GuidNoise: Single-Pair Guided Diffusion for Generalized Noise Synthesis

Changjin Kim[1], **HyeokJun Lee**[2], **YoungJoon Yoo**[1, 2]

[1]SNUAILAB
[2]Dept. of Artificial Intelligence, Chung-Ang University
cjkim@snuailab.ai, {gurwns8926, yjyoo3312}@cau.ac.kr

Abstract

Recent image denoising methods have leveraged generative modeling for real noise synthesis to address the costly acquisition of real-world noisy data. However, these generative models typically require camera metadata and extensive target-specific noisy-clean image pairs, often showing limited generalization between settings. In this paper, to mitigate the prerequisites, we propose a Single-Pair Guided Diffusion for generalized noise synthesis(**GuidNoise**), which uses a **single noisy/clean pair** as the guidance, often easily obtained by itself within a training set. To train GuidNoise, which generates synthetic noisy images from the guidance, we introduce a guidance-aware affine feature modification (GAFM) and a noise-aware refine loss to leverage the inherent potential of diffusion models. This loss function refines the diffusion model's backward process, making the model more adept at generating realistic noise distributions. The GuidNoise synthesizes high-quality noisy images under diverse noise environments without additional metadata during both training and inference. Additionally, GuidNoise enables the efficient generation of noisy-clean image pairs at inference time, making synthetic noise readily applicable for augmenting training data. This self-augmentation significantly improves denoising performance, especially in practical scenarios with lightweight models and limited training data. The code is available at https://github.com/chjinny/GuidNoise.

Introduction

Image denoising is a crucial task in computer vision, enhancing image quality and facilitating higher-level vision tasks. The main challenge lies in effectively removing noise while preserving image details, especially in real-world conditions where noise patterns are influenced by the complexities of modern camera systems such as an image signal processing (ISP) pipeline.

Deep neural networks, particularly convolutional neural networks (CNN), have significantly advanced image denoising (Zhang et al. 2017; Anwar and Barnes 2019). Initially, image-denoising research relied on basic synthetic noise models, primarily trained on typical noise distributions such as Additive White Gaussian Noise (AWGN). These models perform well in controlled environments, but often struggle to generalize to noisy real-world images with complex noise patterns. To address this limitation, datasets consisting of real-world noisy images have been proposed, such as SIDD (Abdelhamed, Lin, and Brown 2018) and PolyU (Xu et al. 2018). However, creating these datasets is costly because it requires an extensive collection of real noisy images and the corresponding acquisition of clean images paired with the target noisy image.

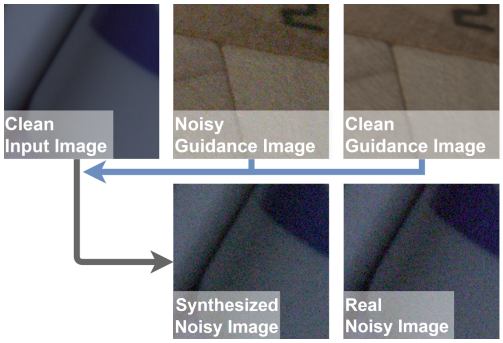


Figure 1: **Illustration of GuidNoise.** Given an input clean image and a noisy-clean guidance image pair, GuidNoise can generate a synthesized noisy image that mimics the real noisy image by capturing the noise distribution of the guidance image. Since the given images can be easily obtained from diverse environments, pseudo-real noisy images can be synthesized arbitrarily.

To reduce the costs associated with preparing noisy image data sets in real-world situations, various generative models have been proposed to synthesize noise by learning from real-world data. Specifically, Generative Adversarial Networks (GAN) (Goodfellow et al. 2020), Normalizing Flow (NF) (Papamakarios et al. 2021), and diffusion (Ho, Jain, and Abbeel 2020a) models have shown promising results (Abdelhamed, Brubaker, and Brown 2019; Yue et al. 2020; Zamir et al. 2020; Jang et al. 2021; Cai et al. 2021a; Kousha et al. 2022; Fu, Guo, and Wen 2023; Kim et al. 2024; Wu et al. 2025). The recent noise synthesis models utilize various techniques using camera metadata or a number of noisy/clean image pairs of the target scene to achieve higher noise modeling performance.

Specifically, a line of approaches (Abdelhamed, Brubaker, and Brown 2019; Kousha et al. 2022; Fu, Guo, and Wen

2023) incorporates camera-specific information to generate noise that closely mimics the characteristics of particular imaging systems, while the others (Yue et al. 2020; Kim et al. 2024) use target noisy images paired with input clean images to capture the intricate patterns of noise distributions. These methods have demonstrated superior performance in generating high-fidelity noise, effectively capturing the complexities of actual camera outputs. However, while these methods perform successfully under the assumption of equivalent environments between training and testing, they often struggle to generalize across diverse conditions, including different devices, ISO settings, shutter speeds, and scene variations. Their reliance on specific metadata or a large number of paired images can limit their adaptation capability to broader datasets or scenarios where such detailed information is unavailable.

To address the limitations, we propose a single-pair guided diffusion model for noise synthesis named **GuidNoise**. As illustrated in Figure 1, GuidNoise trains the diffusion model to generate noise that mimics one of the noise distributions contained in a guidance image pair. At inference time, it requires only a single noisy-clean image pair from the target domain and shows high adaptability across diverse datasets. We note that our method only requires a single guidance pair for the noise synthesis and significantly relaxes the constraint of acquiring clean/noisy image pairs or getting camera metadata. At the inference phase in GuidNoise, a generator can inject a noise distribution contained in a guidance image pair from target domain into every given real clean image, without actual training of the target noise distribution. This flexibility allows us to generate synthetic noisy & clean image pairs by leveraging diverse guidance noisy images where the denoiser fails to remove noise.

The GuidNoise framework is trained by a novel training method, named as guidance-aware affine feature modification (GAFM), to embed fine-grained noise distribution within the diffusion process. In addition, we introduce a noise-aware refine loss that improves the diffusion process, enhancing its capability to generate realistic noise distributions while maintaining adaptation capacity from the guidance pair. Our approach maintains the noisy image quality of previous methods in synthesizing complex noise patterns while overcoming their constraints to enable better generalization across diverse datasets and scenarios. Our contributions are summarized as follows:

- We propose GuidNoise, a single-pair guided diffusion model generating realistic noisy images without relying on camera metadata with using a single guidance noise&clean pair, at an inference phase.
- We propose Guidance-aware Affine Feature Modification (GAFM), an approach to capture fine-grained noise within diffusion models using guidance pairs.
- We introduce a noise-aware refinement loss that enhances the diffusion process and improves its ability to generate realistic noise distributions.
- Our method demonstrates generalized noise synthesis performance across different noisy datasets while maintaining high-quality noise synthesis.

Related Works

Noisy Image Synthesis. The constraints of real-world noisy datasets have driven efforts towards generative noise modeling, aiming to overcome the dependency on real-world datasets by learning to synthesize realistic noise distributions. Generative Adversarial Networks (GAN), Normalizing Flows (NF), and diffusion models have emerged as prominent approaches in this area. Using synthesized noise from the generative model has shown the potential to improve the denoising networks (Cai et al. 2021b).

DANet (Yue et al. 2020) simultaneously generates and removes noise, learning the joint distribution between clean and noisy image pairs from the target scene. CycleISP (Zamir et al. 2020) designs the bidirectional camera imaging pipeline, generating realistic image pairs for RAW and sRGB denoising. C2N (Jang et al. 2021) employs adversarial loss using unpaired noisy-clean images, which allows for training without relying on paired data. However, its modeling from unpaired data often suffers from limitations in noise quality, such as artifacts and color-shift problems. PNGAN (Cai et al. 2021a) tackles the noisy image generation task by breaking it down into image-domain and noise-domain alignment problems using pixel-level modeling. Flow-sRGB (Kousha et al. 2022) extends the normalizing flow-based NoiseFlow (Abdelhamed, Brubaker, and Brown 2019) to the sRGB space, offering improved noise modeling across different image formats. NeCA (Fu, Guo, and Wen 2023) explicitly consists of multiple specialized noise networks based on neighbor-noise correlations. NAFlow (Kim et al. 2024) builds upon this by mapping noise from various camera models into a unified latent space.

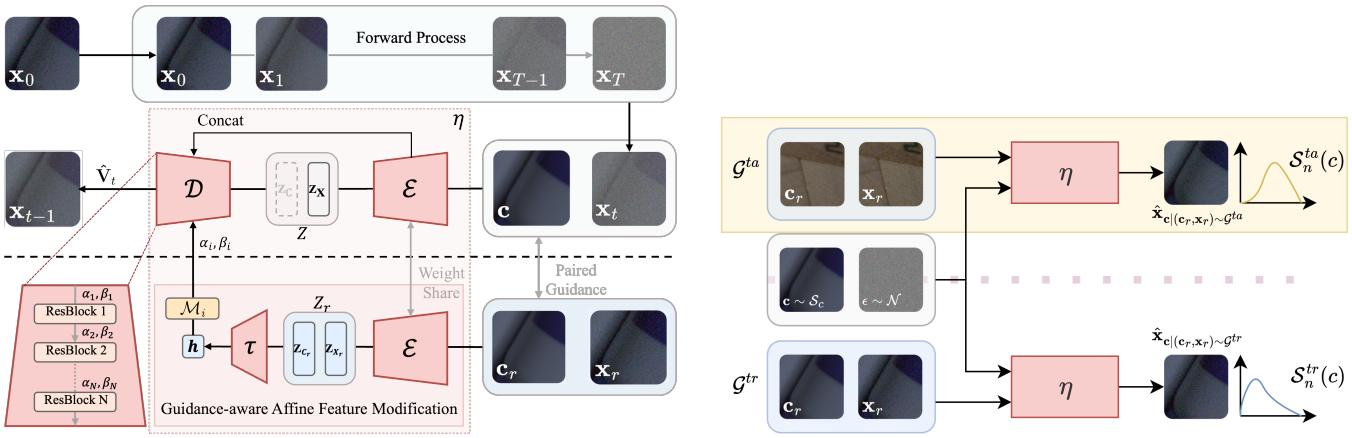
Diffusion models (Ho, Jain, and Abbeel 2020b; Song, Meng, and Ermon 2021; Salimans and Ho 2022) have gained significant interest for their ability to represent complex distributions without the mode collapse observed in GAN, and without the restrictive invertibility requirements of NF models. Unlike NF models, which require invertible transformations and struggle with high-dimensional complex distributions, diffusion models provide the flexibility to better capture intricate noise patterns. RNSD (Wu et al. 2025) is one of the first studies demonstrating the effectiveness of diffusion models in handling the complexities of real-world noise. Additionally, revisiting the dataset preparation pipeline is recently proposed (Li, Jiang, and Iso 2025). GuidNoise generates high-quality noisy images without requiring camera metadata or a paired noisy image corresponding to a target clean image, thereby enhancing generalizability.

Proposed Method

Preliminaries

Noise Modeling. Image noise arises from multiple sources during the imaging process, including the physical limitations of camera sensors and various steps of image processing. Noisy image $\mathbf{x}$ can be defined as a combination of a clean image $\mathbf{c}$ and noise $\mathbf{n}$ from noise space $\mathcal{S}_n$:

$$\mathbf{x} = \mathbf{c} + \mathbf{n}, \mathbf{n} \sim \mathcal{S}_n(\mathbf{c}). \quad (1)$$



(a) Training (b) Inference

Figure 2: **Overview of the proposed method.** The figure shows the training and inference pipelines of *GuidNoise*. The generation model $\eta(\cdot)$ synthesizes noisy images using decoder $\mathcal{D}$, which takes latent feature $\mathbf{Z_x}$ as input. Each residual block processes a concatenation of the previous decoder output and two encoder features, modulated by affine parameters (α_i, β_i).

Diffusion Model. The core idea of diffusion model (Sohl-Dickstein et al. 2015) involves gradually adding noise to an image through a forward diffusion process and then learning a reverse process to reconstruct the original image. Especially, we leverage Denoising Diffusion Probabilistic Models (DDPM) (Ho, Jain, and Abbeel 2020b) for diffusion modeling. In the DDPM, the forward process progressively corrupts a target image with Gaussian diffusion noise, as:

$$\mathbf{x}_t = \sqrt{\alpha_t}\mathbf{x_0} + \sqrt{1-\alpha_t}\epsilon, \quad \text{where } \epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \tag{2}$$

To reconstruct the original image from the noisy counterpart, DDPM directly predicts the noise component added during the forward process with two subsequent improvements. Denoising Diffusion Implicit Models (DDIM) (Song, Meng, and Ermon 2021) and v-prediction (Salimans and Ho 2022). DDIM provides a deterministic sampling,

$$\mathbf{x}_{t-1} = \sqrt{\alpha_{t-1}}\left(\frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\epsilon_\theta(\mathbf{x}_t, t)}{\sqrt{\alpha_t}}\right) + \sqrt{1-\alpha_{t-1}-\sigma_t^2}\epsilon_\theta(\mathbf{x}_t, t) + \sigma_t\epsilon, \tag{3}$$

where α_t and σ_t are time-dependent coefficients. α_t balances the diffusion noise and the target image and σ_t balances between deterministic and stochastic sampling. The v-prediction (Salimans and Ho 2022) estimates the $\mathbf{v}_t$ instead of predicting the diffusion noise ϵ, which improves stability even with fewer sampling steps (Kingma and Gao 2023). This provides a more nuanced combination of the diffusion noise and the target image, allowing better control over the backward process, which defined as:

$$\mathbf{v}_t = \sqrt{\alpha_t}\epsilon - \sqrt{1-\alpha_t}\mathbf{x}_0. \tag{4}$$

The neural network $\mathbf{v}_\theta$ is then trained to predict $\mathbf{v}_t$, following the loss function:

$$\mathcal{L}_{\text{Diffusion}} = \mathbb{E}\left[\|\mathbf{v}_\theta(\mathbf{x}_t, t) - \mathbf{v}_t\|^2\right]. \tag{5}$$

Then $\epsilon_\theta(\mathbf{x}_t, t)$ in Eq. (3) is reparameterized by $\mathbf{v}_\theta(\mathbf{x}_t, t)$ in Eq. (5). This enables a diffusion model to estimate the next step more directly with fewer sampling steps.

Problem Formulation

Noise Synthesis given a Single Guidance Pair. Given clean data space $\mathcal{S}_c$ and its noise space $\mathcal{S}_n$, our goal is to design a diffusion-based noisy image generator $\eta(\cdot)$ as in,

$$\hat{\mathbf{x}} = \eta(\mathbf{c}|(\mathbf{x}_\text{r}, \mathbf{c}_\text{r})), \tag{6}$$

where $\mathbf{c} \in \mathcal{S}_c$ is the clean image to which noise is added, and $(\mathbf{x}_\text{r}, \mathbf{c}_\text{r}) \in \mathcal{G}$ is a guidance noisy-clean image pair, with $\mathcal{G} = \{(\mathbf{x}, \mathbf{c}) \mid \mathbf{x} \sim \mathcal{S}_n(\mathbf{c}),\ \mathbf{c} \in \mathcal{S}_c\}$. Our resultant synthetic noisy image $\hat{\mathbf{x}}$ mimics the real noisy counterpart $\mathbf{x}$ of the input $\mathbf{c}$. Here, we assume that the noise $\mathbf{n}_x = \hat{\mathbf{x}} - \mathbf{c}$ and $\mathbf{n}_r = \mathbf{x}_\text{r} - \mathbf{c}_\text{r}$ both lie in the same noise space $\mathcal{S}_n$. In our proposed synthesize phase, **single-pair guided noise synthesis diffusion model**, we use the generator $\eta(\cdot)$ trained priorly by the training data set, $\mathcal{S}_n^\text{tr}$, having different data and noise domains than our target domains $\mathcal{S}_n^\text{ta}$. However, by using the guidance pair $(\mathbf{x}_\text{r}, \mathbf{c}_\text{r}) \sim \mathcal{G}^\text{ta}$ and given clean image $c \in \mathcal{S}_c^\text{ta}$, our noisy image generator $\eta(\cdot)$ generates noise in target space $\mathcal{S}_n^\text{ta}$. This demonstrates the strong domain adaptation capability of our noise generator. Notably, our method can generate target-domain noise $\mathcal{S}_n^{ta}$ from a clean image $\mathbf{c}$ from any domain, including $\mathcal{S}_c^{tr}$, using just a single guidance pair, further highlighting its robustness across domains.

Single-Pair Guided Noise Synthesis Diffusion

Our noise generation model $\eta(\cdot)$ is based on a conditional U-Net architecture from DDPM. In training phase, $\eta(\cdot)$ takes $\mathbf{x}_t$ from forward process as an input to backward process, along with a clean image $\mathbf{c} \in \mathcal{S}_c^{tr}$ and a guidance pair $(\mathbf{x}_\text{r}, \mathbf{c}_\text{r}) \in \mathcal{G}^\text{tr}$ as additional inputs for guiding the synthetic noisy image generation, as illustrated in Figure 2.

Cascade Decoding Architecture. To implement the architecture, we design a cascade decoding architecture that modifies the original U-Net architecture, forcing each input noise information from the guidance pair to affect different

levels of the generation process. Therefore, all the input information is embedded by the same encoder $\mathcal{E}$ in parallel. The encoder $\mathcal{E}$ consists of multiple encoder blocks $\mathcal{E}_i$, each computing a feature $\mathbf{f}_{\mathbf{x}_t,i}$ for an input image $\mathbf{x}_t$ from the previous feature $\mathbf{f}_{\mathbf{x}_t,i-1}$ and time embedding $\mathbf{t}$ from time step t. The entire encoding process can be summarized as

$$\begin{aligned} &\mathbf{f}_{\mathbf{x}_t,i} = \mathcal{E}_i(\mathbf{f}_{\mathbf{x}_t,i-1}, \mathbf{t}),\ \mathbf{f}_{\mathbf{x}_t,0} = \mathbf{x}_t,\ 1 \le i \le N, \\ &\mathbf{z}_{\mathbf{x}_t} = \mathbf{f}_{\mathbf{x}_t,N} := \mathcal{E}(\mathbf{x}_t, \mathbf{t}),\ \mathbf{F}_{\mathbf{x_t}} = \{\mathbf{f}_{\mathbf{x}_t,1}, \mathbf{f}_{\mathbf{x}_t,2}, \ldots, \mathbf{f}_{\mathbf{x}_t,N}\}. \end{aligned} \tag{7}$$

Here, $\mathbf{z}_{\mathbf{x_t}}$ is the final embedding and $\mathbf{F}_{\mathbf{x_t}}$ is the set of intermediate features.

Guidance-aware Affine Feature Modification. Inspired by the conditional feature modification methods (Perez et al. 2018; Dhariwal and Nichol 2021; Wu et al. 2025), applying the affine transform of the features given the conditional information, we propose a **guidance-aware affine feature modification** method to embed the domain-specific noise information from the guidance pair ($\mathbf{x}_\mathrm{r}$, $\mathbf{c}_\mathrm{r}$). Here, the embedding $\mathbf{z_r}$ from guidance image pair ($\mathbf{x}_\mathrm{r}$, $\mathbf{c}_\mathrm{r}$) is an input into the Noise-aware guidance module τ that processes $\mathbf{z_r}$ to form the guidance embedding $\mathbf{h}$,

$$\mathbf{h} = \tau(\mathbf{z}_\mathrm{r}, \mathbf{t}),\ \mathbf{z}_\mathrm{r} = \mathrm{concat}(\mathcal{E}(\mathbf{x}_\mathrm{r}, t), \mathcal{E}(\mathbf{c}_\mathrm{r}, t)). \tag{8}$$

The concatenated guidance embedding, combined with the time embedding $\mathbf{t}$, serves as guidance input to the decoder $\mathcal{D}$. Note that the decoder $\mathcal{D}$ differs from the encoder $\mathcal{E}$, where the encoder does not use any condition, such as the guidance embedding. By conveying the guidance through $\tau(\cdot)$, we prevent the direct use of raw guidance features in the decoding process. Instead, $\tau(\cdot)$ focuses on capturing the essential characteristics of the noise distribution, which enhances generalization to diverse domains. Specifically, letting $\mathcal{D}_{i,j}$, be the j-th layer in the i-th decoder block and $\mathcal{M}_i(\mathbf{h}, \mathbf{t})$ be a multilayer perceptron (MLP) to convey guidance in t to the i-th decoder block $\mathcal{D}_i$ in the form of an affine coefficient α and β, the entire decoding process is given as:

$$\begin{aligned} &\alpha_i,\ \beta_i = \mathcal{M}_i(\mathbf{h}, \mathbf{t}), \\ &\mathbf{g}_{i,j} = (1+\alpha_i)\mathcal{D}_{i,j}(\mathrm{concat}(\mathbf{g}_{i,j-1}, \mathbf{f}_{\mathbf{x}_t,i}, \mathbf{f}_{\mathbf{c},i})) + \beta_i, \\ &\quad 1 \le j \le L-1, 1 \le i \le N, \\ &\mathbf{g}_{i,L} = \mathcal{D}_{i,L}(\mathbf{g}_{i-1,L}, \mathbf{f}_{\mathbf{x}_t,i-1}, \mathbf{f}_{\mathbf{c},i-1}, \mathbf{h}, \mathbf{t}), \end{aligned} \tag{9}$$

where $\mathbf{g}_{1,0} = \mathbf{z}_{\mathbf{x}_t}$ and $\mathbf{g}_{i\ge 2,0} = \mathbf{g}_{i-1,L}$. Then the final decoder output of the backward process becomes $\mathbf{g}_{N,L}$. Letting the backward process be $\eta(\cdot)$ as in Eq. (5), $\mathbf{g}_{N,L}$ for $\mathbf{x}_t$ becomes $\hat{\mathbf{x}}_t$ that can be expressed by

$$\begin{aligned} \hat{\mathbf{x}}_t &= \eta(\mathbf{x}_t, \mathbf{c}, \mathbf{x}_\mathrm{r}, \mathbf{c}_\mathrm{r}, t) \\ &= \mathcal{D}(\mathbf{z}_{\mathbf{x_t}}, \mathbf{F}_{\mathbf{x_t}}, \mathbf{F}_{\mathbf{c}}, \mathbf{h}, \mathbf{t}). \end{aligned} \tag{10}$$

Finally $\mathbf{x}_{t-1}$ is obtained from Eq. (3) by reparameterizing ϵ_θ to η in Eq. (10). The model is primarily optimized by the diffusion loss in Eq. (5). Note that $\mathbf{v}_\theta(\cdot)$ in Eq. (5) is replaced by $\eta(\cdot)$, i.e., $\mathbf{h}, \mathbf{x}_\mathrm{r}, \mathbf{c}_\mathrm{r}$ are newly explored in our model.

Refine Loss for Preserving Noise Components. As previously mentioned, the standard diffusion loss in Eq. (5) relies on L_2 normalization, which theoretically suppresses high-frequency components—where most noise information resides. To address this issue, we introduce a refinement loss into the backward diffusion process, ensuring distribution alignment between the final synthesized noisy image and the ground truth noisy image.

Specifically, during the sampling process from T to 0, we track the gradients for the last T_{split} steps, where most of the fine-grained noise details are synthesized. The refine loss aims to improve the similarity between the histogram of the synthesized noisy image and the ground truth noisy image. To this end, we use Kullback-Leibler Divergence (KLD) D_{KL} and differentiable histogram (Ustinova and Lempitsky 2016) $\bar{H}$. The refine loss is given by

$$\mathcal{L}_{\text{refine}} = D_{\mathrm{KL}}(\bar{H}(\hat{\mathbf{x}}) \parallel \bar{H}(\mathbf{x})) + \gamma \mathbb{E}\left[\|\hat{\mathbf{x}} - \mathbf{x}\|^2\right], \tag{11}$$

where $\bar{H}(\hat{\mathbf{x}})$ and $\bar{H}(\mathbf{x})$ represent the differentiable histograms of the synthesized noisy image and the ground truth noisy image, respectively. γ is weight parameter of the regularization term. The KL divergence ensures that the distribution of the synthesized noisy image closely matches that of the ground truth by tuning backward process.

By combining the diffusion loss $\mathcal{L}_{\text{Diffusion}}$ in Eq. (5) and the refine loss $\mathcal{L}_{\text{refine}}$ in Eq. (11), the overall loss is given by

$$\mathcal{L} = \mathcal{L}_{\text{Diffusion}} + \lambda \mathcal{L}_{\text{Refine}}, \tag{12}$$

where λ is a weighting factor that controls the contribution of the refine loss.

Inference Phase. During inference, the noisy image generator $\eta(\cdot)$ synthesizes a noisy image $\hat{x}$ for a clean image $\mathbf{c} \in \mathcal{S}_c^{ta}$, guided by a single noisy-clean reference pair $(\mathbf{x}_\mathrm{r}, \mathbf{c}_\mathrm{r}) \sim \mathcal{G}^{ta}$. Unlike the training phase, we no longer require access to any ground-truth noisy image corresponding to $\mathbf{c}$. Instead, the model leverages the noise characteristics extracted from $(\mathbf{x}_\mathrm{r}, \mathbf{c}_\mathrm{r})$ and transfers them to the input image $\mathbf{c}$. By leveraging the guidance-aware affine modulation and refine loss, the model adapts effectively to unseen target domains, generating noise patterns that closely resemble the target domain distribution.

Experiments

Experimental Setup

Evaluation. We compare the similarity of noisy images with several noise synthesis methods: C2N (Jang et al. 2021), Flow-sRGB (Kousha et al. 2022), NeCA-S/NeCA-W (Fu, Guo, and Wen 2023), and NAFlow (Kim et al. 2024). We note that C2N, Flow-sRGB, and NAFlow, as lightweight models, exhibit performance limitations, whereas NeCA includes models with varying parameter sizes, as detailed in the Supplementary Material. The parameter size of the noise synthesis module in our method is of the same order as NeCA. Also, while other synthesis methods have fewer parameters, they demonstrate inferior synthesis quality. To evaluate the quality of synthetic noisy images, we use KLD and AKLD (Yue et al. 2020) as metrics. Furthermore, we train DnCNN (Zhang et al. 2017) from scratch using synthetic datasets to indirectly evaluate the synthetic noisy images. We then train NAFNet (Chen et al. 2022) via a self-augmentation to evaluate enhanced denoising performance. For denoising evaluation, we use PSNR and SSIM metrics.

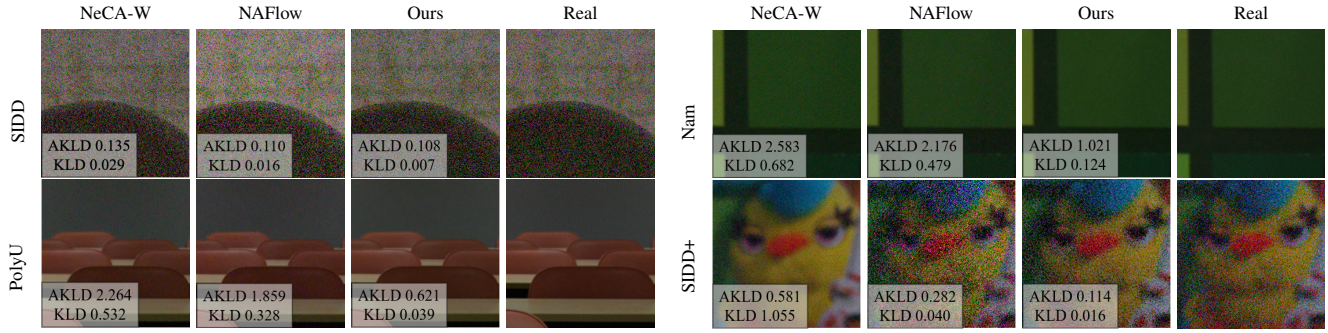


Figure 3: **Qualitative comparison of synthetic noisy imagess** on SIDD-Validation (SIDD), SIDD+, PolyU and Nam dataset.

Camera	Metrics	C2N	Flow-sRGB	NeCA-S	NeCA-W	NAFlow	**Ours**
G4	KLD / AKLD	0.098 / 0.194	0.123 / 0.229	0.403 / 1.680	0.043 / 0.153	0.025 / 0.137	**0.018 / 0.120**
GP	KLD / AKLD	0.693 / 0.428	0.076 / 0.140	0.125 / 0.362	0.043 / 0.126	0.035 / 0.118	**0.010 / 0.111**
IP	KLD / AKLD	0.078 / 0.230	0.077 / 0.216	0.311 / 1.205	0.048 / 0.116	0.034 / 0.152	**0.017 / 0.110**
N6	KLD / AKLD	0.416 / 0.293	0.148 / 0.179	0.282 / 1.058	0.055 / 0.130	0.031 / 0.111	**0.013 / 0.108**
S6	KLD / AKLD	0.683 / 0.418	0.109 / 0.184	0.171 / 0.462	0.052 / 0.194	0.027 / 0.136	**0.013 / 0.117**
Avg	KLD↓ / AKLD↓	0.394 / 0.313	0.112 / 0.193	0.259 / 0.953	0.048 / 0.144	0.031 / 0.131	**0.014 / 0.113**

Table 1: **Quantitative comparison of synthetic noise** on SIDD-Validation dataset.

Method	SIDD+	PolyU	Nam
	KLD↓ / AKLD↓	KLD↓ / AKLD↓	KLD↓ / AKLD↓
C2N	0.192 / 0.302	0.627 / 2.399	0.484 / 2.057
NeCA-S	0.298 / 1.301	1.309 / 3.239	1.161 / 3.144
NeCA-W	0.174 / 0.207	0.139 / 0.795	**0.141** / 0.542
NAFlow	**0.049** / 0.291	0.348 / 1.845	0.456 / 2.040
Ours	0.050 / **0.176**	**0.115 / 0.587**	0.153 / **0.414**

Table 2: **Quantitative comparison of noise similarity** across diverse real-world noisy datasets.

Dataset. We primarily use the real-world noise dataset SIDD (Abdelhamed, Lin, and Brown 2018). For GuidNoise, we use the SIDD-Medium dataset split into 256 × 256 patches. To train DnCNN from scratch, we use subset of SIDD-Medium dataset split into 512 × 512 patches, following the experimental settings in (Fu, Guo, and Wen 2023) and (Kim et al. 2024). We evaluate synthesized noisy images using SIDD-Validation (Abdelhamed, Lin, and Brown 2018). To show our model's generalized performance, we use SIDD+ (Abdelhamed et al. 2020), PolyU (Xu et al. 2018), and Nam (Nam et al. 2016) for image synthesis, and SIDD-Benchmark[1] (Abdelhamed, Lin, and Brown 2018) for image denoising. We crop PolyU and Nam datasets to 512 × 512 pixels. For self-augmentation, we sample the training set from the SIDD-Validation using proportions of 1/16, 1/8, 1/4, and 1/2. with the remaining one as test set.

[1]Our results may slightly differ from prior SIDD benchmarks due to its recent migration to a Kaggle competition.

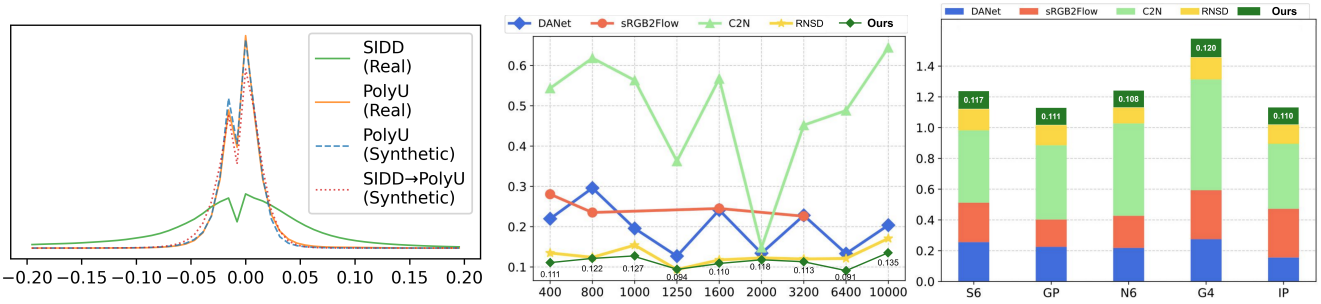
Training. GuidNoise is trained for 300K iterations using the diffusion loss in Eq. (5), with a learning rate of $1e{-}4$ and a batch size of 4. It is then refined for 50K iterations with a learning rate of $1e{-}5$ and a batch size of 1, using both the diffusion loss and the refining loss as defined in Eq. (12). All optimization in GuidNoise is performed using AdamW (Loshchilov and Hutter 2019). We set $\lambda{=}1$, $\gamma{=}0.1$, $T_{\text{split}}{=}2$, and $T{=}50$. To train DnCNN (Zhang et al. 2017) from scratch, we base our approach on the experimental settings in (Fu, Guo, and Wen 2023) and (Kim et al. 2024). We use a learning rate of $1e{-}3$ and run the training for 300 epochs, with a batch size of 32 and random cropping. For optimization, we employ the Adam optimizer (Kingma 2014). To train NAFNet from scratch via self-augmentation, we use a learning rate η_0 of $1e{-}3$ with an exponential scheduler $(\eta_x = \max(0.95^x \cdot \eta_0, 1 \times 10^{-6}))$, which decays the learning rate by 0.95 every 100 iterations initially, then every 1K iterations. We use a batch size of 16 and the PSNR Loss from NAFNet experiments (Chen et al. 2022) for 10K iterations.

Comparison Results

We compare the camera-wise similarity of the synthetic SIDD-Validation datasets with several noise synthesis methods in Table 1. GuidNoise exhibits remarkable AKLD (0.113) score, reduced from NAFlow (0.131) and NeCA-W (0.144). GuidNoise also shows KLD (0.014) score, nearly halving score of NAFlow (0.031) and NeCA-W (0.048). We highlight that C2N is trained on an unpaired dataset. Furthermore, Flow-sRGB results do not include ISO 6400, as the setting is not supported in Flow-sRGB. Therefore, the direct comparisons maybe less suitable.

Dataset	Metrics	C2N	Flow-sRGB	NeCA-S	NeCA-W	NAFlow	**Ours**	Real
SIDD-	PSNR↑	33.72	32.83	35.02	36.20	37.00	**37.07**	37.16
Validation	SSIM↑	0.815	0.861	0.880	0.897	0.895	**0.901**	0.899
SIDD-	PSNR↑	35.05	33.88	35.46	36.61	37.39	**37.48**	37.60
Benchmark	SSIM↑	0.802	0.849	0.872	0.888	0.889	**0.895**	0.890

Table 3: **Quantitative comparison of denoising performance** on SIDD-Validation and SIDD-Benchmark dataset.



(a) Noise distribution from diverse synthesize cases. (b) Average AKLD across ISO. (c) Average AKLD across device.

Figure 4: **Comparison with reported performance results** from RNSD across various camera settings.

Generalizable Noise synthesis. To demonstrate the generalization performance of GuidNoise, we synthesize noisy images using SIDD+, PolyU, and Nam datasets, as shown in Table 2. Note that NeCA-S/W are evaluated using the average score of each device-specific model. Remarkably, despite being trained solely on smartphone-based SIDD dataset, GuidNoise shows significantly improved similarity even on real-world noise datasets captured with DSLR cameras, such as PolyU and Nam. This highlights GuidNoise's generalized performance to capture noise distributions across different camera types without specific training. Specifically, the AKLD reduces from 0.207 (NeCA-W) and 0.291 (NAFlow) to 0.176 on SIDD+, from 0.795 (NeCA-W) to 0.587 on PolyU, and from 0.542 (NeCA-W) to 0.414 on Nam. We present the visualization in Figure 3.

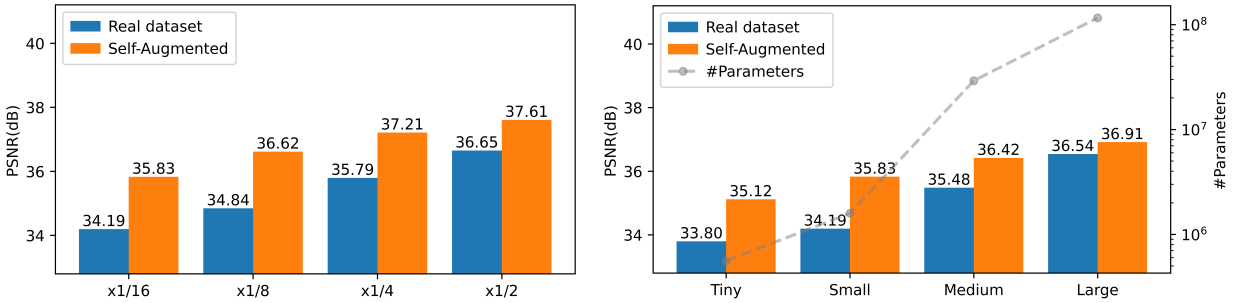
To further investigate the case with a large gap between two noise maps from different settings in Sensor, ISO, scene, etc., we additionally conducted an experimental analysis on **unpaired synthesis scenarios** (i.e., synthetic noise transfer from guide noisy images in the unseen set (PolyU/Nam) to target noisy images in the trained set (SIDD). We measured the noise distributions of the real noises from SIDD training data and synthesized the noises from the PolyU test data in Figure 4a. The results show a large gap between SIDD and real PolyU noises, which arises from a setting gap. To reduce the influence of the setting gap during synthesis, we design the feature-level affine transforms in Eq. (8) (i.e., $(1+\alpha)\mathcal{D}+\beta$) and Figure 2. Consequently, even in the case of a large noise gap in an unpaired synthesis scenario of clean SIDD input and PolyU guide (SIDD→PolyU), we achieve close noise distribution to real PolyU. While our approach uses only paired noisy-clean images (SIDD) for training, our synthesized noisy images in any setting are valid without any assumption on the correlation between input and guide.

In terms of generalizability, RNSD and the proposed GuidNoise exhibit distinct characteristics. While RNSD utilizes clean input along with camera conditions, GuidNoise instead leverages guide image pairs, eliminating the need for explicit camera condition metadata. We added the AKLD results of RNSD to compare our model across different ISO and sensor settings in Figure 4b and 4c, where ours shows consistent performance across ISO levels and sensor types, with better similarity in high ISO cases. Furthermore, our approach shows better performance in PSNR Gap (0.12) than RNSD (0.54), without using camera metadata.

Comparison in Image denoising on synthetic dataset. Furthermore, to indirectly compare the similarity of synthesized noisy images to real noisy images across various noise synthesis methods, we train DnCNN using only the synthetic dataset. The results are shown in Table 3. DnCNN trained with GuidNoise demonstrates remarkable denoising performance on both the SIDD-Validation and SIDD-Benchmark datasets, achieving the highest PSNR and SSIM values among synthetic methods. Specifically, GuidNoise achieves a PSNR of 37.07 and SSIM of 0.901 on SIDD-Validation, as well as a PSNR of 37.48 and SSIM of 0.895 on SIDD-Benchmark. This result closely approximates the performance of real noisy images (Real), where PSNR and SSIM reach 37.16 and 0.899 on SIDD-Validation and 37.60 and 0.890 on SIDD-Benchmark, respectively. We present the visualization of denoised noisy images in the supplementary material. Notably, the denoised images from the denoising model trained with synthesized images from GuidNoise are visually closest to the clean image. This indirectly supports the realistic synthesis performance of GuidNoise.

Discussion

Ablation in Image denoising. As demonstrated above, the components (guidance, refine loss) in GuidNoise contribute to synthesized noise quality, even when using guid-



(a) PSNR comparison across dataset sizes

(b) PSNR comparison across model sizes

Figure 5: **Denoising performance trends by model and dataset size.** Figure 5a presents a performance comparison across dataset sizes for NAFNet-Small. Figure 5b presents a performance comparison across model sizes on the ×1/16 dataset.

Synth. PolyU	Real PolyU	Synth. Nam	Real Nam
34.01 / 0.938	35.71 / 0.956	38.09 / 0.951	41.26 / 0.981

Table 4: **Quantitative comparison** of denoising performance. PSNR/SSIM (↑) were used.

Method	KLD↓	AKLD↓
Baseline $(\mathbf{x}, \mathbf{c})$, $\mathcal{L}_{\text{Diffusion}}$	0.080	0.150
+ Refine loss $\mathcal{L}_{\text{Refine}}$	0.028	0.163
+ **Guidance** $(\mathbf{x}_{\text{r}}, \mathbf{c}_{\text{r}})$	0.050	0.118
+ **Refine loss** $\mathcal{L}_{\text{Refine}}$	**0.014**	**0.113**

Table 5: **Ablation studies.** Evaluating the effectiveness of guidance and refine loss on SIDD-Validation.

ance that differs from the input clean image. This flexibility allow to augment limited real noise datasets by squaring them, which we call **Self-augmentation.** To demonstrate GuidNoise's efficacy in image denoising, we train NAFNet of various sizes on datasets of different scales, with and without self-augmentation. For self-augmented training, each batch contains an equal mix of real and synthetic data.

We evaluate denoising performance using peak PSNR across various size of training datasets ($\times 1/2$, $\times 1/4$, $\times 1/8$, $\times 1/16$ of the original dataset) and model sizes (Tiny, Small, Medium, Large). For each case, we investigate the improvement of denoising performance via self-augmentation from the original dataset. Detailed results are presented with visual trends shown in Figure 5. Self-augmentation improves denoising performance compared to the original real data across various scenarios. It is particularly effective for small models or small datasets in scenarios close to real-world conditions. For NAFNet-small, a self-augmented model trained on 1/8 of real data achieves a PSNR of 36.62dB, nearly equivalent to the 36.65dB PSNR of a model trained on 1/2 of real data. Moreover, with only 1/16 of the real dataset, a self-augmented NAFNet-Small (1.59M parameters) outperforms NAFNet-Medium (29.16M parameters) trained solely on real data, achieving 35.83dB PSNR compared to 35.48dB PSNR, respectively.

Regarding the generalization for denoising, in Table 4, we provide a comparison between real and synthetic settings for PolyU and Nam datasets.

The self-augmentation experiments verify the effectiveness (see Figure 5), where the synthesized noisy images in the unpaired scenarios improve the performance. This demonstrates that self-augmentation via GuidNoise can relax the cost of real noisy datasets or maintain the denoising performance even with few parameters.

Noisy Image Synthesis and Refine Loss. To see the effectiveness of the proposed noise guidance and **refine loss**, we demonstrate the ablation in Table 5. The guidance image reduces KLD from 0.080 to 0.050 and AKLD from 0.150 to 0.118 compared to the baseline which uses only input clean images. The refine loss further reduces KLD from 0.050 to 0.014. These results demonstrate that our cascade decoding architecture and noise-aware guidance significantly improve noise synthesis. Furthermore, a recent study PUCA (Jang et al. 2023) trained in SIDD-Medium achieved an unsupervised performance of PSNR 37.49, slightly higher than our self-augmentation($\times 1/8$) PSNR 37.48. While PUCA uses 320 images, our approach employs a smaller subset ($\times 1/8$) from SIDD-Validation, not SIDD-Medium, so that it consists of only five pairs.

Conclusion

This paper proposes GuidNoise, a single-pair guided diffusion method for enhanced and generalized noise synthesis in image denoising. We mitigate the existing methods' prerequisites by using just a single guidance image pair, easily obtainable from the training set. Our approach introduces a new guidance-aware affine feature modification and noise-aware refine loss that leverages the inherent potential of diffusion models for noisy image generation, allowing the backward process to generate more realistic noise distributions. Additionally, our cascade decoding architecture focuses on the noise distribution of the guidance, thus improving generalization performance across various guidance images. Through these innovations, GuidNoise demonstrates notable noise similarity on various datasets.

Acknowledgements

This work was supported by the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [RS-2021-II211341, Artificial Intelligence Graduate School Program (Chung-Ang University) and RS-2022-II220124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities]. SNUAILAB, corp, also supports this work.

References

Abdelhamed, A.; Afifi, M.; Timofte, R.; and Brown, M. S. 2020. Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 496–497.

Abdelhamed, A.; Brubaker, M. A.; and Brown, M. S. 2019. Noise flow: Noise modeling with conditional normalizing flows. In *Proceedings of the IEEE/CVF 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 (CVPR)*.

Anwar, S.; and Barnes, N. 2019. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, 3155–3164.

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

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

Chen, L.; Chu, X.; Zhang, X.; and Sun, J. 2022. Simple baselines for image restoration. In *European conference on computer vision*, 17–33. Springer.

Dhariwal, P.; and Nichol, A. 2021. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34: 8780–8794.

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

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Ho, J.; Jain, A.; and Abbeel, P. 2020a. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33: 6840–6851.

Ho, J.; Jain, A.; and Abbeel, P. 2020b. Denoising diffusion probabilistic models. *Advances in 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2350–2359.

Jang, H.; Park, J.; Jung, D.; Lew, J.; Bae, H.; and Yoon, S. 2023. PUCA: patch-unshuffle and channel attention for enhanced self-supervised image denoising. *Advances in Neural Information Processing Systems*, 36: 19217–19229.

Kim, D.; Jung, D.; Baik, S.; and Kim, T. H. 2024. sRGB Real Noise Modeling via Noise-Aware Sampling with Normalizing Flows. In *ICLR*.

Kingma, D.; and Gao, R. 2023. Understanding diffusion objectives as the elbo with simple data augmentation. *Advances in Neural Information Processing Systems*, 36: 65484–65516.

Kingma, D. P. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kousha, S.; Maleky, A.; Brown, M. S.; and Brubaker, M. A. 2022. Modeling srgb camera noise with normalizing flows. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 17463–17471.

Li, F.; Jiang, H.; and Iso, D. 2025. Noise Modeling in One Hour: Minimizing Preparation Efforts for Self-supervised Low-Light RAW Image Denoising. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 5699–5708.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1683–1691.

Papamakarios, G.; Nalisnick, E.; Rezende, D. J.; Mohamed, S.; and Lakshminarayanan, B. 2021. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57): 1–64.

Perez, E.; Strub, F.; De Vries, H.; Dumoulin, V.; and Courville, A. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Salimans, T.; and Ho, J. 2022. Progressive Distillation for Fast Sampling of Diffusion Models. In *International Conference on Learning Representations*.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. PMLR.

Song, J.; Meng, C.; and Ermon, S. 2021. Denoising Diffusion Implicit Models. In *International Conference on Learning Representations*.

Ustinova, E.; and Lempitsky, V. 2016. Learning Deep Embeddings with Histogram Loss. In *Neural Information Processing Systems*.

Wu, Q.; Han, M.; Jiang, T.; Jiang, C.; Luo, J.; Jiang, M.; Fan, H.; and Liu, S. 2025. Realistic noise synthesis with diffusion models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 8432–8440.

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 *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part X 16*, 41–58. Springer.

Zamir, S. W.; Arora, A.; Khan, S.; Hayat, M.; Khan, F. S.; Yang, M.-H.; and Shao, L. 2020. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2696–2705.

Zhang, K.; Zuo, W.; Chen, Y.; Meng, D.; and Zhang, L. 2017. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7): 3142–3155.