, and loss weights.

Another widely used approach is adversarial training. GANs [8] introduce a discriminator to encourage outputs that match the target distribution, and conditional GAN formulations have been applied to I2I translation [11]. GAN-based super-resolution, exemplified by SRGAN [16], can improve realism and recover high-frequency details. However, adversarial objectives add extra networks and hyperparameters, and can be sensitive to the generator–discriminator balance, potentially leading to instability or artifacts and increasing engineering overhead when adapting to new tasks or backbones.

2.3 Diffusion Models for Conditional I2I Translation

Diffusion models provide a probabilistic alternative for multi-modal conditional targets. DDPM [10] trains a denoiser across noise levels, and DDIM [38] enables faster sampling via deterministic trajectories. Conditional diffusion models have been adapted to I2I tasks, including super-resolution (SR3) [35] and general I2I translation (Palette) [34]. These approaches often yield strong perceptual quality, but typically require diffusion-specific parameterizations, noise schedules, conditioning modules, and many-step sampling. Latent or pretrained variants further add autoencoders or external generative priors, which can complicate direct reuse of task-specific paired I2I backbones.

Recent prior-based restoration methods further exploit pretrained components, generative priors, semantic priors, or diffusion/flow-based formulations, e.g., ResShift [44], DiffBIR [21], SeeSR [43], and FlowIE [49]. These methods are powerful perceptual restorers, but they differ from our matched supervised setting in priors, model scale, and training regime. We therefore treat them as reference operating points rather than strictly matched train-from-scratch baselines, and compare their fidelity–perception–runtime trade-offs in Sec. 4.

2.4 Rectified Flow

Rectified Flow [23] is closely related to Flow Matching [22] and learns a time-dependent velocity field that transports samples between two endpoint distributions. Let $x_0 \sim \pi_0$ and $x_1 \sim \pi_1$ denote samples drawn from the source and target distributions, respectively. Following [23], the linear interpolation path is given by

$$x_t = (1-t)x_0 + tx_1, \qquad t \in [0,1]. \tag{1}$$

The velocity field is parameterized by a neural network $v_\theta(x_t, t)$ and trained by minimizing

$$\min_\theta \ \mathbb{E}\big[\, \|(x_1 - x_0) - v_\theta(x_t, t)\| \,\big], \tag{2}$$

where the expectation is taken over empirical draws of (x_0, x_1) and the sampling of t. For conditional generation, the same formulation applies by conditioning the velocity field on an external signal c and using $v_\theta(x_t, c, t)$, as in recent rectified-flow models for text-to-image generation [6]. After training, samples are generated by solving the ODE $dx_t = v_\theta(x_t, c, t)\, dt$, which transports π_0 to π_1, or vice versa via time reversal.

Most prior rectified-flow models are developed for unconditional or weakly conditioned synthesis (e.g., text-to-image) [6]. I2I-RFR differs by introducing a supervised image-to-image reformulation that preserves regression backbones and exploits pixel-aligned conditioning, making the ODE state a refinement variable rather than an unconstrained generative latent.

3 Proposed Method

We propose *Image-to-Image Rectified Flow Reformulation (I2I-RFR)*, a practical approach that connects standard supervised image regression models to rectified flow-consistent continuous-time models without redesigning their backbone architecture and training setup. The core idea is to realize rectified-flow transport using a conventional image regression network fed with a pair consisting of the conditioning input and a noise-corrupted target state. This plug-in reformulation requires only minimal changes to existing I2I models, yet provides a continuous-time refinement view that can substantially improve detail preservation across diverse I2I tasks.

3.1 Image-to-Image Rectified Flow Reformulation

Let f_θ be a generic I2I translation network that maps an input image $x \in \mathbb{R}^{C_{\text{in}} \times H \times W}$ to an output image in $\mathbb{R}^{C_{\text{out}} \times H \times W}$. Given paired training data (x, y) with $y \in \mathbb{R}^{C_{\text{out}} \times H \times W}$, conventional supervised training minimizes a pixel-wise regression objective:

$$\min_\theta \ \mathbb{E}\big[\, \|y - f_\theta(x)\|_p^p \,\big], \tag{3}$$

where $\|\cdot\|_p$ denotes the ℓ_p norm and $\|e\|_p^p = \sum_i |e_i|^p$. The parameter $p = 1$ corresponds to the ℓ_1 loss and $p = 2$ corresponds to the squared ℓ_2 loss. In the following, we set $p = 1$ throughout and denote the ℓ_1 norm by $\|\cdot\|$ for simplicity.

Building on a standard I2I regressor, we reformulate the rectified flow training objective in Eq. (2) along the interpolation path in Eq. (1) as

$$\min_{\theta'} \ \mathbb{E}\left[\left\| \frac{y - f_{\theta'}([x; y_t])}{t} \right\|\right]. \tag{4}$$

Algorithm 1 Training of image-to-image rectified flow reformulation

Require: dataset $\mathcal{D}$, I2I network $f_{\theta'}$, learning rate η, lower bound $t_{\min} > 0$ for t
Ensure: θ': updated model parameters
1: **repeat**
2: Sample (x, y) from $\mathcal{D}$
3: $u \sim \mathcal{U}(0, 1)$
4: $\varepsilon \sim \mathcal{N}(0, I)$
5: $t \leftarrow \max\left(u^{1/2},\, t_{\min}\right)$ ▷ Inverse-CDF Beta(2, 1) sampling
6: $y_t \leftarrow (1 - t)\, y + t\, \varepsilon$
7: $\theta' \leftarrow \theta' - \eta \nabla_{\theta'} \left\| \frac{y - f_{\theta'}([x; y_t])}{t} \right\|$ ▷ $[x; y_t]$: Concat(x, $\mathrm{y_t}$)
8: **until** convergence

Here, $[a; b]$ denotes channel-wise concatenation, and y_t is a noisy target obtained by mixing the ground-truth image y with Gaussian noise $\varepsilon \sim \mathcal{N}(0, I)$ using an interpolation parameter $t \in (0, 1]$:

$$y_t = (1 - t)\, y + t\, \varepsilon. \tag{5}$$

In this formulation, the image translation network $f_{\theta'}$ is identical to the original network f_θ except for the number of input channels to accommodate the concatenated input:

$$f_\theta : \mathbb{R}^{C_{\text{in}} \times H \times W} \to \mathbb{R}^{C_{\text{out}} \times H \times W},$$
$$f_{\theta'} : \mathbb{R}^{(C_{\text{in}} + C_{\text{out}}) \times H \times W} \to \mathbb{R}^{C_{\text{out}} \times H \times W}.$$

Consequently, I2I-RFR can be applied to existing regression models by (i) expanding the input-channel dimension and (ii) feeding the noise-corrupted targets alongside the inputs during training. This makes the reformulation compatible with a broad range of carefully engineered I2I architectures for tasks such as image super-resolution, restoration, and enhancement. Furthermore, I2I-RFR applies to video-to-video translation models in the same manner and yields gains on video restoration tasks. The training procedure is summarized in Algorithm 1.

Parameterization and equivalence to velocity-field training. Given paired data $(x, y) \sim \mathcal{D}$, we sample $t \in (0, 1]$ and $\varepsilon \sim \mathcal{N}(0, I)$. To match the rectified-flow notation, we set $x_0 := y$ and $x_1 := \varepsilon$, and form the linear path $x_t = (1 - t)x_0 + tx_1$. Along this path, the ground-truth velocity is constant, $v^\star = x_1 - x_0$, and we also have the identity $x_t - x_0 = t(x_1 - x_0)$. The rectified-flow dynamics are then obtained *implicitly* by defining an induced velocity field $v_{\theta'}(x, x_t, t) = \frac{x_t - f_{\theta'}([x; x_t])}{t}$. Using $x_t - x_0 = t(x_1 - x_0)$, we immediately obtain $(x_1 - x_0) - v_{\theta'}(x, x_t, t) = \frac{f_{\theta'}([x; x_t]) - x_0}{t}$. Therefore, minimizing the t-reweighted regression loss $\mathbb{E}\left[\left\| \frac{x_0 - f_{\theta'}([x; x_t])}{t} \right\|\right]$ is equivalent to minimizing the rectified-flow velocity regression objective $\mathbb{E}[\|(x_1 - x_0) - v_{\theta'}(x, x_t, t)\|]$ in Eq. (2), since $\|a\| = \|-a\|$. Note that the endpoint labeling is purely a convention: swapping (x_0, x_1) corresponds to the time reversal $t \mapsto 1 - t$ and flips the sign of the target velocity

$(v \mapsto -v)$, yielding an equivalent optimization problem. In this paper, we choose x_0 as the data endpoint and x_1 as the noise endpoint mainly for notational simplicity. In practice, for sampling we initialize at the prior endpoint and integrate backward in time. For numerical stability during training, we clamp $t \geq t_{\min}$ to avoid the singularity as $t \to 0$.

Time conditioning in I2I-RFR. Diffusion models and many rectified-flow implementations often condition the network on the noise level by feeding t through a time-embedding module. In I2I-RFR, however, the input is the channel-wise concatenation $[x; y_t]$: the input image x provides strong spatial priors about the target y, while the corruption level is directly reflected in y_t. As a result, explicit time embeddings are often unnecessary in our supervised I2I setting. Empirically, adding time embeddings provides no consistent improvement across our tasks and backbones, and can slightly degrade performance or increase optimization variance. We therefore omit explicit time embeddings in our default model, which simplifies implementation by avoiding an additional embedding module and facilitates the reuse of existing I2I backbones with minimal changes.

Sampling of the interpolation parameter. We construct noisy inputs via $y_t = (1-t)y + t\varepsilon$ and train the regression network $f_{\theta'}([x; y_t])$ with a loss that contains an explicit factor of t^{-p}, e.g., $\left\| \frac{y - f_{\theta'}([x;y_t])}{t} \right\|_p^p$. If t were sampled uniformly on $(0, 1]$, small t values would receive disproportionately large weight through t^{-p} and could induce high gradient variance. To stabilize optimization, we adopt a non-uniform sampling strategy based on a Beta distribution:

$$t \sim \mathrm{Beta}(p + 1,\, 1), \qquad t \leftarrow \max(t, t_{\min}), \tag{6}$$

whose density is proportional to t^p on $[0, 1]$ and thus counterbalances the t^{-p} factor in the loss, making the effective weighting approximately constant. We implement this sampling via inverse-CDF: $u \sim \mathcal{U}(0, 1)$ and $t = u^{1/(p+1)}$, followed by clamping to $t_{\min}$ to avoid numerical issues near $t = 0$. As a practical side effect, this sampling provides sufficient coverage of the high-noise regime ($t \approx 1$), which is important for robust refinement.

Fig. 2 provides an illustrative comparison of t-sampling strategies on low-light enhancement with LEDNet [48]. The plot shows the training loss and the validation PSNR as a function of training iterations for Beta$(2, 1)$, logit-normal$(0, 1)$, and uniform sampling under the same training budget. In this setting, Beta sampling converges more smoothly and achieves the best validation PSNR among the compared strategies. We use Beta sampling in all experiments unless stated otherwise.

Relationship to x_0-prediction. Predicting the clean endpoint, i.e., x_0-prediction, is a recognized parameterization in diffusion and flow-based models, although it is often not the default choice in large-scale generative settings. For example, DDPM reports that directly predicting x_0 led to worse sample quality in early experiments and adopts noise prediction as its default parameterization [10]; many subsequent diffusion models follow this convention [3]. Alternative targets such as v-prediction have also been proposed to improve stability under few-step

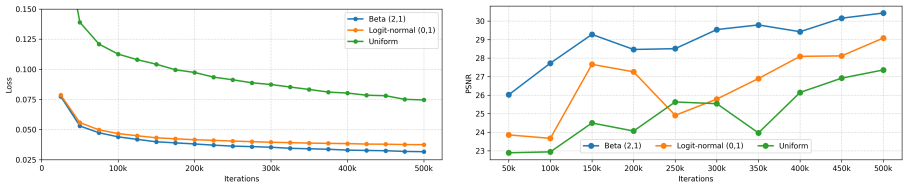


Fig. 2: Comparison of sampling strategies for training (left) and validation (right).

sampling [36]. More broadly, Flow Matching and Rectified Flow are typically formulated as regression of a time-dependent vector field [22, 23], including recent large-scale rectified-flow Transformers for text-to-image generation [6].

Our goal is different from advocating x_0-prediction as a universal parameterization for generative modeling [17]. We target supervised image-to-image translation, where the conditioning image x is pixel-aligned with the target and provides strong structural guidance. In this setting, the ODE state y_t acts more as a refinement variable than as an unconstrained generative latent, making clean-target prediction particularly compatible with existing I2I regression backbones. I2I-RFR preserves this native clean-image prediction interface while inducing the rectified-flow velocity analytically as $v_\theta(x, y_t, t) = (y_t - f_\theta([x; y_t]))/t$. Thus, the network remains a standard direct-target regressor, but the overall model is trained as an RF-consistent ODE refiner. As shown in Sec. 4, this design is not merely cosmetic: replacing the induced-velocity formulation with explicit v-prediction substantially degrades performance for several restoration backbones, suggesting that clean-target prediction is an I2I-specific parameterization choice.

3.2 Inference

Given an input image x, we generate the output by solving the induced rectified-flow ODE with a fixed-step explicit Euler integrator, as summarized in Algorithm 2. We initialize the state at the prior endpoint ($t = 1$) by sampling $y_t \sim \mathcal{N}(0, I)$, and integrate backward in time from $t = 1$ toward $t = 0$ using N uniform steps with step size $\Delta t = 1/N$. At step n, we set $t_n = 1 - \frac{n}{N}$ and evaluate the induced velocity field as $v = \frac{y_{t_n} - f_{\theta'}([x; y_{t_n}])}{t_n}$. We then update $y_{t_{n+1}} \leftarrow y_{t_n} - \Delta t\, v$, which corresponds to an explicit Euler discretization of the backward-time dynamics. Note that this update moves y_{t_n} toward the network prediction $f_{\theta'}([x; y_{t_n}])$, since $y_{t_n} - \Delta t \frac{y_{t_n} - f_{\theta'}([x; y_{t_n}])}{t_n} = y_{t_n} + \Delta t \frac{f_{\theta'}([x; y_{t_n}]) - y_{t_n}}{t_n}$. After N steps, the final state (at $t \approx 0$) is taken as the output $\hat{y}$.

Efficiency. In standard rectified flow, inference typically requires tens of ODE solver steps (e.g., 25 steps). Although this is fewer than the 50–250 steps commonly used by diffusion models with DDIM sampling, it can still be costly for large networks, interactive applications, and video processing. To reduce the number of steps, several distillation-based approaches have been proposed; however, they introduce an additional distillation stage and require careful tuning, leading to substantial engineering overhead. In contrast, our method achieves high-quality translations in only 3–5 steps *without* distillation. We attribute this

Algorithm 2 Inference of Image-to-Image Rectified Flow

Require: input image x, trained I2I network $f_{\theta'}$, number of time steps N
Ensure: output image $\hat{y}$
Sample $y_t \sim \mathcal{N}(0, I)$ ▷ initialization
for $n = 0$ to $N - 1$ **do**
  $t \leftarrow 1 - \frac{n}{N}$
  $v \leftarrow \frac{y_t - f_{\theta'}([x; y_t])}{t}$
  $y_t \leftarrow y_t - \frac{1}{N} v$
end for
$\hat{y} \leftarrow y_t$

efficiency to the informative spatial cues in the input image x, which make the ODE state y_t behave as a refinement variable rather than a fully unconstrained generative latent. By providing x alongside the current state via channel-wise concatenation, a small number of explicit Euler steps are often sufficient.

We use $N = 3$ steps as the default setting for all main experiments. Increasing N can further improve perceptual quality, while PSNR and SSIM may saturate or decrease, making the step count a practical perception–distortion control; detailed step ablations are reported in the supplement. The 3-step refinement increases runtime compared with one-pass regression, but the overhead remains moderate and the parameter increase is negligible: SwinIR 0.0785→0.1467s, LEDNet 0.0683→0.1928s, Restormer 0.3543→1.0436s, and RRTN 0.6408→1.3750s.

4 Results

I2I-RFR is evaluated on diverse image-to-image (I2I) and video-to-video translation tasks, including *image super-resolution*, *deblurring*, *low-light blur enhancement*, *underwater enhancement*, and *video restoration*. For each task, we apply I2I-RFR to strong recent backbones, including task-specialized CNN- and Transformer-based architectures. We also include a canonical U-Net [33] and the Palette [34] denoising U-Net backbone to assess robustness across model capacities and training pipelines. For "Palette backbone + I2I-RFR," we reuse only the Palette backbone and replace its multi-step ϵ-prediction DDPM pipeline with our direct-target objective and 3-step Euler inference. This comparison highlights the practical difference between I2I-RFR and diffusion-based I2I pipelines, which typically require noise schedules, explicit noise-level conditioning, and many-step denoising trajectories.

Unless otherwise specified, our matched comparisons use models trained from scratch under paired supervision or official task-specific checkpoints noted per task. Methods relying on large external generative or semantic priors are excluded from the matched baseline comparisons and are reported separately only as reference operating points in Table 6. The goal is not to exhaustively compete with every task-specific state-of-the-art, but to demonstrate that I2I-RFR

Table 1: Image super-resolution results on standard benchmarks. Best values are in red, and second-best values are in blue. Values in parentheses denote the relative changes compared to the original model.

	Set5			Set14			BSD100			Urban100		
Model	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
NLSN [27]	32.84	0.9020	0.1694	29.05	0.7920	0.2715	27.87	0.7459	0.3595	27.22	0.8176	0.1945
NLSN + I2I-RFR	32.58 (-0.26)	0.8998 (-0.0022)	0.1676 **(-0.0018)**	28.88 (-0.17)	0.7888 (-0.0032)	0.2662 **(-0.0053)**	27.75 (-0.12)	0.7430 (-0.0029)	0.3498 **(-0.0097)**	26.83 (-0.39)	0.8079 (-0.0097)	0.1996 (+0.0051)
SwinIR [18]	32.80	0.9025	0.1687	29.00	0.7922	0.2703	27.89	0.7469	0.3578	27.24	0.8199	0.1948
SwinIR + I2I-RFR	32.84 **(+0.04)**	0.9028 **(+0.0003)**	0.1665 **(-0.0022)**	29.01 **(+0.01)**	0.7916 (-0.0006)	0.2621 **(-0.0082)**	27.85 (-0.04)	0.7461 (-0.0008)	0.3497 **(-0.0081)**	27.19 (-0.05)	0.8180 (-0.0019)	0.1897 **(-0.0051)**
U-Net [33]	32.09	0.8934	0.1767	28.57	0.7806	0.2891	27.53	0.7347	0.3795	25.97	0.7810	0.2414
U-Net + I2I-RFR	31.63 (-0.46)	0.8884 (-0.0050)	0.1665 **(-0.0102)**	28.16 (-0.41)	0.7704 (-0.0102)	0.2606 **(-0.0285)**	27.18 (-0.35)	0.7244 (-0.0103)	0.3404 **(-0.0391)**	25.55 (-0.42)	0.7707 (-0.0103)	0.2218 **(-0.0196)**
Palette [34]	22.28	0.6582	0.2079	21.49	0.5139	0.3059	24.72	0.6063	0.3967	19.58	0.5193	0.3035
Palette backbone + I2I-RFR	32.09 **(+9.81)**	0.8959 **(+0.2377)**	0.1695 **(-0.0384)**	28.47 **(+6.98)**	0.7788 **(+0.2649)**	0.2459 **(-0.0600)**	27.40 **(+2.68)**	0.7339 **(+0.1276)**	0.3299 **(-0.0668)**	26.40 **(+6.82)**	0.7973 **(+0.2780)**	0.1914 **(-0.1121)**

provides broad perceptual improvements across diverse tasks and backbones under simple and stable supervised training. In addition, while I2I-RFR performs strongly with an ℓ_1 loss alone, a variant with commonly used composite losses is also reported to show that the reformulation is compatible with additional objectives such as perceptual losses.

Implementation details. Unless otherwise specified, all models are optimized with Adam using an initial learning rate of 10^{-4} and a cosine learning-rate schedule down to 10^{-6}. Task-dependent settings such as input resolution, batch size, data augmentation, and training iterations follow standard practice for each benchmark and backbone. For I2I-RFR, t is sampled as $t \sim \mathrm{Beta}(2, 1)$ with $t_{\min} = 10^{-3}$, and the default loss is the t^{-1}-reweighted ℓ_1 distance. At inference time, the induced ODE is solved with a fixed-step explicit Euler integrator using $N = 3$ steps for all tasks unless stated otherwise. Classifier-free guidance is not used, and explicit time embedding modules are omitted to keep the architecture changes minimal and preserve direct reuse of existing regression backbones.

Image Super-Resolution. Image super-resolution aims to recover high-resolution (HR) images from low-resolution (LR) inputs. Representative regression-based SR models, NLSN [27] and SwinIR [18], are selected as baselines, where the former is CNN-based and the latter is Transformer-based. I2I-RFR is applied to both models by expanding the input layer to accept a 6-channel input. Both the original models and their I2I-RFR variants are trained on DF2K [1]. During training, HR patches are randomly cropped to 256×256 pixels, and the corresponding LR inputs are generated by $\times 4$ bicubic downsampling. For I2I-RFR, the interpolation state is constructed in the LR domain to match the model input shape: the network is fed with $[x^{\mathrm{LR}}; y_t^{\mathrm{LR}}]$, while the prediction target and the loss are defined in the HR domain, i.e., the model predicts the clean HR image y^{HR} from the LR input state.

Table 1 shows that I2I-RFR often improves LPIPS across backbones, while PSNR and SSIM can remain similar or slightly decrease for some models, reflecting the perception–distortion trade-off in ill-posed SR [2]. In particular, SwinIR with I2I-RFR provides a favorable balance between distortion and perceptual metrics among the matched supervised baselines. Notably, applying I2I-RFR to the Palette denoising backbone substantially improves SR quality, suggesting that the backbone capacity is not the main bottleneck and that the regression-

Table 2: Image deblurring results on RealBlur-J benchmark [31]. Best values are in red, and second-best values are in blue. Values in parentheses denote the relative changes compared to the original model.

Model	PSNR↑	SSIM↑	LPIPS↓
NAFNet [4]	28.62	0.8683	0.1557
NAFNet + I2I-RFR	28.74 (+**0.12**)	0.8737 (+**0.0054**)	0.1449 (**-0.0108**)
Restormer [45]	28.73	0.8677	0.1522
Restormer + I2I-RFR	29.04 (+**0.31**)	0.8812 (+**0.0135**)	0.1408 (**-0.0114**)
FFTFormer [15]	28.19	0.8535	0.1691
FFTFormer + I2I-RFR	28.58 (+**0.39**)	0.8685 (+**0.0150**)	0.1487 (**-0.0204**)
U-Net [33]	28.03	0.8440	0.1848
U-Net + I2I-RFR	27.93 (-0.10)	0.8476 (+**0.0036**)	0.1753 (**-0.0095**)
Palette [34]	26.29	0.6313	0.2826
Palette backbone + I2I-RFR	29.05 (+**2.76**)	0.8831 (+**0.2518**)	0.1436 (**-0.1390**)

preserving transport formulation can be effective even when starting from a diffusion-oriented U-Net.

Image Deblurring. We evaluate I2I-RFR on three representative deblurring backbones, NAFNet [4], Restormer [45], and FFTFormer [15]. Training follows the GoPro protocol [29], and evaluation is performed on RealBlur-J [31], providing a cross-dataset test relative to GoPro training. I2I-RFR is applied by expanding the input channels from 3 to 6 and replacing the pixel loss with its t-reweighted version, without modifying the backbone structure. Metrics include PSNR, SSIM, and LPIPS. As shown in Table 2, I2I-RFR consistently improves perceptual quality over the baseline models.

Low-Light Blur Enhancement. Low-light blur enhancement targets images that are simultaneously under-exposed and blurred (e.g., due to motion or defocus). The goal is to recover a clean, well-lit image by correcting illumination and removing blur to restore sharp edges and fine textures. Because both degradations destroy information and amplify noise, the task is highly ill-posed and requires balancing detail recovery against artifact suppression.

We evaluate I2I-RFR on two strong low-light deblurring models, LEDNet [48] and DarkIR [7], following the LOLBlur training and evaluation protocol [48]. Official checkpoints are used as reference baselines, and the corresponding I2I-RFR variants are trained on the same LOLBlur training split. For the composite-loss setting, the original loss design is kept and only the pixel term is replaced by its t-reweighted counterpart, denoted as I2I-RFR-CL. Evaluation is performed on the LOLBlur test set using PSNR, SSIM, and LPIPS.

Table 3 shows clear gains for LEDNet and perceptual/SSIM improvements for DarkIR, while the composite-loss variant further improves DarkIR across all metrics and achieves state-of-the-art results on LOLBlur. A direct same-backbone adversarial comparison further shows that LEDNet+I2I-RFR improves PSNR, SSIM, and LPIPS over LEDNet+Adv., while avoiding an additional discriminator. This setting is strongly ill-posed due to the combination of under-exposure and blur, and the progressive refinement view helps suppress regression artifacts and recover sharper structures. Moreover, the I2I-RFR-CL variant demonstrates that the reformulation is compatible with task-specific composite objectives: keeping the original loss design and reweighting only the pixel term can further improve perceptual quality for DarkIR.

Table 3: Low-light blur enhancement results on LOLBlur test set. Best values are in red, and second-best values are in blue. Values in parentheses denote the relative changes compared to the original model.

Model	PSNR↑	SSIM↑	LPIPS↓
LEDNet [48]	26.04	0.845	0.159
LEDNet + Adv.	24.55	0.811	0.153
LEDNet + I2I-RFR	27.42 **(+1.38)**	0.874 **(+0.029)**	0.138 **(-0.021)**
LEDNet + I2I-RFR-CL	26.56 **(+0.52)**	0.869 **(+0.024)**	0.140 **(-0.019)**
DarkIR [7]	27.31	0.896	0.111
DarkIR + I2I-RFR	27.19 (-0.12)	0.897 **(+0.001)**	0.106 **(-0.005)**
DarkIR + I2I-RFR-CL	27.72 **(+0.41)**	0.900 **(+0.004)**	0.103 **(-0.008)**
U-Net [33]	20.45	0.764	0.267
U-Net + I2I-RFR	22.48 **(+2.03)**	0.826 **(+0.062)**	0.184 **(-0.083)**
Palette [34]	8.03	0.007	0.936
Palette backbone + I2I-RFR	26.88 **(+18.85)**	0.875 **(+0.868)**	0.141 **(-0.795)**

Table 4: Underwater image enhancement results on LSUI test set and UIEB test set. Best values are in red, and second-best values are in blue. Values in parentheses denote the relative changes compared to the original model.

	LSUI Test Set			UIEB Test Set		
Model	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
Tang et al. [39]	18.34	0.8320	0.1413	14.63	0.7110	0.2339
Tang et al. + I2I-RFR	29.27 **(+10.93)**	0.9381 **(+0.1061)**	0.0838 **(-0.0575)**	19.56 **(+4.93)**	0.8462 **(+0.1352)**	0.1637 **(-0.0702)**
SFGNet [47]	27.54	0.9293	0.0834	18.82	0.8315	0.1738
SFGNet + I2I-RFR	26.09 (-1.45)	0.9190 (-0.0103)	0.1164 (+0.0330)	19.69 **(+0.87)**	0.8515 **(+0.0200)**	0.1640 **(-0.0098)**
SFGNet + I2I-RFR-CL	26.40 (-1.14)	0.9203 (-0.0090)	0.0907 (+0.0073)	19.63 **(+0.81)**	0.8470 **(+0.0155)**	0.1674 **(-0.0064)**
U-Net [33]	27.42	0.9244	0.1110	17.20	0.7957	0.2352
U-Net + I2I-RFR	28.34 **(+0.92)**	0.9322 **(+0.0078)**	0.0893 **(-0.0217)**	17.69 **(+0.49)**	0.8092 **(+0.0135)**	0.2203 **(-0.0149)**
Palette [34]	23.36	0.7424	0.1838	17.73	0.6542	0.3113
Palette backbone + I2I-RFR	29.26 **(+5.90)**	0.9376 **(+0.1952)**	0.0866 **(-0.0972)**	20.00 **(+2.27)**	0.8469 **(+0.1927)**	0.1632 **(-0.1481)**

Underwater Image Enhancement. We evaluate I2I-RFR on the LSUI benchmark [30] using two strong recent baselines: the diffusion-based method by Tang *et al.* [39] and SFGNet [47]. All models are trained on the LSUI training split and evaluated on the LSUI test set; in addition, we report results on the UIEB test set to assess cross-dataset generalization. For Tang *et al.*, the original DDPM training is replaced with I2I-RFR training using the same denoising backbone, enabling a direct architectural comparison. For SFGNet, I2I-RFR augments the input with the noisy target state while keeping the backbone unchanged.

As shown in Table 4, I2I-RFR improves perceptual quality for most baselines and evaluation settings, including clear gains for the diffusion backbone retrained with I2I-RFR and consistent improvements on UIEB. The only exception is SFGNet on the LSUI test set, where I2I-RFR slightly degrades distortion metrics. A plausible explanation is that SFGNet already incorporates strong task-specific priors and tightly coupled frequency- and gradient-based refinement, which may be sensitive to the additional noise-corrupted target state and the t-reweighted objective. In this case, the reformulation can perturb the backbone's carefully balanced internal processing, whereas the composite-loss variant better preserves the original training recipe and recovers most of the gain.

Video Restoration. We evaluate I2I-RFR on old film restoration using RTN [41] and the more recent RRTN [19]. Training follows the RRTN protocol on REDS [28]

Table 5: Video restoration results on DAVIS test set. Best values are in red, and second-best values are in blue. Values in parentheses denote the relative changes compared to the original model.

Model	PSNR↑	SSIM↑	LPIPS↓
Input	20.62	0.7201	0.3717
RTN [41]	24.54	0.8294	0.1562
RTN + I2I-RFR	25.05 (+**0.51**)	0.8330 (+**0.0036**)	0.2100 (+0.0538)
RTN + I2I-RFR-CL	25.52 (+**0.98**)	0.8395 (+**0.0101**)	0.1728 (+0.0166)
RRTN [19]	22.66	0.7453	0.2379
RRTN + I2I-RFR	25.87 (+**3.21**)	0.8659 (+**0.1206**)	0.1699 (**-0.0680**)
RRTN + I2I-RFR-CL	25.80 (+**3.14**)	0.8505 (+**0.1052**)	0.1456 (**-0.0923**)

Table 6: Average SR comparison with prior-based restoration references over Set5, Set14, BSD100, and Urban100. ResShift is trained from scratch using a pretrained VQGAN autoencoder, whereas DiffBIR, SeeSR, and FlowIE use official checkpoints. These methods are treated as reference operating points rather than strictly matched baselines.

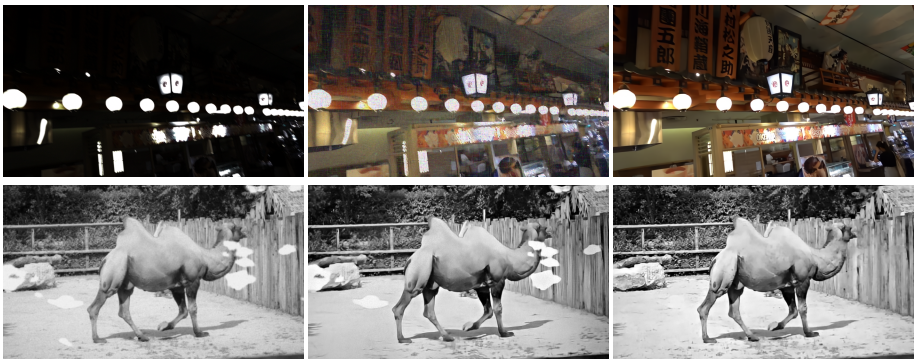
Method	Prior/module	Params	Steps	Time	PSNR↑	SSIM↑	LPIPS↓
SwinIR + I2I-RFR	**None**	**12M**	3	**0.1467s**	**29.22**	**0.8146**	0.2420
ResShift [44]	VQGAN AE	174M	15	0.5876s	26.97	0.7518	**0.1458**
DiffBIR [21] (official)	SD2.1	1682M	50	6.608s	25.17	0.6805	0.2165
SeeSR [43] (official)	SD2/RAM	2510M	50	8.228s	25.49	0.7115	0.1958
FlowIE [49] (official)	SD2.1	1683M	**1**	0.6210s	24.13	0.7189	0.2382

with the same synthetic degradation pipeline, and evaluation is performed on DAVIS with matched degradations. I2I-RFR is applied by expanding the input channels and replacing the pixel loss with its t-reweighted version. For the composite-loss setting, the original loss design is kept and only the pixel term is reweighted, denoted as I2I-RFR-CL. Metrics include PSNR, SSIM, and LPIPS. As shown in Table 5, I2I-RFR improves distortion metrics across both backbones and yields strong perceptual gains for RRTN, with some perception–distortion trade-offs for RTN.

Reference comparison with prior-based restoration methods. Although our main comparisons focus on matched train-from-scratch supervised settings, we also report a compact reference comparison with recent prior-based restoration methods on the SR benchmarks. ResShift is trained from scratch in our setting while using a pretrained VQGAN autoencoder, whereas DiffBIR, SeeSR, and FlowIE are evaluated using official checkpoints. We treat these methods as reference operating points rather than strictly matched baselines, since they rely on external generative or semantic priors and broader training data. As shown in Table 6, ResShift [44] achieves the best LPIPS, confirming the strength of specialized generative priors. In contrast, SwinIR+I2I-RFR achieves substantially higher paired fidelity with only 12M parameters, 3 Euler steps, no latent autoencoder, and no pretrained generative prior. This comparison clarifies the operating point of I2I-RFR: lightweight supervised adaptation and faithful paired restoration, rather than dominance over pretrained generative restorers in every perceptual metric.

Table 7: Parameterization effect. Best values are in red, and second-best are in blue. Values in parentheses indicate the difference from the original baseline.

	PSNR↑	SSIM↑	LPIPS↓
Low-light blur enhancement results on LOLBlur test set.			
LEDNet + I2I-RFR	27.42 **(+1.38)**	0.874 **(+0.029)**	0.138 **(-0.021)**
LEDNet + I2I-RFR v-pred	11.57 (-14.47)	0.060 (-0.785)	0.803 (+0.644)
DarkIR + I2I-RFR	27.19 (-0.12)	0.897 **(+0.001)**	0.106 **(-0.005)**
DarkIR + I2I-RFR v-pred	16.73 (-10.58)	0.431 (-0.465)	0.512 (+0.401)
Old film restoration results on DAVIS test set.			
RTN + I2I-RFR	25.05 **(+0.51)**	0.8330 **(+0.0036)**	0.2100 (+0.0538)
RTN + I2I-RFR v-pred	20.76 (-3.78)	0.7127 (-0.1167)	0.3025 (+0.1463)
RRTN + I2I-RFR	25.87 **(+3.21)**	0.8659 **(+0.1206)**	0.1699 **(-0.0680)**
RRTN + I2I-RFR v-pred	24.72 **(+2.06)**	0.8566 **(+0.1113)**	0.1565 **(-0.0814)**



Input I2I-RFR v-pred I2I-RFR

Fig. 3: Comparison between v-prediction and our direct target prediction in I2I-RFR for low-light blurry image enhancement (top) and video restoration (bottom). v-prediction produces severe artifacts, whereas direct target prediction yields more plausible enhancements. **Zoom in for best view.**

Parameterization Study. I2I-RFR is formulated as a direct prediction of the clean target image. For comparison, the same backbones can also be trained in a velocity-prediction form by regressing the rectified-flow velocity, since the backbone interface is unchanged up to the choice of training target.

The comparison between direct target prediction and v-prediction is conducted on low-light blur enhancement and video restoration. These tasks are representative ill-posed I2I settings in which perceptual fidelity and optimization stability are critical, and they jointly cover both image and video translation. The training data and optimization setup are kept identical for both variants; the only difference lies in the prediction target and the corresponding loss parameterization. As shown in Table 7 and Fig. 3, v-prediction leads to a substantial performance drop in most cases.

This comparison provides a controlled parameterization analysis: the data, backbone, and optimization setup are fixed, and only the prediction target and the corresponding loss are changed. Our formulation predicts the clean image y and induces the velocity analytically, so the network remains in the original clean-image regression regime. In contrast, explicit v-prediction regresses $v^\star =$

Table 8: Effect of time embeddings. Values in parentheses indicate the difference from the original baseline.

	PSNR↑	SSIM↑	LPIPS↓
DarkIR + I2I-RFR	27.19 (-0.12)	0.897 **(+0.001)**	0.106 **(-0.005)**
DarkIR + I2I-RFR + time embedding	26.80 (-0.51)	0.877 (-0.019)	0.127 (+0.016)

$(y_t - y)/t = \varepsilon - y$, a noise-containing velocity field that must preserve residual and noise information. This target is poorly aligned with restoration backbones designed to suppress degradations and output clean images, especially frequency-, gradient-, or degradation-suppression architectures such as LEDNet, DarkIR, and SFGNet. The consistent degradation across multiple backbones suggests a systematic parameterization mismatch rather than an isolated optimization failure. Thus, direct clean-target prediction is an I2I-specific design choice, not a cosmetic alternative to standard RF v-prediction.

Effect of time embeddings. Table 8 shows a case study on low-light deblurring with DarkIR, where we add sinusoidal time embeddings to intermediate features as commonly done in diffusion and rectified-flow models. Time embeddings degrade performance across all metrics in this setting. A likely reason is that *I2I*-RFR already exposes the corruption level through the concatenated input $[x; y_t]$, making explicit time conditioning redundant and sometimes less stable. We therefore omit time embeddings by default.

5 Conclusion

This work introduced *Image-to-Image Rectified Flow Reformulation* (I2I-RFR), a practical plug-in reformulation that equips supervised image-to-image and video-to-video regression models with few-step ODE-based refinement while largely preserving the original backbone interface. By feeding a noise-corrupted target state alongside the input and training with a simple t-reweighted pixel loss, I2I-RFR reuses the inductive biases of task-specialized backbones and improves perceptual quality and detail preservation across diverse benchmarks in a practical few-step inference regime. Future work includes extending the evaluation to broader domains and tasks, such as illustration and other non-photographic data, and further studying how the reformulation interacts with task-specific modules, composite loss designs, and inference depth.

Acknowledgements

This work was supported by JSPS KAKENHI Grant Number JP25K15174.

References

1. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2017)

2. Blau, Y., Michaeli, T.: The perception-distortion tradeoff. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 6228–6237 (2018). https://doi.org/10.1109/CVPR.2018.00652, https://openaccess.thecvf.com/content_cvpr_2018/html/Blau_The_Perception-Distortion_Tradeoff_CVPR_2018_paper.html
3. Cao, H., Tan, C., Gao, Z., Xu, Y., Chen, G., Heng, P.A., Li, S.Z.: A survey on generative diffusion model. arXiv preprint arXiv:2209.02646 (2022)
4. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: European conference on computer vision. pp. 17–33. Springer (2022)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Image super-resolution using deep convolutional networks. IEEE TPAMI **38**(2), 295–307 (2016)
6. Esser, P., Kulal, S., Blattmann, A., Entezari, R., Müller, J., Saini, H., Levi, Y., Lorenz, D., Sauer, A., Boesel, F., Podell, D., Dockhorn, T., English, Z., Rombach, R.: Scaling rectified flow transformers for high-resolution image synthesis. In: Proceedings of the 41st International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 235, pp. 12606–12633. PMLR (2024)
7. Feijoo, D., Benito, J.C., Garcia, A., Conde, M.V.: DarkIR: Robust low-light image restoration. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR). pp. 10879–10889 (June 2025)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A.C., Bengio, Y.: Generative adversarial nets. In: NeurIPS. pp. 2672–2680 (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
10. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Advances in Neural Information Processing Systems. vol. 33, pp. 6840–6851 (2020)
11. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR) (2017)
12. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV. pp. 694–711 (2016). https://doi.org/10.1007/978-3-319-46475-6_43
13. Kim, T.H., Sajjadi, M.S.M., Hirsch, M., Schölkopf, B.: Spatio-temporal transformer network for video restoration. In: ECCV (2018)
14. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: International Conference on Learning Representations (ICLR) (2014)
15. Kong, L., Dong, J., Ge, J., Li, M., Pan, J.: Efficient frequency domain-based transformers for high-quality image deblurring. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5886–5895 (2023)
16. Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., Shi, W.: Photo-realistic single image super-resolution using a generative adversarial network. In: CVPR. pp. 4681–4690 (2017). https://doi.org/10.1109/CVPR.2017.19
17. Li, T., He, K.: Back to basics: Let denoising generative models denoise. arXiv preprint arXiv:2511.13720 (2025)
18. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: SwinIR: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1833–1844 (2021)

19. Lin, S., Simo-Serra, E.: Restoring degraded old films with recursive recurrent transformer networks. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 6718–6728 (2024)
20. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2117–2125 (2017). https://doi.org/10.1109/CVPR.2017.106
21. Lin, X., He, J., Chen, Z., Lyu, Z., Dai, B., Yu, F., Qiao, Y., Ouyang, W., Dong, C.: DiffBIR: Toward blind image restoration with generative diffusion prior. In: Proceedings of the European Conference on Computer Vision (ECCV) (2024). https://doi.org/10.1007/978-3-031-73202-7_25
22. Lipman, Y., Chen, R.T.Q., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: International Conference on Learning Representations (ICLR) (2023), notable Top 25%
23. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations (ICLR) (2023), notable Top 25%
24. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 11451–11461 (2022). https://doi.org/10.1109/CVPR52688.2022.01117
25. Maggioni, M., Boracchi, G., Foi, A., Egiazarian, K.: Video denoising, deblocking, and enhancement through separable 4-d nonlocal spatiotemporal transforms. IEEE TIP **21**(9), 3952–3966 (2012)
26. Mao, X., Li, Q., Wang, Y.: Adarevd: Adaptive patch exiting reversible decoder pushes the limit of image deblurring. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 25681–25690 (2024)
27. Mei, Y., Fan, Y., Zhou, Y.: Image super-resolution with non-local sparse attention. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3517–3526 (2021)
28. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Lee, K.M.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 1996–2005 (June 2019). https://doi.org/10.1109/CVPRW.2019.00251
29. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
30. Peng, L., Zhu, C., Bian, L.: U-shape transformer for underwater image enhancement. IEEE Transactions on Image Processing **32**, 3066–3079 (2023). https://doi.org/10.1109/TIP.2023.3276332
31. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV. p. 184–201. Springer-Verlag, Berlin, Heidelberg (2020). https://doi.org/10.1007/978-3-030-58595-2_12, https://doi.org/10.1007/978-3-030-58595-2_12
32. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10674–10685 (2022). https://doi.org/10.1109/CVPR52688.2022.01042

33. Ronneberger, O., Fischer, P., Brox, T.: U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Lecture Notes in Computer Science, vol. 9351, pp. 234–241. Springer (2015). https://doi.org/10.1007/978-3-319-24574-4_28
34. Saharia, C., Chan, W., Chang, H., Lee, C.A., Ho, J., Salimans, T., Fleet, D.J., Norouzi, M.: Palette: Image-to-image diffusion models. ACM Transactions on Graphics **41**(6), 15:1–15:10 (2022). https://doi.org/10.1145/3528233.3530757
35. Saharia, C., Ho, J., Chan, W., Salimans, T., Fleet, D.J., Norouzi, M.: Image super-resolution via iterative refinement. arXiv preprint arXiv:2104.07636 (2021)
36. Salimans, T., Ho, J.: Progressive distillation for fast sampling of diffusion models. In: International Conference on Learning Representations (ICLR) (2022), spotlight
37. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)
38. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. In: International Conference on Learning Representations (ICLR) (2021)
39. Tang, Y., Kawasaki, H., Iwaguchi, T.: Underwater image enhancement by transformer-based diffusion model with non-uniform sampling for skip strategy. In: Proceedings of the 31st ACM international conference on multimedia. pp. 5419–5427 (2023)
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30, pp. 5998–6008 (2017)
41. Wan, Z., Zhang, B., Chen, D., Liao, J.: Bringing old films back to life. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 17694–17703 (2022)
42. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7794–7803 (2018). https://doi.org/10.1109/CVPR.2018.00813
43. Wu, R., Yang, T., Sun, L., Zhang, Z., Li, S., Zhang, L.: SeeSR: Towards semantics-aware real-world image super-resolution. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 25456–25467 (2024)
44. Yue, Z., Wang, J., Loy, C.C.: ResShift: Efficient diffusion model for image super-resolution by residual shifting. Advances in neural information processing systems **36**, 13294–13307 (2023)
45. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5728–5739 (2022)
46. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: CVPR. pp. 586–595 (2018). https://doi.org/10.1109/CVPR.2018.00068
47. Zhao, C., Cai, W., Dong, C., Zeng, Z.: Toward sufficient spatial-frequency interaction for gradient-aware underwater image enhancement. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 3220–3224. IEEE (2024)
48. Zhou, S., Li, C., Change Loy, C.: LEDNet: Joint low-light enhancement and deblurring in the dark. In: European conference on computer vision. pp. 573–589. Springer (2022)

49. Zhu, Y., Zhao, W., Li, A., Tang, Y., Zhou, J., Lu, J.: FlowIE: Efficient image enhancement via rectified flow. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13–22 (June 2024)