SpiralDiff: Spiral Diffusion with LoRA for RGB-to-RAW Conversion Across Cameras

Supplementary Material

In this supplementary material, we present more implementation details of SpiralDiff and additional experiment results. First, we introduce the derivation of forward and backward processes and the overall training and sampling pipelines in Sec. A. Then we present additional experiment results in Sec. B.

A. Additional Details of SpiralDiff

Derivation of Eq. 6 As discussed in Sec. 3.2, we propose a signal-dependent noise weighting strategy parameterized by a sequence of weight maps $\boldsymbol{w}_{t=1}^{T}$, defined as

$$\boldsymbol{w}_t = \boldsymbol{x}_0 + \eta_t \boldsymbol{e}_0. \tag{10}$$

Here, η_t increases monotonically with t, with boundary conditions $\eta_0 = 0$ and $\eta_T \to 1$. The forward transition distribution of SpiralDiff is given by:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0, \boldsymbol{y}_0) = \mathcal{N}\big(\boldsymbol{x}_t;\, \boldsymbol{x}_{t-1} + \alpha_t \boldsymbol{e}_0,\, \kappa^2(\eta_t \boldsymbol{w}_t^2 - \eta_{t-1}\boldsymbol{w}_{t-1}^2)\boldsymbol{I}\big)\,, \tag{11}$$

where $\boldsymbol{w}_t^2$ denotes the element-wise square of w_t^p for each pixel p and $\boldsymbol{I}$ is the identity matrix. Following the reparameterization of the forward process, $\boldsymbol{x}_t$ can be expressed as

$$\begin{aligned} \boldsymbol{x}_t &= \boldsymbol{x}_0 + \sum_{i=1}^{t} (\boldsymbol{x}_i - \boldsymbol{x}_{i-1}) \\ &= \boldsymbol{x}_0 + \sum_{i=1}^{t} \left(\alpha_i (\boldsymbol{y}_0 - \boldsymbol{x}_0) + \kappa \sqrt{\eta_i \boldsymbol{w}_i^2 - \eta_{i-1}\boldsymbol{w}_{i-1}^2}\, \boldsymbol{\epsilon}_i \right) \\ &= \boldsymbol{x}_0 + \eta_t (\boldsymbol{y}_0 - \boldsymbol{x}_0) + \kappa \sqrt{\eta_t}\, \boldsymbol{w}_t \boldsymbol{\epsilon}_t, \end{aligned} \tag{12}$$

where $\boldsymbol{\epsilon}_i \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$ are i.i.d. Gaussian noise variables, and $\boldsymbol{\epsilon}_t$ denotes the aggregated noise up to step t. This leads to the marginal distribution:

$$q(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{y}_0) = \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{x}_0 + \eta_t \boldsymbol{e}_0, \kappa^2 \eta_t \boldsymbol{w}_t^2 \boldsymbol{I}). \tag{13}$$

Derivation of Eq. 7 According to Bayes's theorem, the reverse transition distribution can be written as

$$\begin{aligned} q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) &= \frac{q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0, \boldsymbol{y}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{y}_0)}{q(\boldsymbol{x}_t \mid \boldsymbol{x}_0, \boldsymbol{y}_0)} \\ &\propto q(\boldsymbol{x}_t \mid \boldsymbol{x}_{t-1}, \boldsymbol{x}_0, \boldsymbol{y}_0) q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_0, \boldsymbol{y}_0). \end{aligned} \tag{14}$$

Incorporating Eq.11, Eq.13 and Eq.14, we consider the distribution at each pixel p as:

$$q(x_{t-1}^p \mid x_t^p, x_0^p, y_0^p) \propto q(x_t^p \mid x_{t-1}^p, x_0^p, y_0^p) q(x_{t-1}^p \mid x_0^p, y_0^p) \tag{15}$$

The two terms are:

$$q(x_t^p \mid x_{t-1}^p, x_0^p, y_0^p) = \mathcal{N}\big(x_t^p;\, x_{t-1}^p + \alpha_t e_0^p,\, \kappa^2 \left(\eta_t (w_t^p)^2 - \eta_{t-1}(w_{t-1}^p)^2\right)\big) \tag{16}$$

$$q(x_{t-1}^p \mid x_0^p, y_0^p) = \mathcal{N}\big(x_{t-1}^p;\, x_0^p + \eta_{t-1} e_0^p,\, \kappa^2 \eta_{t-1} (w_{t-1}^p)^2\big) \tag{17}$$

The exponent of the posterior $q(x_{t-1}^p \mid x_t^p, x_0^p, y_0^p)$ takes the following quadratic form:

$$\begin{aligned}
&-\frac{(x_t^p - x_{t-1}^p - \alpha_t e_0^p)^2}{2\kappa^2(\eta_t(w_t^p)^2 - \eta_{t-1}(w_{t-1}^p)^2)} - \frac{(x_{t-1}^p - x_0^p - \eta_{t-1}e_0^p)^2}{2\kappa^2\eta_{t-1}(w_{t-1}^p)^2} \\
&= -\frac{1}{2}\left[\frac{1}{\kappa^2(\eta_t(w_t^p)^2 - \eta_{t-1}(w_{t-1}^p)^2)} + \frac{1}{\kappa^2\eta_{t-1}(w_{t-1}^p)^2}\right](x_{t-1}^p)^2 \\
&\quad + \left[\frac{x_t^p - \alpha_t e_0^p}{\kappa^2(\eta_t(w_t^p)^2 - \eta_{t-1}(w_{t-1}^p)^2)} + \frac{x_0^p + \eta_{t-1}e_0^p}{\kappa^2\eta_{t-1}(w_{t-1}^p)^2}\right] x_{t-1}^p + \text{const} \\
&= -\frac{(x_{t-1}^p - \mu_{t-1}^p)^2}{2(\sigma_{t-1}^p)^2} + \text{const},
\end{aligned} \tag{18}$$

where μ_{t-1}^p and σ_{t-1}^p denote the mean and standard deviation of the posterior at pixel p, and const collects terms independent of x_{t-1}^p. The closed-form expressions for the posterior parameters are:

$$\begin{aligned}
\mu_{t-1}^p &= \gamma_t^p(x_t^p - \alpha_t e_0^p) + (1-\gamma_t^p)(x_0^p + \eta_{t-1}e_0^p) \\
(\sigma_{t-1}^p)^2 &= \kappa^2\gamma_t^p\left(\eta_t(w_t^p)^2 - \eta_{t-1}(w_{t-1}^p)^2\right) \\
\gamma_t^p &= \frac{\eta_{t-1}(w_{t-1}^p)^2}{\eta_t(w_t^p)^2}
\end{aligned} \tag{19}$$

Therefore, the reverse transition distribution is given by:

$$\begin{aligned}
q(\boldsymbol{x}_{t-1} \mid \boldsymbol{x}_t, \boldsymbol{x}_0, \boldsymbol{y}_0) &= \prod_p q(x_{t-1}^p \mid x_t^p, x_0^p, y_0^p) \\
&= \mathcal{N}(\boldsymbol{x}_{t-1};\ \boldsymbol{\mu}_{t-1},\ \boldsymbol{\Sigma}_{t-1}),
\end{aligned} \tag{20}$$

The mean and covariance are:

$$\begin{aligned}
\boldsymbol{\mu}_{t-1} &= \boldsymbol{\gamma}_t \odot (\boldsymbol{x}_t - \alpha_t \boldsymbol{e}_0) + (\mathbf{1} - \boldsymbol{\gamma}_t) \odot (\boldsymbol{x}_0 + \eta_{t-1}\boldsymbol{e}_0), \\
\boldsymbol{\Sigma}_{t-1} &= \kappa^2\, \boldsymbol{\gamma}_t \odot (\eta_t \boldsymbol{w}_t^2 - \eta_{t-1}\boldsymbol{w}_{t-1}^2), \\
\boldsymbol{\gamma}_t &= \frac{\eta_{t-1}\boldsymbol{w}_{t-1}^2}{\eta_t \boldsymbol{w}_t^2},
\end{aligned} \tag{21}$$

where $\odot$ denotes element-wise multiplication and division is performed element-wise.

Details of $\boldsymbol{w}_t$ Since $\boldsymbol{x}_0$ and $\boldsymbol{y}_0$ lie in the range $[-1, 1]$, the resulting $\boldsymbol{w}_t$ also falls within this interval. In practice, we normalize $\boldsymbol{w}_t$ to $[0, 1]$ by a simple range mapping, and then add a small bias term $b \in (0, 1)$:

$$\hat{\boldsymbol{w}}_t = b + (1-b)\boldsymbol{w}_t \tag{22}$$

This guarantees that $\hat{\boldsymbol{w}}_t$ is strictly bounded away from zero, where b acts as a lower bound on the noise level and prevents numerical instability when computing $\boldsymbol{\gamma}_t$. Consequently, $\hat{\boldsymbol{w}}_t \in [b, 1]$. We set $b = 0.1$ as the default value.

Training and sampling pipelines We present the training and sampling pipelines in Alg. 1 and Alg. 2. During training, we randomly sample a triplet consisting of a RAW image $\mathbf{x}_0$, its corresponding RGB image $\mathbf{y}_0$, and the associated camera label c. Based on $\hat{\boldsymbol{w}}_t$ and $\boldsymbol{e}_0$, the noisy image $\boldsymbol{x}_t$ is sampled according to Eq. 13. The denoising model is optimized to predict $\boldsymbol{x}_0$, denoted as $\hat{\boldsymbol{x}}_{0|t} = f_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t, c)$. During sampling, $\hat{\boldsymbol{w}}_t$ and $\hat{\boldsymbol{e}}_0$ are updated according to the model output $\hat{\boldsymbol{x}}_{0|t}$, then $\boldsymbol{x}_{t-1}$ is sampled by Eq. 20. By iteratively denoising and sampling, the clean image $\boldsymbol{x}_0$ is obtained.

Loss function We follow the same loss function as that in RAW-Diffusion [33], which combines the MSE, L1 and log-L1 loss. The overall loss is formulated as:

$$\begin{aligned}
\mathcal{L} &= \mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{logL1}} \\
&= \sum_t \left(\|\hat{\boldsymbol{x}}_{0|t} - \boldsymbol{x}_0\|_2^2 + \|\hat{\boldsymbol{x}}_{0|t} - \boldsymbol{x}_0\|_1 + \|\log(\hat{\boldsymbol{x}}_{0|t} + \epsilon) - \log(\boldsymbol{x}_0 + \epsilon)\|_1\right),
\end{aligned} \tag{23}$$

where ϵ is a minimal constant.

Algorithm 1: Training

Input: RAW image $\boldsymbol{x}_0$, RGB image $\boldsymbol{y}_0$, camera label c

while *not converged* **do**

Sample $t \sim \mathcal{U}(\{1, \dots, T\})$;

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;

$\boldsymbol{e}_0 = \boldsymbol{y}_0 - \boldsymbol{x}_0$;

$\boldsymbol{w}_t = \boldsymbol{x}_0 + \eta_t \boldsymbol{e}_0$;

$\hat{\boldsymbol{w}}_t = b + (1-b)\boldsymbol{w}_t$;

$\boldsymbol{x}_t = \boldsymbol{x}_0 + \eta_t \boldsymbol{e}_0 + \kappa\sqrt{\eta_t}\hat{\boldsymbol{w}}_t\boldsymbol{\epsilon}$;

Take gradient step on $\nabla_\theta \mathcal{L}(\boldsymbol{x}_0, f_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t, c))$;

return θ

Algorithm 2: Sampling

Input: RGB image $\boldsymbol{y}_0$, camera label c, denoising model f_θ

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;

$\hat{\boldsymbol{w}}_T = b + (1-b)\boldsymbol{y}_0$;

$\boldsymbol{x}_T = \boldsymbol{y}_0 + \kappa\sqrt{\eta_T}\hat{\boldsymbol{w}}_T\boldsymbol{\epsilon}$;

for $t = T, \dots, 1$ **do**

Sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$;

$\hat{\boldsymbol{x}}_{0|t} = f_\theta(\boldsymbol{x}_t, \boldsymbol{y}_0, t, c)$;

$\hat{\boldsymbol{e}}_{0|t} = \boldsymbol{y}_0 - \hat{\boldsymbol{x}}_{0|t}$;

$\boldsymbol{w}_t = \hat{\boldsymbol{x}}_{0|t} + \eta_t \hat{\boldsymbol{e}}_{0|t}$;

$\boldsymbol{w}_{t-1} = \hat{\boldsymbol{x}}_{0|t} + \eta_{t-1} \hat{\boldsymbol{e}}_{0|t}$;

$\hat{\boldsymbol{w}}_t = b + (1-b)\boldsymbol{w}_t$;

$\hat{\boldsymbol{w}}_{t-1} = b + (1-b)\boldsymbol{w}_{t-1}$;

$\boldsymbol{\gamma}_t = \frac{\eta_{t-1}\hat{\boldsymbol{w}}_{t-1}^2}{\eta_t \hat{\boldsymbol{w}}_t^2}$;

$\boldsymbol{\mu}_{t-1} =$
$\boldsymbol{\gamma}_t(\boldsymbol{x}_t - \alpha_t \hat{\boldsymbol{e}}_{0|t}) + (1-\boldsymbol{\gamma}_t)(\hat{\boldsymbol{x}}_{0|t} + \eta_{t-1}\hat{\boldsymbol{e}}_{0|t})$;

$\boldsymbol{\Sigma}_{t-1} = \kappa^2 \boldsymbol{\gamma}_t(\eta_t \hat{\boldsymbol{w}}_t^2 - \eta_{t-1}\hat{\boldsymbol{w}}_{t-1}^2)\boldsymbol{I}$;

$\boldsymbol{x}_{t-1} = \boldsymbol{\mu}_{t-1} + \sqrt{\boldsymbol{\Sigma}_{t-1}} \odot \boldsymbol{\epsilon}$;

return $\boldsymbol{x}_0$

B. Additional Experiments and Results

This section provides additional quantitative evaluations.

Tab. 8 reports the comparison across existing RGB-to-RAW conversion methods on the over-exposed test set, where SpiralDiff consistently outperforms previous approaches.

Table 8. Quantitative comparison results on the over-exposed test set.

Method	FiveK Canon		FiveK Nikon		NOD Nikon		NOD Sony	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
CycleISP [47]	27.75	0.9833	30.19	0.9872	35.86	0.9957	32.52	0.9890
InvISP [40]	25.15	0.9699	29.11	0.9764	36.66	0.9905	30.85	0.9782
ReRAW [3]	26.36	0.9664	27.51	0.9520	38.56	0.9691	34.96	0.9713
RAW-Diffusion [33]	30.60	0.9739	32.95	0.9850	40.05	0.9926	35.58	0.9792
SpiralDiff	**31.10**	**0.9856**	**34.42**	**0.9908**	**40.79**	**0.9973**	**36.11**	**0.9919**

Tab. 9 presents the ablation study on the LoRA rank r used in CamLoRA. We observe that $r = 8$ achieves the best performance and adopt it as the default setting in all experiments.

Table 9. Ablation study on the LoRA rank r. We set $r = 8$ as the default setting for all experiments.

Rank	FiveK Canon		FiveK Nikon		NOD Nikon		NOD Sony	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$r = 0$	41.99	0.9931	43.19	0.9947	52.06	0.9987	49.52	0.9974
$r = 2$	42.64	0.9928	43.52	0.9948	52.23	0.9987	49.78	0.9977
$r = 4$	42.34	0.9931	**43.95**	0.9948	52.47	0.9988	49.82	0.9977
$r = 8$	**42.46**	**0.9934**	43.82	**0.9950**	**52.62**	0.9988	**50.08**	0.9977
$r = 16$	42.00	0.9932	42.82	0.9945	52.36	**0.9989**	49.60	**0.9978**

Tab. 10 reports the plug-in experiment. RAW-Diffusion is built on the DDPM [17], and we replace this original diffusion

process with our SpiralDiff while leaving the network architecture and training settings unchanged. The improvement, especially on the FiveK dataset with diverse illumination conditions, validates the effectiveness of our signal-dependent noise weighting strategy and shows that it can be integrated into other RGB-to-RAW reconstruction pipelines.

Table 10. *RAW-Diffusion (DDPM)* denotes the original RAW-Diffusion model with its DDPM-based diffusion process, while *RAW-Diffusion (SpiralDiff)* replaces only this diffusion process with our SpiralDiff formulation, keeping the network architecture unchanged.

Variant	FiveK Canon		FiveK Nikon		NOD Nikon		NOD Sony	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RAW-Diffusion (DDPM)	39.96	0.9890	39.68	0.9866	50.52	**0.9954**	47.31	**0.9908**
RAW-Diffusion (SpiralDiff)	**41.53**	**0.9912**	**42.34**	**0.9920**	**50.54**	**0.9954**	**47.32**	0.9907

We conduct ablation experiments on a real-ISP dataset from the NTIRE'25 RGB-to-RAW conversion track [11], where RGB images are rendered by smartphone ISP pipelines (iPhone and Samsung). Since the official test set is not publicly available, we randomly split the provided training set into 85%/15% for training and evaluation. As shown in Tab. 11, SpiralDiff improves performance on both subsets, and further gains are obtained when integrating CamLoRA.

Table 11. Results on the real-ISP dataset. Models are trained in the *combined* setting.

Method	iPhone		Samsung	
	PSNR	SSIM	PSNR	SSIM
Baseline	30.79	0.9040	34.84	0.9362
SpiralDiff	31.56	0.9094	35.22	0.9394
SpiralDiff + CamLoRA	**31.57**	**0.9100**	**35.96**	**0.9433**

Table 12. Object detection on NOD.

Setting	#Real	#Syn	AP
(a)	1000	0	33.7
(b)	1000	732	35.9
(c)	1732	0	36.2

We further conduct object detection experiments in a data-rich setting. Tab. 12 reports three training settings: (a) 1000 real RAW images from NOD-train, (b) the same 1000 real images augmented with 732 synthetic RAW images converted from NOD-val RGB using SpiralDiff, and (c) 1732 real RAW images from NOD-train+val. Setting (b) outperforms (a) and approaches (c), indicating that our synthetic RAW data serves as a low-cost substitute for augmenting real RAW data in object detection. We also observe a trade-off when using cross-source synthetic data. When training with all real NOD-RAW together with synthesized CS-RAW, performance decreases on the NOD dataset but increases on the BDD dataset. This also happens when performing the same augmentation in the RGB domain. Overall, in data-rich scenarios, same-source synthetic data improves in-dataset performance, whereas cross-source synthetic data enhances generalization at the cost of degraded in-dataset performance.