Inverting Image Signal Processing Pipeline with Diffusion Models

Xinman Liu, Xuanchi Ren, Ziyi Wu

Abstract—In this paper, we study the problem of converting sRGB images back to RAW format. RAW images are the direct outputs from cameras without going through the image signal processing (ISP) pipeline. Since they contain additional information compared to processed sRGB images, RAW image is a highly valuable image format that enables several editing and computer vision tasks. However, due to its large file size, users often only have access to their processed and compressed counterparts. There has been tremendous efforts in designing methods to reverse the ISP pipeline. In this work, we treat this problem as an image-to-image translation task, and leverage the powerful diffusion models to solve it. We test our algorithm on two camera datasets. Quantitative and qualitative results show that our model is competitive with state-of-the-art methods. We also study the generation quality-speed trade-off by experimenting with different sampling strategies. Our code is available here.

Index Terms—Image Signal Processing (ISP), Diffusion Models

1 Introduction

For professional users, RAW images are usually preferred over RGB images since they contain unprocessed scene irradiance. Such information is desirable for attaining more plausible visual effects and various image editing tasks. Recently, researchers point out that RAW images are also valuable for computer vision tasks, such as intrinsic image decomposition [1], image super-resolution [2], [3], image denoising [4], [5], [6], and reflection removal [7], [8]. However, since RAW images are memory-intensive, saving a pair of RAW and RGB images is not feasible, as they are often discarded after the image signal processing (ISP) pipeline. To enable users to get access to the RAW one, inverting the sRGB images to RAW images becomes an important problem in computational photography.

Due to the great advantages of RAW images, there has been several works studying such reverse ISP mapping [3], [4], [9], [10], [11], [12], [13], which can be categorized into two classes. Traditional methods utilize additional information such as parameters of the ISP functions [11], or priors about the camera [4] to compute the reverse process. Recently, data-driven methods demonstrate their excellent performance using deep neural networks [3], [12], [13]. They treat the sRGB-RAW mapping as an image-to-image translation problem, and apply advanced generative models such as GANs [14], [15] and Normalizing-Flows [16], [17].

Inverting the ISP pipeline is a challenging problem since it is a lossy pipeline, converting 12 or 14-bit RAW data to 8-bit RGB data. ISP steps such as denoising, tone mapping, and quantization all lead to inevitable information loss. Especially, the over-exposed regions totally lose the corresponding data, resulting in a harder inversion. Moreover, there is also a lossy image compression process in the digital camera to save the RGB in the JPEG format. The current state-of-the-art method, Invertible ISP [13], utilizes the normalizing-flow-based models [16] with a differentiable JPEG simulator. However, due to the capacity limitation of flow-based models (as they need to preserve the invertibility of network), their performance still has room for improvement, especially for the over-exposed areas.

Recently, diffusion models have demonstrated great success in image generation tasks [18], [19], [20], [21]. By decomposing the image generation process into multiple rounds of denoising operation, diffusion models are able to synthesize images of high quality. Later works [22], [23] also show that these models are good at conditional generation, thus suitable for image-to-image translation tasks. This inspires us to apply diffusion models on inverting the ISP process. With their incredible generative power, in this work, we seek to develop a fully end-to-end framework to directly synthesize RAW images from the sRGB ones. In summary, this project makes the following contributions:

- To the best of our knowledge, our method is the first attempt in RAW image reconstruction via diffusion models. Quantitative and qualitative results on two camera datasets show that we achieve competitive performance with state-of-the-art approaches;
- Diffusion models are notoriously slow and memory-consuming in the generation process. We study the quality-speed trade-off with different sampling algorithms in our experiments, and provide insights in which algorithm to choose under different scenarios.

2 Related Work

2.1 RAW Image Reconstruction

There has been several works researching reconstructing RAW images from their sRGB counterparts [3], [4], [9], [10], [11], [12], [13], which can be mainly categorized into two classes depending on whether they still follow the traditional steps in the ISP pipeline.

Methods following ISP pipeline. Nguyen et al. [11] store the parameters of ISP into a 64KB overhead in the metadata

- *X. Liu (1004370637), X. Ren (1009173403), and Z. Wu (1007807526) are with University of Toronto.*
- *Authors are listed alphabetically.*

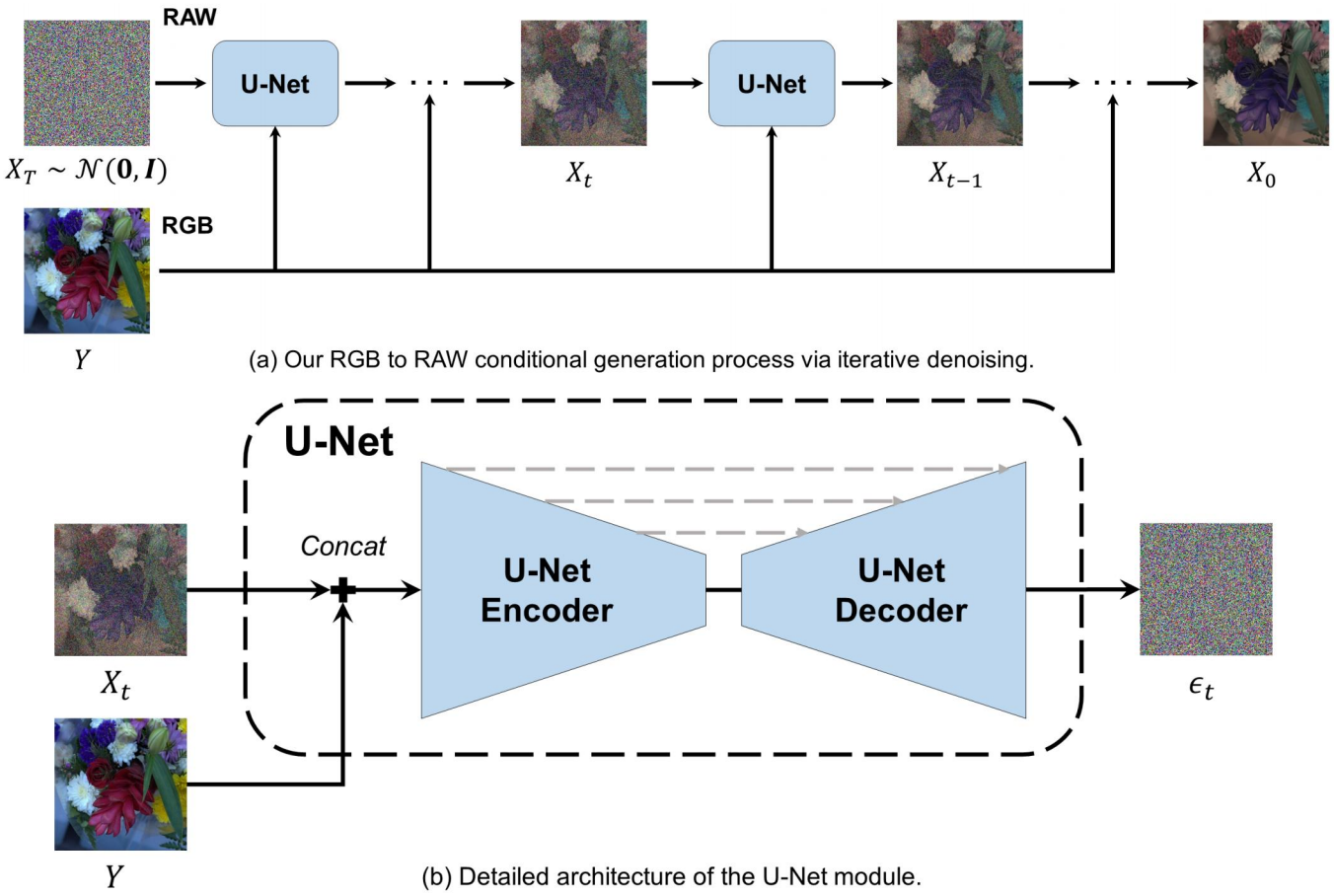

Fig. 1. Overview of our framework. The top figure (a) shows the iterative RAW image reconstruction procedure, where a U-Net performs denoising conditioned on the RGB image. The bottom figure (b) illustrates the architecture of U-Net with skip-connections, which takes in the concatenation of a noisy RAW image X_t and an RGB image Y, and predicts the noise ϵ_t, which is then used to obtain X_{t-1}.

of their processed JPEG images, which can be used to map RGB images back to RAW data. Brooks et al. [4] leverages neural networks to learn camera-specific priors, and use it to reverse ISP step by step. These methods are interpretable as they follow the manually designed ISP pipeline. However, their performance is bounded by the inevitable error accumulation issue in each step, thus underperforming end-to-end learning-based methods.

Methods re-designing ISP pipeline. These approaches replace traditional ISP steps with learned neural network modules. Afifi et al. [9] propose to model the RAW recovery process with camera-independent CIE-XYZ color space. CycleISP [3] models the RGB-RAW-RGB pipeline in a cycle manner, and can perform bi-direction image conversion. Closet to our work, InvISP [13] proposes to leverage the invertible neural network [16] to merge RAW-to-RGB and RGB-to-RAW mapping in a single model. Similarly, we also replace all the irreversible intermediate ISP steps with a deep generative model. Unlike previous works, we target to utilize the generative power of diffusion models [18] to compensate for the gap between RAW data and RGB data, especially for the over-exposed part.

2.2 Diffusion Models

Recently, diffusion models have achieved tremendous progress in generation tasks, including images [18], [19], [20], videos [24], [25], [26], and 3D shapes [27], [28], showing their great ability in density estimation and sample quality. The generative process of diffusion models is formulated as an iterative denoising procedure [18] with a powerful U-Net [29]. Later works show that diffusion models are also good at conditional generation tasks, such as text-to-image generation [21], [30], image super-resolution [22], and image inpainting [23]. The later two tasks belong to the family of image-to-image translation problem, where the input and target data have the same shape, and RGB to RAW conversion also belongs to it. While researchers demonstrate that diffusion models can achieve amazing performance in several domains, there is no work applying them to the task of RAW image reconstruction to fully dig into the potential of their generative power for the lost information.

3 Method

In this section, we describe our diffusion model based RGB to RAW image reconstruction pipeline. Given an image $\boldsymbol{Y} \in \mathbb{R}^{H \times W \times 3}$ in the sRGB space, we aim to synthesize the corresponding demosaiced RAW image $\boldsymbol{X}_0 \in \mathbb{R}^{H \times W \times 3}$, which can then be transformed via the Bayer sampling function to obtain the target RAW data $\boldsymbol{Z} \in \mathbb{R}^{H \times W \times 1}$. Below we first review some basic concepts of diffusion models (Section 3.1), then detail our conditional generation framework (Section 3.2). An overview of our framework is presented in Figure 1.

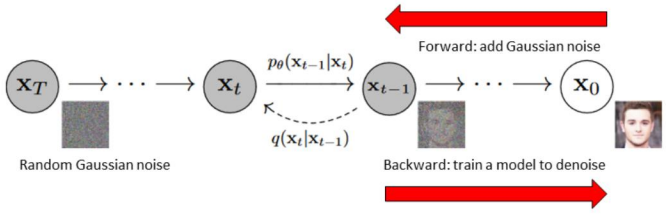

Fig. 2. Forward and reverse process of a typical diffusion model. Image adopted from Figure 2 of [18].

3.1 Review of Diffusion Models

Diffusion models are probabilistic models that learn a data distribution $p_\theta(\boldsymbol{X}_0)$ by gradually denoising a standard Gaussian distribution, in the form of $p_\theta(\boldsymbol{X}_0) = \int p_\theta(\boldsymbol{X}_{0:T})\, d\boldsymbol{X}_{1:T}$. Here, $\boldsymbol{X}_{1:T}$ are intermediate denoising results with the same shape as $\boldsymbol{X}_0 \sim q(\boldsymbol{X})$, and θ are learnable parameters of the deep denoising network, usually realized as a U-Net [29].

Figure 2 shows the graphical model of a typical diffusion model [18]. The joint distribution $q(\boldsymbol{X}_{1:T}|\boldsymbol{X}_0)$ is called the *forward process* or *diffusion process*, which is a fixed Markov Chain that gradually adds Gaussian noise to the clean data $\boldsymbol{X}_0$. The noise is controlled by a pre-defined variance schedule $\{\beta_t\}_{t=1}^T$:

$$q(\boldsymbol{X}_{1:T}|\boldsymbol{X}_0) = \prod_{t=1}^{T} q(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) \tag{1}$$

$$\begin{aligned} q(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}) &= \mathcal{N}(\sqrt{1-\beta_t}\boldsymbol{X}_{t-1}, \beta_t \boldsymbol{I}) \\ &= \sqrt{1-\beta_t}\boldsymbol{X}_{t-1} + \beta_t \epsilon_t \\ &\text{where } \epsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I}) \end{aligned} \tag{2}$$

Thanks to the nice property of Gaussian distributions, an good property of this formulation is that $\boldsymbol{X}_t$ can be sampled directly from $\boldsymbol{X}_0$ in closed form without adding the noise t times. Let $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, we have:

$$\begin{aligned} q(\boldsymbol{X}_t|\boldsymbol{X}_0) &= \mathcal{N}(\sqrt{\bar{\alpha}_t}\boldsymbol{X}_0, (1-\bar{\alpha}_t)\boldsymbol{I}) \\ &= \sqrt{\bar{\alpha}_t}\boldsymbol{X}_0 + (1-\bar{\alpha}_t)\epsilon_t \end{aligned} \tag{3}$$

We can now train a model to reverse this process and thus generate target data from random noise $\boldsymbol{X}_T \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. The *reverse process* $p_\theta(\boldsymbol{X}_{0:T})$ is also defined as a Markov Chain with a learned Gaussian transition:

$$\begin{aligned} p_\theta(\boldsymbol{X}_{0:T}) &= p(\boldsymbol{X}_T) \prod_{t=1}^{T} p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t) \\ p_\theta(\boldsymbol{X}_{t-1}|\boldsymbol{X}_t) &= \mathcal{N}(\mu_\theta(\boldsymbol{X}_t, t), \Sigma_\theta(\boldsymbol{X}_t, t)) \end{aligned} \tag{4}$$

In practice, we do not learn the variance and usually set it to $\Sigma_t = \beta_t \boldsymbol{I}$. Also, instead of learning the mean μ_θ directly, we learn to predict the noise ϵ_t in Equation (2). See [18] for how we can compute $\boldsymbol{X}_{t-1}$ given $\boldsymbol{X}_t$ and the predicted $\hat{\epsilon}_t$.

The training process of diffusion models is thus simple given Equation (3). At each step, we sample a batch of clean data $\boldsymbol{X}_0$ from the training set, timestamps t uniformly from $\{1, ..., T\}$, and random Gaussian noise $\epsilon_t \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$. We then create the noisy version of data $\boldsymbol{X}_t$ by applying Equation (3). A denoising model ϵ_θ is utilized to predict the noise via $\hat{\epsilon}_t = \epsilon_\theta(\boldsymbol{X}_t, t)$. The entire network is trained end-to-end via an MSE loss:

$$\mathcal{L}_{DM} = \mathbb{E}_{\boldsymbol{X},t,\epsilon}[||\epsilon_t - \epsilon_\theta(\boldsymbol{X}_t, t)||^2] \tag{5}$$

3.2 RAW Reconstruction with Diffusion Models

The diffusion model introduced in the above section can perform unconditional generation tasks excellently. However, our task requires generating RAW images conditioned on their sRGB counterparts. Figure 1 (a) demonstrates the desired generation process of our framework, where the denoising network is guided by the RGB image $\boldsymbol{Y}$ to synthesize the RAW data $\boldsymbol{X}_0$.

Inspired by previous works [22], [23] that perform similar image-to-image translation tasks, we employ channel-wise concatenation with the RGB image $\boldsymbol{Y}$ to guide the denoising process towards generating the corresponding RAW image $\boldsymbol{X}_0$. As shown in Figure 1 (b), we simply concatenate $\boldsymbol{Y}$ and $\boldsymbol{X}_t$ as the input to the denoiser ϵ_θ, and predict the added noise ϵ_t. The training loss in Equation (5) is thus modified to:

$$\mathcal{L}_{CDM} = \mathbb{E}_{(\boldsymbol{X},\boldsymbol{Y}),t,\epsilon}[||\epsilon_t - \epsilon_\theta(\boldsymbol{X}_t, \boldsymbol{Y}, t)||^2] \tag{6}$$

3.3 Implementation Details

Here we illustrate some of our design choice and training details of the model.

Diffusion model. We follow DDPM [18] to use a U-Net [29] as the denoising network ϵ_θ. It consists of four residual block stages in both the encoder and decoder, with skip connections in-between to facilitate local and global feature fusion. Each stage is implemented as $N_{res} = 1$ residual block [31] (i.e. the `BasicBlock` module used in ResNet-18), with Group Normalization [32] and Swish [33] activation function. In addition, all residual stages (except the first one) are followed by a 2x down- or up-sampling layer, and a spatial attention layer, which is realized as a scaled dot-product attention [34] that takes in all entries of the feature maps. This module performs global interactions of features, and is proved vital for the expressiveness of diffusion models [19]. We set the base channel number as $N_\epsilon = 64$ in U-Net, which is multiplied by $\{1, 2, 3, 4\}$ in the four stages. For the noise variance schedule $\{\beta_t\}_{t=1}^T$ in the forward process, we follow LDM [21] to use a linear schedule that linearly increase from $\beta_1 = 0.0015$ to $\beta_T = 0.0195$ over $T = 1000$ steps. In our own research, we find this schedule generally performs better in conditional generation tasks than the one used in the original DDPM paper.

Model training. Diffusion models are memory-consuming due to the spatial attention module, which scales quadratically with the training image resolution. Similarly, our closest baseline InvISP [13] also faces such memory issue, as they adopt a normalizing flow network [16] whose feature maps have the same dimensionality as input images to preserve invertibility. Therefore, they design a patch-based protocol, where the model is trained on small patches randomly cropped from the image. At test time, an image is split into regular patch grids as model inputs, and merged back for metrics evaluation. In this work, we adopt the same setting as InvISP and train our model on patches of shape

Fig. 3. Sample RGB-RAW images from the MIT-Adobe FiveK dataset. Note that the RAW data has undergone demosaicing and gamma correction.

64×64. We use the Adam [35] optimizer and train the model for 20k steps. The learning rate is first linearly warmed-up to 2×10^{-4} in the first 1k steps, and decayed to 2×10^{-6} throughout the training in a cosine schedule. We also clip the maximum L_2 norm of gradients to 1 for stabilizing training. We train our model with a batch size of 256. The codebase is implemented in PyTorch [36] and trained on 4 NVIDIA V100 GPUs, each with 32GB memory.

TABLE 1
Quantitative evaluation between our method and baselines.

Method	PSNR↑	
	NIKON D700	Canon EOS 5D
UPI	30.12	26.31
CycleISP	30.19	34.48
InvGrayscale	33.28	38.00
U-Net	41.17	41.14
InvISP	**44.19**	**45.73**
Ours	40.10	41.41

4 Experiments

In this section, we evaluate the RAW reconstruction results of our diffusion model. We first detail the datasets used in our experiments and baselines we compare with (Section 4.1). Then, we present quantitative (Section 4.2) and qualitative (Section 4.3) results to verify the effectiveness of our method. Finally, ablation study (Section 4.4) is conducted to investigate several design choices we made and the quality-speed trade-off in the sample generation process.

4.1 Experimental Setup

Datasets. We use the same RAW-RGB dataset as InvISP [13], which is a subset from the MIT-Adobe FiveK dataset [37]. Specifically, we collect 590 RAW data from the NIKON D700 camera and 777 RAW data from the Canon EOS 5D camera. Following InvISP, we use the 85:15 train-test data split, and train two diffusion models on the two camera data separately. The ground-truth sRGB images are rendered with representative ISP steps in modern digital cameras. Notably, since demosaicing and gamma correction are reversible steps, the ground-truth RAW images are processed by them in advance. Sample RGB-RAW image pairs of the dataset can be found in Figure 3.

Evaluation Metrics. We adopt PSNR to measure the reconstruction quality of RAW images. We also consider common image quality metrics such as SSIM [38] and perceptual distance [39]. However, they are not well-defined in the RAW data space. So we do not report them.

Baselines. We adopt the baselines from [13] and copy the numbers from its paper. *UPI* leverages learned camera priors to invert the ISP pipeline step-by-step. *CycleISP* [3] trains a CNN model to perform the RGB-RAW-RGB bi-directional conversion jointly. *InvGrayscale* [40] and *U-Net* [13] both apply a U-Net structure to reconstruct RAW data from sRGB images. *InvISP* [13] applies an inherently invertible Normalizing Flow network to learn the ISP process. For fair comparison, we adopt the InvISP variant without differentiable JPEG simulation.

4.2 Quantitative Results

Table 1 shows quantitative results regarding the PSNR metrics on the MIT-Adobe FiveK dataset [37]. Our method achieves competitive performance on both types of cameras. The Canon EOS 5D part is larger than the NIKON D700 part, and thus, our model is able to outperform U-Net [29] on the former part. This demonstrates the great potential of applying the diffusion model on RAW image reconstruction and indicates that it would be better to train diffusion models with large-scale datasets.

Fig. 4. Visualization of our result on NIKON D700 dataset. Our diffusion-based method is capable of synthesizing plausible RAW images.

4.3 Qualitative Results

Fig. 4 provides quantitative results in resolution 1024×1024. We can synthesize plausible RAW images based on only RGB images in diverse scenarios. Though our model is trained on the patch-level, we can apply our method to arbitrary high-resolution images via running our model on overlapping patches and merging them via weighted sum.

4.4 Ablation Study

We conduct ablation study on the NIKON D700 subset.

TABLE 2
Ablation study on model design on NIKON D700 subset.

Method	PSNR ↑
Ours (Full Model)	**40.10**
Residual block number $N_{res} = 2$	40.00
U-Net base channel $N_\epsilon = 32$	12.84
DDPM variance schedule $\{\beta_t\}_{t=1}^T$	37.61
Patch size 128×128	15.41

4.4.1 Model Design Choices

Table 2 shows the model performance regarding different settings. Using two residual blocks in each stage leads to slightly worse result, which may because doubling the model size leads to overfitting. In contrast, using half the number of U-Net base channels significantly degrades the performance. This proves that our current model architecture is a good fit for the dataset. For the diffusion process, using DDPM's variance schedule makes PSNR drop by 3. This coincides with previous research [21] that shows better conditional generation quality with the schedule we use. Finally, we experiment on patches of shape 128×128. However, we have to use a much smaller batch size due to large memory consumption, which results in low PSNR.

TABLE 3
Ablation study on sampling strategy of diffusion models. All speed is measured on 8 NVIDIA V100 GPUs with a batch size of 512 patches.

Method	PSNR ↑	Time per Image (s) ↓
Ours (DDPM)	**40.10**	421.76
DDIM	27.71	166.12
DPM-Solver	28.49	**10.34**

4.4.2 Sampling Strategy

Diffusion models are notoriously slow in generation speed due to its iterative sampling procedure. To generate an image, DDPM needs to perform forward pass $T = 1000$ times. This is even worse in our task since we work on 64×64 patches, and an image will be split into more than 1,000 patches. There is a line of research studying accelerating the generation process of diffusion models. We examine three sampling strategies here, namely, naive DDPM, DDIM [41], and DPM-Solver [42]. For DDIM, we follow the original implementation and use 200 sampling steps. For DPM-Solver, we follow the recommended setting to use a 3rd-order single-step solver with 20 sampling steps.

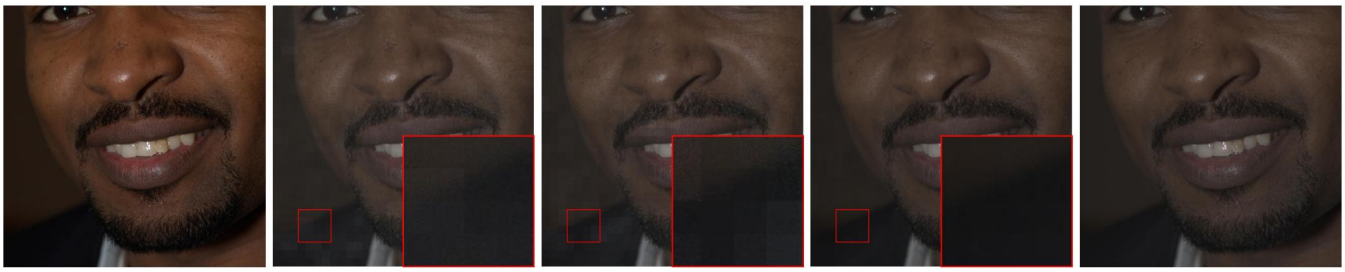

Fig. 5. Qualitative comparison between different sampling strategies. Though DDPM consumes much more time than other sampling strategies, it achieves the best RAW reconstruction quality. Note the blockly artifacts highlighted in the red windows.

Table 3 shows the generation quality and speed trade-off with different sampling strategies. Our vanilla DDPM sampling leads to the best performance, but also requires over 7 minutes to generate just one image. DDIM accelerates the sampling by $2.5\times$, while degrading the PSNR drastically. Similarly, DPM-Solver also leads to large PSNR drop, but it also provides over $40\times$ speedup.

Qualitatively, as shown in Figure 5, the global appearance of the reconstructed RAW images by different methods are not that distinct as presented in the numerical results. However, when zooming in into local details, we observe severe blockly artifacts in images generated by DDIM and DPM-Solver. This is because the noise is not completely removed due to their smaller sampling steps, which leads to color perturbations. Since the visual results are merged from multiple 64×64 patches, such color inconsistency is exaggerated, causing significant performance drop.

Overall, the selection of sampling strategy depends on the actual use case. For tasks that require high accuracy reconstruction like RAW image reconstruction, vanilla DDPM provides the best performance. While for applications such as content generation in gaming, fast algorithms such as DPM-Solver can be a good fit.

5 Conclusion and Future Works

In this paper, we propose a diffusion model based method to invert the traditional ISP pipeline in modern digital cameras. It treats the sRGB to RAW image reconstruction task as a general image to image translation problem, and leverages the strong generative capability of diffusion models to tackle it. Extensive experiments on two camera datasets demonstrate the effectiveness of our method both quantitatively and qualitatively. We also conduct ablation study to verify our design choices, and present preliminary investigations in the effect of different sampling algorithms.

Limitations and Future Works. Our base U-Net model is directly adopted from previous diffusion model papers without special designs to accommodate the inverse ISP task. One can explore injecting priors about the ISP process to the network design, such as modules for inverting the denoising step. Another limitation is that we train our model on patches of shape 64×64 due to the huge memory consumption of diffusion models. This eliminates the global effects in RAW to RGB conversions, as the model cannot perform long-range reasoning between patches. We tried to train a model with larger patch size 128, but failed due to the quadratically increasing memory requirement. One possible solution is to adopt the Latent Diffusion Model (LDM) framework [21], which first trains a VAE model to convert images to latent feature maps, and then learns the diffusion generation process in the latent space. LDM demonstrated impressive generation quality in both unconditional and conditional cases. However, LDM often require tens of thousands of training data, while the paired RGB-RAW datasets are often small (less than 1,000 pairs). One viable solution is to freeze a pre-trained LDM such as Stable-Diffusion[1], and only fine-tune or re-train a new image encoder/decoder.

Overall, we believe the task of inverting ISP pipeline holds great potential for several photography and computer vision applications, and that our work is a new step towards this goal.

References

[1] S. Bell, K. Bala, and N. Snavely, "Intrinsic images in the wild," *TOG*, 2014.
[2] X. Zhang, Q. Chen, R. Ng, and V. Koltun, "Zoom to learn, learn to zoom," in *CVPR*, 2019.
[3] S. W. Zamir, A. Arora, S. H. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao, "Cycleisp: Real image restoration via improved data synthesis," in *CVPR*, 2020.
[4] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, "Unprocessing images for learned raw denoising," in *CVPR*, 2019.
[5] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ToG*, 2016.
[6] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll, "Burst denoising with kernel prediction networks," in *CVPR*, 2018.
[7] C. Lei and Q. Chen, "Robust reflection removal with reflection-free flash-only cues," in *CVPR*, 2021.
[8] C. Lei, X. Huang, M. Zhang, Q. Yan, W. Sun, and Q. Chen, "Polarized reflection removal with perfect alignment in the wild," in *CVPR*, 2020.
[9] M. Afifi, A. Abdelhamed, A. Abuolaim, A. Punnappurath, and M. S. Brown, "CIE XYZ net: Unprocessing images for low-level computer vision tasks," *T-PAMI*, 2022.
[10] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-image hdr reconstruction by learning to reverse the camera pipeline," in *CVPR*, 2020.
[11] R. M. Nguyen and M. S. Brown, "Raw image reconstruction using a self-contained srgb–jpeg image with small memory overhead," *IJCV*, 2018.
[12] A. Punnappurath and M. S. Brown, "Learning raw image reconstruction-aware deep image compressors," *T-PAMI*, 2019.
[13] Y. Xing, Z. Qian, and Q. Chen, "Invertible image signal processing," in *CVPR*, 2021.
[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, 2020.

1. https://huggingface.co/CompVis/stable-diffusion

[15] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *TKDE*, 2021.

[16] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *ICLR*, 2017.

[17] I. Kobyzev, S. J. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *T-PAMI*, 2020.

[18] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020.

[19] A. Q. Nichol and P. Dhariwal, "Improved denoising diffusion probabilistic models," in *ICML*, 2021.

[20] P. Dhariwal and A. Nichol, "Diffusion models beat gans on image synthesis," in *NeurIPS*, 2021.

[21] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[22] C. Saharia, J. Ho, W. Chan, T. Salimans, D. J. Fleet, and M. Norouzi, "Image super-resolution via iterative refinement," *T-PAMI*, 2022.

[23] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, and M. Norouzi, "Palette: Image-to-image diffusion models," in *ACM SIGGRAPH 2022 Conference Proceedings*, 2022.

[24] J. Ho, T. Salimans, A. Gritsenko, W. Chan, M. Norouzi, and D. J. Fleet, "Video diffusion models," in *NeurIPS*, 2022.

[25] W. Harvey, S. Naderiparizi, V. Masrani, C. Weilbach, and F. Wood, "Flexible diffusion modeling of long videos," in *NeurIPS*, 2022.

[26] U. Singer, A. Polyak, T. Hayes, X. Yin, J. An, S. Zhang, Q. Hu, H. Yang, O. Ashual, O. Gafni *et al.*, "Make-a-video: Text-to-video generation without text-video data," *arXiv preprint arXiv:2209.14792*, 2022.

[27] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," *arXiv preprint arXiv:2209.14988*, 2022.

[28] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," *arXiv preprint arXiv:2211.10440*, 2022.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

[30] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. Denton, S. K. S. Ghasemipour, B. K. Ayan, S. S. Mahdavi, R. G. Lopes *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," in *NeurIPS*, 2022.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[32] Y. Wu and K. He, "Group normalization," in *ECCV*, 2018.

[33] P. Ramachandran, B. Zoph, and Q. Le, "Swish: A self-gated activation function. arxiv 2017," *arXiv preprint arXiv:1710.05941*, 2017.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[36] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *NeurIPS*, 2019.

[37] V. Bychkovsky, S. Paris, E. Chan, and F. Durand, "Learning photographic global tonal adjustment with a database of input/output image pairs," in *CVPR*, 2011.

[38] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *TIP*, 2004.

[39] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018.

[40] M. Xia, X. Liu, and T.-T. Wong, "Invertible grayscale," *TOG*, 2018.

[41] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," in *ICLR*, 2020.

[42] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps," in *NeurIPS*, 2022.