# Multi-view Self-supervised Disentanglement for General Image Denoising

Hao Chen[*,1] Chenyuan Qu[*,1] Yu Zhang[2] Chen Chen[3] Jianbo Jiao[1]

[1]University of Birmingham [2]Shanghai Jiao Tong University [3]University of Central Florida

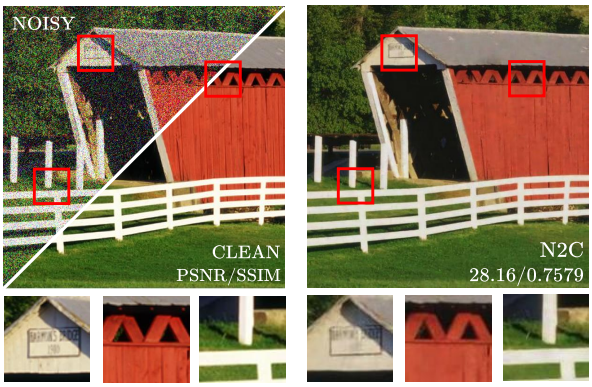
Project page: https://chqwer2.github.io/MeD/

## Abstract

*With its significant performance improvements, the deep learning paradigm has become a standard tool for modern image denoisers. While promising performance has been shown on seen noise distributions, existing approaches often suffer from generalisation to unseen noise types or general and real noise. It is understandable as the model is designed to learn paired mapping (*e.g. *from a noisy image to its clean version). In this paper, we instead propose to learn to disentangle the noisy image, under the intuitive assumption that different corrupted versions of the same clean image share a common latent space. A self-supervised learning framework is proposed to achieve the goal, without looking at the latent clean image. By taking two different corrupted versions of the same image as input, the proposed **M**ulti-view **S**elf-supervised **D**isentanglement (MeD) approach learns to disentangle the latent clean features from the corruptions and recover the clean image consequently. Extensive experimental analysis on both synthetic and real noise shows the superiority of the proposed method over prior self-supervised approaches, especially on unseen novel noise types. On real noise, the proposed method even outperforms its supervised counterparts by over **3 dB**.*

## 1. Introduction

Image restoration is a critical sub-field of computer vision, exploring the reconstruction of image signals from corrupted observations. Examples of such ill-posed low-level image restoration problems include image denoising [16, 25, 26, 29, 33, 35, 38], super-resolution [2, 8, 19, 30, 37], and JPEG artefact removal [7, 12, 31], to name a few. Usually, a mapping function dedicated to the training data distribution is learned between the corrupted and clean images to address the problem. While many image restoration systems perform well when evaluated over the same corruption distribution that they have seen, they are often required to be deployed in settings where the environment is unknown and off the training distribution. These settings, such as medical imaging, computational lithography, and remote sensing, require image restoration methods that can handle complex and unknown corruptions. Moreover, in many real-world image-denoising tasks, ground truth images are unavailable, introducing additional challenges.




(a) Noisy/ Clean (b) N2C [21], *Supervised*

(c) LIR [9], *Self-supervised* (d) Ours, *Self-supervised*

Figure 1. **Denoising performance on unseen Speckle Noise with $\hat{v} = 50$.** The models were trained with Gaussian noise $\sigma \in [5, 50]$. (a) The noisy and clean images, with ground-truth clean patches shown below. (b) *Noise-to-Clean (N2C)* [21] is trained with clean images. (c) *LIR* [9] is self-supervised but needs unpaired clean images as training data. (d) Our approach is fully self-supervised, training with only the noisy input data.

[*]Equal contribution.

**Limitations of existing methods:** Current low-level corruption removal tasks aim to address the inquiry of "what is the clean image provided a corrupted observation?" However, the ill-posed nature of this problem formulation poses a significant challenge in obtaining a unique resolution [5].

To mitigate this limitation, researchers often introduce additional information, either explicitly or implicitly. For example, in [15], *Laine et al.* explicitly use the prior knowledge of noise as complementary input, generating a new invertible image model. Alternatively, Learning Invariant Representation (LIR) [9] implicitly enforces the interpretability in the feature space to guarantee the admissibility of the output. However, these additional forms of information may not always be practical in real-world scenarios or may not result in satisfactory performance.

**Main idea and problem formulation:** Our motivation for tackling this ill-posed nature stems from the solution in the 3D reconstruction of utilising multiple views to provide a unique estimation of the real scene [1]. Building on this motivation, we propose a training scheme that is explicitly built on multi-corrupted views and perform **M**ulti-view s**e**lf-supervised **D**isentanglement, abbreviated as *MeD*.

Under this new multi-view setting, we reformulate the task problem as **"what is the shared latent information across these views?"** instead of the conventional "what is the clean image?" By doing so, MeD can effectively leverage the scene coherence of multi-view data and capture underlying common parts without requiring access to the clean image. This makes it more practical and scalable in real-world scenarios. An example of the proposed method with comparison to prior works is shown in Figure 1, indicating its effectiveness over the state-of-the-art.

Specifically, given any scene image $x^k$ $X; k\ 2$ N sampled uniformly from a clean image set $X$, MeD produces two contaminated views:

$$y_1^k \ , \ T_1(x^k); \ y_2^k \ , \ T_2(x^k); \tag{1}$$

forming two independent corrupted image sets $fY_1g$, $fY_2g$, where $y_1^k \ 2 \ Y_1; \ y_2^k \ 2 \ Y_2$. The $T_1$ and $T_2$ represent two random independent image degradation operations.

We parameterise our scene feature encoder $G^X$ and decoder $D^X$ with and . Considering the image pair $fy_1^k; \ y_2^k g_{k2\mathrm{N}}$, the core of the presented method can be summarised as:

$$G^X(y_1^k) \ , \ z_x^{k;i} \ , \ G^X(y_2^k); \tag{2}$$

$$\hat{x}^k \ , \ D^X(z_x^{k;i}); \tag{3}$$

where $z_x^{k;i}$ represents the shared scene latent between $y_1^k$ and $y_2^k$ with $i$ referring to the input image index of $y_i$. A clean image estimator $D^X$ forms an all-deterministic reverse mapping from $z_x^{k;i}$ to reconstruct an estimated clean image $\hat{x}^k$. Similarly, the noise latent $u^{k;i}$ is factorised from a corrupted view with a corruption encoder $E^N$. Afterwords, the resulting corruption is reconstructed from $u^{k;i}$ through the use of a corruption decoder, represented by $F^N$.

The *disentanglement* is then performed between $fz_x^{k;i}; \ u^{k;j} g_{i \neq j}$ on a cross compose decoder $R^Y$ with parameter , which can be formulated as:

$$\hat{y}_1^k \ , \ R^Y(z_x^{k;2}; u^{k;1}): \tag{4}$$

It should be noted that Equation (4) is performed over latent features $u$ and $z$ from different views. When assuming that $z_x^k$ remains constant across views, the reconstructed view $\hat{y}_1^k$ is determined by the $u^{k;1}$.

**Contributions.** The contributions of our work are summarised as follows:

- We propose a new problem formulation to address the ill-posed problem of image denoising using only noisy examples, in a different paradigm than prior works.
- We introduce a disentangled representation learning framework that leverages multiple corrupted views to learn the shared scene latent, by exploiting the coherence across views of the same scene and separating noise and scene in the latent space.
- Extensive experimental analysis validates the effectiveness of the proposed *MeD*, outperforming existing methods with more robust performance to unknown noise distributions, even better than its supervised counterparts.

## 2. Related Work

**Single-view image restoration:** In [8], *Dong et al.* were the first to employ a deep network in super-resolution. Later, a range of single view-based models expanded the idea of supervised deep learning to handle image restoration tasks, such as deblurring [14], JPEG artefacts [12], inpainting [17, 34] and denoising [16, 26, 35]. Recently, it is receiving increasing interest in relaxing the prerequisite of supervised learning with corrupted/ clean image pairs. In the context of image denoising, the "*corrupted/clean*" pair denotes a corrupted input image and its corresponding clean image for calculating the loss. To tackle the issue of the lack of clean data, several methods have been proposed, such as the Noise2Noise (N2N) method [16] and Recorrupted-to-Recorrupted (R2R) [26], which train deep networks on pairs of noisy images. Noise2Void (N2V) [13], Noise2Self (N2S) [4], and the method proposed by *Laine et al.* [15] are based on the blind-spot strategy that discards some pixels in the input and predicts them using the remaining. In the field

Figure 2. Method Overview. This figure illustrates the main steps of our proposed method, MeD, which first generates scene features (cubes) and distortion features (cuboids). The colour of them indicates their image source. In the right section, the features are rearranged and utilised for the four forward paths, from top to bottom, which are the reconstructions of noisy input ($\hat{y}_1^k$), scene ($\hat{x}_2^k$), input image ($y_2^k$) and shared scene ($\hat{x}^k$). It is noteworthy that $y_2^k$ is reconstructed using $z_x^{k;1}$ from $y_1^k$ and $u^{k;2}$ from $y_2^k$ for feature disentanglement. Additionally, the reconstruction of $\hat{x}^k$ relies on mixed scene features to facilitate learning of invariant scene latent. Moreover, the reconstruction paths for ($\hat{y}_2$, $\hat{x}_1$, and $y_1$) are not depicted here, as they differ from the given paths only in their sub-indexes.

of single-image denoising, some methods such as DIP [29] and S2S [27], have achieved remarkable denoising results using only one noisy image.

These methods, however, often inevitably compromise image quality to noise reduction, resulting in over-smoothed output. This trade-off is further exacerbated under domain shifts when dealing with unknown noise distributions.

Restoration based on multiple views: Existing multi-view variants of image restoration methods mainly focus on sequential data such as video or burst images. For example, Tico [28] builds a paradigm that separates the unique and common features within the multi-frame to produce a denoised estimate. Liu et al. [20] models degrading elements as foreground and estimate background using video data. Deep Burst Denoising (DBD) [11] performs multi-view denoising based on burst images. Each image is taken in a short exposure time and serves as a corrupted observation of the clean image.

Unlike the above-mentioned methods, our MeD aims to use multiple static observations simultaneously to learn the latent representation of a clean scene that is shared by multiple discrete views.

Self-supervised feature disentanglement: Another notable path of work has attempted to disentangle underlying invariant content from distorted images. For instance, UDE-GAN [22] utilises unpaired clean/blurred images to disentangle content and blurring effects in the feature space and yield improved deblurring performance. Similarly, LUR [9] used unpaired input to isolate invariant content features through self-supervised inter-domain transformation.

However, these methods are limited to synthetic noise and do not extend well in real-world scenarios due to their reliance on clean images. In contrast, our method is purely based on multiple noisy views of the same static scene and aims to disentangle the scene from corruption, without the need for clean image supervision.

## 3. Methodology

Our primary objective is to identify the commonalities among different views in the denoising process. To achieve this, we aim to discover the shared scene $z_x^k$ that is degradation-agnostic over various corrupted views $\{y_j^k\}_{j=1}^N$ via our proposed training schema, namely Multi-view self-supervised Disentanglement (MeD). A graphic depiction of MeD is shown in Figure 2, composed of the representation learning process in the left panel, and four distinct reconstruction pathways in the right panel.

The detailed design of the proposed schema will be introduced in the following subsections. Section 3.1 explains the restoration of noise and scene. Section 3.2 details the reconstruction of noisy input using a cross-feature combination. Section 3.3 elaborates on the reconstruction of the

scene using mixed scene latent.

We will start our introduction by outlining three essential properties that a multi-view representation disentanglement technique should exhibit.

Pre-assumed properties: Suppose the scene latent space and corruption latent space are symbolised by $Z_x$ and $U$, respectively.

(1) Independence: For any scene latent $z_x^k \in Z_x$, it is expected to be independent of any corruption latent $u^{k;i} \in U$.

(2) Consistency: There exists one shared latent code in $Z_x$ that is capable of representing the shared clean component of all instances in the set $\{y_i^k\}$.

(3) Composability: Recovery of the corrupted view $y_i^k$ can be achieved using the feature pairs $z_x^k$, $u^{k;i}$, and the index of the recovered view is determined by the index of the corruption latent, which represents the unique component within that particular view.

A key step of our method is to realise these pre-requisitions by determining how to implement the latent space assumption. As shown in the left panel of Figure 2, to infer our latent space assumption, MeD is comprised of two encoders and three decoders: A shared content latent encoder $G^X$ and its decoder $D^X$, an auxiliary noise latent encoder $E^N$ and its decoder $F^N$, and a cross disentanglement decoder $R^Y$.

### 3.1. Main Forward Process

Given two corrupted views of the same image $y_1^k = T_1(x^k)$ and $y_2^k = T_2(x^k)$, the encoder $G^X$ mainly perform the scene feature space encoding that can be formulated as:

$$z_x^{k;1} = G^X(y_1^k);\ z_x^{k;2} = G^X(y_2^k); \tag{5}$$

where $z_x^{k;1}$ and $z_x^{k;2}$ are the estimation of the scene feature corresponding to the inputs $y_1^k$ and $y_2^k$.

The process of clean image reconstruction is then completed by the $D^X$:

$$\hat{x}_1^k = D^X(z_x^{k;1});\ \hat{x}_2^k = D^X(z_x^{k;2}). \tag{6}$$

Similar to the process of estimating scene features, the estimation of distortion features by $E^N$, followed by the reconstruction of noise with $F^N$, can be described as follows:

$$\begin{aligned} u_n^{k;1} &= E^N(y_1^k);\ u_n^{k;2} = E^N(y_2^k); \\ \hat{n}_1^k &= F^N(u_n^{k;1});\ \hat{n}_2^k = F^N(u_n^{k;2}). \end{aligned} \tag{7}$$

We adhere to the methodology introduced by N2N [16] to use noisy images as supervisory signals. The objective function of the aforementioned process can be simplified to:

$$\begin{aligned} &\underset{;}{\operatorname{argmin}}\, L^X,\ \|\hat{x}_1^k - y_2^k\|; \\ &\underset{;}{\operatorname{argmin}}\, L^N,\ \|(y_1^k - \hat{n}_1^k) - y_2^k\|. \end{aligned} \tag{8}$$

The objective of $\hat{x}_2^k$ and $\hat{n}_2^k$ are the same as above, with only a subscript difference. It should be noted that, although our objective functions are similar to that of N2N, our goal is not simply to find and remove noise, but rather to capture the common features shared across multiple views.

### 3.2. Cross Disentanglement

For general latent codes $z_x^k$ to sufficiently represent scene information in the image space, it is natural to assume that these codes exhibit a certain degree of freedom, allowing them to intersect with the noise space. Consequently, there is no guarantee of complete isolation between $z_x^k$ and $u^k$. To meet the requirements of properties (1) and (3), we use an additional decoder $R^Y$ to reconstruct a corrupted view over a cross-feature combination, e.g. $z_x^{k;1}$ from $y_1$ and $u_n^{k;2}$ from $x_2$, which can be represented as:

$$\hat{y}_1^k = R^Y(z_x^{k;2}; u_n^{k;1});\ \hat{y}_2^k = R^Y(z_x^{k;1}; u_n^{k;2}). \tag{9}$$

This realisation explicitly requires $z_x^{k;i}$ to represent the common part and $u_n^{k;j}$ to represent the unique part within the corrupted views. We then optimise $\{\ ;\ ;\ \}$ from $\{G^X; E^N; R^Y\}$ using the following objective:

$$\underset{;;}{\operatorname{argmin}}\, L^C,\ \|\hat{y}_1^k - y_1^k\| + \|\hat{y}_2^k - y_2^k\|. \tag{10}$$

Generally, it is possible for there to be a trivial solution from $u_n^{k;i}$ to $y_i^k$ in Equation (9) such as, when $u_n^{k;1}$ is extracted from $y_1^k$ and used to reconstruct it as well. However, Equation (7) explicitly requires $u_n^{k;1}$ to rebuild the noise, which prevents the collapse of $u_n^{k;1}$ in expressing $y_1^k$.

### 3.3. Bernoulli Manifold Mixture

The aforementioned latent constraint might appear to be restricted at first, but in fact, it enables us to capture a large number of degrees of freedom in latent space implementation. For instance, it is possible to have multiple scene features that correspond to a single scene. However, in such cases, the mapping from input to scene features becomes ambiguous. To tackle this issue, we propose the use of the Bernoulli Manifold Mixture (BMM) as a means of leveraging the shared structure within the scene latent.

Given the assumption of property (2), the acquired scene features $z_x^{k;1}$ and $z_x^{k;2}$ are expected to be identical and interchangeable with one another, as they both refer to the same scene feature. BMM establishes a new explicit connection

between the scene features of multi-views, which can be expressed in the equation as:

$$\hat{z}_x^k \triangleq \mathrm{Mix}_p(z_x^{k;1}, z_x^{k;2}); \tag{11}$$

where the $\hat{z}_x^k$ is an estimation of the true underlying scene feature. Let $b$ be a sample instance drawn from a Bernoulli distribution with probability $p \in (0, 1)$, the function $\mathrm{Mix}_p(\cdot)$ described in Equation (11) denotes:

$$\mathrm{Mix}_p(m, n) \triangleq bm + (1 - b)n. \tag{12}$$

By establishing this new connection (Equation 11), we can enhance the interchangeability between $z_x^{k;1}$ and $z_x^{k;2}$ by optimising the reconstruction performance on $\hat{z}_x^k$. In Lemma 1. Assuming $z_x^{k;i} \sim N_x(\cdot, \cdot)$, where $N_x$ denotes multivariate Gaussian distributions and then and is the mean and the covariance matrix.
For a given function $G^X(\cdot)$, assume $\forall k, i, m, n \in N$, the following property holds:

$$E\left[G^X(z_x^{k;i})\right] = E\left[G^X(\mathrm{Mix}_p(z_x^{k;m}, z_x^{k;n}))\right]. \tag{13}$$

Proof. Assume $z_x^{k;m}, z_x^{k;n} \in R^{dim}$ are i.i.d., we may also factorise $b \in R^{dim}$. Write

$$\hat{z}_x^k \triangleq \mathrm{Mix}_p(z_x^{k;m}, z_x^{k;n}) \tag{14}$$

so that we can have

$$\hat{z}_x \sim N_x((b + 1 - b)\mu, (b^2 + (1 - b)^2)\Sigma) \sim N_x(\mu, \Sigma) \tag{15}$$

with the fact that Bernoulli sample $b^2 = b$, the mixed feature $\hat{z}_x$ is in the same representation distribution as $z_x^{k;i}$.

In MeD, we denote $\hat{z}_x = G^X(\hat{z}_x^k)$, and the objective for implementing Equation (13) can be formulated as follows:

$$\arg\min_{\cdot;\cdot} L^M \triangleq \| \hat{z}_x - \mathrm{Mix}_p(y_1^k, y_2^k) \|; \tag{16}$$

where the is the weight parameter. Here, the target is using a mixed version of $y_1^k$ and $y_2^k$. The choice of this is driven by the intuition that the hybrid version would better align with the aforementioned blended features.

# 4. Experiments

To evaluate the effectiveness of our proposed method, we assess our method against several representative self-supervised denoising methods, including Noise2Noise (N2N) [16], Noise2Self (N2S) [4] and Recorrupted2Recorrupted (R2R) [26], and the invariant feature learning method LIR [9]. Moreover, we also evaluate our approach against two supervised baseline methods (Noise2Clean (N2C) [21] and multi-frame method DBD [11]) to further validate its effectiveness. Comparisons to more methods, including methods using only one noisy image, are presented in the supplementary material.

We start our experiments by denoising synthetic additive white Gaussian noise (AWGN) in Section 4.2, and then move on to testing unseen noise levels and noise types in Section 4.3 and Section 4.4, respectively. Furthermore, we evaluate the performance in real-world scenarios in Section 4.5. In Section 4.6, we expand our experiments to incorporate more views to study their impact on performance. Finally, in Section 4.7, we apply our method to other tasks, e.g. image super-resolution and inpainting, to demonstrate its generalisation ability.

## 4.1. Experimental Setups

Noted that, the nature of feature disentanglement requires no leak from input to output, however, the global residual connection of the original DnCNN [35] cannot satisfy. Thus, we incorporate the Swin-Transformer (Swin-T) [21] instead of the traditionally used DnCNN in our experiments. Nevertheless, as Swin-T is not originally designed for image restoration, we make some modifications to enforce local dependence across the image. Specifically, we add one Convolution Layer each before the patch embedding and after the patch unembedding of Swin-T, as inspired by SwinIR [18]. The resulting modified network backbone is denoted as Swin-Tx.

To ensure a fair comparison, we use the Swin-Tx backbone for all methods in our study, except for DBD. As DBD did not release the code, we follow the instructions presented in the paper and make our best effort to re-implement it. However, we observe that the two-view DBD could not converge efficiently, which is consistent with the findings in the paper. Therefore, we limit our evaluation to the four-view DBD, denoted as DBD$_4$. Furthermore, we replace the U-Net backbone originally used in the LIR method with Swin-Tx to maintain consistency in our evaluation. This results in an average improvement of approximately 1 dB in PSNR for Gaussian denoising.

In all experiments, all methods were trained using only DIV2K [3] and the same optimisation parameters, except for LIR and DBD$_4$, which used manually selected parameters obtained through experiments. For more training and evaluation details including the choice of parameters, please refer to the supplementary material. Code is available at: https://github.com/chqwer2/Multi-view-Self-supervised-Disentanglement-Denoising.

Remark: In tables, the best results are highlighted in bold, while the second best is underlined

Table 1. Quantitative comparison of different methods on CBSD68 Dataset [23] for Synthetic Gaussian noise. The experiments were conducted on xed and random variance, respectively. The best results are highlighted in bold, while the second best is underlined

| Training Schema | Test ^ | Noisy/ Clean N2C [21] | Noisy/ Clean $DBD_4$ [11] | Noisy/ Noisy N2N [16] | Noisy/ Noisy N2S [4] | Noisy/ Noisy R2R [26] | Invariant Feature LIR [9] | Invariant Feature MeD (ours) |
|---|---|---|---|---|---|---|---|---|
| Gaussian = 25 | 15 | 33.36/ 0.9020 | 33.57/ 0.9092 | 32.64/ 0.8805 | 32.77/ 0.8780 | 29.74/0.7865 | 31.06/ 0.8632 | 33.11/ 0.8880 |
| | 25 | 30.83/ 0.8494 | 31.31/ 0.8548 | 30.68/ 0.8334 | 30.99 0.8405 | 30.45/0.8183 | 30.01/ 0.8024 | 30.57/ 0.8496 |
| | 50 | 24.76/ 0.5519 | 25.12/ 0.5583 | 24.59/ 0.5385 | 22.13/ 0.3928 | 24.02/0.5133 | 21.97/ 0.3578 | 25.67/ 0.6026 |
| | 75 | 20.75/ 0.3376 | 21.09/ 0.3412 | 20.60/ 0.3162 | 17.86/ 0.1998 | 19.10/0.2641 | 16.23/ 0.1689 | 23.09/ 0.4320 |
| Gaussian 2 [5; 50] | 15 | 33.47/ 0.9027 | 33.12/ 0.8915 | 33.45/ 0.8945 | 31.28/ 0.8187 | 20.76/ 0.2508 | 30.85/ 0.8431 | 33.62/ 0.9026 |
| | 25 | 30.87/ 0.8538 | 30.64/ 0.8491 | 30.77/ 0.8423 | 29.65/ 0.7801 | 23.91/ 0.4552 | 28.92/ 0.8069 | 30.91/ 0.8573 |
| | 50 | 27.41/ 0.7361 | 27.13/ 0.7290 | 27.15/ 0.7219 | 27.00/ 0.7114 | 26.92/ 0.6911 | 25.13/ 0.6191 | 27.48/ 0.7530 |
| | 75 | 25.05/ 0.6223 | 24.97/ 0.6205 | 24.80/ 0.5908 | 24.89/ 0.6023 | 23.83/ 0.5132 | 22.37/ 0.4212 | 25.40/ 0.6645 |

Table 2. Quantitative result of generalisation performance experiment on CBSD68 [23]. All methods use Gaussian for pre-trained methods and then Gaussian 2 [5; 50] for ne-turning. The better result in each method is highlighted in bold.

| Fine-tuning Method / Pretraining Method | N2C [21] N2C | N2C [21] MeD | N2N [16] N2N | N2N [16] MeD | LIR [9] LIR | LIR [9] MeD | MeD MeD |
|---|---|---|---|---|---|---|---|
| Gaussian, ^ 2 [15; 75] | 29.20/ 0.7797 | 29.53/ 0.8081 | 29.04/ 0.7642 | 29.21/ 0.7890 | 26.42/ 0.6640 | 27.25/ 0.7036 | 29.60/ 0.8101 |
| Local Var Gaussian | 35.62/ 0.9308 | 35.85/ 0.9439 | 35.66/ 0.9256 | 35.73/ 0.9310 | 29.26/ 0.8170 | 30.51/ 0.8387 | 35.91/ 0.9762 |
| Poisson Noise | 40.49/ 0.9736 | 42.80/ 0.9776 | 41.35/ 0.9736 | 42.27/ 0.9813 | 31.23/ 0.8672 | 33.47/ 0.8932 | 45.05/ 0.9826 |
| Speckle, ^ 2 [25; 50] | 33.36/ 0.9004 | 33.40/ 0.9044 | 33.32/0.8931 | 33.38 0.8907 | 28.28/ 0.7713 | 29.82/ 0.8229 | 33.48/ 0.9115 |
| S&P, ^ 2 [0:3; 0:5] | 28.85/ 0.8267 | 30.73/ 0.8372 | 28.59/ 0.8003 | 29.45/ 0.8255 | 26.69/ 0.7241 | 27.62/ 0.7460 | 30.84/ 0.9135 |
| Average | 33.50/ 0.8822 | 34.46/ 0.8942 | 33.59/ 0.8714 | 34.00/ 0.8835 | 28.38/ 0.7687 | 29.73/ 0.8009 | 34.98/ 0.9188 |

### 4.2. AWGN Noise Removal

We rst investigate the denoising generalisation of the methods using synthetic zero-mean additive white Gaussian noise (AWGN). The experiments are divided into two parts. The rst segment employs xed variance AWGN, whereas the second segment employs varied variance Gaussian for training in a separate manner. Table 1 summarises the quantitative results evaluated on CBSD68 Dataset [23] at variance levels of 15, 25, 50, and 75.

Analysis: In the xed variance setting, MeD performs inferior compared to the other methods on lower noise levels of 15 and 25. However, as the methods face more severe corruption, MeD outperforms all self-supervised and supervised methods, showing our greater advantage of handling severe noise. For instance, at = 75, MeD outperforms the second-best method (N2C) by around 2 dB. These results suggest that MeD has a remarkable ability to generalise to a range of unseen noise levels in Gaussian noise.

In the context of random variance, it has been observed that MeD exhibits superior performance across all four noise levels compared to other methods, including supervised methods. These ndings imply that MeD can bene t more from varying training noise than other methods. More experiments and details on AWGN noise removal can be found in the supplementary material.

### 4.3. Generalisation on Unseen Noise Removal

In the previous subsection, we demonstrated the remarkable generalisation ability of our model in the case of Gaussian noise. Here, we aim to extend this investigation to other types of unseen noise and evaluate the denoising generalisation ability of our method. Speci cally, we consider Poisson noise, Speckle noise, Local Variance Gaussian noise, and Salt-and-Pepper noise. For a more detailed synthetic process, please refer to the supplementary material.

First, we demonstrate qualitative comparisons of denoising unseen noise types using models trained only with Gaussian = 25 in Figure 3. Next, in order to further verify the denoising generalisation ability of MeD, we employ its scene encoder and decoder as pre-trained models to be compared against other methods. It should be noted that the pre-training and ne-tuning methods employed in this study may differ, as shown in Table 2. The pre-training of all test models was conducted on a Gaussian sigma value of 25, followed by ne-tuning with a Gaussian sigma range of 5 to 50. Since the training schema of N2S, R2R, and $DBD_4$ differs from MeD, we do not include them in this section. However, evaluations of these methods on unseen noise are still presented in Section 4.4 under different settings.

Analysis: Qualitative results in Figure 3 show that under Gaussian = 25 training settings, our method surpasses other methods in denoising unseen noise types. Additionally, Table 2 shows that the approaches using pre-trained MeD models outperform their self-transfer models for N2C, N2N, and LIR, with improvements of up to 2 dB in some cases. On average, the MeD pre-trained models show a performance gain of around 0.5 dB across all methods, highlighting the potential of MeD as a powerful pre-training

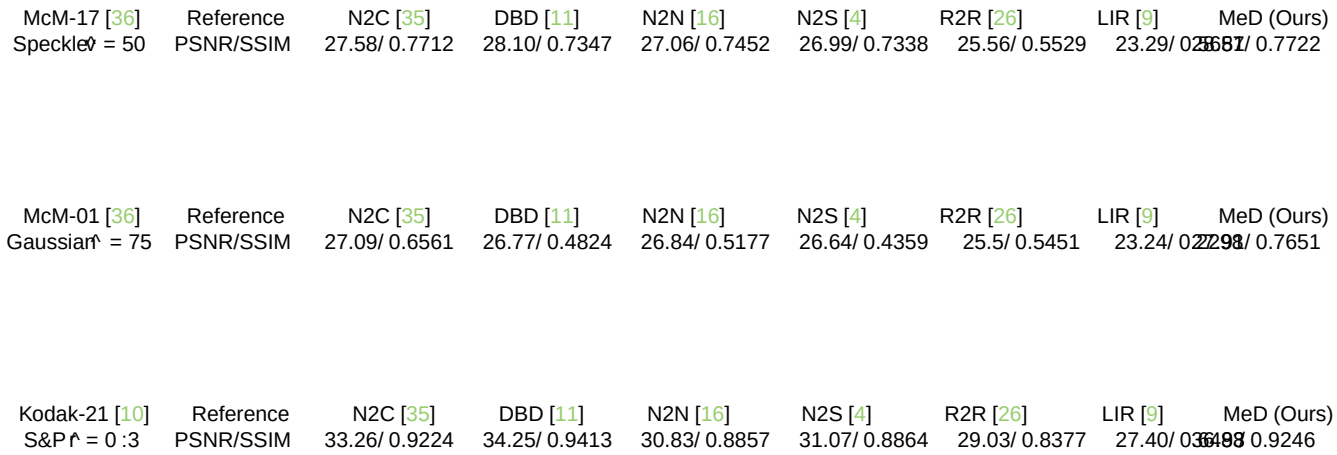



Figure 3. Qualitative denoising results on unseen noise types. All the methods are trained with Gaussian. The quantitative PSNR/SSIM results are provided underneath the respective images. Best viewed in colour (zoom-in for a better comparison).

Table 3. Analysis of Noise Pool on CBSD68 [23]. All methods were trained using randomly drawn noise from the Noise Pool.

| Test Noise | Noisy/ Clean | | Noisy/ Noisy | | | Invariant Feature | |
|---|---|---|---|---|---|---|---|
| | N2C [21] | DBD$_4$ [11] | N2N [16] | N2S [4] | R2R [26] | LIR [9] | MeD (ours) |
| Gaussian, $\sigma \in [15, 75]$ | 29.24/ 0.7754 | 29.05/ 0.7616 | 29.23/ 0.7634 | 28.58/ 0.7589 | 26.43/ 0.6639 | 26.67/ 0.6866 | 29.61/ 0.8178 |
| Local Var Gaussian (LVG) | 36.64/ 0.9442 | 36.18/ 0.9307 | 36.65/ 0.9235 | 33.24/ 0.8858 | 34.70/ 0.8779 | 31.61/ 0.8627 | 37.99/ 0.9568 |
| Poisson Noise | 45.72/ 0.9764 | 44.23/ 0.9606 | 45.64/ 0.9799 | 46.31/ 0.9808 | 44.45/ 0.9491 | 43.27/ 0.9292 | 48.10/ 0.9916 |
| Speckle, $v \in [25, 50]$ | 35.58/ 0.9417 | 35.24/ 0.9385 | 35.34/ 0.9475 | 35.13/ 0.9596 | 34.20/ 0.9078 | 33.98/ 0.8810 | 37.21/ 0.9715 |
| S&P, $r \in [0.3, 0.5]$ | 38.85/ 0.9165 | 37.10/ 0.8884 | 38.89/ 0.9289 | 38.22/ 0.9330 | 36.17/ 0.9087 | 33.43/ 0.8202 | 42.33/ 0.9695 |
| Gaussian $\sigma = 25$ + Speckle $v = 25$ | 30.19/ 0.8279 | 29.24/ 0.8156 | 30.32/ 0.8317 | 29.51/ 0.8050 | 28.78/ 0.7744 | 29.20/ 0.7871 | 31.92/ 0.8726 |
| Gaussian $\sigma = 50$ + Speckle $v = 25$ | 27.30/ 0.7251 | 26.55/ 0.7126 | 27.23/ 0.7331 | 26.91/ 0.7081 | 26.49/ 0.6935 | 26.19/ 0.6941 | 29.68/ 0.7928 |
| LVG + Poisson | 31.78/ 0.9087 | 31.10/ 0.8842 | 31.60/ 0.7617 | 30.15/ 0.8086 | 28.52/ 0.7144 | 27.33/ 0.7234 | 34.29/ 0.9325 |
| Poisson + Speckle $v = 25$ | 31.39/ 0.9069 | 30.86/ 0.8782 | 31.52/ 0.8935 | 30.58/ 0.9067 | 30.34/ 0.8897 | 29.93/ 0.8554 | 33.04/ 0.9258 |
| Average | 34.08/ 0.8803 | 33.28/ 0.8634 | 34.05/ 0.8626 | 33.18/ 0.8607 | 32.23/ 0.8193 | 31.29/ 0.8044 | 36.02/ 0.9145 |

method for image denoising. It is noteworthy that the self-transfer MeD model exhibits the best denoising performance across all validation noise types, even outperforming the supervised method, N2C. This is particularly evident in Poisson noise, where MeD surpasses N2C by 3 dB. These results highlight the generalisation ability of our approach in handling unseen noise.

### 4.4. Experiments on General Noise Pool

Here we further investigate the generalisation ability of our method by introducing our general Noise Pool. The Noise Pool comprises the five aforementioned types of noise, each with a diverse range of noise levels. During training, we randomly sample from the noise pool to provide the model with noisy images. This novel approach resimulates a realistic scenario where noise is unknown and can originate from various sources to some extent.

Specifically, we evaluated all methods using the random noise pool approach to train and test on combined or single noise types. The results are summarised in Table 3.

Analysis: In Table 3, our MeD approach outperforms all other methods significantly on all the test noise types. For example, when a test noise containing a combination of Gaussian noise with $\sigma = 50$ and Speckle noise with $v = 25$ is used, other methods exhibit an approximate performance of 27 dB. However, MeD achieves significantly better results with a performance of 29.68 dB. And on average, MeD exhibits a performance that is approximately 2 dB better than other methods. Our findings show that utilising a comprehensive noise pool for training purposes can effectively improve the generalisation capability. Furthermore, the remarkable denoising generalisation ability of our MeD approach, in comparison to other methods, is particularly advantageous for real-world applications.