LEARNING FROM UNPAIRED DATA: A VARIATIONAL BAYES APPROACH

Anonymous authors
Paper under double-blind review

ABSTRACT

Collecting the paired training data is a difficult task in practice, but the unpaired samples broadly exist. Thus, current approaches aim at generating synthesized training data from the unpaired samples by exploring the relationship between the corrupted and clean data. In this work, we propose LUD-VAE, a method to learn the joint probability density function from the data sampled from marginal distributions. Our method is based on the variational inference framework and maximizes the evidence lower bound (ELBO), the lower bound of the joint probability density function. Furthermore, we show that the ELBO is computable without paired samples under the inference invariant assumption. This property provides the mathematical rationale of our approach in the unpaired setting. Finally, we apply our method to the real-world image denoising and super-resolution tasks and train the models using the synthetic data generated by the LUD-VAE. Experimental results on four datasets validate the advantages of our method over other learnable approaches.

1 INTRODUCTION

In recent years, deep learning based models have been an astonishing success in image restoration, which requires large quantities of paired noisy-clean training data. However, in practice, collecting such paired data is difficult or even impossible due to the high complexity of tasks, such as real-world image denoising (Abdelhamed et al., 2018; 2019; Guo et al., 2019) and super-resolution (Lugmayr et al., 2019a; Cai et al., 2019a;b). On the contrary, unpaired data broadly exists and is easily accessible in many situations. For example, it is easy to obtain many images of different resolutions or noisy and clean images through the internet. Consequently, designing deep learning methods with unpaired data is of significant research interest and deserves deep exploration.

Currently, there are two common strategies along this line. One is the unsupervised image restoration methods with a single noisy image (Ulyanov et al., 2018; Quan et al., 2020; Zheng et al., 2021) and noisy image datasets Wu et al. (2020); Laine et al. (2019), which do not consider clean images. As a result, those methods are either time-consuming and or inferior to supervised methods. The other strategy is to learn the degradation model from the unpaired noisy-clean datasets. After learning a generative model to construct the synthetic paired training data, it trains an image restoration model using conventional supervised deep learning algorithms. The main difficulty of these methods is to develop effective methods such that the synthetic paired data is close to the underlying paired data. Mathematically, assume $\mathbf{y} = \mathcal{D}(\mathbf{x}, \mathbf{n})$, where $\mathbf{x}$ represents the clean image, $\mathbf{y}$ represents the noisy image, $\mathbf{n}$ represents the unknown random noise, and $\mathcal{D}$ represents the unknown degradation process, the goal is to learn the stochastic degradation process $\mathcal{D}$ and $\mathbf{n}$ through unpaired samplings of $\mathbf{x}$ and $\mathbf{y}$. Current methods mainly adopt generative adversarial networks (GANs) (Goodfellow et al., 2014) and draw on the idea of cycle-consistency constraint in Cycle-GAN (Bulat et al., 2018; Lugmayr et al., 2019a; Zhu et al., 2017) and domain adversarial objectives (Bell-Kligler et al., 2019; Fritsche et al., 2019; Wei et al., 2021). However, these methods often require careful adjustment of different losses, and the heuristic constraint of cycle consistency is too weak for our problem and lacks theoretical rigorousness. More importantly, those GAN-based methods usually obtain a deterministic mapping while ignoring the randomness in the degradation generation process. Very recently, the DeFlow (Wolf et al., 2021) method, which models the unpaired degradation process using a flow model, has shown promising performance in super-resolution. However, the training process of the DeFlow method encounters unstable problems.

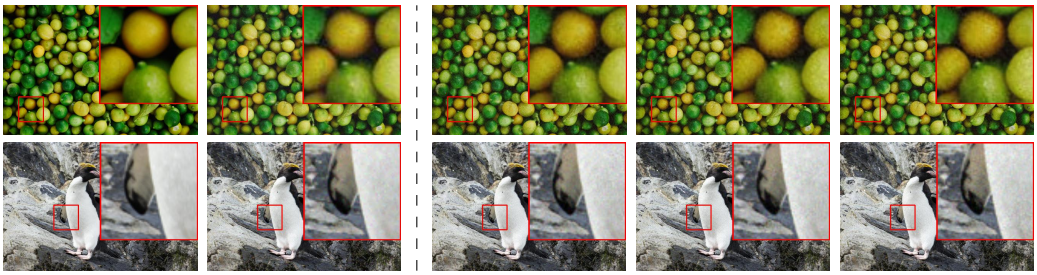

Figure 1: Synthetic noisy images obtained by LUD-VAE learned from unpaired noisy-clean data. The first row is from the AIM2019 (Lugmayr et al., 2019b) dataset, and the second row is from the NTIRE2020 (Lugmayr et al., 2020) dataset.

In this work, we propose LUD-VAE, a variational auto-encoder (VAE) (Kingma & Welling, 2013) based learning method for unpaired data. In concrete, given $\mathbf{x} \sim p(\mathbf{x})$ and $\mathbf{y} \sim p(\mathbf{y})$, our goal is to approximate the joint distribution $p(\mathbf{x}, \mathbf{y})$ with a carefully designed generative graph. This graph consists of two independent latent variables $\mathbf{z}$, which represents the image information, and $\mathbf{z_n}$, which represents the degradation information. The generation relationship is as follows: $\mathbf{x}$ is generated from $\mathbf{z}$, and $\mathbf{y}$ is generated from $\mathbf{z}$, $\mathbf{z_n}$. Using the idea from VAE, we introduce an encoder network for the inference and a decoder network for the generative process such that we can model the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$. In addition, we impose the inference invariant condition on $\mathbf{z}$ that requires the same latent representations of paired $\mathbf{x}$ and $\mathbf{y}$ and prove that the ELBO can be computed via unpaired noisy and clean data. This property gives us a clear and explainable loss for training the networks.

We apply our LUD-VAE model to the problem of real-world image denoising and super-resolution. LUD-VAE is used to learn the degradation model with unpaired data and synthesize paired training data for the downstream supervised models. We demonstrate the model performance of on two real-world super-resolution datasets: AIM2019 (Lugmayr et al., 2019b) and NTIRE2020 (Lugmayr et al., 2020), and one real-world image denoising dataset: SIDD (Abdelhamed et al., 2018). Our model outperforms other GAN-based approaches. Compared with DeFlow, the results of LUD-VAE are very competitive and using much fewer network parameters.

2 Related work

Unpaired degradation modeling. Learning the degradation model from unpaired data can be considered as the image-to-image transfer task, a long-standing problem in computer vision. Most of the existing works employ the GANs (Goodfellow et al., 2014), mainly using cycle-consistency Bulat et al. (2018); Lugmayr et al. (2019a) proposed in Cycle-GAN (Zhu et al., 2017) and domain adversarial (Bell-Kligler et al., 2019; Fritsche et al., 2019; Wei et al., 2021) to characterize the conditional relationship between $p(\mathbf{x})$ and $p(\mathbf{y})$. However, these methods often use heuristic losses, lack theoretical guarantees, and need elaborate fine-tuning of those losses. Meanwhile, GAN-based methods may have issues such as unstable training and model collapse. Our model is based on variational inference, each term in our loss function can be derived from the evidence lower bound, and there are no adversarial losses as in GANs. There are also handcrafted methods to synthesize degradation images (Ji et al., 2020), but lack generalization ability. Recently, Wolf et al. (2021) proposed the DeFlow model, a flow-based degradation modeling method without paired data, which has achieved excellent performance on real-world super-resolution tasks.

Unpaired learning with VAEs. Variational auto-encoder (VAE) (Kingma & Welling, 2013) based unpaired learning for degradation modeling is currently less developed; here, we investigate some related topics. Zhao & Chen (2021) proposed to train an Energy-Based Model (EBM) in the latent space of a trained VAE to realize image-to-image transfer. However, this method uses Markov Chain Monte Carlo (MCMC) algorithms to sample from the latent space; thus, the generation speed is relatively slow. Meanwhile, this method has no theoretical constraints to ensure the rationality of the transformation of $\mathbf{x}$ to $\mathbf{y}$, which is mainly used for the transformation between human faces or

animal images. Zheng et al. (2021) proposed a single image based unsupervised denoising method with VAEs. The problem is that this method requires training a new network for each image, which is time-consuming. Moreover, this algorithm has too many hyperparameters, and it needs to be carefully adjusted for different Noise types. Prakash et al. (2020) proposed a dataset based unsupervised denoising algorithm for microscopy images with VAEs, which needs to know the noise distribution and requires multiple re-samplings to boost the performance. In addition, it only verifies the effectiveness for microscopy images and text images, whereas the texture and content of natural images are more complicated. In this work, we learn the degradation model with unpaired data, design an explainable and reasonable loss with successful applications in real-world image restoration tasks.

3 OUR METHODOLOGY

In this section, we present our method for learning the unknown degradation model using unpaired noisy and clean images. Formally, assume the data $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$ are i.i.d. sampled from $p(\mathbf{x})$ and $p(\mathbf{y})$ respectively, our goal is to generate paired samples from the conditional distribution $p(\mathbf{y} \mid \mathbf{x})$. In the following context, we assume clean images $\mathbf{x}$ lie in the source domain, and noisy images $\mathbf{y}$ lie in the target domain.

3.1 BASIC IDEA

To find the transformation from $\mathbf{x}$ to $\mathbf{y}$, one straightforward idea is to estimate the conditional density $p(\mathbf{y} \mid \mathbf{x})$. However, due to the lack of paired data, it is difficult to model the conditional density function directly. Thus, we consider to model the joint density function $p(\mathbf{x}, \mathbf{y})$ instead. To achieve unpaired learning, our basic idea is to decouple the joint density function into the source domain and target domain. Since the independent assumption for $\mathbf{x}, \mathbf{y}$ can not hold, we impose the conditional independence by assuming the joint random variable $(\mathbf{x}, \mathbf{y})$ has two latent variables: $\mathbf{z}$ and $\mathbf{z_n}$. For a paired data $(\mathbf{x}, \mathbf{y})$ sampled from $p(\mathbf{x}, \mathbf{y})$, we assume the image content comes from latent variable $\mathbf{z}$ and the degradation information comes from $\mathbf{z_n}$. The generating relationship is given in Figure 2. Under the above assumptions, the conditional joint distribution becomes

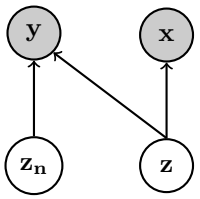

Figure 2: Graphical model of the image generation process. $\mathbf{z}$ represents the content information of the image, and $\mathbf{z_n}$ represents the degradation information.

$$p(\mathbf{x}, \mathbf{y} \mid \mathbf{z}, \mathbf{z_n}) = p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{y} \mid \mathbf{z}, \mathbf{z_n}), \tag{1}$$

and the conditional log-likelihood is

$$\begin{aligned} \sum_i \log p(\mathbf{x}_i, \mathbf{y}_i \mid \mathbf{z}, \mathbf{z_n}) &= \sum_i \log p(\mathbf{x}_i \mid \mathbf{z}) + \log p(\mathbf{y}_i \mid \mathbf{z}, \mathbf{z_n}) \\ &= \sum_i \log p(\mathbf{x}_i \mid \mathbf{z}) + \sum_j \log p(\mathbf{y}_j \mid \mathbf{z}, \mathbf{z_n}), \end{aligned} \tag{2}$$

which removes the dependence of paired data. Inspired by the above observation, we can rigorously obtain the approximated joint distribution in the next subsection.

3.2 PROPOSED LUD-VAE METHOD

To conduct the inference of the graphical model given by Figure 2, we apply the variational inference framework. Since the generation/inference process is too complicated and usually represented by deep neural networks, traditional Coordinate Ascent Variational Inference (CAVI) and Expectation Maximization (EM) (Bishop, 2006) are not applicable, see Appendix A for the details. To avoid the direct update of the inference model $q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})$, we adopt the idea from VAE (Kingma & Welling, 2013) by parameterizing the $q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})$ with an encoder network, which is the proposed LUD-VAE method.

The log-likelihood function $\log p(\mathbf{x}, \mathbf{y})$ has the decomposition:

$$\log p(\mathbf{x}, \mathbf{y}) = \mathbb{E}_{q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})} \log \frac{p(\mathbf{z}, \mathbf{z_n} \mathbf{x}, \mathbf{y})}{q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})} + D_{\mathrm{KL}} \left(q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y}) \| p(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y}) \right), \tag{3}$$

where

$$\mathbb{E}_{q(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y})} \log \frac{p(\mathbf{z},\mathbf{z_n}\mathbf{x},\mathbf{y})}{q(\mathbf{z},\mathbf{z_n} \mid \mathbf{x},\mathbf{y})} = \mathbb{E}_{q(\mathbf{z},\mathbf{z_n}|\mathbf{x},\mathbf{y})} \log p(\mathbf{x},\mathbf{y} \mid \mathbf{z},\mathbf{z_n}) - D_{\mathrm{KL}} q(\mathbf{z},\mathbf{z_n} \mid \mathbf{x},\mathbf{y}) \| p(\mathbf{z_n},\mathbf{z}) \tag{4}$$

is called the evidence lower bound (ELBO). Instead of maximizing the intractable log-likelihood, we maximize the ELBO that is a lower bound of the log-likelihood. Suppose the image information is contained in the pair data $(\mathbf{x},\mathbf{y})$ and the degrading information is only contained by the noisy data $\mathbf{y}$, we choose the inference model has the decomposition

$$q(\mathbf{z_n},\mathbf{z} \mid \mathbf{x},\mathbf{y}) = q(\mathbf{z} \mid \mathbf{x},\mathbf{y}) q(\mathbf{z_n} \mid \mathbf{y}). \tag{5}$$

Moreover, the graphical model in Figure 2 gives

$$p(\mathbf{z_n},\mathbf{z}) = p(\mathbf{z_n}) p(\mathbf{z}), \quad p(\mathbf{x},\mathbf{y} \mid \mathbf{z},\mathbf{z_n}) = p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{y} \mid \mathbf{z},\mathbf{z_n}). \tag{6}$$

Combining Eq. 5 with Eq. 6, the ELBO in Eq. 4 satisfies

$$\begin{aligned} \mathrm{ELBO} = & \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y})} \log p(\mathbf{x} \mid \mathbf{z}) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x},\mathbf{y}) q(\mathbf{z_n}|\mathbf{y})} \log p(\mathbf{y} \mid \mathbf{z_n},\mathbf{z})}_{\text{Reconstruction}} \\ & \underbrace{- D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x},\mathbf{y}) \| p(\mathbf{z})) - D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}) \| p(\mathbf{z_n}))}_{\mathrm{KL}}. \end{aligned} \tag{7}$$

Due to the existence of $q(\mathbf{z} \mid \mathbf{x},\mathbf{y})$ in Eq. 7, it still needs the pair information. Thus, to decouple it, we impose the inference invariant condition on $q(\mathbf{z} \mid \mathbf{x},\mathbf{y})$ as

$$q(\mathbf{z} \mid \mathbf{x},\mathbf{y}) = q(\mathbf{z} \mid \mathbf{x}) = q(\mathbf{z} \mid \mathbf{y}). \tag{8}$$

This assumption shows that for paired data $(\mathbf{x},\mathbf{y})$, the latent image information $\mathbf{z}$ can be obtained from either clean image $\mathbf{x}$ or noisy image $\mathbf{y}$. In practice, this assumption is not strong and can be easily satisfied using a pre-trained network or predefined operations. For instance, the noisy data and the clean data are the same in low frequency counterpart, this assumption can be approximately satisfied by passing the input image through a low-pass filter. More discussion on this assumption is present in Section 3.3. Now, we are ready to propose the objective function of our unpaired learning model in the next proposition and the proof is given in Appendix B.

Proposition 1 *Assume the inference model $q(\mathbf{z} \mid \mathbf{x},\mathbf{y})$ satisfies the inference invariant condition in Eq. 8, then maximize the* ELBO *in Eq. 7 with paired data $(\mathbf{x},\mathbf{y}) \sim p(\mathbf{x},\mathbf{y})$ is equivalent to maximize the following objective function:*

$$\begin{cases} \displaystyle\sum_i \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i \mid \mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}_i) \| p(\mathbf{z})), \\ \displaystyle\sum_j \mathbb{E}_{q(\mathbf{z}|\mathbf{y}_j) q(\mathbf{z_n}|\mathbf{y}_j)} \log p(\mathbf{y}_j \mid \mathbf{z_n},\mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{y}_j) \| p(\mathbf{z})) - D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}_j) \| p(\mathbf{z_n})), \end{cases} \tag{9}$$

where $\{\mathbf{x}_i\}$ and $\{\mathbf{y}_j\}$ are i.i.d. samples from $p(\mathbf{x})$ and $p(\mathbf{y})$ respectively.

Remark 1 *The recent DeFlow model (Wolf et al., 2021) can also estimate the joint density $p(\mathbf{x},\mathbf{y})$ by directly maximizing two marginal log-likelihood functions $\log p(\mathbf{x})$ and $\log p(\mathbf{y})$. Their work assumes that there are two latent variables $\mathbf{z_x}$ and $\mathbf{z_y}$ where $\mathbf{z_x}$ and $\mathbf{z_y}$ are not independent, which is the main difference with our method. In this case, the log-likelihood of the joint density cannot be estimated by maximizing the marginal log-likelihood functions and requires the paired data information. Please see the Appendix C for more details. It is worth mentioning that the DeFlow method introduces a domain invariant function and the conditional flow model. This modified model may help decorrelate the latent representations, but the exact mechanism is not clear.*

3.3 METHOD DETAILS

In this section, we give the detailed settings in our LUD-VAE model.

Generate synthetic paired data. After the training procedure, there are three methods to generate paired data.

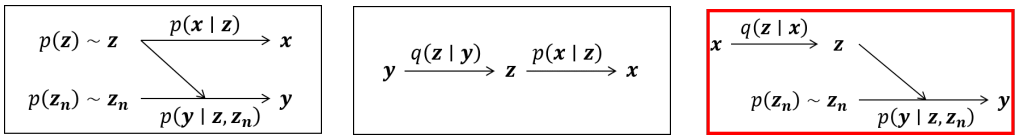

Figure 3: Three methods to generate paired data. Method 3 is used in LUD-VAE.

1. Sample $\mathbf{z_n} \sim p(\mathbf{z_n})$, $\mathbf{z} \sim p(\mathbf{z})$, then generate $(\mathbf{x}, \mathbf{y})$ from $p(\mathbf{y} \mid \mathbf{z}, \mathbf{z_n})$ and $p(\mathbf{x} \mid \mathbf{z})$.
2. Sample $\mathbf{y} \sim p(\mathbf{y})$, inference the latent variable $\mathbf{z}$ with $q(\mathbf{z} \mid \mathbf{y})$, then generate the corresponding $\mathbf{x}$ from $p(\mathbf{x} \mid \mathbf{z})$.
3. Sample $\mathbf{x} \sim p(\mathbf{x})$, inference the latent variable $\mathbf{z}$ with $q(\mathbf{z} \mid \mathbf{x})$, sample $\mathbf{z_n} \sim p(\mathbf{z_n})$, then generate the corresponding $\mathbf{y}$ from $p(\mathbf{y} \mid \mathbf{z}, \mathbf{z_n})$.

See Figure 3 for the graphical explanations. In LUD-VAE, we adopt method 3 to generate paired training data for the downstream tasks since sampling from the prior distribution $p(\mathbf{z})$ may be difficult and obtaining the latent variable $\mathbf{z}$ from the clean image is better than that from degraded data.

Choice of latent space. There are two latent variables $\mathbf{z_n}, \mathbf{z}$ in our model, and the choice of their prior distributions is critical. It is known that in the traditional VAE models usually generate unrealistic and blurry Dosovitskiy & Brox (2016) samples, unless using the hierarchical structure (Vahdat & Kautz, 2020; Child, 2020). To alleviate this problem, inspired by the idea in VQ-VAE (Oord et al., 2017), we choose $p(\mathbf{z})$ as a discrete space, which can facilitate the model's reconstruction of data, and assume $p(\mathbf{z_n})$ as a traditional continuous space $\mathcal{N}(\mathbf{0}, \mathbf{I})$, which is convenient for sampling the latent variables. This choice facilitates the inference method 3 to generate the synthetic paired data as it does not need to sample form $p(\mathbf{z})$. Thus, we only adopt the stage-one in VQ-VAE and do not need to train a PixelCNN (Oord et al., 2016) for sampling from $p(\mathbf{z})$. For the degradation part, we use the latent space assumption of the traditional VAE, which makes it easy to sample the degradation, and we use the diagonal Gaussian distributions for the degradation inference model. The reparameterization trick is used to train the neural networks end-to-end with backpropagation algorithm.

Inference invariant condition. The assumption for the establishment of Proposition 1 is to assume that our inference model satisfies the inference invariant condition in Eq. 8, which means that for paired clean and noisy data $(\mathbf{x}, \mathbf{y}) \sim p(\mathbf{x}, \mathbf{y})$, their latent variable $\mathbf{z}$ is the same. This assumption makes sense since we assume that $\mathbf{z}$ represents the image information, and for paired clean-noisy data, their latent image information should be the same. To satisfy this constraint, we pass the input data through a low-pass filter h. It can be considered that for paired clean-noisy data $\mathbf{x}, \mathbf{y}$, they have the same low frequency parts *i.e.,* $h(\mathbf{x}) = h(\mathbf{y})$, then

$$q(\mathbf{z} \mid h(\mathbf{x}), h(\mathbf{y})) = q(\mathbf{z} \mid h(\mathbf{x})) = q(\mathbf{z} \mid h(\mathbf{y})). \tag{10}$$

We choose $q(\mathbf{z} \mid \mathbf{x}) = q(\mathbf{z} \mid h(\mathbf{x}))$, $q(\mathbf{z} \mid \mathbf{y}) = q(\mathbf{z} \mid h(\mathbf{y}))$, then the inference invariant condition hold from Eq. 10. In practice, we implement h by simple convolutions: $h(\mathbf{x}) = \mathbf{x} * \mathbf{G}_\sigma^s$, where $\mathbf{G}_\sigma^s$ is a Gaussian blur kernel with size s and variance σ^2.

Remark 2 *It is noted that the inference invariant condition is flexible, and $h(\cdot)$ can be considered as a pre-processing operator. If there are paired samples $\{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, we can define $h(\mathbf{x}_i) = \mathbf{x}_i$ and $h(\mathbf{y}_i) = \mathbf{x}_i$, $\forall i = 1, ...N$, the inference invariant condition is automatically satisfied. Thus, our model can handle both paired and unpaired learning tasks.*

KL anneal. When training the VAE models, one commonly encountered problem is the KL-Vanishing (Bowman et al., 2015), which means that the VAE models make the data and the latent variable to be independent. In this case, the inference model $q(\mathbf{z} \mid \mathbf{x})$ equals the prior $p(\mathbf{z})$, the KL divergence equals 0, and data reconstruction does not depend on the latent variable. In our model, there are two latent spaces of different nature. One is a discrete latent space representing image information $p(\mathbf{z})$, and the other is a continuous latent space representing degradation information $p(\mathbf{z_n})$. Since we are using method 3 in Figure 3 to generate paired data, we need to prevent

$q(\mathbf{z_n} \mid \mathbf{y})$ from the KL-Vanishing problem. There are many methods to achieve this (Kingma et al., 2016; Chen et al., 2016; Shen et al., 2018), here we apply the the KL annealing method Bowman et al. (2015). We multiply a weight coefficient α in front of $D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}_i) \| p(\mathbf{z_n}))$ in Eq. 9, and choose α equals to the current number of iterations divide by $5e8$.

Model architecture. We use the hierarchical VQ-VAE (Razavi et al., 2019) with two layers as the main body of our model, which has better reconstruction quality. For the degradation part, we use a traditional VAE with the same encoder/decoder structure as VQ-VAE. The model architecture is illustrated in Figure 4, the degraded image $\mathbf{y}$ is encoded by both noise encoder and image encoder and reconstructed by the combination of noise decoder and image decoder, while the clean image $\mathbf{x}$ is only encoded by the image encoder/decoder. Latent variables $\mathbf{z_n}$ and $\mathbf{z}$ interact in the latent space, indicating that the degradation process may be signal dependent.

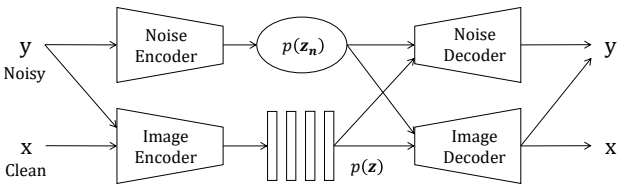

Figure 4: The model architecture used in our paper.

4 Experiments and Results

We evaluate the performance of our LUD-VAE model on real-world image denoising and super-resolution tasks. First, we use LUD-VAE to learn the unknown degradation model under the unpaired learning settings, generate the synthetic training dataset, and then use an off-the-shelf supervised learning algorithm to learn the restoration model. All experiments are evaluated in the sRGB space.

4.1 Datasets and evaluation metrics

Two real-world super-resolution datasets and one real-world denoising dataset are chosen for evaluating our method:

AIM19: Track 2 of the AIM 2019 real-world super-resolution challenge (Lugmayr et al., 2019b) provides a dataset of unpaired noisy-clean images. The noisy images are synthesized with an unknown combination of noise and compression, which mainly manifests as structural and low frequency noises. The task is to learn a super-resolution model from the unpaired dataset, which restores high-resolution clean images from the low-resolution noisy inputs. The challenge also provides a validation set of 100 paired images, where different models can be compared with quantitative metrics. We refer to this dataset as the AIM19 dataset.

NTIRE20: Track 1 of NTIRE 2020 super-resolution challenge (Lugmayr et al., 2020) follows the same setting as the AIM19 dataset, where it features a completely different type of degradation, namely highly correlated high-frequency noise. As AIM19, a validation set contains 100 paired images exists, enabling a quantitative-based evaluation. We refer to this dataset as the NTIRE20 dataset.

SIDD: The smartphone image denoising dataset (SIDD) (Abdelhamed et al., 2018) provides 30,000 noisy images from 10 scenes under different lighting conditions using five representative smartphone cameras and generates their ground truth images. We use the SIDD-Small Dataset and ignored the original index of clean noisy images to set up an unpaired dataset. It also provides the validation and benchmark datasets, each of which is cropped into 32 blocks of size 256×256, resulting in a total of 1024 image blocks in each dataset. We refer to this dataset as the SIDD dataset.

For both three datasets, we report the peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) (Wang et al., 2004). For AIM19 and NTIRE20 datasets, we also compute the LPIPS (Zhang et al., 2018) distance, which is based on the comparison between features of a neural network, here we use a pre-trained AlexNet (Krizhevsky et al., 2012) model.

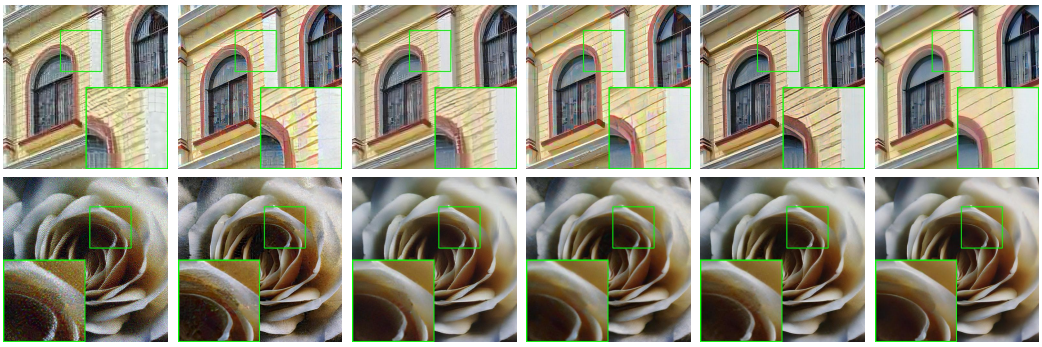

Figure 5: Visual comparison on real-world super-resolution: AIM19 (top) and NTIRE20 (bottom).

Table 1: Quantitative comparison on real-world super-resolution datasets AIM19 and NTIRE20.

	AIM19			NTIRE20			Num of
	PSNR ↑	SSIM ↑	LPIPS ↓	PSNR ↑	SSIM ↑	LPIPS ↓	Parameters
ESRGAN-B	21.69	0.5517	0.517	20.45	0.3241	0.675	-
ESRGAN-FS	20.81	0.5242	0.387	21.07	0.4356	0.414	1.62M
Impressionism	21.99	0.6060	0.420	25.27	0.6731	0.229	-
DASR	21.06	0.5658	0.375	23.70	0.5748	0.328	1.70M
DeFlow-NP	21.06	0.5842	0.346	24.81	0.6777	0.225	62.94M
LUD-VAE	22.13	0.6165	0.377	25.74	0.7135	0.228	10.07M
DeFlow	22.25	0.6214	0.349	25.87	0.7005	0.218	62.94M
CinCGAN	21.60	0.6129	0.461	24.83	0.6752	0.509	53.22M

4.2 Implementation details

We train all LUD-VAE models for 200k iterations with the Adam (Kingma & Ba, 2014) optimizer. The learning rate is fixed to $1e-4$. We use a batch size of 16, containing random crops of size 160×160. Batches are sampled randomly such that images from each domain are drawn with the same possibility. Random flips and rotates are used as data augmentation. For the super-resolution task, we train the LUD-VAE models using the bicubic downsampled images as clean dataset $\{\mathbf{x}_i\}$ and the noisy images as degraded dataset $\{\mathbf{y}_j\}$. After the training process, we use LUD-VAE to transfer the clean dataset to the degraded dataset, forming a paired training set with the high-resolution clean images. For the AIM19 dataset, we normalize the noisy image, such that it has the same channel-wise mean and standard deviation as the clean domain as the DeFlow model (Wolf et al., 2021), and then de-normalize the synthetic noisy image to constitute the training dataset. For the denoising task, we first learn the noise model with unpaired data and then transform the clean image into the corresponding noisy image to obtain the paired training set. For AIM19, we use a Gaussian blur kernel of size $s = 9$ and $\sigma = 4$; for NTIRE20 we use kernel size $s = 9$ and $\sigma = 3$; for SIDD we use kernel size $s = 21$ and $\sigma = 10$ since the noise level in this dataset is larger.

4.3 Real-world super-resolution

We compare LUD-VAE with four unpaired degradation models namely ESRGAN-FS (Fritsche et al., 2019) the winner of the AIM 2019 real-world super-resolution challenge (Lugmayr et al., 2019b), Impressionism (Ji et al., 2020) the winner of the NTIRE 2020 real-world super-resolution challenge (Lugmayr et al., 2020), DASR (Wei et al., 2021) a recently proposed GAN-based method, and DeFlow (Wolf et al., 2021) a flow-based unpaired degradation modeling method. For a fair comparison, we used the settings without the pre-trained network to retrain the DeFlow model, denoted as DeFlow-NP. We use LUD-VAE and these four methods to learn the unknown degradation model, then bicubic downsample the high-resolution images and generate the low-resolution noisy images to obtain the paired training dataset. In addition, we use the bicubic downsampled low-resolution data without degradation as the baseline, denoted as ESRGAN-B. When we have the

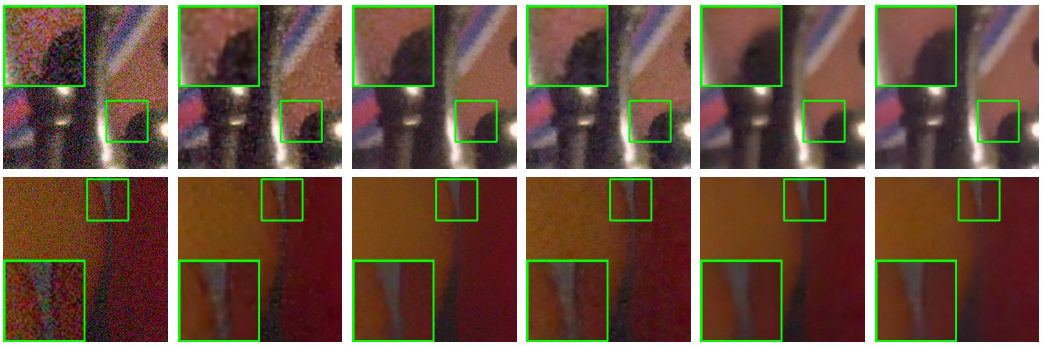

Figure 6: Visual comparison on real-world image denoising dataset SIDD benchmark.

Table 2: Quantitative comparison on real-world image denoising dataset SIDD benchmark (top) and SIDD validation (bottom).

	AWGN	MCWNNM	N2S	DeFlow-NP	LUD-VAE	DeFlow	DnCNN
PSNR ↑	32.12	33.37	29.56	33.54	34.63	33.81	36.54
SSIM ↑	0.868	0.875	0.808	0.875	0.915	0.897	0.927
PSNR ↑	32.06	33.40	30.72	33.53	34.64	33.82	36.83
SSIM ↑	0.809	0.815	0.787	0.817	0.868	0.846	0.870

training dataset, we use the real-world super-resolution model ESRGAN (Wang et al., 2018) to get the final super-resolution results for all methods. We use the training code from Impressionism and train the ESRGAN model for 60k iterations, then choose the final model with the best LPIPS score on the validation dataset every 5k iterations. We also compare with a recently proposed unsupervised super-resolution model CinCGAN (Yuan et al., 2018).

The quantitative results are given in Table 1. For the evaluation metrics, PSNR and SSIM focus on the restoration of the overall content of the image, while LPIPS pays more attention to the image details, so these two types of metrics are mutually exclusive to each other. Most models tend to perform well on only one type of metric, where LUD-VAE performs uniformly well on these three metrics. For the AIM19 dataset, the degradation type mainly behaves as low-frequency and structured noise, while the NTIRE20 dataset is primarily high-frequency noise. We found that LUD-VAE can learn these different types of degradation models without paired data.

Comparison with DeFlow/DeFlow-NP. We find that our model has inferior quantitative performance than the DeFlow model that uses the pre-trained model and better PSNR and SSIM than the DeFlow-NP model that does not use the pre-trained model. This may be because the pre-trained network is trained on paired low-resolution and high-resolution data, which provides strong prior knowledge, and the quantitative metrics are worse on DeFLow-NP without this prior. In addition, the amount of parameters in DeFlow is much larger than our model, which will bring difficulties to practical use. From the visual results in Figure 5, our result removes the noise and is smoother and clearer than the DeFlow-NP model.

Comparison with GAN-based methods. We find that our model is more robust and effective. This may be because GAN-based methods often need to fine-tune different loss functions, and the cycle-consistency constraint is too weak to attain the theoretical guarantee for image restoration tasks. In addition, we find that the performance of LUD-VAE on PSNR and SSIM is better than LPIPS. This may be because we are using the VAE model and the loss function is the Mean Squared Error (MSE) loss, which lacks attention to the image details, making the LPIPS score worse. In the LUD-VAE model, we do not use any heuristic loss function, such as perceptual loss or GAN-style loss, to further improve the quantitative score. The visual results in Figure 5 show that the performance of the GAN-based method is unstable; some images are still noisy, while our approach is more robust.

Table 3: Ablation study of LUD-VAE model. Left: Different generation methods on SIDD benchmark (top) and SIDD validation (bottom) datasets. Right: Validate inference invariant condition on AIM19 dataset.

	Method 2	Method 3
PSNR ↑	33.96	34.63
SSIM ↑	0.902	0.915
PSNR ↑	34.08	34.64
SSIM ↑	0.856	0.868

	PSNR ↑	SSIM ↑	LPIPS ↓
No blur $h(\mathbf{x}) = \mathbf{x}$	21.51	0.5208	0.476
Kernel size $s = 9$, $\sigma = 2$	21.16	0.4631	0.460
Kernel size $s = 9$, $\sigma = 4$	22.13	0.6165	0.377
Kernel size $s = 9$, $\sigma = 6$	21.45	0.5874	0.361

4.4 Real-world image denoising

We compare LUD-VAE with the unsupervised denoising method MCWNNM (Xu et al., 2017), the dataset-based denoising method N2S (Batson & Royer, 2019), the degradation modeling method DeFlow/DeFlow-NP (Wolf et al., 2021), and the fully supervised method DnCNN (Zhang et al., 2017). We also set up a baseline method for Additive White Gaussian Noise degradation, denoted as AWGN. Since the SIDD dataset contains images with different noise levels, we synthetic the degraded images with random noise levels using different degradation methods. For AWGN, we randomly apply Gaussian noise with zero mean and standard deviation $\sigma \in [0.05, 0.5]$ to each image; for DeFLow/DeFlow-NP and LUD-VAE we randomly apply synthetic noise with the noise level parameter $t \in [1, 4]$ to each image. For AWGN, LUD-VAE, and DeFlow/DeFlow-NP, we use the DnCNN (Zhang et al., 2017) for downstream denoising tasks for 50k iterations with an initial learning rate $1e-4$ and halved in 25k iteration. We test each method on the validation set every 500 iterations, and choose the final model with the best PSNR to evaluate on the benchmark set.

The quantitative results are given in Table 2. We find that LUD-VAE has achieved the best results except for the supervised method DnCNN. DeFlow does not perform very well on this dataset; the reason may be the incomplete loss function of the model (see Appendix C) and the instability of the training process. The dataset-based denoising method N2S performed poorly because the noise does not satisfy the spatial uncorrelated assumption. The visual results in Figure 6 show that our results are closer to DnCNN's, with complete noise removal than the other methods.

4.5 Ablation study and discussion

Generation method. We compared different methods in Figure 3 for generating paired data in our model. For method 1, we need to train an additional PixelCNN to sample from the prior distribution $p(\mathbf{z})$, which is inconvenient for practical use; thus, we do not adopt this method. For method 2 and method 3, we evaluate them on the SIDD dataset. The results are shown in Table 3 (left), and we find that method 3 achieves better performance than method 2. One possible reason is that method 2 is a deterministic mapping, while method 3 is stochastic. Since one clean image has many different degraded counterparts, method 3 is more suitable for the downstream denoising model.

Inference invariant. We verify the inference invariant condition on the AIM19 dataset. The quantitative results are in table 3 (right). From the table, we find that if the inference invariant condition, in case "No blur" and Kernel size $s = 9$, $\sigma = 2$, does not hold, the model cannot learn the unknown degradation process, resulting in poor performance. Moreover, we find that increasing the size of the Gaussian blur kernel will result in worse PSNR and SSIM, but better LPIPS. This may be because using a large blur kernel will cause excessive loss of high-frequency information in the image, making the downstream super-resolution model pay more attention to restoring the image's high-frequency content, making LPIPS better PSNR and SSIM becomes worse.

5 Conclusion

In this paper, we propose LUD-VAE, a degradation modeling method using unpaired data based on variational inference. We establish the equivalency between paired and unpaired learning for LUD-VAE is under the inference invariant condition. We use LUD-VAE to generate synthetic training datasets for the downstream supervised learning method and evaluate it on real-world denoising and super-resolution tasks. The experimental results show that our model is lightweight and has very competitive performance.

REFERENCES

Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *CVPR*, pp. 1692–1700, 2018.

Abdelrahman Abdelhamed, Radu Timofte, and Michael S Brown. Ntire 2019 challenge on real image denoising: Methods and results. In *CVPRW*, pp. 0–0, 2019.

Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *ICML*, pp. 524–533. PMLR, 2019.

Sefi Bell-Kligler, Assaf Shocher, and Michal Irani. Blind super-resolution kernel estimation using an internal-gan. *arXiv:1909.06581*, 2019.

Christopher M Bishop. Pattern recognition. *Mach Learn*, 128(9), 2006.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *arXiv:1511.06349*, 2015.

Adrian Bulat, Jing Yang, and Georgios Tzimiropoulos. To learn image super-resolution, use a gan to learn how to do image degradation first. In *ECCV*, pp. 185–200, 2018.

Jianrui Cai, Shuhang Gu, Radu Timofte, and Lei Zhang. Ntire 2019 challenge on real image super-resolution: Methods and results. In *CVPRW*, pp. 0–0, 2019a.

Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *ICCV*, pp. 3086–3095, 2019b.

Xi Chen, Diederik P Kingma, Tim Salimans, Yan Duan, Prafulla Dhariwal, John Schulman, Ilya Sutskever, and Pieter Abbeel. Variational lossy autoencoder. *arXiv:1611.02731*, 2016.

Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv:2011.10650*, 2020.

Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. *NIPS*, 29:658–666, 2016.

Manuel Fritsche, Shuhang Gu, and Radu Timofte. Frequency separation for real-world super-resolution. In *ICCVW*, pp. 3599–3608. IEEE, 2019.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *NIPS*, 27, 2014.

Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *CVPR*, pp. 1712–1722, 2019.

Xiaozhong Ji, Yun Cao, Ying Tai, Chengjie Wang, Jilin Li, and Feiyue Huang. Real-world super-resolution via kernel estimation and noise injection. In *CVPRW*, pp. 466–467, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv:1312.6114*, 2013.

Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. *NIPS*, 29:4743–4751, 2016.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS*, 25:1097–1105, 2012.

Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. *NIPS*, 32:6970–6980, 2019.

Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Unsupervised learning for real-world super-resolution. In *ICCVW*, pp. 3408–3416. IEEE, 2019a.

Andreas Lugmayr, Martin Danelljan, Radu Timofte, Manuel Fritsche, Shuhang Gu, Kuldeep Purohit, Praveen Kandula, Maitreya Suin, AN Rajagoapalan, Nam Hyung Joon, et al. Aim 2019 challenge on real-world image super-resolution: Methods and results. In *ICCVW*, pp. 3575–3583. IEEE, 2019b.

Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Ntire 2020 challenge on real-world image super-resolution: Methods and results. In *CVPRW*, pp. 494–495, 2020.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. Conditional image generation with pixelcnn decoders. *arXiv:1606.05328*, 2016.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. *arXiv:1711.00937*, 2017.

Mangal Prakash, Alexander Krull, and Florian Jug. Fully unsupervised diversity denoising with convolutional variational autoencoders. *arXiv:2006.06072*, 2020.

Yuhui Quan, Mingqin Chen, Tongyao Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. In *CVPR*, pp. 1890–1898, 2020.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In *NIPS*, pp. 14866–14876, 2019.

Xiaoyu Shen, Hui Su, Shuzi Niu, and Vera Demberg. Improving variational encoder-decoders in dialogue generation. In *AAAI*, volume 32, 2018.

Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, pp. 9446–9454, 2018.

Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *arXiv:2007.03898*, 2020.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCVW*, pp. 0–0, 2018.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Trans Image Process*, 13(4):600–612, 2004.

Yunxuan Wei, Shuhang Gu, Yawei Li, Radu Timofte, Longcun Jin, and Hengjie Song. Unsupervised real-world image super resolution via domain-distance aware training. In *CVPR*, pp. 13385–13394, 2021.

Valentin Wolf, Andreas Lugmayr, Martin Danelljan, Luc Van Gool, and Radu Timofte. Deflow: Learning complex image degradations from unpaired data with conditional flows. In *CVPR*, pp. 94–103, 2021.

Xiaohe Wu, Ming Liu, Yue Cao, Dongwei Ren, and Wangmeng Zuo. Unpaired learning of deep image denoising. In *ECCV*, pp. 352–368. Springer, 2020.

Jun Xu, Lei Zhang, David Zhang, and Xiangchu Feng. Multi-channel weighted nuclear norm minimization for real color image denoising. In *ICCV*, pp. 1096–1104, 2017.

Yuan Yuan, Siyuan Liu, Jiawei Zhang, Yongbing Zhang, Chao Dong, and Liang Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *CVPRW*, pp. 701–710, 2018.

Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Trans Image Process*, 26(7):3142–3155, 2017.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pp. 586–595, 2018.

Yang Zhao and Changyou Chen. Unpaired image-to-image translation via latent energy transport. In *CVPR*, pp. 16418–16427, 2021.

Dihan Zheng, Sia Huat Tan, Xiaowen Zhang, Zuoqiang Shi, Kaisheng Ma, and Chenglong Bao. An unsupervised deep learning approach for real-world image denoising. In *ICLR*, 2021.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, pp. 2223–2232, 2017.

A VARIATIONAL INFERENCE FOR PROPOSED MODEL

To inference though the graphical model in Figure 2, traditional methods include Coordinate ascent variational inference (CAVI) and Expectation maximization (EM) algorithms. For CAVI in Algorithm 1, it requires to know the generation process $p(\mathbf{x} \mid \mathbf{z})$, $p(\mathbf{y} \mid \mathbf{z}, \mathbf{z_n})$ and the prior distribution $p(\mathbf{z})$, $p(\mathbf{z_n})$. However, in image generation, we do not know the the two distributions, so the alternate iteration process in Algorithm 1 cannot be realized. For EM in Algorithm 2, in the E step, it needs to update the inference model $q(\mathbf{z})$ and $q(\mathbf{z_n})$ by the true posterior distribution $p(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})$, which is also unavailable in image generation. In this paper, we use the idea of VAEs Kingma & Welling (2013), which avoids updating $q(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})$ directly and needing to be aware of $p(\mathbf{z}, \mathbf{z_n} \mid \mathbf{x}, \mathbf{y})$.

Algorithm 1 Coordinate ascent variational inference

Input: Dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$.
Output: Inference model $q(\mathbf{z})$ and $q(\mathbf{z_n})$.

1: Initialize $q(\mathbf{z})$ and $q(\mathbf{z_n})$.
2: **while** ELBO has not converged **do**
3: $\quad q(\mathbf{z}) \propto \exp\{\mathbb{E}_{q(\mathbf{z_n})} p(\mathbf{z_n}) p(\mathbf{z}) \prod_{i=1}^N p(\mathbf{x}_i \mid \mathbf{z}) \prod_{i=1}^N p(\mathbf{y}_i \mid \mathbf{z}, \mathbf{z_n})\}$
4: $\quad q(\mathbf{z_n}) \propto \exp\{\mathbb{E}_{q(\mathbf{z})} p(\mathbf{z_n}) p(\mathbf{z}) \prod_{i=1}^N p(\mathbf{x}_i \mid \mathbf{z}) \prod_{i=1}^N p(\mathbf{y}_i \mid \mathbf{z}, \mathbf{z_n})\}$
5: **return** $q(\mathbf{z})$ and $q(\mathbf{z_n})$.

B PROOF OF PROPOSITION 1

Suppose $q(\mathbf{z} \mid \mathbf{x}, \mathbf{y})$ satisfies Eq. 8, then for a paired dataset $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$, where $(\mathbf{x}_i, \mathbf{y}_i) \sim p(\mathbf{x}, \mathbf{y})$, we have

$$
\begin{aligned}
\text{ELBO} =& \sum_i^N \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i)} \log p(\mathbf{x}_i \mid \mathbf{z}) + \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i) q(\mathbf{z_n}|\mathbf{y}_i)} \log p(\mathbf{y}_i \mid \mathbf{z_n}, \mathbf{z}) \\
& - D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}_i, \mathbf{y}_i) \| p(\mathbf{z})) - D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}_i) \| p(\mathbf{z_n})) \\
=& \sum_i^N \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i)} \log p(\mathbf{x}_i \mid \mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}_i, \mathbf{y}_i) \| p(\mathbf{z})) \\
& + \sum_i^N \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i,\mathbf{y}_i) q(\mathbf{z_n}|\mathbf{y}_i)} \log p(\mathbf{y}_i \mid \mathbf{z_n}, \mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}_i, \mathbf{y}_i) \| p(\mathbf{z})) \\
& - D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}_i) \| p(\mathbf{z_n})) \\
=& \sum_i \mathbb{E}_{q(\mathbf{z}|\mathbf{x}_i)} \log p(\mathbf{x}_i \mid \mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{x}_i) \| p(\mathbf{z})) \\
& + \sum_j \mathbb{E}_{q(\mathbf{z}|\mathbf{y}_j) q(\mathbf{z_n}|\mathbf{y}_j)} \log p(\mathbf{y}_j \mid \mathbf{z_n}, \mathbf{z}) - \frac{1}{2} D_{\mathrm{KL}}(q(\mathbf{z} \mid \mathbf{y}_j) \| p(\mathbf{z})) \\
& - D_{\mathrm{KL}}(q(\mathbf{z_n} \mid \mathbf{y}_j) \| p(\mathbf{z_n}))
\end{aligned} \tag{11}
$$

So maximizing the ELBO in Eq. 7 with paired data is equivalent to maximize Eq. 9, where we can shuffle the index to achieve unpaired training, thus Proposition 1 holds.

Algorithm 2 Expectation maximization

Input: Dataset $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$.
Output: Inference model $q(\mathbf{z})$, $q(\mathbf{z_n})$, and generating model $p(\mathbf{x}, \mathbf{y} \mid \mathbf{z}, \mathbf{z_n})$.

1: **while** not converged **do**
2: E-step: Let $q(\mathbf{z}) = p(\mathbf{z} \mid D)$ and $q(\mathbf{z_n}) = p(\mathbf{z_n} \mid D)$.
3: M-step: maximize ELBO $= \mathbb{E}_{q(\mathbf{z},\mathbf{z_n})} \log \frac{p(\mathbf{z},\mathbf{z_n}\mathbf{x},\mathbf{y})}{q(\mathbf{z},\mathbf{z_n})}$ w.r.t. $p(\mathbf{z}, \mathbf{z_n}\mathbf{x}, \mathbf{y})$.
4: **return** $q(\mathbf{z})$, $q(\mathbf{z_n})$, and $p(\mathbf{x}, \mathbf{y} \mid \mathbf{z}, \mathbf{z_n})$.

C Discussions on maximal likelihhood

Considering the generative model in Figure 7 that is suggested by the DeFlow method, where $\mathbf{z_x} \sim \mathcal{N}(0, I)$, $\mathbf{z_y} = \mathbf{z_x} + \mathbf{u}$, $\mathbf{u} \sim \mathcal{N}(\mu_u, \Sigma_u)$. In the DeFlow method, it maximizes the log-likelihood function of the two marginal densities:

$$\max_\theta \sum_i \log p(\mathbf{x}_i) + \sum_j \log p(\mathbf{y}_j). \tag{12}$$

where

$$\begin{aligned} \log p(\mathbf{x}) &= \log |\det Df_\theta(\mathbf{x})| + \log \mathcal{N}\left(f_\theta(\mathbf{x}); 0, I\right) \\ \log p(\mathbf{y}) &= \log |\det Df_\theta(\mathbf{y})| + \log \mathcal{N}\left(f_\theta(\mathbf{y}); \mu_u, I + \Sigma_u\right) \end{aligned} \tag{13}$$

by the change of variables formula, where f_θ is an invertible normalizing flow. In the next, we show that the objective function in Eq. 12 is incomplete for representing the log-likelihood $\log p(\mathbf{x}, \mathbf{y})$. Define

$$F(\mathbf{x}, \mathbf{y}) = (f_\theta(\mathbf{x}), f_\theta(\mathbf{y})) = (\mathbf{z_x}, \mathbf{z_y}), \tag{14}$$

then $(\mathbf{z_x}, \mathbf{z_y}) \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \mu_u \end{bmatrix}, \begin{bmatrix} I & I \\ I & I + \Sigma_u \end{bmatrix}\right)$ as $\mathbf{z_y} = \mathbf{z_x} + \mathbf{u}$. Using the change of variables formula, we have

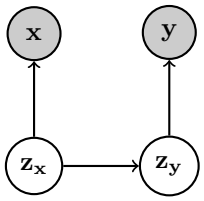

Figure 7: Generative process in DeFlow model.

$$p(\mathbf{x}, \mathbf{y}) = |\det DF(\mathbf{x}, \mathbf{y})| \cdot \mathcal{N}\left(F(\mathbf{x}, \mathbf{y}); \begin{bmatrix} 0 \\ \mu_u \end{bmatrix}, \begin{bmatrix} I & I \\ I & I + \Sigma_u \end{bmatrix}\right). \tag{15}$$

Since

$$F\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \begin{bmatrix} f_\theta(\mathbf{x}) \\ f_\theta(\mathbf{y}) \end{bmatrix} \Rightarrow DF\left(\begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}\right) = \begin{bmatrix} Df_\theta(\mathbf{x}) & 0 \\ 0 & Df_\theta(\mathbf{y}) \end{bmatrix}, \tag{16}$$

then

$$p(\mathbf{x}, \mathbf{y}) = \frac{|\det Df_\theta(\mathbf{x})|\,|\det Df_\theta(\mathbf{y})|}{\sqrt{(2\pi)^{2n} \det \Sigma_u}} \exp\left\{-\frac{1}{2} \begin{bmatrix} f_\theta(\mathbf{x}) \\ f_\theta(\mathbf{y}) - \mu_u \end{bmatrix}^T \begin{bmatrix} I + \Sigma_u^{-1} & -\Sigma_u^{-1} \\ -\Sigma_u^{-1} & \Sigma_u^{-1} \end{bmatrix} \begin{bmatrix} f_\theta(\mathbf{x}) \\ f_\theta(\mathbf{y}) - \mu_u \end{bmatrix}\right\}, \tag{17}$$

where n is the dimension of random variables $\mathbf{z_x}$ and $\mathbf{z_y}$. Then the log-likelihood function $\log p(\mathbf{x}, \mathbf{y})$ can be decomposed into

$$\begin{aligned} \log p(\mathbf{x}, \mathbf{y}) = &\log \mathcal{N}\left(f_\theta(\mathbf{x}); 0, I\right) + \log |\det Df_\theta(\mathbf{x})| \\ &+ \log \mathcal{N}\left(f_\theta(\mathbf{y}); \mu_u, I + \Sigma_u\right) + \log |\det Df_\theta(\mathbf{y})| \\ &+ \log \mathcal{N}\left(f_\theta(\mathbf{x}); -\mu_u, \Sigma_u\right) + f_\theta^T(\mathbf{x}) \Sigma_u^{-1} f_\theta(\mathbf{y}) \\ &+ \frac{1}{2} \log\left((2\pi)^n \det(\Sigma_u)\right) - \frac{1}{2} \mu_u^T \Sigma_u^{-1} \mu_u. \end{aligned} \tag{18}$$

Using marginal distribution notation, the maximal likelihood estimation is

$$\begin{aligned} \max_\theta \sum_i \log p(\mathbf{x}_i, \mathbf{y}_i) = \max_\theta &\sum_i \log p(\mathbf{x}_i) + \sum_j \log p(\mathbf{y}_j) \\ &+ \sum_i \log \mathcal{N}\left(f_\theta(\mathbf{x}_i); -\mu_u, \Sigma_u\right) + f_\theta^T(\mathbf{x}_i) \Sigma_u^{-1} f_\theta(\mathbf{y}_i) \\ &+ \frac{1}{2} \log\left((2\pi)^n \det(\Sigma_u)\right) - \frac{1}{2} \mu_u^T \Sigma_u^{-1} \mu_u. \end{aligned} \tag{19}$$

The additional term $\sum_i f_\theta^T(\mathbf{x}_i)\Sigma_u^{-1} f_\theta(\mathbf{y}_i)$ in Eq. 19 requires the paired information. But, in the DeFlow model, it further introduces conditional marginal likelihood and its relationship with conditional likelihood is still unknown and deserves further exploration. Inspired by the above derivations, we propose our generative graph in which the two latent variables are independent that is relatively easy for constructing an approximation of the log-likelihood.

D MORE VISUAL RESULTS

We show more visual results of real-world super-resolution and denoising. See Figure 8 in page 15 for results from AIM19 dataset; see Figure 9 in page 16 for results from NTIRE20 dataset; see Figure 10 in page 17 for results from SIDD dataset.

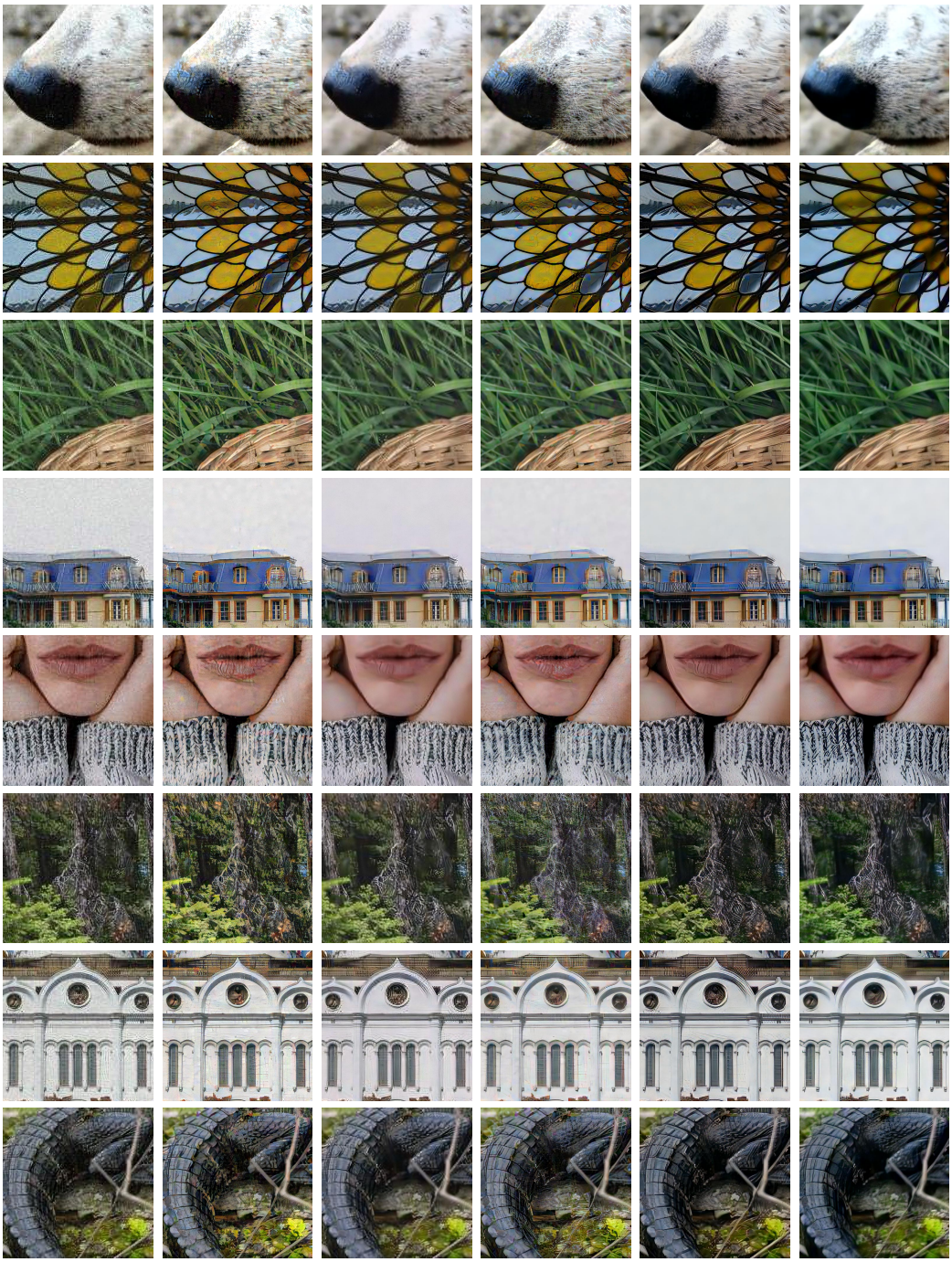

Figure 8: Visual results of real-world super-resolution from AIM19 dataset.

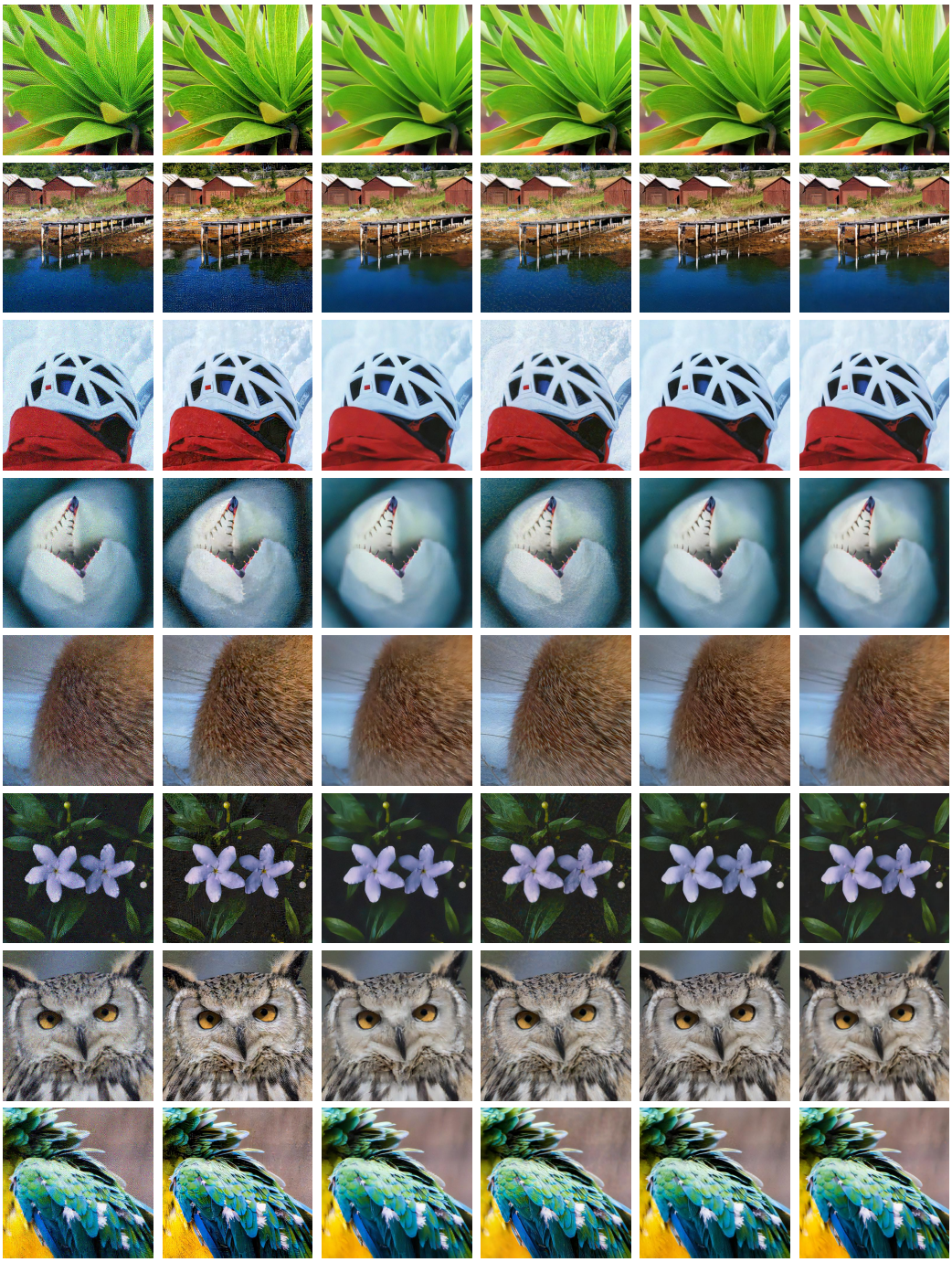

Figure 9: Visual results of real-world super-resolution from NTIRE20 dataset.

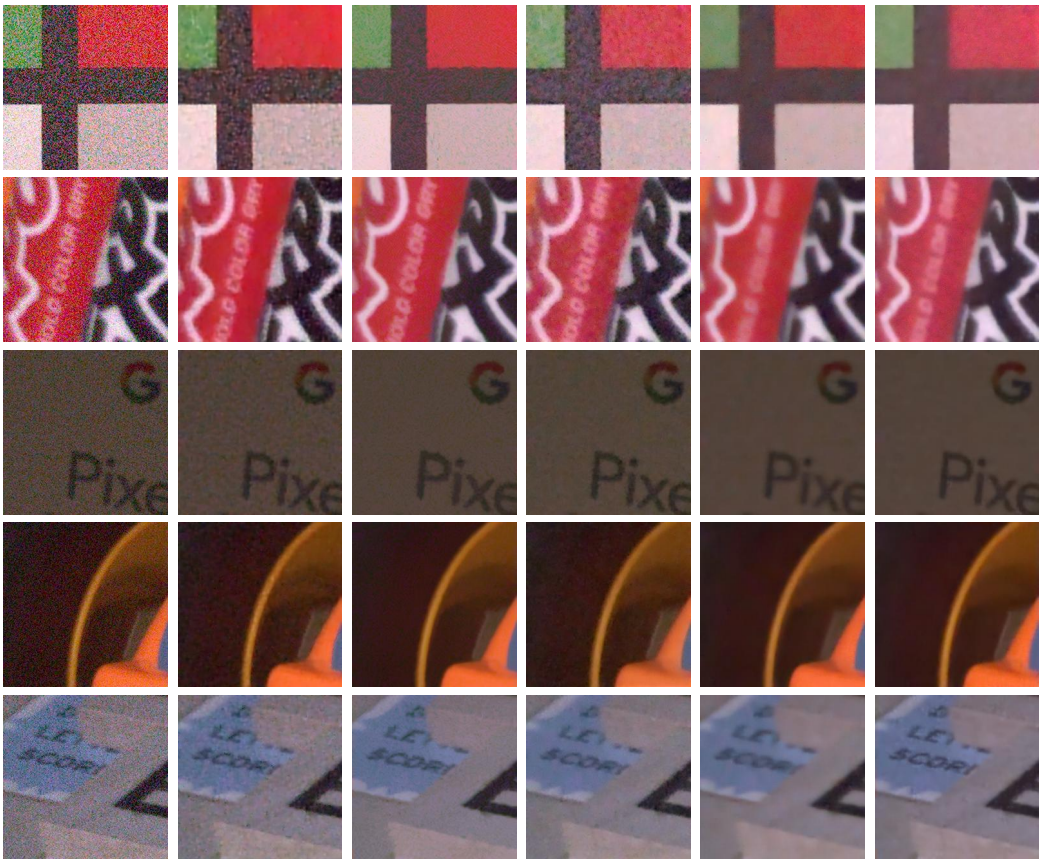

Figure 10: Visual results of real-world image denoising from SIDD dataset.