Realistic Noise Synthesis with Diffusion Models

Qi Wu[1*], Mingyan Han[1*], Ting Jiang[1], Chengzhi Jiang[1], Jinting Luo[1], Man Jiang[1], Haoqiang Fan[1], Shuaicheng Liu[2†]

[1]Megvii Technology Inc.
[2]University of Electronic Science and Technology of China
{wuqiresearch, hanmy628, tjhedlen, morven126, k531756653, himandy1006}@gmail.com,
fhq@megvii.com, liushuaicheng@uestc.edu.cn

Abstract

Deep denoising models require extensive real-world training data, which is challenging to acquire. Current noise synthesis techniques struggle to accurately model complex noise distributions. We propose a novel Realistic Noise Synthesis Diffusor (RNSD) method using diffusion models to address these challenges. By encoding camera settings into a time-aware camera-conditioned affine modulation (TCCAM), RNSD generates more realistic noise distributions under various camera conditions. Additionally, RNSD integrates a multi-scale content-aware module (MCAM), enabling the generation of structured noise with spatial correlations across multiple frequencies. We also introduce Deep Image Prior Sampling (DIPS), a learnable sampling sequence based on depth image prior, which significantly accelerates the sampling process while maintaining the high quality of synthesized noise. Extensive experiments demonstrate that our RNSD method significantly outperforms existing techniques in synthesizing realistic noise under multiple metrics and improving image denoising performance.

Code — https://github.com/wuqi-coder/RNSD

Introduction

In deep learning, image denoising (Wang et al. 2022; Zamir et al. 2021, 2022; Guo et al. 2019; Zhang, Zuo, and Zhang 2018; Kim et al. 2020) is an ill-posed problem that often necessitates supervised training with extensive data pairs. A noisy image $\mathbf{y}$ in the RGB domain can be modeled as its noise-free version $\mathbf{s}$ plus noise $\mathbf{n}$ after Image Signal Processing (ISP) using the following equation:

$$\mathbf{y} = \mathbf{ISP}(\mathbf{s} + \mathbf{n}). \tag{1}$$

In contrast to the linearly modelable and spatially independent noise in RAW, noise in the RGB domain exhibits: **Irregular and Diverse Noise Distribution.** ISP post-processing parameters—such as AWB, CCM, and GAMMA—result in non-uniform noise variations across scenes, channels, ISO levels, and pixels, due to their dependence on sensor, ISO, scene, and exposure settings; **Structural and Spatial Correlation of Noise.** Spatially dependent ISP operations, including demosaicing, denoising, and sharpening, introduce local structural patterns to the noise, which increases its correlation with the signal-to-noise ratio.

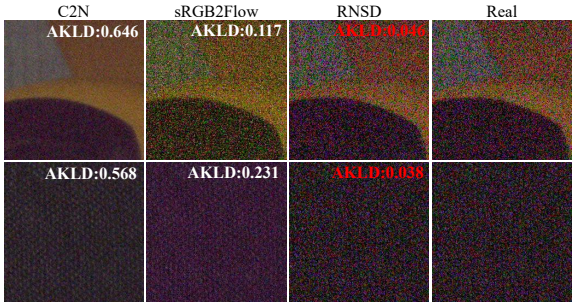

Figure 1: Subjective results and AKLD (Yue et al. 2020) of various noise synthesis methods, including sRGB2Flow (Kousha et al. 2022), DANet (Yue et al. 2020), and C2N (Jang et al. 2021).

Most datasets (Plotz and Roth 2017; Xu et al. 2018; Nam et al. 2016) rely on multi-frame averaging, which is not only challenging to obtain but also fails to provide diverse noise types and cannot address structural noise. Some approaches (Foi et al. 2008; Foi 2009; Brooks et al. 2019) model noise as Gaussian white noise, neglecting the spatial correlations present in real noise. GAN-based methods (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; San-Roman, Nachmani, and Wolf 2021; Bansal et al. 2022; Chen et al. 2022) have attempted to model real noise distributions but often struggle with instability and mode collapse due to the lack of a rigorous likelihood function, leading to a mismatch between generated and real noise distributions. In contrast, diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; San-Roman, Nachmani, and Wolf 2021; Bansal et al. 2022; Chen et al. 2022) offer more stable and varied image generation due to their rigorous likelihood derivations. However, they have yet to be successfully applied to synthetic noise generation, possibly due to insufficient conditioning designs for complex noise distributions with spatial correlations.

In this paper, we introduce Realistic Noise Synthesize

*These authors contributed equally.
†Corresponding author

Diffusor (RNSD), a novel method for synthesizing realistic rgb noise data based on the diffusion model. RNSD has the capability to generate a large amount of noise images that closely resemble the distribution of real-world noise by clean images from various public datasets. The augmented data generated by RNSD significantly enhances the performance of existing denoising models, both in terms of noise reduction and image fidelity.

Specifically, RNSD uses real noisy images $\mathbf{y}$ as initial state $\mathbf{x_0}$ to build Diffusion for noise generation. To effectively accommodate diverse noise distributions, we propose a time-ware camera conditioned affine modulation, termed **TCCAM**. This module encodes varying camera settings and employs a time-adaptive conditioned affine transformation during the sampling process, permitting RNSD to synthesize diversity and realistic noise.

Additionally, we constructed a multi-scale content-aware module, **MCAM**, which integrates multi-scale guidance information of clean image into the diffusion network. This module efficiently guides the generation of signal-correlated and spatial-correlated noise.

Based on the depth image prior that the network first learns low-frequency components and then high-frequency components, we developed Deep Image Prior Sampling (**DIPS**). Unlike DDIM, DIPS uses a distillation-based single-step model with decay sampling, reducing a 1000-step model to just 5 steps with only a 4% accuracy loss, significantly improving sampling efficiency.

To summarize, our main contributions are as follows:

- We first propose a real noise data synthesis approach **RNSD** based on the diffusion model.
- We design a time-ware camera conditioned affine modulation, **TCCAM** that can better control the distribution and level of generated noise.
- By constructing the multi-scale content-aware module **MCAM**, the coupling of multi-frequency information is introduced, enabling the generation of more realistic noise with spatial correlation.
- **Deep Image Prior Sampling (DIPS)**: Leveraging the depth image prior that the network learns low-frequency before high-frequency components, DIPS improves sampling efficiency by reducing a 1000-step model to just 5 steps with only a 4% accuracy loss, compared to DDIM.
- Our approach achieves state-of-the-art results on multiple benchmarks and metrics, significantly enhancing the performance of denoising models.

Related Work

Noise Models. In digital imaging, noise sources include read noise, shot noise, fixed-pattern noise and so on. Some methods (Foi et al. 2008; Foi 2009) use the Gaussian-Poisson model, where read noise is signal-independent and modeled as Gaussian, while shot noise, which is signal-dependent, is modeled as Poisson noise. Shot noise is often approximated as Gaussian for simplicity (Brooks et al. 2019; Guo et al. 2019). However, complex ISP operations like demosaicing, tone mapping, HDR, and sharpening can disrupt noise regularity and introduce complex spatial correlations, complicating traditional noise modeling.

GAN-based Methods. GANs (Karras et al. 2017; Karras, Laine, and Aila 2019; Brock, Donahue, and Simonyan 2018; Shaham, Dekel, and Michaeli 2019) are known for their strong data distribution fitting in image generation. Real noise can be viewed as a data distribution, leading to efforts in synthesizing noise with GANs. Jiang *et al.*(Jang et al. 2021) demonstrated that GANs can synthesize real noise using Gaussian noise inputs and unsupervised conditions. Cai *et al.*(Cai et al. 2021) used a pre-training network to align generated content and noise domains. Despite their potential, GANs often face instability and poor convergence due to the absence of explicit maximum likelihood (Lucic et al. 2018; Mescheder, Geiger, and Nowozin 2018).

Diffusion Methods. Diffusion models (Ho, Jain, and Abbeel 2020; Song, Meng, and Ermon 2020; San-Roman, Nachmani, and Wolf 2021; Bansal et al. 2022; Chen et al. 2022) handle the complex and diverse distribution of real noise, which can be influenced by sensor type, ISO, and ISP. Unlike GANs, diffusion models avoid mode collapse and provide more diverse results. However, they have not been effectively applied to synthetic noise generation, potentially due to the lack of conditioning designs for handling complex and varied noise distributions.

Methodology

We propose a novel diffusion-based method for synthesizing realistic noisy data, termed Real Noise Synthesis via Diffusion (**RNSD**) (Fig. 2 (a)). Our method leverages real noisy images as initial conditions and incorporates a Time-aware Camera Conditioned Affine Modulation (**TCCAM**) (Fig. 2 (b)) to control the results. Additionally, we introduce a Multi-scale Content-aware Module (**MCAM**) (Fig. 2 (c)) to guide the generation of signal-correlated noise. Finally, we designed a learnable accelerated sampling method based on depth image prior(**DIPS**) (Ulyanov, Vedaldi, and Lempitsky 2018) as shown in algorithm 2.

Noise Generation via Diffusion

Traditional diffusion models are usually trained on noise-free style data, which can sample target domain images from any Gaussian noise distribution. In contrast, we treat images with real noise distributions as target domain images. As shown in Fig. 2 (a), by replacing $\mathbf{x}_0$ with real noise distribution data $\mathbf{y}$ and through simple settings, the diffusion model sample real noise from any Gaussian noise distribution.

Specifically, we adopt the probability model of DDPM (Ho, Jain, and Abbeel 2020). In the forward process, a T-step Markov chain is used to minimize the prior probability $q(\mathbf{x}_T|\mathbf{x}_0)$, that is diffusing $\mathbf{x}_0$ to a pure Gaussian distribution $\mathbf{x}_T$ with a variance noise intensity β_t:

$$\begin{aligned} q(\mathbf{x}_T|\mathbf{x}_0) &= \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1}), \\ q(\mathbf{x}_t|\mathbf{x}_{t-1}) &= \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}), \end{aligned} \tag{2}$$

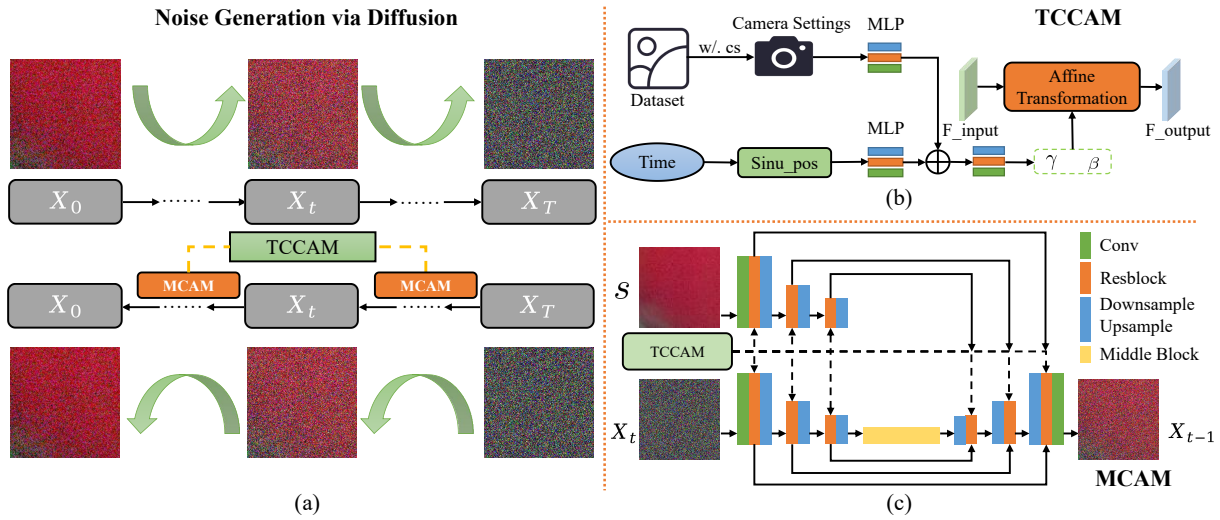

Figure 2: (a) the pipeline of noise generation via diffusion. (b) The pipeline of our TCCAM. (c) The architecture of the UNet with MCAM we designed.

Algorithm 1: RNSD Training

repeat
  $\alpha_t = 1 - \beta_t$, $\overline{\alpha}_t = \prod_{s=1}^{t} \alpha_s$, $t \sim \text{Uniform}(1, \ldots, T)$
  $\mathbf{x_0} \sim \mathbf{q}(\mathbf{x_0})$, $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
  $\mathbf{x}_t = \sqrt{\overline{\alpha}_t}\mathbf{x}_0 + \sqrt{1-\overline{\alpha}_t}\epsilon$
  Compute gradient descent step on:
  $\nabla_\theta \|\epsilon - \epsilon_\theta(\mathbf{x}_t, \mathbf{MCAM(s)}, \mathbf{TCCAM}(t, \mathbf{cs}))\|$
until converged

where $\mathbf{I}$ is a unit covariance matrix. For adjacent two steps, with the help of reparameterization, $\mathbf{x}_t$ can be viewed as sampled from the prior distribution $q(\mathbf{x}_t|\mathbf{x}_{t-1})$, which is regarded as a Gaussian distribution formed by $\mathbf{x}_{t-1}$ and β_t.

The general sampling process is obtained by inversely solving a Gaussian Markov chain process, viewed as a joint probability distribution $p_\theta(\mathbf{x}_{0:T})$, which can be understood as gradual denoising from the above Gaussian distribution $\mathbf{x}_T$ to obtain the sampled result $\mathbf{x}_0$:

$$\begin{aligned} p_\theta(\mathbf{x}_{0:T}) &= p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t), \\ p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}; \mu_\theta(\mathbf{x}_t, t), \Sigma_t), \end{aligned} \tag{3}$$

where μ_θ is the mean of the posterior distribution estimated by the network and Σ_t is a variance obtained by β_t forward calculation. We introduce additional information clean image $\mathbf{s}$ and camera settings information $\mathbf{cs}$ to make the whole process more controllable, mathematically, the process is:

$$\begin{aligned} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) &= \mathcal{N}(\mathbf{x}_{t-1}; \mu_\mathbf{t}, \Sigma_t), \\ \mu_\mathbf{t} &= \mu_\theta(\mathbf{x}_t, \mathbf{s}, \mathbf{cs}, t), \end{aligned} \tag{4}$$

Considering the varying influence of camera settings($\mathbf{cs}$) information across different sampling steps and the spatial correlation of noise in the RGB domain, we formulate enhanced conditioning mechanisms **TCCAM** and **MCAM** to achieve tighter coupling between the noise generation module and the diffusion-based image sampling framework.

Algorithm 2: RNSD Sampling(DIPS)

$\mathbf{x}_T \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, maximum steps T, accelerated steps S
one-step model ψ_θ, closer initial step N
last step $t_{last} = \frac{10}{log(S)}$, density parameter r
for $i = S, ..., 1$ **do**
  $t = t_{last} + (N - t_{last})\frac{e^{r(\frac{i-1}{S-1})}-1}{e^r-1}$
  $t_{next} = t_{last} + (N - t_{last})\frac{e^{r(\frac{i-2}{S-1})}-1}{e^r-1}$
  if t==N **then**
    $\epsilon = \psi_\theta(x_T, t, \mathbf{MCAM(s)}, \mathbf{TCCAM}(t, \mathbf{cs}))$
    $x_t = \sqrt{\bar{\alpha}_t}(\frac{x_T - \sqrt{1-\bar{\alpha}_T}\epsilon}{\sqrt{1-\bar{\alpha}_T}}) + \sqrt{1-\bar{\alpha}_t}\epsilon$
  end if
  $\epsilon = \epsilon_\theta(x_t, t, \mathbf{MCAM(s)}, \mathbf{TCCAM}(t, \mathbf{cs}))$
  if i==1 **then**
    $t_{next} = 0$
  end if
  $x_{t_{next}} = \sqrt{\bar{\alpha}_{t_{next}}}(\frac{x_t - \sqrt{1-\bar{\alpha}_t}\epsilon}{\sqrt{1-\bar{\alpha}_t}}) + \sqrt{1-\bar{\alpha}_{t_{next}}}\epsilon$
end for
$\mathbf{y} = \mathbf{x}_0$
return $\mathbf{y}$

TCCAM: Time-aware Camera Conditioned Affine Modulation

As shown in algorithm. 1, normal diffusion models learn network parameters ϵ_θ to predict the added noise component ϵ_t from $\mathbf{x}_0$ to $\mathbf{x}_t$ during the forward process. However, real noise distributions are determined by multiple factors, including ISO gain, shutter speed, color temperature, brightness, and others. Without distinguishing noise distributions under different conditions, it is challenging to learn a generalized distribution from complex noise based on spatial

photometric variations, ISO changes, and sensor variations. The fundamental issue is that noise distributions vary significantly under different conditions. For instance, noise across different sensors can exhibit entirely different distributions. During learning, the network tends to converge to the overall expectation of the dataset, leading to fixed-mode noise patterns that cause discrepancies between generated and target noise. To address this, we introduce five factors as shown in Algorithm 1:

$$\mathbf{cs} = \phi(iso, ss, st, ct, bm), \tag{5}$$

where iso is ISO, ss is shutter speed, st is sensor type, ct is color temperature, and bm is brightness mode. These factors are embedded using an encoding method (ϕ) as a feature vector of camera settings $\mathbf{cs}$ to control noise generation. This explicit prior narrows the learning domain of the network, enabling it to approximate more complex and variable noise distributions. The influence of camera settings should vary with sampling steps. For example, sensor type (st), strongly correlated with ISP, determines the basic form of noise and its impact is usually coupled with high-frequency information in image content. As t samples from T to 0, recovering image content from low to high frequency, the impact of camera settings gradually increases. To address this, we propose a **TCCAM** with a dynamic setting mechanism, where the weights of different factors vary with sampling steps. As shown in Fig. 2 (b), the process is:

$$\begin{aligned} \gamma, \beta &= MLP_3(MLP_1(sinu\_pos(t)) + MLP_2(\mathbf{cs})) \\ \mathbf{F}_{output} &= \gamma * \mathbf{F}_{input} + \beta \end{aligned} \tag{6}$$

where a Multilayer Perceptron (MLP) is used to encode camera settings together with sampling steps using sinusoidal position encoding ($sinu\_pos$) to generate affine parameters β and γ at each layer of UNet. This method enables a dynamic setting influence mechanism by applying affine transformations to each layer of features $\mathbf{F}_{input}$ in UNet.

MCAM: Multi-scale Content-aware Module

Real noise distribution is inherently linked to image content, varying across brightness regions due to photon capture and ISP processing. Inspired by Zhou et al.'s(Zhou et al. 2020) insights on noise's spatial frequency characteristics, we propose a Multi-Scale Content-aware Module (MCAM) (Fig.2(c)) to model noise-image coupling across different frequencies. Mathematically, our approach is as follows:

$$\begin{aligned} \mathbf{F}_{\mathbf{x}_{t_i}} &= encoder_i(\mathbf{x}_t), \\ \mathbf{F}_{\mathbf{s}_i} &= encoder_i(\mathbf{s}), i = 1, 2, 3, \\ \mathbf{F}_{\mathbf{o}_i} &= decoder_i(Concat(\mathbf{F}_i, \mathbf{F}_{\mathbf{s}_i}, \mathbf{F}_{\mathbf{x}_{t_i}})), \end{aligned} \tag{7}$$

where symmetric but non-shared weight encoders are used to extract features from both $\mathbf{x}_t$ and the clean image $\mathbf{s}$ at three downsampling stages of the encoder. Alongside standard skip connections between $\mathbf{F}_i$ and $\mathbf{F}_{\mathbf{x}_{t_i}}$, we incorporate multi-scale features of $\mathbf{F}_{\mathbf{s}_i}$ in the three upsampling stages.

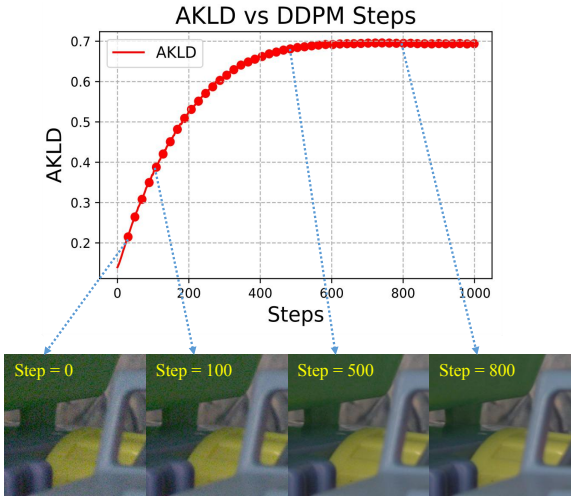

Figure 3: Illustration of the motivation behind DIPS. Upper figure illustrates the variation of AKLD with respect to the steps in DDPM. Lower figure presents a selection of images generated at 0, 100, 500, and 800 steps.

Deep Image Prior Sampling(DIPS)

Based on observations from the depth image prior paper (Ulyanov, Vedaldi, and Lempitsky 2018), where networks first learn clean low-frequency components before high-frequency noise, we noticed a similar pattern during the DDPM sampling process. The decline of AKLD shows an increasingly large gradient as sampling progresses from 1000 steps, indicating that RNSD transitions from low-frequency content to high-frequency noise, as shown in Fig. 3. Since DDIM uses uniform sampling steps, it doesn't align well with our noise estimation task. Consequently, reducing the number of steps leads to a significant performance drop, as demonstrated in our experiments.

To address this, we propose a new sampling method:

$$t = t_{last} + (T - t_{last}) \frac{e^{r(\frac{i-1}{S-1})} - 1}{e^r - 1}, \quad i = S : 1 \tag{8}$$

where T is the sampling step of DDPM, and S is the sampling step of DIPS. t_{last} is last sampling step before step 0. Boundary effect makes last steps' generation weak, so placed in important steps as total sampling steps decrease. r controls the gradient of sampling density (generally set as). For $T = 1000$ and $S = 10$, get sampling sequence [1000, 572, 327, 186, 106, 59, 33, 18, 9, 4, 0], which becomes denser as decreases.

Our basic version, DIPS (DIPS-Basic), reduces sampling steps S to 30 while maintaining quality. Additionally, we find that low-frequency learning from 1000 to 200 steps can be effectively replaced by a single-step model, enabling sampling from a closer truncation step N = 200 instead of T = 1000. The specific formula for this is:

$$\nabla_\theta \| \psi_\theta(x_T, t_N) - \epsilon_\theta(x_N, t_N) \| \tag{9}$$

where a single-step model ψ_θ is distilled from a pre-trained model ϵ_θ, enabling us to reduce initial sampling position

and build deterministic mappings to achieve 5-step sampling while maintaining quality. As follows in Algorithm 2, this method is referred to as **DIPS-Advanced**.

Experiments

Experimental Settings

Datasets. To comprehensively demonstrate the effectiveness of our work, we adopt the following different datasets:

- **SIDD**: The SIDD dataset (Abdelhamed, Lin, and Brown 2018), utilized for training and evaluation, includes subsets such as SIDD small with 160 image pairs from 5 smartphone cameras, and SIDD medium with double the noise sampling. The SIDD validation set (1280 patches from unseen sensor settings) is used to test noise model and denoising models.

- **DND**: DND benchmark (Plotz and Roth 2017) provides 50 reference images and their realistic noisy counterparts generated using accurate sensor noise models. We use it to noise model generalization after RNSD data augmentation.

- **LSDIR**: LSDIR dataset (Li et al. 2023) contains 84,991 high-quality clean samples.

In all experiments, we train RNSD on SIDD small. We augment the noise sampling and scene sampling of the denoising dataset by using RNSD to generate noisy samples from the clean samples of SIDD small and 1000 randomly selected high-quality samples from LSDIR, respectively. The SIDD validation set and DND benchmark are used to assess the accuracy of RNSD noise distribution, as well as the performance and generalization of RNSD-enhanced denoising models.

Evaluation and Metrics. We use PGap and AKLD from (Yue et al. 2020) to assess noise generation. Lower PGap indicates better noise generation, while lower AKLD signifies more similar distributions. The Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) are also used to evaluate the performance of the denoising model.

Implementation Details. We train the DDPM (Ho, Jain, and Abbeel 2020) diffusion model of our noise generation system with 1000 steps, a gradient accumulation step size of 2, and Adam optimizer (lr = 8×10^{-5}). Training samples are 128×128 crops from original images, with a batch size of 16. Models are trained on an NVIDIA GeForce RTX 2080 Ti GPU for 2×10^5 iterations. For testing, we use Exponential Moving Average Decay (EMA) with decay 0.995. Our RNSD model achieves an inference time of 0.15 seconds to generate a batch of 16 128×128 image patches using 5 sampling steps with DIPS on an NVIDIA GeForce RTX 2080 Ti GPU. Evaluation uses DIPS-Basic 30-step, matching DDPM's 1000-step accuracy. When training the denoising networks from scratch, we largely keep the original training hyper-parameters consistent with the baseline denoising models. For finetuning, we reduced the learning rate to 1×10^{-6}, maintaining other parameters, and finetuned for 1×10^6 iterations before evaluation.

Qualitative Comparison

Visual Analysis of Noisy Images. We compare RNSD with baselines such as C2N (Jang et al. 2021), DANet (Yue et al. 2020), and R2F(sRGB2Flow)(Kousha et al. 2022), as shown in Fig.4. RNSD accurately mimics real-world noise patterns across sensors and ISO settings, synthesizing realistic noise while preserving color and tonal accuracy. Fig. 5 shows the controllability of RNSD in incorporating camera settings such as ISO level, shutter speed, and smartphone camera type. The noise intensity in synthesized images increases with ISO level and decreases with shutter speed. Noise patterns also vary depending on the smartphone camera type, showcasing RNSD's ability to modulate noise profiles based on camera metadata.

Metrics	GRDN	C2N	sRGB2Flow	DANet	NeCA	PNGAN	RNSD
AKLD↓	0.443	0.314	0.237	0.212	0.156	0.153	**0.117**
PGap↓	2.28	6.85	6.3	2.06	0.97	0.84	**0.54**

Table 1: The AKLD and PGap performances of various methods on the SIDD validation set.

Method	C2N	NoiseFlow	sRGB2Flow	GMDCN	NeCA	RNSD	Real
PSNR↑	33.98	33.81	34.74	36.07	37.65	**38.11**	38.40

Table 2: Denoising performance comparison between DnCNN trained on purely synthetic noise data and baseline.

Visual Analysis of Denoising Performance. It is evident from Fig. 6 that the DnCNN (Zhang et al. 2017) trained on synthetic noise samples generated by RNSD significantly outperforms other methods in terms of denoising effectiveness, closely matching the performance of the DnCNN trained with realistic noise data.

Quantitative Comparison

Noise Generation. We assess the realism of noise distributions synthesized by different methods using the public evaluation metrics AKLD and PGap on the SIDD validation set. We compare RNSD with baseline techniques including GRDN (Kim, Chung, and Jung 2019), C2N (Jang et al. 2021), sRGB2Flow (Kousha et al. 2022), DANet (Yue et al. 2020), PNGAN (Cai et al. 2021) and NeCA (Fu, Guo, and Wen 2023). As shown in Table 1, our method outperforms the state-of-the-art (SOTA) with a PGap reduced by 0.30 and an AKLD improved by 0.036, indicating more realistic and stable noise synthesis.

Additionally, we evaluate our method using another publicly available metric (Jang et al. 2021) by training the DnCNN network (Zhang et al. 2017) from scratch with synthetic noise generated by RNSD. We compare its performance with C2N (Jang et al. 2021), NoiseFlow (Abdelhamed, Brubaker, and Brown 2019), sRGB2Flow (Kousha et al. 2022), GMDCN (Song et al. 2023), and NeCA (Fu, Guo, and Wen 2023). As shown in Table 2, our synthetic noise improves DnCNN's denoising PSNR by 0.46 dB compared to the SOTA, approaching the performance of real-data training (38.11 dB vs. 38.40 dB).

Enhance Denoising Performance with Data Augmentation. In the actual training process of denoising models, the performance of the model often decreases due to insufficient noise samples and limited sampling scenarios in the dataset. Our RNSD method can improve model performance by increasing noise samples and augmenting scene samples, re-

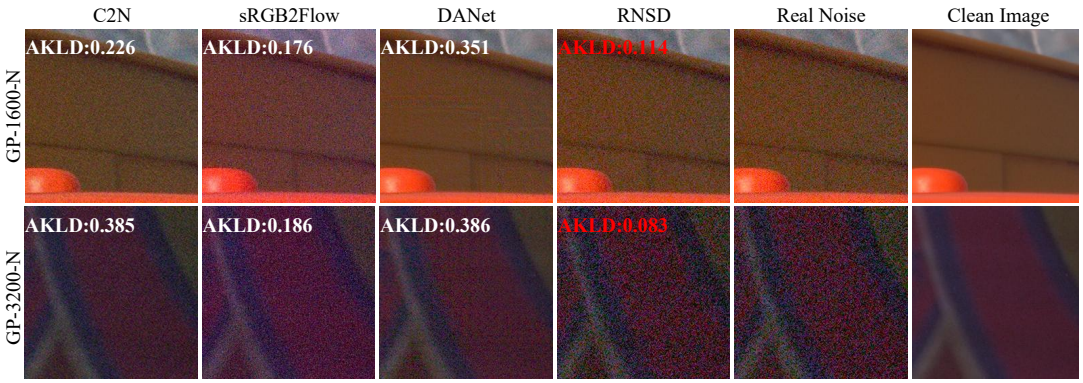

Figure 4: Noisy synthesis samples from different methods, including C2N, sRGB2Flow, DANet and RNSD.

		SIDD validation set					
	Method	DnCNN-B		RIDNet		NAFNet	
	Metric	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Increasing noise samples	SIDD small	37.70	0.947	38.20	0.951	38.95	0.955
	SIDD small+C2N	38.00	0.950	38.47	0.953	39.01	0.956
	SIDD small+DANet	38.05	0.951	38.41	0.953	39.05	0.956
	SIDD small+sRGB2flow	37.79	0.950	38.28	0.952	39.39	0.957
	SIDD small+RNSD	38.27	0.952	38.74	0.954	39.56	0.958

Table 3: Denoising performance improvement brought by existing open-source data synthesis methods for data augmentation.

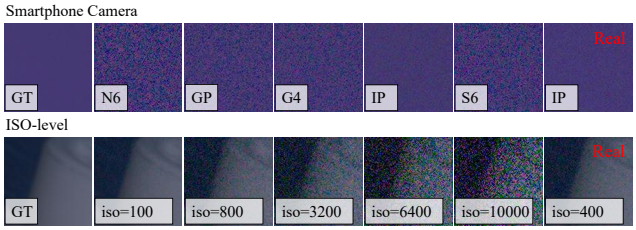

Figure 5: Noise synthesis results with camera settings.

	RIDNet	MIRNet	NBNet	Restormer	NAFNet	RIDNet*	NAFNet*
SIDD	38.77	39.72	39.75	40.02	40.30	39.10	**40.36**
DND	39.29	39.89	39.89	40.03	38.43	39.43	39.05

Table 4: PSNR comparison on SIDD and DND, with * indicating fine-tuning using RNSD noise augmentation.

spectively. Therefore, we verify the effectiveness of RNSD through two experimental setups.

- **Increasing noise samples**: SIDD Medium has twice the noise samples of SIDD Small, so we use SIDD Small to simulate insufficient sampling. RNSD is only trained on SIDD small. We augment SIDD small with different noise generation methods to train denoising models from scratch. As shown in Table. 3, augmenting SIDD Small with RNSD improves denoising performance, boosting PSNR/SSIM by 0.57dB/0.005, 0.54dB/0.003, and 0.61dB/0.003 for DnCNN-B, RIDNet, and NAFNet, respectively. The results validate that our synthesized diverse noise by RNSD is beneficial for training denoising models.

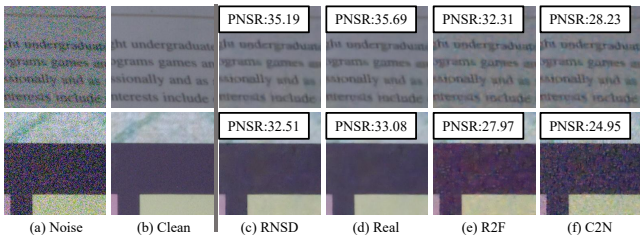

Figure 6: Denoising results on SIDD validation set from DnCNN trained on noisy images from (d) real noisy of SIDD Meduim, (c) RNSD, and (e, f) two of other baselines.

- **Augmenting scene samples**: We use SIDD Medium for ample noise sampling, fine-tuning RIDNet and NAFNet pretrained on SIDD Medium with 1000 clean samples from LSDIR to augment scene samples. We evaluate these enhanced baselines on the SIDD and DND (Plotz and Roth 2017) datasets. Results in Table 4 show significant performance improvements: RIDNet gains 0.33 dB and NAFNet gains 0.06 dB up to state-of-the-art levels. Notably, NAFNet shows a 0.62 dB improvement on the DND dataset. Our model, trained on a small subset of SIDD, demonstrates that augmenting training data enhances both denoising performance and generalization.

In summary, quantitative results demonstrate that RNSD effectively augments data, boosting denoising performance and generalization. This validates its ability to enhance model robustness via increased noise and sample diversity.

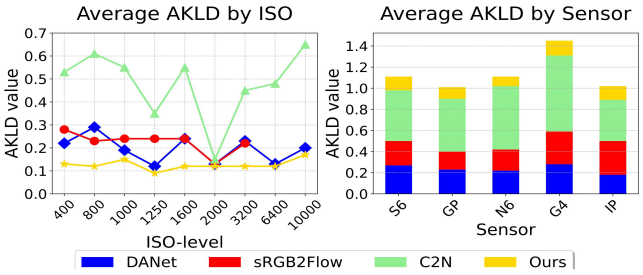

Figure 7: Left: Average AKLD across ISO levels. Right: Average AKLD across sensor types.

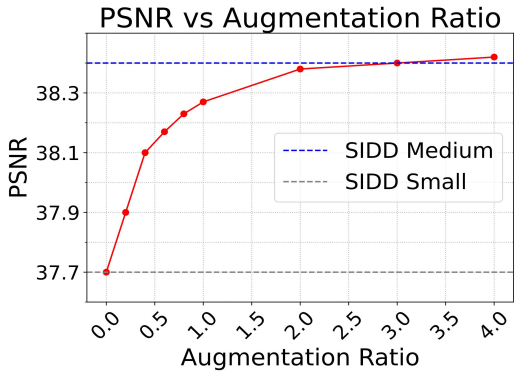

Figure 8: Noise Augmentation results at Different Ratios

Ablation Studies

Performance Evaluation Across Different Camera Settings. In order to evaluate the performance of RNSD across different camera settings, we consider ISO settings and sensor type as examples. Fig. 7 shows that our noise synthesis method outperforms sRGB2Flow, C2N, and DANet in terms of lower AKLD values across all ISO levels and all sensors. This indicates our method's capability to effectively capture significant noise distribution variations and learn a more realistic noise model.

RNSD Performance on Limited Training Data. We assessed RNSD's performance with limited data firstly, training models on 10, 20, and 80 pairs from SIDD Small. RNSD achieved an AKLD close to DANet's (0.212) with 320 pairs when trained with fewer than 20 pairs. We used these models to synthesize additional clean-noisy pairs from SIDD, augmenting training data for DnCNN as detailed in Table 5. Evaluation on the SIDD validation set revealed significant gains, particularly with fewer than 20 pairs, underscoring RNSD's efficacy with minimal data.

Impact of Noise Augmentation Ratios. In Fig. 8, we show that augmenting noise ratios from SIDD Small through RNSD progressively improves denoising accuracy. A 4x increase in training samples yields a 0.72 dB PSNR boost, highlighting RNSD's effectiveness in diversifying noise during training and enhancing model generalizability. Improvements plateau at a 3x augmentation ratio, indicating diminishing returns. Overall, RNSD's noise augmentation significantly improves real-world denoising performance.

	10 pairs	20 pairs	80 pairs	Total (160 pairs)
Baseline/RNSD	35.49/37.08	36.23/37.45	37.34/37.92	37.70/38.27
AKLD	0.312	0.213	0.178	0117

Table 5: Performance vs Training Cost

Method	AKLD↓
Baseline	0.169
+ concat camera settings	0.137
+ TCCAM	0.126
+ MCAM	**0.117**

Table 6: Ablation study of the TCCAM and MCAM.

Steps	3	5	10	30	200
DDIM	0.507	0.356	0.210	0.131	0.117
DIPS-Basic	0.466	0.208	0.124	0.117	0.117
DIPS-Advanced	0.138	0.122	0.121	0.120	0.117

Table 7: Ablation study of DIPS.

RNSD's Modules. RNSD introduces TCCAM and MCAM for realistic noise synthesis. As shown in Table 6, TCCAM encodes camera settings and sampling steps into a condition tensor, reducing AKLD from 0.169 to 0.0.126, enhancing controllability. MCAM, combined with TCCAM, further reduces AKLD to 0.117, validating its role in decoupling content and camera representations. In summary, our ablation studies validate the crucial roles of RNSD's three modules in guiding realistic noise synthesis.

Ablation of Deep Prior Sampling (DIPS). We compared DIPS-Advanced, DIPS-Basic, and DDIM at various step counts (3, 5, 10, 30, 200) in Table 7. The DIPS-Advanced demonstrated remarkable efficiency, achieving results in just 5 steps with only a 4% accuracy loss compared to the full 1000-step DDPM. DIPS-Basic, while also effective, showed a more incremental improvement in accuracy as step counts increased, placing it second in performance. DDIM, known for its deterministic approach to sampling, was tested at reduced step counts but struggled to maintain the accuracy levels of DIPS, especially at the lower end of the step spectrum.This study shows DIPS's superior efficiency and accuracy, outperforming DDIM and making it ideal for real-time applications.

Conclusion

In this paper, we first introduce RNSD, a novel real noise synthesis method based on diffusion models. TCCAM encodes camera settings and sampling steps into time-aware affine transformation parameters, ensuring training stability and controllability while maintaining result diversity. The MCAM module uses dual multi-scale visual features to generate noise with multi-frequency spatial correlations that match real noise. Additionally, our nearly lossless accelerated sampling method, DIPS, is tailored for synthetic noise tasks. Our approach achieves state-of-the-art performance across multiple benchmarks and significantly enhances denoising models' performance and generalization in augmentation experiments.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (NSFC) under Grant Nos. 62372091, 62071097 and in part by Sichuan Science and Technology Program under Grant Nos. 2023NSFSC0462, 2023NSFSC0458, 2023NSFSC1972.

References

Abdelhamed, A.; Brubaker, M. A.; and Brown, M. S. 2019. Noise flow: Noise modeling with conditional normalizing flows. In *International Conference on Computer Vision*, 3165–3173.

Abdelhamed, A.; Lin, S.; and Brown, M. S. 2018. A high-quality denoising dataset for smartphone cameras. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1692–1700.

Bansal, A.; Borgnia, E.; Chu, H.-M.; Li, J. S.; Kazemi, H.; Huang, F.; Goldblum, M.; Geiping, J.; and Goldstein, T. 2022. Cold diffusion: Inverting arbitrary image transforms without noise. *arXiv preprint arXiv:2208.09392*.

Brock, A.; Donahue, J.; and Simonyan, K. 2018. Large scale GAN training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*.

Brooks, T.; Mildenhall, B.; Xue, T.; Chen, J.; Sharlet, D.; and Barron, J. T. 2019. Unprocessing images for learned raw denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 11036–11045.

Cai, Y.; Hu, X.; Wang, H.; Zhang, Y.; Pfister, H.; and Wei, D. 2021. Learning to generate realistic noisy images via pixel-level noise-aware adversarial training. *Conference and Workshop on Neural Information Processing Systems*, 34: 3259–3270.

Chen, S.; Sun, P.; Song, Y.; and Luo, P. 2022. Diffusiondet: Diffusion model for object detection. *arXiv preprint arXiv:2211.09788*.

Foi, A. 2009. Clipped noisy images: Heteroskedastic modeling and practical denoising. *Signal Processing*, 89(12): 2609–2629.

Foi, A.; Trimeche, M.; Katkovnik, V.; and Egiazarian, K. 2008. Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10): 1737–1754.

Fu, Z.; Guo, L.; and Wen, B. 2023. sRGB Real Noise Synthesizing with Neighboring Correlation-Aware Noise Model. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1683–1691.

Guo, S.; Yan, Z.; Zhang, K.; Zuo, W.; and Zhang, L. 2019. Toward convolutional blind denoising of real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1712–1722.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising diffusion probabilistic models. *Conference and Workshop on Neural Information Processing Systems*, 33: 6840–6851.

Jang, G.; Lee, W.; Son, S.; and Lee, K. M. 2021. C2n: Practical generative noise modeling for real-world denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2350–2359.

Karras, T.; Aila, T.; Laine, S.; and Lehtinen, J. 2017. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4401–4410.

Kim, D.-W.; Chung, J. R.; and Jung, S.-W. 2019. GRDN:Grouped Residual Dense Network for Real Image Denoising and GAN-based Real-world Noise Modeling. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*, 0–0.

Kim, Y.; Soh, J. W.; Park, G. Y.; and Cho, N. I. 2020. Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3482–3492.

Kousha, S.; Maleky, A.; Brown, M. S.; and Brubaker, M. A. 2022. Modeling sRGB Camera Noise with Normalizing Flows. In *IEEE Conference on Computer Vision and Pattern Recognition*, 17463–17471.

Li, Y.; Zhang, K.; Liang, J.; Cao, J.; Liu, C.; Gong, R.; Zhang, Y.; Tang, H.; Liu, Y.; Demandolx, D.; Ranjan, R.; Timofte, R.; and Van Gool, L. 2023. LSDIR Dataset: A Large Scale Dataset for Image Restoration.

Lucic, M.; Kurach, K.; Michalski, M.; Gelly, S.; and Bousquet, O. 2018. Are gans created equal? a large-scale study. *Conference and Workshop on Neural Information Processing Systems*, 31.

Mescheder, L.; Geiger, A.; and Nowozin, S. 2018. Which training methods for GANs do actually converge? In *International conference on machine learning*, 3481–3490. PMLR.

Nam, S.; Hwang, Y.; Matsushita, Y.; and Kim, S. J. 2016. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1683–1691.

Plotz, T.; and Roth, S. 2017. Benchmarking denoising algorithms with real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1586–1595.

San-Roman, R.; Nachmani, E.; and Wolf, L. 2021. Noise estimation for generative diffusion models. *arXiv preprint arXiv:2104.02600*.

Shaham, T. R.; Dekel, T.; and Michaeli, T. 2019. Singan: Learning a generative model from a single natural image. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4570–4580.

Song, J.; Meng, C.; and Ermon, S. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*.

Song, M.; Zhang, Y.; Aydın, T. O.; Mansour, E. A.; and Schroers, C. 2023. A Generative Model for Digital Camera Noise Synthesis. *arXiv preprint arXiv:2303.09199*.

Ulyanov, D.; Vedaldi, A.; and Lempitsky, V. 2018. Deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 9446–9454.

Wang, Z.; Cun, X.; Bao, J.; Zhou, W.; Liu, J.; and Li, H. 2022. Uformer: A General U-Shaped Transformer for Image Restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 17683–17693.

Xu, J.; Li, H.; Liang, Z.; Zhang, D.; and Zhang, L. 2018. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*.

Yue, Z.; Zhao, Q.; Zhang, L.; and Meng, D. 2020. Dual Adversarial Network: Toward Real-world Noise Removal and Noise Generation. In *European Conference on Computer Vision*, 41–58.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; and Yang, M.-H. 2022. Restormer: Efficient Transformer for High-Resolution Image Restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5728–5739.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2021. Multi-stage progressive image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 14821–14831.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. In *IEEE Transactions on Image Processing*, 3142–3155.

Zhang, K.; Zuo, W.; and Zhang, L. 2018. FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, 27(9): 4608–4622.

Zhou, Y.; Jiao, J.; Huang, H.; Wang, Y.; Wang, J.; Shi, H.; and Huang, T. 2020. When awgn-based denoiser meets real noises. In *AAAI Conference on Artificial Intelligence*, volume 34, 13074–13081.