TTT-MIM: Test-Time Training with Masked Image Modeling for Denoising Distribution Shifts

Youssef Mansour[1,2], Xuyang Zhong[1,3], Serdar Caglar[1], and Reinhard Heckel[1,2]

[1] Technical University of Munich, Germany
[2] Munich Center for Machine Learning, Germany
[3] City University of Hong Kong, Hong Kong
y.mansour@tum.de, xuyang.zhong@my.cityu.edu.hk,
serdar.caglar@tum.de, reinhard.heckel@tum.de

Abstract. Neural networks trained end-to-end give state-of-the-art performance for image denoising. However, when applied to an image outside of the training distribution, the performance often degrades significantly. In this work, we propose a test-time training (TTT) method based on masked image modeling (MIM) to improve denoising performance for out-of-distribution images. The method, termed TTT-MIM, consists of a training stage and a test time adaptation stage. At training, we minimize a standard supervised loss and a self-supervised loss aimed at reconstructing masked image patches. At test-time, we minimize a self-supervised loss to fine-tune the network to adapt to a single noisy image. Experiments show that our method can improve performance under natural distribution shifts, in particular it adapts well to real-world camera and microscope noise. A competitor to our method of training and finetuning is to use a zero-shot denoiser that does not rely on training data. However, compared to state-of-the-art zero-shot denoisers, our method shows superior performance, and is much faster, suggesting that training and finetuning on the test instance is a more efficient approach to image denoising than zero-shot methods in setups where little to no data is available. Our GitHub page is: https://github.com/MLI-lab/TTT_Denoising.

Keywords: Test Time Training · Distribution Shifts · Masked · Efficient

1 Introduction

The goal of image denoising is to estimate a clean image based on a noisy observation. Convolutional neural networks (such as UNet [26], DnCNN [41], and NAFNet [6]) as well as architectures incorporating attention mechanisms (such as the Restormer [38] and Swin-UNet [5]) trained end-to-end to map a noisy image to a clean image give state-of-the-art performance and significantly outperform classical methods, in particular if the noise is non-Gaussian [4].

However, in practice, the test data a neural network is applied to often comes from a slightly different distribution than the training data. For image reconstruction tasks, such as accelerated magnetic resonance imaging, this is known to induce a significant performance drop [9]. In image denoising, different image domains, noise levels, and noise types can all be regarded as distribution shifts. A particularly interesting distribution shift in image denoising is training on synthetic noise and testing on natural noise,

since it is easy to generate training data with synthetic noise. Methods trained on artificial noise such as Gaussian or Poisson noise suffer from a performance drop when applied to natural noise. This is due to the fact that synthetic noise is insufficient for simulating real camera noise, which is signal-dependent and substantially altered by the camera's imaging system [39].

In this paper, we propose a method for improving the performance of neural network-based denoisers under distribution shifts by adapting the model's parameters to a given instance at test time. Existing test-time adaption methods such as Gaintuning [22] and Lidia [31] assume knowledge of the noise statistics (distribution and variance) of the test image from the new distribution. However, such methods are not applicable to natural noise, since unlike synthetic noise, natural noise can not be accurately estimated with a noise model. Our method does not require a noise model, and is therefore suitable for adapting a network trained on artificial noise to work well on natural noise.

More recent works, such as SS-TTA [12] and Meta-transfer Learning [13] train separate models for real and synthetic noise, i.e., a network is trained on synthetic noise and adapted to synthetic noise of different distribution or level, and a separate network is trained and adapted to natural noise. This means that the severe distribution shift of training on synthetic noise and adapting to camera noise is not considered. Therefore, there is no unified model to denoise a test image of unknown noise distribution. Moreover, [12] utilizes additive Gaussian noise during test time, which results in poor performance for natural camera noise. Unlike prior work, we provide a single unified model that can be adapted to a single noisy image of unknown degradation with no assumptions on the noise distribution in under a second.

Our method is motivated by the test-time-training framework proposed for image classification by Sun et al. [28]. Our pipeline is illustrated in Figure 1. We train a neural network with two heads on two tasks; the main head is trained with a supervised denoising loss, and the auxiliary head is trained with a self-supervised loss, which is reconstructing a masked version of the original noisy image. It was proposed by He et al [14] (masked auto-encoder) and Xie et al. [36] (SimMIM) for computer vision tasks, and is originally inspired by masked language modeling in natural language processing. Training networks to reconstruct masked patches has exhibited immense success in extracting meaningful representations.

In Sun et al.'s TTT method for classification [28], only the model and the auxiliary head are adapted at test time by finetuning on a self-supervised loss, while the main head is kept constant, since a label is required for the supervised loss the main head is trained with. However, in our method, we also adapt the main head during test time with an additional self-supervised loss, which we show to be critical for performance. We propose a pseudo-denoising (PD) loss for adapting the main head, which uses the reconstructed image from the auxiliary head as the pseudo-clean reference image.

Our method successfully adapts a model trained on ImageNet images corrupted with artificial Gaussian noise to perform well on real-world camera (SIDD [1], DND [24], PolyU [37]) and microscope (FMDD [42]) noise based on a single test image. This is of practical significance since natural noise datasets are not available for some denoising tasks.

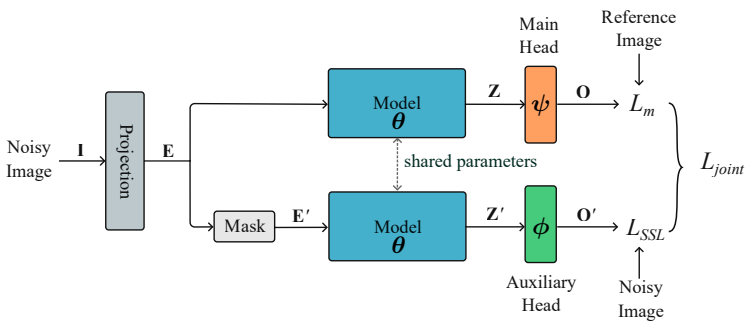

Fig. 1: Overview of our pipeline. The model we utilized is the UNet [26]. We first use a 1×1 conv layer to map the image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ to an embedding with more channels $\mathbf{E} \in \mathbb{R}^{H \times W \times D}$. After that, masking $\mathbf{E}$ yields a masked embedding $\mathbf{E}'$. The volumes $\mathbf{E}$ and $\mathbf{E}'$ are input to the model, and the model yields the feature maps $\mathbf{Z}$ and $\mathbf{Z}' \in \mathbb{R}^{H \times W \times D}$. Finally, we use two different heads to generate the noise residual $\mathbf{O}$ and reconstructed image $\mathbf{O}'$, respectively. The denoised image $\mathbf{D}$ is obtained as $\mathbf{I} - \mathbf{O}$. During training, the reference image in the main branch is the ground truth, but at test time adaptation, the reference image is the pseudo clean image $\mathbf{O}'$.

We show that our method outperforms the state-of-the-art zero shot denoising algorithm Self2Self (S2S) [25], and is five orders of magnitude faster. Zero shot methods are a competing approach for a setup with little or no training data, but suffer from long inference times, as a separate network is trained for each test image. Our comparison to zero shot methods suggests that it is better (in terms of performance and speed) to train on a dataset and then adapt to the given out-of-distribution test instance, as opposed to performing zero shot denoising.

Our method still works on other distribution shifts such as different domains. Specifically, we adapt the model trained on ImageNet to work well on medical images, such as CT [27] and MRI [40]. We also consider the distribution shift of varying noise types by adapting the model trained on Gaussian noise to Poisson and Salt & Pepper noise.

To summarize, our main contribution is that we enable denoising of a single out-of-domain image without any knowledge of the noise distribution or level in a very short time. We achieve this by adapting a single model trained on synthetic noise to the input noisy image. Our method requires only a few iterations and takes less than a second to denoise. Since we adapt to a single image, our method can be regarded as a fast zero-shot denoiser. Learning based zero-shot methods usually take long since they train a separate network from scratch for each given image. Our work shows that adapting a single pretrained network at test time can achieve equal or better performance, while significantly reducing the time.

2 Related Works

2.1 Image Denoising

Image denoising is a sub-task of image restoration, which aims to restore a degraded image. Traditionally image restoration algorithms and denoising algorithms are not based

on learning. Today, neural networks trained end-to-end give state-of-the art image quality. The network architecture used for denoising is critical for performance. Most existing works can be categorized into three architectures CNN-based [26,41], Transformer-based [5, 19, 33, 38], and MLP-based [21, 29] models.

2.2 Self-Supervised Image Denoising

Since the acquisition of clean images is expensive, and in some domains such as medical imaging or astronomy is not even possible, a variety of self-supervised training methods have been proposed for training image denoisers with only noisy images. Noise2Noise [18] enables denoising without clean images, but only using pairs of two noisy images of the same scene. Neighbour2Neighbour [16] extends Noise2Noise to allow denoising using single noisy images. Noise2Self [3] masks pixels and trains a model to predict each masked pixel by its neighboring pixels. Self2Self [25] follows Noise2Self but in a zero shot setup, where a separate network is trained from scratch for each test noisy image.

2.3 Masked Image Modeling

Masked image modeling aims to learn representations from images by training a network to predict masked patches. Context encoder [23] was the first work in this direction to inpaint large missing rectangle regions using convolutional networks. With the success of vision Transformers, iGPT [7], ViT [11] and BEiT [2] showed the potential of masked image modeling on Transformer-based models by introducing special designs on some components. In contrary to these complex designs, MAE [14] and SimMIM [36] directly predict the masked patches. MAE only feeds the embeddings of unmasked patches into the encoder, and predicts the masked patches using a light decoder. SimMIM feeds all embeddings (masked + unmasked) into the encoder.

2.4 Test-Time Training

Test-time training (TTT), also called test-time adaptation, adapts trained machine learning models based on new test samples, with the goal of improving the distribution shift performance. TTT for classification tasks [17, 28], has exhibited the ability to adapt a trained classifier to new test domains in an unsupervised manner, i.e., with no labels from the new test set. For instance, TENT [32] minimizes the entropy of the output and only updates the batch-norm statistics and affine parameters.

TTT approaches for denoising include GainTuning [22] and Lidia [31]. Lidia requires a noise model and is therefore tested only on Gaussian noise. Gaintuning also requires a noise model, but is tested on real microscope noise that can be approximated with a Gaussian-Poisson model, but can't be tested on camera noise of unknown noise model. Other works like SS-TTA [12] and Meta-transfer Learning [13] do not consider severe distribution shifts, i.e., training and testing sets are not very different. This results in several models, where each model is used for a certain type of noise distribution. While this may work well, it requires prior knowledge of the test images to pick the corresponding model.

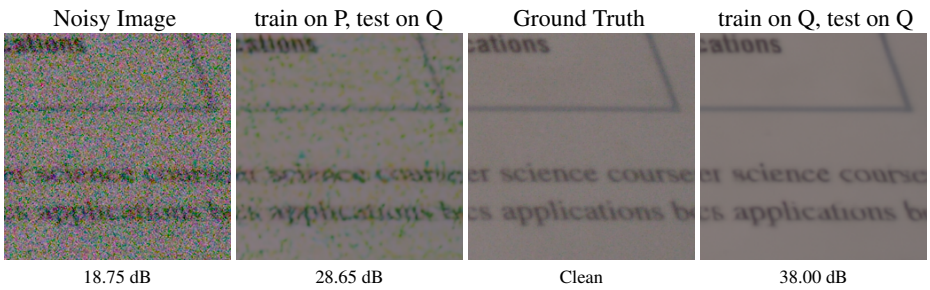

Fig. 2: Visualizing the effect of the distribution shift, where a model trained on one distribution is tested on an image from another distribution.

Contrary to previous work, we provide a single model that can be adapted to any test image from an unknown distribution. The main challenge of our work is to formulate an appropriate self-supervised objective function that does not rely on ground truths or prior assumptions on the test set, such that the parameters can be tuned at test time with no knowledge of the new distribution.

3 Problem Statement

We propose a method to improve the denoising performance under distribution shifts. Distribution shifts refer to a mismatch between the training and test set. For example, let distribution P be images from Imagenet corrupted with Gaussian noise, and distribution Q images from the SIDD [1], which is a dataset of real-world camera noisy images. If we train a model on images from Q, and test it on a different image from Q, we get 38.00 dB. However, if we train on P and test on Q, we get 28.65 dB. This drop in performance is known as the gap induced due to a distribution shift. See figure 2 for a visualization. Our aim is to close this gap by adapting the model trained on P to the test image from Q.

4 Method

Our method consists of a training phase during which the model is trained on pairs of noisy images and their corresponding ground truth data, and a test time adaptation phase, where the model is adapted to a noisy image to produce a clean image.

As seen in Figure 1, the model has two branches. During training, the main branch is trained with a standard denoising supervised loss, and the auxiliary branch is trained with a self-supervised reconstructing loss.

At test time, the auxiliary branch is adapted with the same self-supervised loss as in training, and the main branch is adapted with a pseudo denoising self-supervised loss, which uses the output of the auxiliary head as a label. The elements of our method are described in the following sections.

4.1 Model and Mask

We use the encoder-decoder-convolutional-based UNet [26] as the backbone model, whose exact architecture can be found in the supplementary material. The trainable parameters of the model are denoted by $\boldsymbol{\theta}$.

Following [35], who utilized masked image modeling for pretraining vision downstream tasks, we first use a 1×1 convolutional layer to project the noisy image $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$ to an image with more channels, which yields the embedding $\mathbf{E} \in \mathbb{R}^{H \times W \times D}$. We then randomly mask $\mathbf{E}$ and obtain a masked embedding $\mathbf{E}'$. The random masking is implemented patch-wise, i.e., the mask patch size determines the size of each square masked patch, and the mask ratio determines the fraction of the masked-off area. Formally, the process of masking is defined as:

$$\mathbf{E}' = (\mathbb{1} - \mathbf{M}) \odot \mathbf{E} + \mathbf{M} \odot \boldsymbol{t}, \tag{1}$$

where $\mathbb{1}$ is a matrix full of 1s, and $\mathbf{M}$ is a binary patch-wise mask matrix, with the same size as $\mathbf{E}$. The mask ratio is 0.6 and the mask patch size is 7 here. The vector $\boldsymbol{t}$ is a learnable token vector of length D, that is expanded to $H \times W \times D$ when computing its dot product with $\mathbf{M}$. The expansion occurs by repeating $\boldsymbol{t}$ $H \times W$ times. The dot product of $\mathbf{M}$ with $\boldsymbol{t}$ is equivalent to multiplying each channel of $\mathbf{M}$ with a learnable scalar. After that, $\mathbf{E}$ and $\mathbf{E}'$ are fed into the model, respectively. The model outputs the feature maps $\mathbf{Z}$ and $\mathbf{Z}' \in \mathbb{R}^{H \times W \times D}$. A main and an auxiliary head, parameterized by $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ respectively, generate the noise residual $\mathbf{O}$ and reconstructed image $\mathbf{O}'$, respectively. The denoised image $\mathbf{D}$ is the difference between the noisy image $\mathbf{I}$ and the noise residual $\mathbf{O}$, i.e., $\mathbf{D} = \mathbf{I} - \mathbf{O}$. Both heads are identical and are a stack of convolutional layers followed by leaky ReLU and group normalization layers. Their exact architecture is in the supplementary material.

4.2 Joint Training

The goal of the main branch is to denoise the image, therefore it is trained with a supervised denoising loss $\mathcal{L}_m$, that utilizes the ground truth image $\mathbf{Y}$. We use the residual loss [41], where the model learns to fit the noise. $\mathcal{L}_m$ is defined as:

$$\mathcal{L}_m(\boldsymbol{\theta}, \boldsymbol{\psi}) = \frac{1}{HWC} ||\mathbf{O}_{\boldsymbol{\theta},\boldsymbol{\psi}} - (\mathbf{I} - \mathbf{Y})||_1, \tag{2}$$

where H, W and C are the height, width and number of channels of the image, respectively.

As for the auxiliary branch, its goal is to learn meaningful features in a self-supervised manner, such that it can be adapted at test time without a ground truth. The original TTT for classification paper [28] used rotation prediction as a self supervised loss. However, the recent advancements in representation learning have shown that reconstructing masked patches [14] achieves SOTA performance for extracting useful representations. We therefore utilize the masked image modeling loss [36], which aims to reconstruct the noisy image from its masked version, while calculating the loss only over the masked pixels. The auxiliary loss $\mathcal{L}_{SSL}$ is:

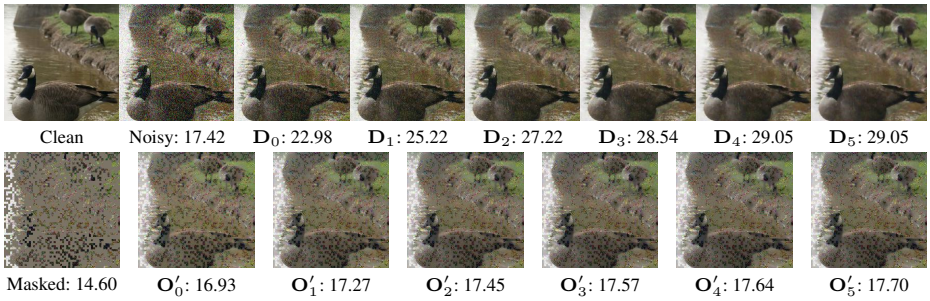

Fig. 3: Visualization of the test time adaptation process. The scores denote the PSNR values w.r.t. the clean image. Noisy and masked images are the inputs to the main and auxiliary branches respectively. $\mathbf{D}_k$ is the denoised image obtained by $\mathbf{D}_k = \mathbf{I} - \mathbf{O}_k$. $\mathbf{O}_k$ and $\mathbf{O'}_k$ are the outputs of the main and auxiliary heads respectively at each gradient update iteration of the $\mathcal{L}_{TTT}$ loss. Iteration 0 represents the initial outputs of the heads before any adaptation.

$$\mathcal{L}_{SSL}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \frac{1}{M}||\mathbf{M} \odot (\mathbf{O}'_{\boldsymbol{\theta},\boldsymbol{\phi}} - \mathbf{I})||_1, \tag{3}$$

where M is the sum of the mask matrix $\mathbf{M}$. It should be noted that $\mathbf{M}$ is randomly generated at each iteration.

Finally, we formulate the joint loss $\mathcal{L}_{joint}$ consisting of the supervised loss $\mathcal{L}_m$ computed over the main branch, and the self-supervised loss $\mathcal{L}_{SSL}$ over the auxiliary branch:

$$\mathcal{L}_{joint}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}) = \frac{1}{2}(\mathcal{L}_m(\boldsymbol{\theta}, \boldsymbol{\psi}) + \mathcal{L}_{SSL}(\boldsymbol{\theta}, \boldsymbol{\phi})). \tag{4}$$

4.3 Test-Time Adaptation

At test-time, we adapt the trained weights to the test image. We adapt the auxiliary branch by minimizing the self-supervised loss from eq. (3). However, the main branch can not be adapted with the supervised loss from training, since it requires ground truths. The original TTT for classification paper [28] kept the main head constant at test time. We also adapt the main head at test-time, which we show in the ablation studies to be crucial for performance.

We propose to use the output of the auxiliary head as a label for adapting the main head. We call the loss to tune the main head the Pseudo Denoising (PD) loss, as the auxiliary head's output can be thought of as a pseudo label. Pseudo labelling is widely used in classification [15], but to our knowledge has not been utilized in image recovery. The PD loss is:

$$\mathcal{L}_{PD}(\boldsymbol{\theta}, \boldsymbol{\psi}) = \frac{1}{M}||\mathbf{M} \odot [\mathbf{O}_{\boldsymbol{\theta},\boldsymbol{\psi}} - (\mathbf{I} - \mathbf{O}')]||_1. \tag{5}$$

The $\mathcal{L}_{PD}$ loss is similar to the $\mathcal{L}_m$ loss in eq. (2) used during training. The main difference is that the ground truth $\mathbf{Y}$ is replaced by the reconstructed image $\mathbf{O}'$, and

the loss is calculated only over the masked pixels. Since both $\mathbf{O}$ and $\mathbf{O}'$ are outputs of the same model, minimizing $\mathcal{L}_{PD}$ will lead to mode collapse, which is a trivial solution being the minimizer of the loss function. Following SimSiam [8], we apply stop gradient on $\mathbf{O}'$, which treats $\mathbf{O}'$ as fixed and not differentiable.

Finally, the test-time adaptation loss $\mathcal{L}_{TTT}$ combines the losses on the main and auxiliary branches together, and can be written as:

$$\mathcal{L}_{TTT}(\boldsymbol{\theta}, \boldsymbol{\psi}, \boldsymbol{\phi}) = \mathcal{L}_{SSL}(\boldsymbol{\theta}, \boldsymbol{\phi}) + \alpha \mathcal{L}_{PD}(\boldsymbol{\theta}, \boldsymbol{\psi}), \tag{6}$$

where α is a scaling factor set to 0.01.

Figure 3 visualizes the adaptation process at test time. The noisy image and its masked version are fed into the main and auxiliary branches to output the noise residual $\mathbf{O}_0$ and reconstructed image $\mathbf{O}'_0$ respectively. The subscript denotes the iteration number. Iteration 0 is the heads' outputs before adaptation. Then each iteration denotes a gradient update step of the $\mathcal{L}_{TTT}$ loss.

Minimizing $\mathcal{L}_{TTT}$ during test time till convergence leads to overfitting on the self-supervised losses and dramatically changes the model's initially trained parameters. We therefore early stop at 8 iterations. This value was chosen based on a small validation set from ImageNet and then fixed for all other datasets and tasks. Different datasets and noise types have different optimal early stopping iterations, which can be determined by cross validation if a few labels from the dataset are available. However, we assume no prior availability or knowledge of the test set, and therefore fix the early stopping iteration at 8 for all datasets and tasks. A detailed discussion on the optimal iteration number is in section 6.

Using the auxiliary head's output as a label might seem counter intuitive, since, as seen in the second row of figure 3, the reconstructed images are noisy. However, using noisy labels together with early stopping is common in image recovery. It was shown by Ulyanov *et al.* in Deep Image Prior (DIP) [30], that a network trained with early stopping to map random noise to a noisy image, will denoise the image, since the network has an inductive bias towards natural images, and can fit the image before the noise.

Our approach differs from DIP in that we utilize better input and labels. DIP takes random noise as input, and the noisy image as label, whereas our input is the noisy image, and our label is the auxiliary head's output $\mathbf{O}'$. Moreover, our label is dynamic, in that it gets updated with every iteration, as the model is adapted to the test image. Since the noisy image (our input) is a better estimate of the clean image compared to random noise (DIP's input), and $\mathbf{O}'$ (our label) is adapted to become less noisy than the noisy image (DIP's label), our method requires only 8 gradient descent iterations to denoise an image, while DIP requires thousands of iterations. Furthermore, our method significantly outperforms DIP as shown in the next section.

It is natural to question our choice of the PD loss for adapting the main branch, and consider using another self-supervised loss, such as $\mathcal{L}_{SSL}$, which is used for the auxiliary branch. Reconstructing masked patches causes loss of details and a blur in the reconstructed patches as seen in the second row of figure 3. An alternative is the blind spot networks, which mask the image pixel wise instead of patch wise, and have

Method	Natural Noise				Gaussian 0.005		ImageNet				Avg.
	SIDD	DND	PolyU	FMDD	CT	fastMRI	G0.01	G0.02	S&P	Poisson	
input noisy	25.49	29.98	36.69	39.10	36.86	23.54	20.41	17.57	18.00	27.76	27.27
train on P, test on Q	28.98	35.08	38.06	43.71	43.73	27.87	28.03	23.00	23.46	32.66	32.17
finetune on Q, test on Q	37.78	38.77	39.35	45.14	51.45	32.60	30.39	30.75	32.46	34.89	37.75
TTT-MIM (ours)	**33.58**	**36.91**	**38.33**	**44.70**	46.05	29.87	**29.65**	**27.35**	25.86	32.91	**34.26**
Self2Self [25]	30.43	-	37.70	39.19	42.82	31.99	29.61	27.20	**26.07**	**33.21**	33.14
DIP [30]	30.81	-	37.14	43.25	46.35	31.32	25.81	23.67	23.35	27.54	32.14
ZS-N2N [20]	30.42	-	36.91	43.45	**46.65**	**32.19**	28.90	26.51	24.30	29.40	33.19
gap closed by TTT-MIM	52.3%	49.6%	34.8%	69.2%	30.0%	42.3%	68.6%	56.1%	26.7%	11.2%	43.5%

Table 1: Results for adapting to a single image in terms of PSNR. Best results are highlighted in bold. DND results for the zero-shot methods are omitted due to the high computational cost. For consistency, average results in last column do not include DND. G0.01 and G0.02 denote Gaussian noise with variance 0.01 and 0.02 respectively. S&P denotes Salt & Pepper noise.

achieved impressive self-supervised denoising performance. In the ablation studies we show that our loss outperforms the standard blind spot loss Noise2Self [3].

5 Experiments

We first train the model with joint training on 38400 ImageNet [10] images corrupted with Gaussian noise with constant variance 0.005 (pixel range 0-1). We then validate the ability of our test-time-training method to adapt to three different classes of distribution shifts.

- The first and perhaps most interesting class of distribution shifts is natural camera and microscope noise. Specifically we test on the SIDD [1], DND [24], and PolyU [37] real world camera noise datasets and the FMDD [42] natural microscope noise dataset.
- The second class of distribution shifts is images from a different domain corrupted with the same Gaussian noise (0.005 variance) the model has been trained on. Specifically, we evaluate on CT images from a CT challenge [27] and MRI images from fastMRI [40].
- The third class of distribution shifts is varying noise types and levels applied to the same dataset (ImageNet) the model has been trained on. For this third class of distribution shifts, we consider Gaussian noise with variance 0.01 and 0.02, Salt & Pepper noise, and finally Poisson noise. More details on the datasets can be found in the supplementary material.

We refer to the training set as distribution P (which is ImageNet corrupted with Gaussian noise of variance 0.005 throughout), and to the test set as distribution Q. We apply our test time adaptation method (referred to as **TTT-MIM**) to the model trained on P before testing it on Q.

To put our results in context, we consider the following methods for comparison:

- **Train on P, test on Q**: The model jointly trained on the training distribution P is directly evaluated on the test distribution Q without any adaptation. This gives an idea on the severity of the distribution shift.

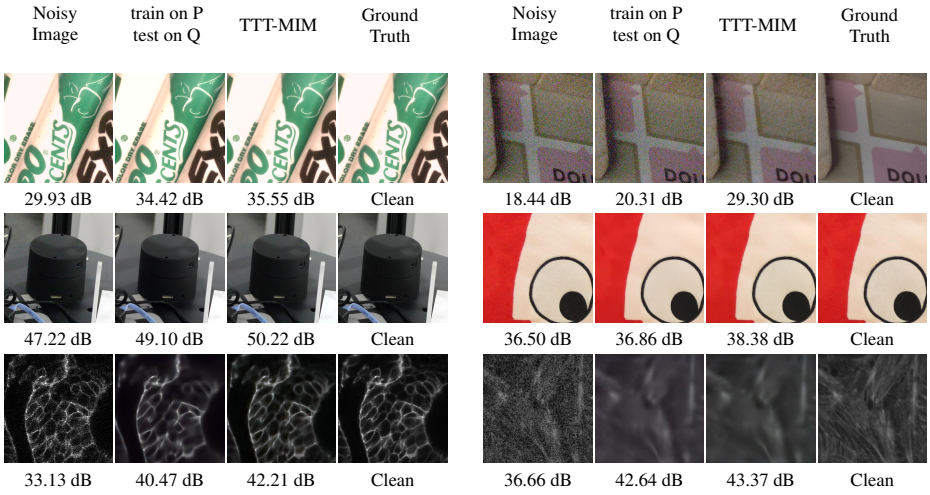

Fig. 4: Natural noise. Datasets from top to bottom are: SIDD, PolyU, and FMDD.

– **finetune on Q, test on Q**: This is a baseline method that is trained on P and then supervisedly finetuned on Q. This baseline indicates the performance the method can obtain when clean data from Q is available. However, this performance is impossible to reach in practice since we do not have access to ground truths from Q.

Regarding the baseline **finetune on Q, test on Q**, we also trained on Q instead of training on P and finetuning on Q. If abundant noisy-clean image pairs from Q are available, then training on Q results in better performance than training on P and finetuning on Q. However, for some datasets we considered, only few data from Q is available, and for those datasets training on P then finetuning on Q performed better. For consistency, we therefore only report the results of **finetune on Q, test on Q** throughout.

As additional baselines, we compare to Self2Self (S2S) [25], which is a state-of-the-art zero-shot single-image denoising method. We also compare to DIP [30], which motivated our loss function at test time. We finally compare to the recent zero-shot denoiser ZS-N2N [20], which achieves good performance at impressive speed. Similar to our method, S2S, DIP, and ZS-N2N are all blind denoisers that do not require the noise distribution or level. We do not compare to existing test time adaption methods since they are not applicable to natural noise [22, 31], or have no available code online [12, 13].

5.1 Adaptation performance to single images

We show the results for adapting to a single image in table 1. We also display sample images in figures 4 and 5 to show the performance of our method visually.

Since the compute needed for the zero shot methods is expensive (as we show in section 5.3), the results reported in table 1 are based on only 10 images randomly sampled

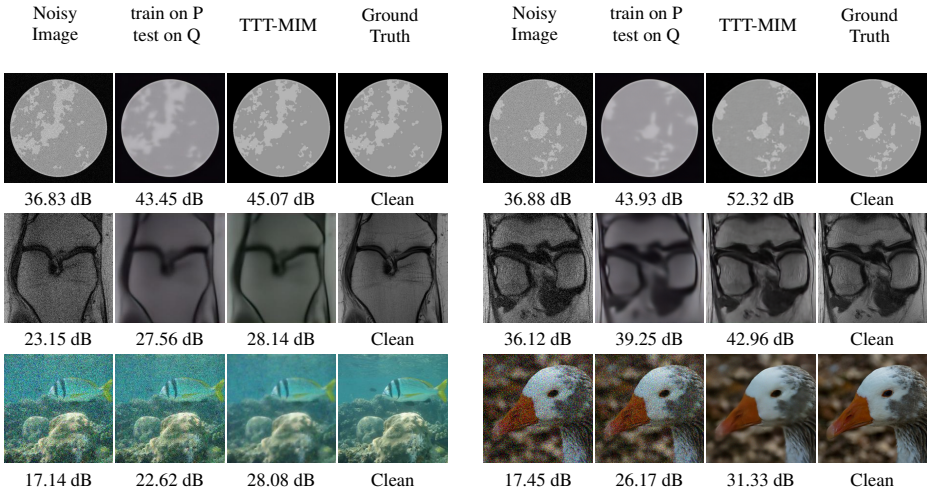

Fig. 5: Artificial Gaussian noise. Datasets from top to bottom are : CT, MRI, and ImageNet.

from each test set for each distribution shift. The model is adapted to each single image independently, and then reinitialized to the initially trained weights before adapting to the next image.

The DND consists only of a test set of thousand noisy images of size 512×512 with no corresponding clean images, and no training set. The evaluation on DND can only be obtained by uploading the denoised images to the DND website and retrieving the scores from there. The DND evaluation website requires to submit the entire test set for evaluation. Therefore, the DND results are based on the entire test set, and we were unable to retrieve results for S2S, DIP, and ZS-N2N, since that would require thousands of GPU hours.

We define the performance drop due to the distribution shift as the gap. More specifically, the gap is the difference between the scores of **train on P, test on Q** and **finetune on Q, test on Q**. We define the performance gain due to our method as the difference between the scores of **train on P, test on Q** and **TTT-MIM**. The gap closed by **TTT-MIM** denotes the performance gain due to our method relative to the gap.

It should be noted that since DND [24] consists only of noisy images, the baseline **finetune on Q, test on Q** is not directly applicable to DND, as there is no ground truths to finetune on. However, SIDD images are very similar to the DND ones. As a result, we finetune on SIDD, then test on DND.

The results show that the test-time-training method significantly improves performance on natural noise datasets, when adapting to images of other domains, and when adapting to Gaussian noise with a different noise level. However, its performance gain for artificial Salt & Pepper and Poisson noise is only minimal.

This is likely due to the real-world noise being slightly similar to the Gaussian noise the model was initially trained on, while the Poisson and S&P noise are very different and therefore the network fails to learn the new noise structure from one image only.

Nevertheless, we show in the ablation studies in table 2 that with a batch of images, our method still works for the different artificial noise types and exhibits even better performance for the other distribution shifts.

5.2 Comparison to zero shot methods

We compare our method to the recent zero-shot methods ZS-N2N [20] and Self2Self [25] (S2S), and to DIP [30]. Zero-shot methods overcome the distribution shift problem by not relying on any training data. Instead, they train a separate network from scratch for each incoming test image, and therefore are time consuming. We compare our method to S2S, DIP, and ZS-N2N to show the advantages of our approach over zero shot methods.

Our method outperforms DIP, S2S, and ZS-N2N on all natural noise datasets and also on the average of all tasks. Moreover, our experiments, show that S2S is hyperparameter sensitive, especially, to the learning rate. For dark images such as the FMDD or CT images, S2S's default learning rate causes trivial solutions (all-black images). This requires careful hyperparameter tuning before running S2S.

The advantage of our method over zero-shot methods is not only in performance, but also in the speed, as seen in the next section.

5.3 Computational cost

In addition to the improved performance, our method offers significant reduction in computational cost compared to zero shot methods. Self2Self requires 150k iterations until convergence, which takes 1.2 hours to denoise one 256×256 RGB image on GPU. DIP is more efficient and needs around 3-5k iterations, which brings the denoising down time to about 7 minutes for one image. ZS-N2N also requires 3k iterations, but since it utilizes a lightweight network, it takes half a minute to denoise an image. Our proposed scheme of adapting an already trained network to the test image instead of training a new network from scratch provides substantial decrease in the denoising time. Since we only execute 8 update iterations, our method requires less than a second to denoise an image.

5.4 Ablation Studies

Next, we present some ablation studies. We perform the studies on a batch of images instead of on single images, which means the model is adapted to a collection of images as a whole. This is in contrast to the previous section, where the model is adapted to each individual image in isolation, and then reset to its initially trained weights before being adapted to the next image.

Adapting to a batch of images might be of practical importance, since a collection of test images from the same domain might be available. Zero-shot methods do not benefit from multiple test images, since they train a network from scratch for each image independently. However, we show that our method exhibits performance gains through batch-wise adaptation.

We consider the following ablated versions of our method:

Method	Natural Noise				Gaussian 0.005		ImageNet				Avg.
	SIDD	DND	PolyU	FMDD	CT	fastMRI	G0.01	G0.02	S&P	Poisson	
	4000	1000	100	48	1000	398	12000				
input noisy	29.98	29.98	35.79	39.72	36.86	23.39	20.43	17.62	17.92	27.78	27.95
train on P, test on Q	31.60	35.08	37.11	43.15	43.63	27.60	28.64	23.18	23.46	34.19	32.76
finetune on Q, test on Q	42.31	38.77	39.19	45.54	56.21	32.01	32.01	30.10	35.64	36.58	38.84
only GN	31.24	34.91	37.10	43.24	43.58	28.73	31.16	28.60	26.33	33.97	33.89
w/o main head	31.21	32.91	36.69	43.87	45.61	30.39	**31.76**	28.75	27.39	**34.97**	34.36
w/ N2S loss	36.52	30.97	37.00	45.51	51.78	**31.42**	31.13	**29.44**	27.35	34.64	35.58
TTT-MIM	**39.69**	**37.04**	**38.56**	**45.53**	**51.81**	31.20	30.81	28.93	28.36	33.26	**36.52**
NB2NB [16]	32.43	30.91	31.68	32.15	33.08	30.55	28.79	26.98	**28.50**	33.12	30.82
gap closed	75.4%	53.1 %	69.6 %	99.8%	65.0%	86.6%	92.7%	90.4%	37.1%	32.7%	70.2%

Table 2: Ablation studies for adapting to a batch of images in terms of PSNR. The dataset size is under each dataset. Best results are highlighted in bold. G0.01 and G0.02 denote Gaussian noise with variance 0.01 and 0.02 respectively. S&P denotes Salt & Pepper noise. The gap closed is calculated using the best version from the ablated versions. However, in most cases TTT-MIM, which is our method's default version, either performs best or falls slightly short of the best performing version.

- **only GN**: only update the affine parameters in the group normalization layers [34] by minimizing $\mathcal{L}_{SSL}$.
- **w/o main head**: update the parameters of the whole model and auxiliary head with $\mathcal{L}_{SSL}$, but keep the main head fixed.
- **w/ N2S loss**: same as w/o main head, but also update the main head with the Noise2Self [3] loss. The exact formulation of the N2S loss is in the supplementary material.
- **w/ PD loss (TTT-MIM)**: default version of our method. Same as w/o main head, but also update the main head with $\mathcal{L}_{PD}$.

We report the ablation results in table 2. The batch size, i.e., size of the test set is indicated below each dataset.

From the results, we can observe that our method significantly improves the denoising performance on all distribution shifts. Our method works particularly well on real world noise, and varying image domains, which are distribution shifts of practical significance.

By inspecting the scores from the ablated versions of our method, we can see the importance of the choices we made when designing our method. The original TTT paper for classification [28] proposed to keep the main head fixed. By checking the results for w/o main head in table 2, we see the necessity for adapting the main head as well. Moreover, our proposed PD loss surpasses the standard N2S loss in almost all distribution shifts. In very rare cases, the N2S loss is only marginally better.

For reference, we add an extra baseline; Neighour2Neighbour (NB2NB) [16], which is a self-supervised blind denoising method that requires a dataset of noisy images, but doesn't require ground truth images. NB2NB works by decomposing a noisy image into four downsampled noisy images and then training a network to map one of the downsampled images to another one; an idea motivated by Noise2Noise [18].

One requirement for Noise2Noise and in turn NB2NB to work well is a large training set. The original papers of both Noise2Noise and NB2NB trained on 50k images. However, as seen in table 2, except for ImageNet, the test set sizes we use are much smaller. Therefore in most cases, NB2NB's performance is significantly worse than our method. Only on the ImageNet experiments, were the test set is large (12k images), NB2NB's performance approaches our method's performance and surpasses it slightly only on Salt & Pepper noise.

6 Limitations

In the adaptation phase, our approach executes eight iterations of gradient descent to adapt the pretrained model to each specific test image. This number is fixed and used across all datasets, which is suboptimal. The optimal number of iterations is a hyperparameter that differs slightly across different datasets. To assess the significance of employing the optimal iteration number, experiments were conducted in accordance with the configuration detailed in Table 1, but using ground truth images to determine the best performing iteration number for each dataset. This procedure yielded an average improvement of 0.5 dB in the performance of our method. These findings highlight the potential for further optimization of our approach through the development of a technique capable of determining the optimal iteration number in the absence of ground truth images.

7 Conclusion

In this work, we propose TTT-MIM, a test-time training method based on masked image modeling, to improve the performance of neural networks trained end-to-end for denoising under distribution shifts. A unified model pretrained on Gaussian noise, is adapted at test time to single test images from unknown distortion. Adaptation requires only 8 gradient descent iterations, which takes less than a second on any GPU.

Our method enables a model trained on ImageNet corrupted with synthetic Gaussian noise to work well on real-world camera and microscope noise, medical images, and ImageNet images corrupted with other types of noise, such as Poisson and Salt & Pepper. Our method's good performance on natural noise and medical images is of great practical significance, since it is relatively easy to build a training set with synthetic noise, but typically expensive to construct a dataset with natural noise or medical images.

Our results show that our approach outperforms zero-shot methods and is much faster. A main take away of our work is that in the regime of low to no training data, supervised pretraining and adapting to the incoming test sample is comparable to zero-shot denoising in terms of image quality, but significantly superior considering the computational cost.

Acknowledgements

Y.M. and R.H. are supported by the Institute of Advanced Studies at the Technical University of Munich, the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - 456465471, 464123524, the German Federal Ministry of Education and Research, and the Bavarian State Ministry for Science and the Arts. The authors of this work take full responsibility for its content.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 2, 5, 9
2. Bao, H., Dong, L., Wei, F.: Beit: Bert pre-training of image transformers. In: International Conference on Learning Representations (ICLR) (2022) 4
3. Batson, J., Royer, L.: Noise2self: Blind denoising by self-supervision. In: International Conference on Machine Learning (ICML) (2019) 4, 9, 13
4. Broaddus, C., Krull, A., Weigert, M., Schmidt, U., Myers, G.: Removing structured noise with self-supervised blind-spot networks. In: International Symposium on Biomedical Imaging (ISBI) (2020) 1
5. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. In: European Conference on Computer Vision Workshops (ECCVW) (2022) 1, 4
6. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: IEEE/CVF European Conference on Computer Vision (ECCV) (2022) 1
7. Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., Sutskever, I.: Generative pretraining from pixels. In: International Conference on Machine Learning (ICML) (2020) 4
8. Chen, X., He, K.: Exploring simple siamese representation learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 8
9. Darestani, M.Z., Liu, J., Heckel, R.: Test-time training can close the natural distribution shift performance gap in deep learning based compressed sensing. In: International Conference on Machine Learning (ICML) (2022) 1
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2009) 9
11. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (ICLR) (2021) 4
12. Fahim, M.A.N.I., Boutellier, J.: Ss-tta: Test-time adaption for self-supervised denoising methods. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2023) 2, 4, 10
13. Gunawan, A., Nugroho, M.A., Park, S.J.: Test-time adaptation for real image denoising via meta-transfer learning. In: arXiv preprint (2022) 2, 4, 10
14. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 4, 6

15. Hu, Z., Yang, Z., Hu, X., Nevatia, R.: Simple: Similar pseudo label exploitation for semi-supervised classification. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 7
16. Huang, T., Li, S., Jia, X., Lu, H., Liu, J.: Neighbor2neighbor: Self-supervised denoising from single noisy images. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 4, 13
17. Kundu, J.N., Venkat, N., Babu, R.V., et al.: Universal source-free domain adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 4
18. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2Noise: Learning image restoration without clean data. In: International Conference on Machine Learning (ICML) (2018) 4, 13
19. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021) 4
20. Mansour, Y., Heckel, R.: Zero-shot noise2noise: Efficient image denoising without any data. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 9, 10, 12
21. Mansour, Y., Lin, K., Heckel, R.: Image-to-image mlp-mixer for image reconstruction. In: arXiv preprint (2022) 4
22. Mohan, S., Vincent, J.L., Manzorro, R., Crozier, P., Fernandez-Granda, C., Simoncelli, E.: Adaptive denoising via gaintuning. In: Neural Information Processing Systems (NeurIPS) (2021) 2, 4, 10
23. Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2016) 4
24. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 2, 9, 11
25. Quan, Y., Chen, M., Pang, T., Ji, H.: Self2self with dropout: Learning self-supervised denoising from single image. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (2020) 3, 4, 9, 10, 12
26. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention (MICCAI) (2015) 1, 3, 4, 6
27. Sidky, E.Y., Pan, X.: Report on the aapm deep-learning sparse-view ct grand challenge. In: Medical Physics (2022) 3, 9
28. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: International Conference on Machine Learning (ICML) (2020) 2, 4, 6, 7, 13
29. Tu, Z., Talebi, H., Zhang, H., Yang, F., Milanfar, P., Bovik, A., Li, Y.: Maxim: Multi-axis mlp for image processing. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4
30. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018) 8, 9, 10, 12
31. Vaksman, G., Elad, M., Milanfar, P.: Lidia: Lightweight learned image denoising with instance adaptation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2020) 2, 4, 10
32. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. In: International Conference on Learning Representations (ICLR) (2021) 4

33. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 4
34. Wu, Y., He, K.: Group normalization. In: European Conference on Computer Vision (ECCV) (2018) 13
35. Xie, Z., Geng, Z., Hu, J., Zhang, Z., Hu, H., Cao, Y.: Revealing the dark secrets of masked image modeling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 6
36. Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., Hu, H.: Simmim: A simple framework for masked image modeling. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 4, 6
37. Xu, J., Li, H., Liang, Z., Zhang, D., Zhang, L.: Real-world noisy image denoising: A new benchmark. In: arXiv preprint (2018) 2, 9
38. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 1, 4
39. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 2
40. Zbontar, J., Knoll, F., Sriram, A., Muckley, M.J., Bruno, M., Defazio, A., Parente, M., Geras, K.J., Katsnelson, J., Chandarana, H., Zhang, Z., Drozdzal, M., Romero, A., Rabbat, M., Vincent, P., Pinkerton, J., Wang, D., Yakubova, N., Owens, E., Zitnick, C.L., Recht, M.P., Sodickson, D.K., Lui, Y.W.: fastMRI: An Open Dataset and Benchmarks for Accelerated MRI. In: arXiv preprint (2018) 3, 9
41. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising. In: IEEE Transactions on Image Processing (2017) 1, 4, 6
42. Zhang, Y., Zhu, Y., Nichols, E., Wang, Q., Zhang, S., Smith, C., Howard, S.: A poisson-gaussian denoising dataset with real fluorescence microscopy images. In: CVPR (2019) 2, 9