Decoupled Residual Denoising Diffusion Models for Unified and Data Efficient Image-to-Image Translation

Supplementary Material

A. Derivations and Proofs

A.1. Proofs of Proposition 3.1

In this section, we give a proof to the aforementioned **proposition 3.1**.

Proposition 3.1. *Let P and Q be two distinct probability distributions over a space $\mathcal{X}$. Suppose that we inject Gaussian noise $\mathcal{N}(0, \sigma^2)$ (with $\sigma \neq 0$) to both distributions and denote P_σ and Q_σ as the resulting distributions. Then, the Kullback-Leibler (KL) divergence between P_σ and Q_σ is less than the KL divergence between P and Q:*

$$D_{KL}(P_\sigma \parallel Q_\sigma) < D_{KL}(P \parallel Q) \tag{13}$$

Proof: Let P and Q be two distinct probability distributions ($P \neq Q$) with corresponding probability density functions $p(x)$ and $q(x)$. The KL divergence is defined as:

$$D_{KL}(P \parallel Q) = \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \tag{14}$$

where the integral is taken over the entire support of x. To add Gaussian noise to each data distribution in a random manner, we consider a Gaussian kernel $K(y|x)$, which represents the probability of output y given input x. The Gaussian kernel is defined as:

$$K(y|x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y-x)^2}{2\sigma^2} \right), \tag{15}$$

where σ^2 is the variance of the noise. Adding Gaussian noise, the distributions for P and Q are modified to P_σ and Q_σ, with the corresponding probability densities:

$$p_\sigma(y) = \int K(y|x)p(x)dx, \tag{16}$$

$$q_\sigma(y) = \int K(y|x)q(x)dx. \tag{17}$$

According to the definition:

$$\begin{aligned} D_{KL}(P_\sigma \parallel Q_\sigma) &= \int p_\sigma(y) \log \left(\frac{p_\sigma(y)}{q_\sigma(y)} \right) dy \\ &= D_{KL}(P(Y) \parallel Q(Y)) \end{aligned} \tag{18}$$

Define the joint density $p(x, y) = p(x)K(y|x)$ and $q(x, y) = q(x)K(y|x)$. Note that due to Gaussian noise being independent, $K(y|x)$ does not depend on p or q, so $p(x|y) = K(y|x)$ for both P and Q. Here, KL divergence $D_{KL}(P \parallel Q)$ is related to x, while $D_{KL}(P_\sigma \parallel Q_\sigma)$ is related to y. We can now write the joint KL divergence as:

$$D_{KL}(P(X,Y) \parallel Q(X,Y)) = \int p(x,y) \log \left(\frac{p(x,y)}{q(x,y)} \right) dxdy \tag{19}$$

Since

$$\frac{p(x,y)}{q(x,y)} = \frac{p(x)K(y|x)}{q(x)K(y|x)} = \frac{p(x)}{q(x)} \tag{20}$$

$$\begin{aligned} &D_{KL}(P(X,Y) \parallel Q(X,Y)) \\ &= \iint p(x)K(y|x) \log \left(\frac{p(x)}{q(x)} \right) dxdy \quad \text{(Eq. 20)} \\ &= \int p(x) \log \left(\frac{p(x)}{q(x)} \right) dx \quad \left(\int K(y|x)dy = 1 \right) \\ &= D_{KL}(P \parallel Q) \end{aligned} \tag{21}$$

The joint KL divergence can be decomposed as:

$$\begin{aligned} &D_{KL}(P(X,Y) \parallel Q(X,Y)) = D_{KL}(P(Y) \parallel Q(Y)) \\ &+ D_{KL}(P(X|Y) \parallel Q(X|Y)|P(Y)) \end{aligned} \tag{22}$$

where $D_{KL}(P(Y) \parallel Q(Y))$ is the KL divergence of the marginals P_σ and Q_σ, i.e., $D_{KL}(P_\sigma \parallel Q_\sigma)$. The second term $D_{KL}(P(X|Y) \parallel Q(X|Y)|P(Y))$ is positive ($P \neq Q$), hence

$$D_{KL}(P(X,Y) \parallel Q(X,Y)) < D_{KL}(P(Y) \parallel Q(Y)) \tag{23}$$

According to Eq. 18 and Eq. 21, we transfer Eq. 23 to:

$$D_{KL}(P_\sigma \parallel Q_\sigma) < D_{KL}(P \parallel Q) \tag{24}$$

This means that, for most cases, if $P \neq Q$, adding Gaussian noise with $\sigma \neq 0$ will decrease the KL divergence between P and Q.

A.2. Proofs in Our Method

In this section, we give a detailed explanation to section 3.2, including detailed explanation of forward sampling process and proof of reverse sampling process.

Reverse Sampling steps of Residual-Removal Stage. Given Eq. 3 and Eq. 5, we have:

$$
\begin{aligned}
I_{t-1}^{(2)} &= I_0^\theta + \bar{\alpha}_{t-1} I_{\text{res}}^\theta + \sigma\epsilon_t \\
&= (I_t^{(2)} - \bar{\alpha}_t I_{\text{res}}^\theta) + \bar{\alpha}_{t-1} I_{\text{res}}^\theta + \sigma\epsilon_t \\
&= I_t^{(2)} - (\bar{\alpha}_t - \bar{\alpha}_{t-1}) I_{\text{res}}^\theta + \sigma\epsilon_t \\
&= I_t^{(2)} - \bar{\alpha}_t I_{\text{res}}^\theta + \sigma\epsilon_t \\
&= I_t^{(2)} - \alpha_t I_{\text{res}}^\theta
\end{aligned} \tag{25}
$$

Finally, we have

$$
I_{t-1}^{(2)} = I_t^{(2)} - \alpha_t I_{\text{res}}^\theta(I_t^{(2)}, I_{in}, t). \quad (Eq.\ 6) \tag{26}
$$

Reverse Sampling steps of Denoising Stage. Given

$$
I_t^{(1)} = I_0^{(1)} + \bar{\beta}_t\,\varepsilon, \quad (Eq.\ 2) \tag{27}
$$

and

$$
\begin{aligned}
&p_\theta(I_{t-1}^{(1)} \mid I_t^{(1)}) := q_\sigma\big(I_{t-1}^{(1)} \mid I_t^{(1)}, I_0^{(1)}(\theta)\big) \quad (Eq.\ 7) \\
&= \mathcal{N}(I_{t-1}; I_0^{(1)}(\theta) + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\frac{(I_t^{(1)} - I_0^{(1)}(\theta))}{\bar{\beta}_t}, \sigma_t^2\mathbf{I}),
\end{aligned} \tag{28}
$$

$$
\begin{aligned}
&I_{t-1}^{(1)} \\
&= I_0^{(1)}(\theta) + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\ \frac{(I_t^{(1)} - I_0^{(1)}(\theta))}{\bar{\beta}_t} + \sigma_t\varepsilon_t \\
&= (I_t^1 - \bar{\beta}_t\epsilon_t) + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\ \frac{(I_t^1 - (I_t^1 - \bar{\beta}_t\epsilon_t))}{\bar{\beta}_t} + \sigma_t\varepsilon_t \\
&= (I_t^{(1)} - \bar{\beta}_t\epsilon_t) + \epsilon_t\sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2} + \sigma_t\varepsilon_t \\
&= I_{t-1}^{(1)} = I_t^{(1)} - (\bar{\beta}_t - \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2})\,\epsilon_\theta(I_t^{(1)}, t) + \sigma_t\varepsilon_t.
\end{aligned} \tag{29}
$$

where $\sigma_t^2 = \eta\beta_t^2\bar{\beta}_{t-1}^2/\bar{\beta}_t^2$. When generation process is deterministic ($\eta = 0$), we have:

$$
I_{t-1}^{(1)} = I_t^{(1)} - (\bar{\beta}_t - \bar{\beta}_{t-1})\, I_\epsilon^\theta(I_t^{(1)}, t) \tag{30}
$$

Derivation of Eq. 7 (Eq. 28). Similar to proof in RDDM [34] A.2, we have:

$$
q(I_t|I_0) = \mathcal{N}(I_t; I_0, \bar{\beta}_t^2\mathbf{I}). \tag{31}
$$

Similar to the evolution from DDPM [17] to DDIM [54], we can prove the statement with an induction argument for t from T to 1. Assuming that Eq. 31 holds at T, we just need to verify $q(I_{t-1}|I_0)$ at $t-1$ from $q(I_t|I_0)$ at t using Eq. 31. Given:

$$
q(I_t|I_0) = \mathcal{N}(I_t; I_0, \bar{\beta}_t^2\mathbf{I}), \tag{32}
$$

$$
q_\sigma(I_{t-1}|I_t, I_0) = \mathcal{N}(I_{t-1}; I_0 + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\frac{(I_t - I_0)}{\bar{\beta}_t}, \sigma_t^2\mathbf{I}), \tag{33}
$$

$$
q(I_{t-1}|I_0) := \mathcal{N}(\tilde{\mu}_{t-1}, \tilde{\Sigma}_{t-1}) \tag{34}
$$

Similar to obtaining $p(y)$ from $p(x)$ and $p(y|x)$ using Eq.2.113-Eq.2.115 in [4], the values of $\tilde{\mu}_{t-1}$ and $\tilde{\Sigma}_{t-1}$ are derived as follows:

$$
\tilde{\mu}_{t-1} = I_0 + \sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}\frac{(I_0) - (I_0)}{\bar{\beta}_t} = I_0, \tag{35}
$$

$$
\tilde{\Sigma}_{t-1} = \sigma_t^2\mathbf{I} + \left(\frac{\sqrt{\bar{\beta}_{t-1}^2 - \sigma_t^2}}{\bar{\beta}_t}\right)^2 \bar{\beta}_t^2\mathbf{I} = \bar{\beta}_{t-1}^2\mathbf{I}. \tag{36}
$$

Therefore, $q(I_{t-1}|I_0) = \mathcal{N}(I_{t-1}; I_0, \bar{\beta}_{t-1}^2\mathbf{I})$. In fact, the case ($t = T$) already holds, thus Eq. 31 holds for all t. We can derive Eq. 7 and Eq. 28 from Eq. 33.

A.3. Derivation of Training Objectives

In this section, we give a proof to the training objectives. According to Eq. 5, we derive the training objective of residual-removal process as follows:

$$
\begin{aligned}
&L_{res}(\theta) \\
&= D_{KL}\left(q(I_{t-1}^{(2)} \mid I_t^{(2)}, I_0^{(2)}(\theta), I_{\text{res}}^\theta)\right) \parallel p_\theta\left(I_{t-1}^{(2)} \mid I_t^{(2)}\right) \\
&= \mathbb{E}\left[\left\|I_t - \alpha_t I_{res} - \left(I_t - \alpha_t I_{res}^\theta\right)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|I_{\text{res}} - I_{\text{res}}^\theta(I_t^{(2)}, t, I_{in})\right\|_1\right]. \quad (Eq.\ 10)
\end{aligned} \tag{37}
$$

According to Eq. 7, we derive the training objective of denoising process as follows:

$$
\begin{aligned}
&L_\epsilon(\theta) \\
&= D_{KL}\left(q(I_{t-1}^{(1)} \mid I_t^{(1)}, I_0^{(1)}(\theta)\right) \parallel p_\theta\left(I_{t-1}^{(1)} \mid I_t^{(1)}\right) \\
&= \mathbb{E}\left[\left\|I_t - \frac{\beta_t^2}{\bar{\beta}_t}\epsilon - \left(I_t - \frac{\beta_t^2}{\bar{\beta}_t}\epsilon^\theta\right)\right\|^2\right] \\
&= \mathbb{E}\left[\left\|\epsilon - \epsilon_\theta(I_t^{(1)}, t)\right\|_1\right]. \quad (Eq.\ 10)
\end{aligned} \tag{38}
$$

A.4. Derivation of Decoupled SDE

This section introduces decoupling paradigm in SDE-based diffusion models and show the process of generating samples with reverse-time SDEs. The forward process formula is as follows:

$$
\mathrm{d}x = \theta_t\,(\mu - x)\,\mathrm{d}t + \sigma_t\,\mathrm{d}w, \tag{39}
$$

where t denote the continuous time variable, w is a standard Wiener process, μ is the state mean, and θ_t, σ_t are time-dependent positive parameters that characterize the speed of the mean-reversion and the stochastic volatility, respectively. In 39, $\theta_t\,(\mu - x)\,\mathrm{d}t$ is the drift term, which governs the evolutionary trend of the state, while $\sigma_t\,\mathrm{d}w$ denotes the random noise disturbance affecting the state. We then reverse the SDE to derive an image restoration SDE. During the testing phase, only the score $\nabla_x \log p_t(x)$ needs to be predicted in this formula:

$$\mathrm{d}x = \left[\theta_t\,(\mu - x) - \sigma_t^2\,\nabla_x \log p_t(x)\right]\mathrm{d}t + \sigma_t\,\mathrm{d}\hat{w}. \quad (40)$$

We decouple the forward process into a two-stage procedure, adding noise and degradation-specific information respectively, as follows:

$$\mathrm{d}x^{(1)} = \sigma_t\,\mathrm{d}w \quad (41)$$

$$\mathrm{d}x^{(2)} = \theta_t\,(\mu - x^{(2)})\,\mathrm{d}t + \sigma_t\,\mathrm{d}w, \quad (42)$$

Finally, we can reverse the SDE by predicting the score in Formula 40 for each of the two stages. Given an initial state x_0, for any state x_i at discrete time $i > 0$, the optimum residual reversing solution x_{i-1}^* in (39) is given by:

$$\begin{aligned} x_{i-1}^{(2)*} = {} & \frac{1 - e^{-2\,\bar{\theta}_{i-1}}}{1 - e^{-2\,\bar{\theta}_i}} e^{-\theta_i'}(x_i^{(2)} - \mu) \\ & + \frac{1 - e^{-2\,\theta_i'}}{1 - e^{-2\,\bar{\theta}_i}} e^{-\bar{\theta}_{i-1}}(x_0^{(2)} - \mu) + \mu. \end{aligned} \quad (43)$$

For noise reversing $x_{i-1}^{(1)*}$ is given by:

$$x_{i-1}^{(1)*} = \frac{1 - \mathrm{e}^{-2\bar{\theta}_{i-1}}}{1 - \mathrm{e}^{-2\bar{\theta}_i}} \mathrm{e}^{-\theta_i}(x_i^{(1)} - x_0^{(1)}) + x_0^{(1)}. \quad (44)$$

A.5. Noise-Level Selection via Dual MMD Distances

In this section, we give a detailed exploration of aforementioned Eq. 12 in Section 4.5.

$$A(\sigma) = \Delta(P_s^\sigma, P_t^\sigma), \quad B(\sigma) = \Delta(P_s^\sigma, P_s) \quad (45)$$

Here, $A(\sigma)$ and $B(\sigma)$ represent the measures of distance (in this case, based on MMD) between the distributions P_s^σ and P_t^σ for $A(\sigma)$, and P_s^σ and P_s for $B(\sigma)$, where: - P_s^σ is the distribution of the source after adding Gaussian noise $\mathcal{N}(0, \sigma^2)$ with σ being the noise strength parameter. - P_t^σ and P_s are the target and reference distributions used for comparison.

$$\begin{aligned} \Delta = {} & \mathbb{E}_{x,x'\sim P}\left[\exp\left(-\frac{\|x - x'\|^2}{2\sigma^2}\right)\right] \\ & + \mathbb{E}_{y,y'\sim Q}\left[\exp\left(-\frac{\|y - y'\|^2}{2\sigma^2}\right)\right] \\ & - 2\mathbb{E}_{x\sim P, y\sim Q}\left[\exp\left(-\frac{\|x - y\|^2}{2\sigma^2}\right)\right], \end{aligned} \quad (46)$$

In the above equation, Δ is the Maximum Mean Discrepancy between two distributions, P and Q. Here, σ is a hyperparameter controlling the width of the Gaussian kernel, which affects the sensitivity of the kernel to the differences between samples from the distributions. The term $\|x - x'\|^2$ is the squared Euclidean distance between two samples, and P and Q represent the two distributions being compared.

$$\widehat{A}(\sigma) = \frac{1}{R}\sum_{r=1}^{R} \widehat{A}_r(\sigma), \qquad \widehat{B}(\sigma) = \frac{1}{R}\sum_{r=1}^{R} \widehat{B}_r(\sigma). \quad (47)$$

In this equation, $\widehat{A}(\sigma)$ and $\widehat{B}(\sigma)$ are the average estimates of $A(\sigma)$ and $B(\sigma)$, computed over R independent samples. Each sample, $\widehat{A}_r(\sigma)$ and $\widehat{B}_r(\sigma)$, is calculated for the r-th sample from the dataset. Here, R represents the total number of samples used in the averaging process.

$$\widetilde{A}(\sigma) = \frac{\widehat{A}(\sigma) - \min_{\tau\in[0,\infty)} \widehat{A}(\tau)}{\max_{\tau\in[0,\infty)} \widehat{A}(\tau) - \min_{\tau\in[0,\infty)} \widehat{A}(\tau) + \epsilon}, \quad (48)$$

$$\widetilde{B}(\sigma) = \frac{\widehat{B}(\sigma) - \min_{\tau\in[0,\infty)} \widehat{B}(\tau)}{\max_{\tau\in[0,\infty)} \widehat{B}(\tau) - \min_{\tau\in[0,\infty)} \widehat{B}(\tau) + \epsilon}. \quad (49)$$

The equations above represent the normalized versions of $\widehat{A}(\sigma)$ and $\widehat{B}(\sigma)$, denoted as $\widetilde{A}(\sigma)$ and $\widetilde{B}(\sigma)$, respectively. The normalization is done to scale the values of $\widehat{A}(\sigma)$ and $\widehat{B}(\sigma)$ to a range between 0 and 1. The small constant ε is added to the denominator to avoid division by zero and ensure numerical stability.

$$J(\sigma;\lambda) \;=\; \lambda\,\widetilde{A}(\sigma) \;+\; \big(1-\lambda\big)\,\widetilde{B}(\sigma), \qquad \lambda \in [0,1]. \quad (50)$$

The function $J(\sigma;\lambda)$ is the weighted trade-off between $\widetilde{A}(\sigma)$ and $\widetilde{B}(\sigma)$, controlled by the parameter λ. This trade-off allows us to balance the importance of the two terms, with λ taking values in the range $[0, 1]$.

$$\sigma_J^\star \;=\; \arg\min_{\sigma\in[0,\infty)} \; J(\sigma;\lambda). \quad (51)$$

Finally, the optimal $\sigma_J^\star$ is selected by minimizing the trade-off function $J(\sigma;\lambda)$ over the range $\sigma \in [0, \infty)$. The goal is to find the value of σ that minimizes the trade-off between $\widetilde{A}(\sigma)$ and $\widetilde{B}(\sigma)$, optimizing the performance based on the specific task.

B. Experiment Settings and Dataset

Experiment Settings. All experiments are conducted on NVIDIA A6000 48GB GPUs. The network is optimized with L2 loss. Data augmentation includes random horizontal and vertical flips for all tasks and histogram equalization for low-light images. During training, 256×256 patches are randomly cropped from

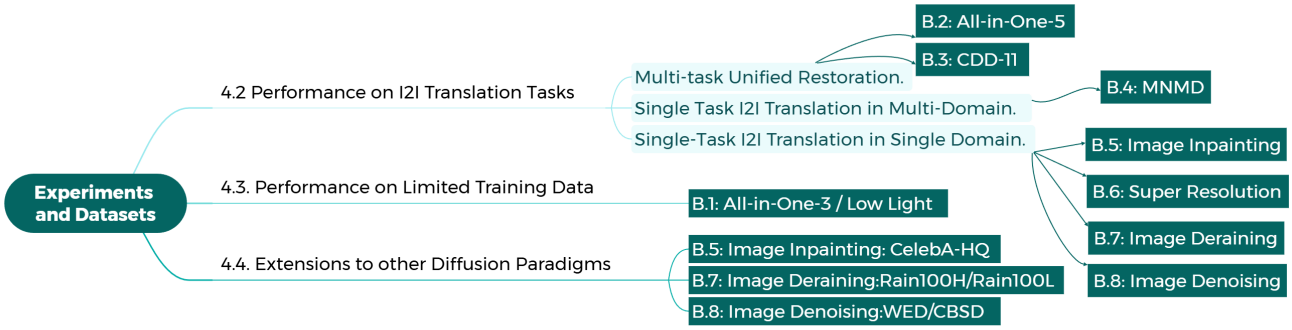

Figure 7. *Mindmap of experiments and corresponding datasets.*

Table 4. *Details for All-in-One image restoration benchmarks.*

Task	Training Dataset	Size	Testing Dataset	Size
All-in-One-3	WED+BSD400 [2, 41]	5,144	CBSD68 [2]	68
	RESIDE-OTS [26]	72,135	SOTS [26]	492
	Rain100L [49]	200	Rain100L	100
All-in-One-5	GoPro [44]	2,111	GoPro	1,111
	WED+BSD400 [2, 41]	5,144	CBSD68 [2]	68
	LoLV1 [61]	485	LoLV1	15
	RESIDE-OTS [26]	72,135	SOTS [26]	492
	Rain100L [49]	200	Rain100L	100
MNMD	UC-Merced [45]	17,010	UC-Merced	630
	WED+BSD400	46,296	CBSD68	204
	BrainWeb [21]	4,689	BrainWeb-test	174
CDD-11	CDD-11 [15]	20,790	CDD-11	2,310
Low-Light	LoLV1 [61]	485	LoLV1	15
	LOL-VE [49]	400	LOL-VE	100

augmented images as network inputs. Unless otherwise specified, the batch size is set to 8, the initial learning rate is 8e-5, and the total number of training steps is 300,000.

Model Architecture. Our residual removal model sets the basic U-Net [51] architecture and the hyper-parameters are as follows: $C = 64$, channel multiplier = $(1, 2, 4, 8)$. Our denoising model follows the U-Net architecture in [12], and the parameters are as follows $C = 128$, channel multiplier = $(1, 1, 2, 2, 4, 4)$.

Datasets. Overall dataset structures are displayed in Fig. 7. The detailed dataset introduction are provided later from B.1 to B.8.

Metrics. To comprehensively evaluate restoration performance, we adopt both distortion-based metrics (PSNR, SSIM [60]) and perceptual metrics (LPIPS [68], FID [16]). Distortion metrics assess pixel-level fidelity, while perceptual metrics compare feature space similarities to better reflect human visual perception. Following standard practice in All-in-One image restoration, all metrics are computed on the RGB channels of full-resolution images.

B.1. All-in-One-3 & Low Light

In this section, we introduce the datasets we used for experiment "4.3.Performance on Limited Training Data".

All-in-One-3. All-in-One-3 is a common setting for unified multiple image restoration tasks training, which contains "Noise+Haze+Rain".

Image dehazing: For image dehazing, we use the outdoor synthesis dataset of RESIDE-OTS [26] with 72,135 pairs for training and SOTS-Outdoor [26] dataset for testing with 492 image pairs. These pairs are captured in real-world outdoor scenes and are specifically designed to benchmark the performance of dehazing algorithms under varying weather and lighting conditions, ensuring effective evaluation of dehazing performance across diverse environments.

Image deraining: we use the Rain100L [49] dataset with 200 pairs of images for training and 100 pairs for testing. Detailed introduction to Rain100L is provided in Appendix B.7.

Image denoising: We conduct training using a merged dataset of BSD400 [2] and WED [41] with 400 and 4,744 clear images, respectively. Noisy images are generated with Gaussian noise (σ = (15, 25, 50)). Testing is performed on CBSD68 [2] datasets with 68 samples. Detailed introduction of BSD400 [2] and WED [41] is provided in Appendix B.8.

Low-Light. Following previous image restoration settings [58], we combine data samples in LOLv1 and VE-LOL-L [33] as Low-Light benchmark. **VE-LOL-L**: This dataset is specifically designed to benchmark low-light image enhancement techniques. The VELOLL dataset consists of 400 synthetically paired images for training and 100 paired images for testing, with each pair consisting of a low-light image and its corresponding well-exposed reference image. **LOLv1**: See details in Appendix B.2.

B.2. All-in-One-5

All-in-One-5 contains datasets from the aforementioned three-task setting(All-in-One-3) as well as additional datasets: GoPro [44] for motion deblurring, and LOLv1 [61]

for lowlight image enhancement. The overall dataset contains "Noise+Haze+Rain+Light+Blur" in total.

Image Deblurring. The GoPro [44] dataset is a standard benchmark for dynamic scene deblurring, created to enable supervised learning for blind deblurring models. The authors recorded high-frame-rate video and used the high-fps frames as sharp reference images. Instead of convolving with simple uniform kernels, they synthesized realistic, spatially varying motion blur by averaging consecutive sharp frames. The dataset release provides 2,111 blurry-sharp image pairs for training and 1,111 pairs for evaluation.

Low Light Enhancement. LOLv1 [61] is a standard supervised benchmark for low-light enhancement. It consists of paired low-light and well-exposed reference images. The authors collected these pairs by capturing the same scenes under low and normal lighting conditions using a variety of consumer cameras and phones. This process yielded real captures (not purely synthetic) that include realistic sensor noise and color shifts, making the dataset a robust resource. We use 485 image pairs from LoLV1 for training and 15 pairs for testing.

B.3. CDD-11

The CDD-11 (Composite Degradation Dataset) is a synthetic dataset designed for training and evaluating image restoration models under composite degradation conditions. CDD-11 was introduced in the OneRestore [15], from which highlights the importance of dealing with multiple degradation types simultaneously, such as low-light, haze, rain, and snow, rather than addressing each degradation type in isolation. The dataset consists of 11 different degradation conditions, including combinations like "low_haze," "haze_rain," "low_rain," and "low_snow," which containing 20,790 image pairs for training and 2,310 image pairs for testing in total. CDD-11 serves as a benchmark for evaluating models that aim to perform restoration under mixed degradation conditions, reflecting real-world scenarios more accurately than datasets with only single degradation types.

B.4. Multi-Noise and Multi-Domain

In this section, we introduce a novel denoising benchmark designed for the single task I2I in multi-domains. We name this new benchmark **M**ulti-**N**oise and **M**ulti-**D**omain (MNMD). As shown in Tab. 4, MNMD consists of image pairs from various sources, including 46,296 natural images (from WED and BSD400 [49]), 17,010 remote sensing images (from UC-Merced [45]), and 4,689 medical images (from BrainWeb [7, 21, 22]). To better simulate noise-related degradations under different conditions, we introduce three types of noise, each with multiple intensity levels: **Gaussian noise** is added as $x' = x + n$, where $n \sim \mathcal{N}(0, \sigma^2)$. The parameter $\sigma = (15, 25, 50)$ controls the standard deviation, reflecting the noise strength; **Salt-and-Pepper noise** is applied by randomly selecting a fraction d (scale 0.014, 0.039, 0.154) of pixels and replacing each with either the minimum or maximum possible value, thereby simulating random pixel corruption; and **Poisson noise** is simulated by replacing each pixel with a value drawn from a Poisson distribution whose mean is $x \times$ peak, with x being the original pixel value and peak (26, 102, 283) acting as a noise scaling factor.

For evaluation, we use 204 noisy natural images from CBSD68 [2], as well as 630 and 174 images from UC-Merced and BrainWeb, respectively. The test images are degraded with Gaussian noise (σ = 15), salt-and-pepper noise (density = 0.039), and Poisson noise (scale = 102). This setup forms a comprehensive test set for evaluating denoising performance across multiple domains and noise types.

The UC-Merced Land Use Dataset (UC-Merced [45] is a widely-used remote sensing image dataset designed for land use scene classification, featuring 21 distinct categories of US urban and semi-urban scenes. Each category contains 100 images, resulting in a total of 2,100 images in the dataset. The images have a resolution of 256×256 pixels and were manually extracted from the United States Geological Survey (USGS) National Map Urban Area Imagery collections for various US urban areas. The medical images are stem from **BrainWeb** [7, 21, 22], which is a website for synthesizing medical images. Detailed introduction of **BSD400** [2] and **WED** [41] is provided in Appendix B.8.

B.5. Image Inpainting

The CelebA-HQ [19] (CelebFaces High Quality) dataset is a high-resolution version of the CelebA dataset, specifically designed for training and evaluating face-related image processing tasks, including image inpainting. CelebA-HQ consists of 30,000 high-resolution facial images, each with a resolution of 256x256 pixels. The dataset includes a wide variety of celebrity faces with various attributes such as age, gender, and facial expressions, making it suitable for tasks like face generation and image inpainting. For image inpainting tasks, we adopt the CelebA-HQ dataset for both training and testing, with our training set containing 28,000 image pairs and our test set consisting of 2,000 image pairs. Each image in the dataset is paired with a mask that specifies the region to be inpainted, allowing models to learn to fill in missing parts of the face.

B.6. Super-Resolution

For super-resolution, we adopt the FFHQ [20] dataset for training and testing. FFHQ (Flickr-Faces-HQ) dataset is a high-quality facial image dataset consists of 70,000 samples, which is primarily used for computer vision and deep learning research, particularly in applications like facial image generation, editing, and expression recognition. Released by NVIDIA in 2018, the dataset aims to provide a high-

resolution standard for facial image generation and recognition tasks. In our experiment, we input 16x16 images and output high-resolution 128x128 images. During training, we employ bicubic interpolation to upsample the 16x16 input images to 128x128, using the upsampled images as input for training.

B.7. Deraining

For deraining, we adopt the **Rain100** dataset [64] for both training and testing. Rain100 consists of two subsets, Rain100H (Heavy Rain) and Rain100L (Light Rain), each containing paired rainy and ground-truth clean images. The full dataset contains 2,200 image pairs: 300 for Rain100L and 1,900 for Rain100H. For our experiments, we use 100 image pairs from each subset for testing and the remaining pairs for training.

B.8. Denoising

For image denoising, we leverage two widely used datasets: BSD400 [2] and WED [41] for training, while CBSD68 [2] and Urban100 [18] for testing. Our training set consists of 5,144 image pairs (400 from BSD400 and 4,744 from WED), while our testing set consists of 168 image pairs (68 from CBSD68 and 100 from Urban100).

Training datasets. We use BSD400 and WED to build our training set. BSD400 is a widely used dataset of 400 natural images, usually adopted as a training set in image-restoration research. The WED (Waterloo Exploration Database) is a much larger collection created for image-quality assessment, containing thousands of diverse, high-quality natural images. Combining WED with BSD400 provides a larger and more varied pool of clean images, which is essential for training models that can generalize well to diverse real-world noise. Many recent denoising pipelines [8, 48, 66] merge these two datasets to expand their training data, and we follow this paradigm to utilize BSD400+WED as our training set.

Testing datasets. Our models are evaluated on two challenging test sets, CBSD68 and Urban100. CBSD68 is the color version of the BSD68 test set, a benchmark of 68 natural images commonly used to evaluate color-image denoising. Urban100 is a 100-image dataset of high-resolution urban scenes, initially developed for single-image super-resolution but now widely adopted as a challenging benchmark for denoising. Urban scenes often feature strong, repeating geometric patterns (like bricks and windows) that are difficult to reconstruct, making Urban100 a robust test for a model's ability to recover fine structural detail.

C. Additional Experiments

C.1. Implementation Details of Methods in Experiments

In this section, we introduce the training details of our comparison methods.

DRDD(Ours). Without specific mentioned, training settings follow experiments setting details in Appendix B. In Experiment "4.3.Performance on Limited Training Data". We start training denoising U-NET from pre-trained parameters, where $C = 256$, channel multiplier = $(1, 1, 2, 2, 4, 4)$.

RDDM. [34] All experimental settings are kept consistent with those in the paper for both image deraining and image inpainting tasks.

DiffuIR. It [69] uses a selective hourglass mapping strategy within a diffusion model to handle multiple image restoration tasks with a single, efficient model. All experimental settings are kept consistent with those in the paper.

AdAIR [9] adaptively restores images suffering from various degradations by mining and modulating frequency components, enabling unified and effective all-in-one image restoration. For training stage, it is conducted with a batch size of 32 in the all-in-one setting and a batch size of 8 in the single-task setting. The model is trained on cropped image patches of size 128 × 128 pixels for 150 epochs, which is approximately equivalent to 300,000 steps in DRDD.

DA-CLIP [39] employs an image controller to identify degradation types and adaptively restore images affected by diverse distortions, thereby providing a unified and effective solution for multi-task image restoration. For training stage, we use a batch size of 16 in training for All-in-One-5 task. The model is trained for 300,000 steps on cropped image patches of size 256 × 256 pixels.

DFPIR. [57] It uses a unified model with a degradation-aware feature perturbation mechanism, which introduces channel-wise and attention-wise perturbations to mitigate task interference. We follow the experiment settings in the paper. **IR-SDE** [38] adaptively restores images suffering from various degradations by modeling the degradation process with mean-reverting stochastic differential equations, enabling unified image restoration. For training stage, all tasks are trained with a batch size of 8 for a total of 500,000 steps. For the image inpainting task, the model is trained on cropped image patches of size 64 × 64 pixels, whereas for the other tasks, is trained on 128 × 128 pixels.

De-IRSDE is a decoupling version of IRSDE [38]. All experimental settings are kept consistent with those of IRSDE [38].

VLUNET [67] adaptively restores images suffering from various degradations by leveraging vision-language models to automatically select degradation-specific transforms, enabling unified and effective all-in-one image restoration within a deep unfolding framework. For training stage, we

Table 5. Comparison of single task image restoration approaches on deraining and denoising with two metrics (SSIM / LPIPS). Best results are highlighted in **Bold**.

(a) Deraining Results

Method	Rain100H	Rain100L
	SSIM / LPIPS	SSIM / LPIPS
RDDM*	.8806 / .1170	.9432 / .0250
IRSDE*	.9041 / .0470	.9805 / .0140
DRDD (Ours)	**.9375 / .0400**	**.9839 / .0180**

(b) Denoising on CBSD68 and Kodak24 with $\sigma \in \{15, 25, 50\}$

Method	CBSD68			Kodak24		
	σ=15 SSIM / LPIPS	σ=25 SSIM / LPIPS	σ=50 SSIM / LPIPS	σ=15 SSIM / LPIPS	σ=25 SSIM / LPIPS	σ=50 SSIM / LPIPS
AdAIR	.9340 / .0534	.8898 / .0967	.8003 / .1895	.9269 / .0712	.8841 / .1114	.8036 / .1966
VLUNet	.9341 / .0568	.8902 / .0984	**.8039** / .1938	.9270 / .0745	.8838 / .1146	**.8079** / .2040
DRDD (Ours)	**.9346 / .0502**	**.8920 / .0877**	.8013 / **.1750**	**.9274 / .0680**	**.8879 / .1032**	.8047 / **.1880**

Table 6. *Performance comparison of inpainting methods at 256×256 (center), 256×256 (irregular), and 64×64 (center) resolutions.* Best results and the second-best results are highlighted in **red** and blue.

Method	256×256 (Center)		256×256 (Irregular)		64×64 (Center)	
	LPIPS↓	FID↓	LPIPS↓	FID↓	LPIPS↓	FID↓
RDDM [34]	0.0862	15.67	0.0963	8.52	0.0568	14.75
CTSDG [14]	0.0798	11.64	0.0722	7.87	0.0498	15.68
MISF [29]	0.0764	11.72	0.0803	7.68	0.0695	20.43
TransRef [35]	0.0745	**9.13**	0.0692	7.80	0.0490	10.17
DRDD(Ours)	**0.0542**	10.30	**0.0528**	**7.15**	**0.0382**	**10.02**

Table 7. *Performance comparison of RDDM and DRDD on Edges2Handbags and Edges2Shoes dataset.*

Table A2	Edges2Handbags-256×256			Edges2Shoes-256×256		
	SSIM↑	FID↓	LPIPS↓	SSIM↑	FID↓	LPIPS↓
RDDM	0.645	5.72	0.256	0.652	23.57	0.178
DRDD	**0.723**	**4.76**	**0.247**	**0.782**	**9.658**	**0.154**

use a batch size of 8 in training for all tasks. The model is trained for 200 epochs on cropped image patches of size 128 × 128 pixels, which is roughly equivalent to 400,000 steps in DRDD.

TransRef. [35] For training stage, we use a batch size of 8 for training across all tasks. The model is trained on the original image pixels for 400,000 steps.

CTSDG. [14] For training stage, we use a batch size of 4 for training across all tasks. The model is trained on cropped image patches of size 128 × 128 pixels for 150,000 steps.

C.2. More Single I2I Tasks in Single Domain.

Image Inpainting. To validate the generalization capability of the proposed framework across different tasks, we conducted inpainting experiments under various settings. Specifically, we evaluated the inpainting performance at different resolutions using two mask patterns (center and irregular) on the CelebA-HQ [19] dataset. It is worth noting that irregular masks more closely reflect the restoration demands of complex occlusions encountered in real-world scenarios, whereas rectangular masks are typically used to simulate cases in which local image information is entirely missing.

Figure 8. *Visual comparison of RDDM and DRDD on Edges2Handbags dataset.*

Furthermore, we conducted experiments at both 64×64 and 256×256 resolutions to further confirm the model's adaptability to different input scales. As shown in Tab. 6, our method achieved the best or second-best results across all configurations, demonstrating its superior performance and robustness under diverse task conditions.

Image Deraining. We compare DRDD's performance on image deraining through Rain100H and Rain100L datasets with RDDM [34] and DiffuIR [69]. Results are shown in Tab. 5.

Edges to Objects. As shown in Tab. 7 and Fig. 8, we evaluated DRDD on style transfer task using edges2handbags & edges2shoes datasets. DRDD significantly outperforms its baseline RDDM. Visual results confirm that the injected noise does not adversely affect DRDD.

Image Denoising. We compare DRDD's performance on image denoising through BSD400 [2] and WED [41] with AdaIR[9] and VLUNet [67]. Results are shown in Tab. 5.

Super Resolution. To further demonstrate the generalization ability of our method across different tasks, we conducted a qualitative experiment on the FFHQ [20] dataset to evaluate DRRD's performance in the image super-resolution task.

C.3. Ablation Studies.

We compare DRDD against a matched "coupled" baseline with the same architecture. Although only a single neural network is used, we assign it the combined number of parameters of two decoupled neural networks, along with the

Table 8. *Ablation Studies and Further Investigaitons on All-in-One-5 daytaset.* SSIM (↑) and LPIPS (↓) are reported on the full RGB images. The **best** performances are highlighted.

Method	*Dehazing*		*Deraining*		*Denoising*		*Deblurring*		*Low-Light*		Average	
	SOTS		Rain100L		$BSD68_{\sigma=25}$		GoPro		LOLv1			
DRDD w.o Denoising Network	.9690	.0149	.9706	.0244	.8771	.1136	.8241	.1923	.8469	.1251	.8971	.0941
DRDD with General Denoising Network	.9652	.0145	.9701	.0248	.8867	.1054	.8712	.1748	.8490	.1232	.9084	.0885
Entangled Baseline	.9601	.0136	.9727	.0231	.8818	.1153	.8405	.1849	.8283	.1555	.8966	.0985
DRDD (Ours)	**.9715**	**.0122**	**.9777**	**.0153**	.8891	**.0977**	**.8806**	**.1342**	**.8640**	**.1033**	**.0916**	**.0762**

double cumulative inference steps. We test it on All-in-One-5 benchmark and results are displayed in Tab. 8.

C.4. Further Investigations of Denoising Model

Investigating the Training of Denoising Models on Isolated Datasets. Based on the properties we previously discussed, the denoising model can be trained on a isolate dataset. To further validate this, We train the denoising model for 300,000 steps on the All-in-One-3 dataset and combined it with the residual removal model, which was trained for 300,000 steps on the All-in-One-5 dataset. The results in Tab. 8 show that despite the denoising model never having encountered low-light or deblurring data, it still learned a certain level of generalized denoising capability from the other datasets. Such a strategy can significantly reduce the number of training-required parameters.

Investigating Inference Without Denoising Models. In our decoupled model, the residual removal model is responsible for directional semantic elimination in the noise domain, while the denoising model is tasked with translating the data back to the noise-free domain and performing refined restoration. The necessity for a dedicated denoising model, rather than simply subtracting the noise added in the first step, arises because during directional semantic transformation, the residual removal model also exerts some influence on the noise. This perspective has been confirmed by experiment results in Tab. 8, DRDD without Denoising Network.

C.5. Cost and Efficiency

C.5.1. Parameters, Flops and Inference Time

In this section, we present the key performance metrics of our model. The floating-point operations (FLOPs) were computed by performing a forward inference using a randomly generated matrix of size 256×256×3. For measuring inference time, the model was trained on the AIOIR5 architecture and evaluated on the Rain100L dataset. The reported inference time was obtained under the condition where the model runs for two steps on each of its two sub-modules: residual removing and denoising.

C.5.2. Influence of sampling steps

We conducted experiments to investigate the influence of different sampling steps on the results using the All-in-One-3 dataset. The results, shown in Tab. 10, indicate that there is no significant difference between 2-step and 10-step inference when using DDIM sampling.

Table 9. *Computational resource comparisons: Parameters, FLOPs, and Runtime.* FLOPs are measured on the patch size of 256 × 256 × 3, while Runtime are measured on Rain100L testing set. The sampling steps of denoising model and residual-removal model are both 2.

	Method	Para (M)	FLOPs (G)	Step	Latency (s)	SSIM↑	LPIPS↓	FID↓
	AdAIR	29	138	1	0.24	0.909	0.089	26.1
	VLUNet	123	143	1	0.74	0.904	0.096	27.9
Diff.	DiffUIR	138	284	3	0.75	0.869	0.117	33.7
	DA-CLIP	174	380	3	0.76	0.876	0.108	20.0
	RDDM - S	138	278	4	0.92	0.878	0.105	27.8
	RDDM - L	138×2	278×2	4	1.84	0.897	0.098	23.6
	DRDD - S	7+35	32+69	2 + 2	0.10+0.23	0.908	0.080	19.6
	DRDD - L	7+138	32+278	2 + 2	0.10+0.45	**0.916**	**0.073**	**18.3**

Table 10. *Performance of different sampling steps.* SSIM (↑) and LPIPS (↓) are reported.

Sampling Steps		Dehazing		Denoise		Deraining	
de-res	de-noise	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓	SSIM↑	LPIPS↓
2	2	0.9787	0.0104	0.8919	0.0929	0.9767	0.0149
2	5	0.9790	0.0103	0.8900	0.0940	0.9774	0.0146
5	2	0.9791	0.0101	0.8918	0.0930	0.9767	0.0149
5	5	0.9794	0.0100	0.8899	0.0940	0.9774	0.0146
10	10	0.9792	0.0101	0.8838	0.0951	0.9777	0.0144

Table 11. Performance comparison between our method and other universal image restoration methods on real unseen datasets. Best results are highlighted in **red**, while the second-best results are blue.

Method	Low-Light	Deraining	Denoising	Deblurring
	NIQE↓	NIQE↓	PSNR↑ / SSIM↑ / LPIPS↓	PSNR↑ / SSIM↑ / LPIPS↓
VLUNet	4.40	**3.73**	24.6 / 0.490 / 0.664	26.8 / 0.822 / 0.175
DiffuIR	**3.89**	4.49	28.6 / 0.674 / 0.569	28.4 / 0.857 / 0.186
DRDD(Ours)	4.31	3.80	**34.8 / 0.865 / 0.274**	**28.5 / 0.861 / 0.172**

C.6. Generalization Ability

To further validate the proposed method's generalization capability on unseen data distribution, we evaluated its performance on unseen data across four representative image restoration tasks: low-light enhancement, deraining, denoising, and deblurring. The corresponding evaluation metrics are presented in Table 11. Specifically, the low-light enhancement task was conducted using a combined dataset comprising MEF [40], NPE [59], and DICM [25]; the deraining task employed the Practical [64] dataset; the denoising task utilized the SIDD [1] dataset; and the deblurring task

was assessed using the RealBlur [50] dataset.

C.7. PSNR results

We provide the PSNR results of All-in-One-5 model as below. As shown in Tab. 1 of the paper, DRDD consistently achieves the best FID, LPIPS among all the methods, the best PSNR, SSIM among diffusion-based approaches, while remaining highly competitive on PSNR, SSIM with non-diffusion methods.

Table C.7	Task	Low-Light	Deraining	Denoising	Deblurring	Dehazing
	Dataset	LOLv1	Rain100L	CBSD68	GoPro	SOTS
Non-Diff.	DFPIR	**23.80**	37.50	31.26	28.80	**31.24**
	AdAIR	22.94	37.85	31.29	28.11	29.93
	VLU-NET	22.25	**38.36**	**31.39**	**28.85**	30.56
Diff.	DiffuIR	19.33	34.88	30.12	26.48	30.27
	DA-CLIP	20.12	35.86	25.17	27.34	26.88
	DRDD (Ours)	**23.00**	**36.86**	**31.47**	**29.08**	**30.56**

C.8. Comparing data pruning results with more methods

We provide comparison with AdaIR [9] and RDDM [34] on pruned All-in-One-3 dataset, SSIM and LPIPS are reported.

Table C.8	25%	50%	75%	100%
	SSIM↑ / LPIPS↓	SSIM↑ / LPIPS↓	SSIM↑ / LPIPS↓	SSIM↑ / LPIPS↓
RDDM	.929 / .058	.935 / .056	.941 / .049	.942 / .047
AdAIR	.944 / .050	.948 / .046	.949 / .045	.951 / .042
DRDD (Ours)	**.947 / .041**	**.949 / .040**	**.950 / .039**	**.951 / .038**

C.9. Sensitivity Analysis of Noise Injection Level

Noise Injection level is relatively insensitive within the range of 0.8–1.3 (calculated via Eq. 12 across different datasets), as validated by Fig. 6 on the All-in-One-5 dataset and Table A5 (PSNR metric) on the Rain100H and Edges2Bags datasets. Outside this 0.8–1.3 range, sensitivity increases. If optimal performance is required, experimentation in 0.8–1.3 range is necessary.

Table C.9	Recommend σ	0.1	0.5	0.8	1.0	1.5	2.0
Rain100H	0.8-1.2	29.64	30.97	32.01	32.33	31.94	31.68
Edges2Bags	0.8-1.1	16.34	19.22	20.05	19.81	18.76	17.92

D. More Visual Comparisons

As shown in Fig. 9 - Fig.10, we provide additional visual examples of DRDD on various image-to-image transformation tasks, including restoration and inpainting. These results serve as supplementary visualizations to those presented in the main paper.

Figure 9. *Visual results of state-of-the-art methods and our proposed DRDD.* (a) Comparison of haze image restoration results on the SOTS dataset [26]. (b) Comparison of noise restoration results on the CBSD68 dataset [2]. (c) Super-Resolution result in FFHQ [20]. Zoom in for best view.

Figure 10. *Irregular Mask inpainting results of state-of-the-art methods and our proposed DRDD.* Zoom in for best view.