GAN2GAN: GENERATIVE NOISE LEARNING FOR BLIND DENOISING WITH SINGLE NOISY IMAGES

Sungmin Cha\textsuperscript{1}, Taeelon Park\textsuperscript{1}, Byeongjoon Kim\textsuperscript{2}, Jongduk Baek\textsuperscript{2} and Taesup Moon\textsuperscript{3}\textsuperscript{*}
Sungkyunkwan University\textsuperscript{1}, Yonsei University\textsuperscript{2}, Seoul National University\textsuperscript{3}, South Korea
\{csm9493,ptel1236\}@skku.edu, bjkim2006@naver.com, jongdukbaek@yonsei.ac.kr, tsmoon@snu.ac.kr

ABSTRACT

We tackle a challenging blind image denoising problem, in which only single distinct noisy images are available for training a denoiser, and no information about noise is known, except for it being zero-mean, additive, and independent of the clean image. In such a setting, which often occurs in practice, it is not possible to train a denoiser with the standard discriminative training or with the recently developed Noise2Noise (N2N) training; the former requires the underlying clean image for the given noisy image, and the latter requires two independently realized noisy image pair for a clean image. To that end, we propose GAN2GAN (Generated-Artificial-Noise to Generated-Artificial-Noise) method that first learns a generative model that can 1) simulate the noise in the given noisy images and 2) generate a rough, noisy estimates of the clean images, then 3) iteratively trains a denoiser with subsequently synthesized noisy image pairs (as in N2N), obtained from the generative model. In results, we show the denoiser trained with our GAN2GAN achieves an impressive denoising performance on both synthetic and real-world datasets for the blind denoising setting: it almost approaches the performance of the standard discriminatively-trained or N2N-trained models that have more information than ours, and it significantly outperforms the recent baseline for the same setting, e.g., Noise2Void, and a more conventional yet strong one, BM3D. The official code of our method is available at \url{https://github.com/csm9493/GAN2GAN}.

1 INTRODUCTION

Image denoising is one of the oldest problems in image processing and low-level computer vision, yet it still attracts lots of attention due to the fundamental nature of the problem. A vast number of algorithms have been proposed over the past several decades, and recently, the CNN-based methods, \textit{e.g.}, [Cha & Moon (2019); Zhang et al. (2017); Tai et al. (2017); Liu et al. (2018)], became the throne-holders in terms of the PSNR performance. The main approach of the most CNN-based denoisers is to apply the discriminative learning framework with (clean, noisy) image pairs and known noise distribution assumption. While being effective, such framework also possesses a couple of limitations that become critical in practice; the assumed noise distribution may be mismatched to the actual noise in the data or obtaining the noise-free clean target images is not always possible or very expensive, \textit{e.g.}, medical imaging (CT or MRI) or astrophotographs.

Several attempts have been made to resolve above issues. For the noise uncertainty, the so-called \textit{blind training} have been proposed. Namely, a denoiser can be trained with a composite training set that contains images corrupted with multiple, pre-defined noise levels or distributions, and such blindly trained denoisers, \textit{e.g.}, DnCNN-B in [Zhang et al. (2017)], were shown to alleviate the mismatch scenarios to some extent. However, the second limitation, \textit{i.e.}, the requirement of clean images for building the training set, still remains. As an attempt to address this second limitation, [Lehtinen et al. (2018)] recently proposed the Noise2Noise (N2N) method. It has been shown that a denoiser, which has a negligible performance loss, can be trained without the clean target images, as long as two independent noisy image realizations for the same underlying clean image are available. Despite its

\textsuperscript{*}Corresponding author (E-mail: tsmoon@snu.ac.kr)
effectiveness, the requirement of the two independently realized noisy image pair for a single clean image, which may hardly be available in practice, is a critical limiting factor for N2N.

In this paper, we consider a setting in which neither of above approach is applicable, namely, the pure unsupervised blind denoising setting where only single distinct noisy images are available for training. Namely, nothing is known about the noise other than it being zero-mean, additive, and independent of the clean image, and neither the clean target images for blind training nor the noisy image pairs for N2N training is available. While some recent work, e.g., [Krull et al. (2019); Batson & Royer (2019); Laine et al. (2019)], took the self-supervised learning (SSL) approach for the same setting, we take a generative learning approach. The crux of our method is to first learn a Wasserstein GAN [Arjovsky et al. (2017)]-based generative model that can 1) learn and simulate the noise in the given noisy images and 2) generate rough, initially denoised images. Using such generative model, we then synthesize noisy image pairs by corrupting each of the initially denoised images with the simulated noise twice and use them to train a CNN denoiser as in the N2N training (i.e., Noisy N2N). We further show that iterative N2N training with refined denoised images can significantly improve the final denoising performance. We dubbed our method as GAN2GAN (Generated-Artificial-Noise to Generated-Artificial-Noise) and show that the denoiser trained with our method can achieve (sometimes, even outperform) the performance of the standard supervised-trained or N2N-trained blind denoisers for the white Gaussian noise case. Furthermore, for mixture/correlated noise or real-world noise in microscopy/CT images, for which the exact distributions are hard to know a priori, we show our denoiser significantly outperforms those standard blind denoisers, which are mismatch-trained with white Gaussian noise, as well as other baselines that operate in the same condition as ours: the SSL baseline, N2V [Krull et al. 2019], and a more conventional BM3D [Dabov et al. (2007)].

2 RELATED WORK

Several works have been proposed to overcome the limitation of the vanilla supervised learning based denoising. As mentioned above, Noise2Self (N2S) [Batson & Royer, 2019] and Noise2Void (N2V) [Krull et al. 2019] recently applied self-supervised learning (SSL) approach to train a denoiser only with single noisy images. Their settings exactly coincide with ours, but we show later that our GAN2GAN significantly outperforms them. More recently, Laine et al. (2019) improved N2V by incorporating specific noise likelihood models with Bayesian framework, however, their method required to know the exact noise model and could not be applied to more general, unknown noise settings. Similarly, Soltanayev & Chun (2018) proposed SURE (Stein’s Unbiased Risk Estimator)-based denoiser that can also be trained with single noisy images, but it worked only with the Gaussian noise. Their work was extended in Zhussip et al. (2019), but it required noisy image pairs as in N2N as well as the Gaussian noise constraint. Chen et al. (2018) devised GCBD method to learn and generate noise in the given noisy images using W-GAN [Arjovsky et al. (2017)] and utilized the unpaired clean images to build a supervised training set. Our GAN2GAN is related to Chen et al. (2018), but we significantly improve their noise learning step and do not use the clean data at all. Table 1 summarizes and compares the settings among the above mentioned recent baselines. We clearly see that only our GAN2GAN and N2V do not utilize any “sidekicks” that other methods use.

Table 1: Summary of different settings among the recent baselines.

<table>
<thead>
<tr>
<th>Alg.</th>
<th>Requirements</th>
<th>Clean image</th>
<th>Noisy “pairs”</th>
<th>Noise model</th>
</tr>
</thead>
<tbody>
<tr>
<td>N2N [Lehtinen et al. (2018)]</td>
<td>X</td>
<td>✓</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>HQ SSL [Laine et al. (2019)]</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>SURE [Soltanayev &amp; Chun (2018)]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Ext. SURE [Zhussip et al. (2019)]</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>GCBD [Chen et al. (2018)]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>N2V [Krull et al. (2019)]</td>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td></td>
</tr>
<tr>
<td>GAN2GAN (Ours)</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

Additionally, there are recently published papers on blind image denoising but these also have a difference with ours. Anwar & Barnes (2019); Zhang et al. (2018) suggest effective CNN architectures for denoising, however, they only consider the setting in which clean images are necessary for training. Zamir et al. (2020) considers the denoising of specific camera settings, and it also requires clean sRGB images as well as the knowledge of the noise level. Thus, it cannot be applied to the complete blind setting as ours, in which no information on the specific noise distribution or clean images is available.
More classical denoising methods are capable of denoising solely based on the single noisy images by applying various principles, e.g., filtering-based [Buades et al. (2005), Dabov et al. (2007), optimization-based [Elad & Aharon (2006), Mairal et al. (2009), Wavelet-based [Donoho & Johnstone (1995), and effective prior-based [Zoran & Weiss (2011)]. Those methods typically are, however, computationally intensive during the inference time and cannot be trained from a separate set of noisy images, which limits their denoising performance. Another line of recent work worth mentioning is the deep learning-based priors or regularizers, e.g., [Ulyanov et al. (2018), Yeh et al. (2018); Lunz et al. (2018), but their PSNRs still fell short of the supervised trained CNN-based denoisers.

3 Motivation

In order to develop the core intuition for motivating our method, we first consider a simple, single-letter Gaussian noise setting. Let \( Z = X + N \) be the noisy observation of \( X \sim \mathcal{N}(0, \sigma_X^2) \), corrupted by the \( N \sim \mathcal{N}(0, \sigma_N^2) \). It is well known that the minimum MSE (MMSE) estimator of \( X \) given \( Z \) is \( f_{\text{MMSE}}(Z) = \mathbb{E}(X|Z) = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_N^2} Z \). We now identify the optimality of N2N in this setting.

**N2N** Assume that we have two i.i.d. copies of the noise \( N \): \( N_1 \) and \( N_2 \). Then, let \( Z_1 = X + N_1 \) and \( Z_2 = X + N_2 \) be the two independent noisy observation pairs of \( X \). The N2N in this setting corresponds to obtaining the MMSE estimator of \( Z_2 \) given \( Z_1 \),

\[
f_{\text{N2N}}(Z_1) \triangleq \arg \min_f \mathbb{E}(Z_2 - f(Z_1))^2 = \mathbb{E}(Z_2|Z_1) = \mathbb{E}(X + N_2|Z_1) \overset{(a)}{=} \mathbb{E}(X|Z_1) = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_N^2} Z_1,
\]

in which (a) follows from \( N_2 \) being independent of \( Z_1 \). Note (1) has the exact same form as \( f_{\text{MMSE}}(Z) \), hence, estimating \( X \) with \( f_{\text{N2N}}(Z) \) also achieves the MMSE, in line with [Lehtinen et al. (2018)].

**“Noisy” N2N** Now, consider the case in which we again have the two i.i.d. \( N_1 \) and \( N_2 \), but the noisy observations are of a noisy version of \( X \). Namely, let \( X' = X + N_0 \), in which \( N_0 \sim \mathcal{N}(0, \sigma_0^2) \), and denote \( Z'_1 = X' + N_1 \) and \( Z'_2 = X' + N_2 \) as the noisy observation pairs. Then, we can define a “Noisy” N2N estimator as the MMSE estimator of \( Z'_2 \) given \( Z'_1 \),

\[
f_{\text{Noisy N2N}}(Z'_1, y) \triangleq \arg \min_f \mathbb{E}(Z'_2 - f(Z'_1))^2 = \mathbb{E}(X'|Z'_1) = \frac{\sigma_X^2(1 + y)}{\sigma_X^2(1 + y) + \sigma_N^2} Z'_1,
\]

in which we denote \( y \triangleq \sigma_0^2/\sigma_X^2 \) and assume that \( 0 \leq y < 1 \). Note clearly (2) coincides with (1) when \( y = \sigma_0^2 = 0 \). Following N2N, (2) is essentially estimating \( X' \) based on \( Z' = X' + N \). An interesting subtle question is what happens when we use the mapping \( f_{\text{Noisy N2N}}(Z, y) \) for estimating \( X \) given \( Z = X + N \), not \( X' \) given \( Z' \). Our theorem below, of which proof is in the Supplementary Material (S.M.), shows that for a sufficiently large \( \sigma_0^2 \), \( f_{\text{Noisy N2N}}(Z, y) \) gives a better estimate of \( X \) than \( X' \).

**Theorem 1** Consider the single-letter Gaussian setting and \( f_{\text{Noisy N2N}}(Z, y) \) obtained in (2). Also, assume \( 0 < y < 1 \). Then, there exists some \( y_0 \) s.t. \( \forall y \in (y_0, 1), \mathbb{E}(X - f_{\text{Noisy N2N}}(Z, y))^2 < \sigma_0^2 \).

Theorem [1] provides a simple, but useful, intuition that motivates our method; if simulating the noise in the images is possible, we may carry out the N2N training iteratively, provided that a rough noisy estimate of the clean image is initially available. Namely, we can first simulate the noise to generate noisy observation pairs of the initial noisy estimate, then do the Noisy N2N training with them to obtain a denoiser that may result in a better estimate of the clean image when applied to the actual noisy image subject to denoising (as in Theorem [1]). Then, we can refine the estimates by iterating the Noisy N2N training with the generated noisy observation pairs of the previous step’s estimate of the clean image, until convergence.

To check whether above intuition is valid, we carry out a feasibility experiment. Figure [1] shows the denoising results on BSD68 [Roth & Black (2009)] for Gaussian noise with \( \sigma = 25 \). The blue line is the PSNR of the N2N model trained on the cleaned images, and the red line is the PSNR of the Noisy N2N model trained on the noisy images.
model trained with noisy observation pairs of the clean images in the BSD training set, setting as an upper bound. The orange line, in contrast, is the PSNR of the Noisy N2N model that is trained with the noisy observation pairs of the noisy estimates for the clean images, which were set to be another Gaussian noise-corrupted training images. The standard deviations ($\sigma_0$) of the Gaussian for generating the noisy estimates are given in the horizontal axis, and the corresponding PSNRs of the estimates are given in the parentheses. Although Noisy N2N clearly lies much lower than the N2N upper bound, we note its PSNR is still higher than that of the initial noisy estimates, which is in line with Theorem 1. Now, if we iterate the Noisy N2N with the previous step’s denoised images (i.e., Noisy-N2N/Noisy-N2N for second/third iterations, respectively), we observe that the PSNR significantly improves and approaches the ordinary N2N for most of the initial $\sigma_0$ values. Thus, we observe the intuition from Theorem 1 generalizes well to the image denoising case in an ideal setting, where the noise can be perfectly simulated, and the initial noisy estimates are Gaussian corrupted versions. The remaining question is whether we can also obtain similar results for the blind image denoising setting. We show our generative model-based approach in details in the next section.

4 Main Method: Three Components of GAN2GAN

To concretely describe our method, we first set the notations. We assume the noisy image $Z$ is generated by $Z = x + N$, in which $x$ denotes the underlying clean image and $N$ denotes the zero-mean, additive noise that is independent of $x$. For training a denoiser, we do not assume either the distribution or the covariance of $N$ is known. Moreover, we assume only a database of $n$ distinct noisy images, $D = \{Z(i)\}_{i=1}^n$, is available for learning a denoiser. A CNN-based denoiser is denoted as $X_\phi(Z)$ with $\phi$ being the model parameter, and we use the standard quality metrics, PSNR/SSIM, for evaluation. Our method consists of three parts: 1) smooth noisy patch extraction, 2) training a generative model, and 3) iterative GAN2GAN training of $X_\phi(Z)$, each of which we elaborate below.

4.1 Smooth Noisy Patch Extraction

The first step is to extract the noisy image patches from $D$ that correspond to smooth, homogeneous areas. Our extraction method is similar to that of the GCBD proposed in (Chen et al., 2018), but we make a critical improvement. The GCBD determines a patch $p$ (of pre-determined size) is smooth if it satisfies the following for all of its smaller sub-patches, $q_j$, with some hyperparameters $\mu, \gamma \in (0, 1)$:

$$||E(q_j) - E(p)|| \leq \mu E(p), \quad ||V(q_j) - V(p)|| \leq \gamma V(p),$$

in which $E(\cdot)$ and $V(\cdot)$ are the empirical mean and variance of the pixel values.

![Figure 2: Comparison of smooth noisy patch extraction rules.](image)

(a) Histograms of empirical $\hat{\sigma}$.  
(b) Examples of extracted noise patches ($\sigma = 25$)

While (3) works for extracting smooth patches to some extent, as we show in Figure 2(b), it does not rule out choosing patches with high-frequency repeating patterns, which are far from being smooth. Thus, we instead use the 2D discrete wavelet transform (DWT) for a new extraction rule; namely, we determine $p$ is smooth if its four sub-band decompositions obtained by DWT, $\{W_k(p)\}_{k=1}^4$, satisfy

$$\frac{1}{4} \sum_{k=1}^4 \left| \hat{\sigma}(W_k(p)) - E[\hat{\sigma}_W(p)] \right| \leq \lambda E[\hat{\sigma}_W(p)],$$

in which $\hat{\sigma}(\cdot)$ is the empirical standard deviation of the wavelet coefficients, $E[\hat{\sigma}_W(p)] \triangleq \frac{1}{4} \sum_{k=1}^4 \hat{\sigma}(W_k(p))$, and $\lambda \in (0, 1)$ is a hyperparameter. This rule is much simpler than (3), which
The model architecture with three generators and two critics.

The two plots in Figure 2(a) show the normalized histograms of the empirical standard deviations, $\hat{\sigma}$, of the extracted patches when true $\sigma = \{25, 50\}$, respectively. We clearly observe that while the $\hat{\sigma}$’s for (4) are mostly concentrated on true $\sigma$, those of (3) have much higher variation. In addition, Figure 2(b) visualizes the randomly sampled patches of which $\hat{\sigma}$’s were above the 90-th percentile among the extracted patches for each rule (when $\sigma = 25$). Again, it is obvious that (4) also may result in selecting the patches with high-frequency patterns, whereas (3) is much more effective for extracting accurate noise patches. Later, we show (in Figure 5) that such improved quality of the noise patches by our (4) plays an essential role; namely, our pure unsupervised learning based denoiser using (4) even outperforms the clean target image based denoiser in [Chen et al., 2018] using (3).

4.2 Training a W-GAN based generative model

Equipped with $D = \{Z^{(i)}\}_{i=1}^{n}$ and the extracted noise patches $N = \{n^{(j)}\}_{j=1}^{N}$, we train a generative model, which can learn and simulate the noise as well as generate initial noisy estimates of the clean images, hence, realize the Noisy N2N training explained in Section 3. As shown in Figure 3, our model has three generators, $\{g_1, g_2, g_3\}$, and two critics, $\{f_{w_1}, f_{w_2}\}$, in which the subscripts stand for the model parameters. The loss functions associated with the components of our model are:

$$L_n(\theta_1, w_1) \triangleq E_n [f_{w_1}(n)] - E_r [f_{w_1}(g_\theta_1(r))]$$

$$L_z(\theta_1, \theta_2, w_2) \triangleq E_z [f_{w_2}(Z)] - E_{z,r} [f_{w_2}(g_\theta_1(Z) + g_\theta_2(r))]$$

$$L_{\text{cyc}}(\theta_2, \theta_3) \triangleq E_Z [\|z - g_\theta_3(g_\theta_2(Z))\|_1].$$

The loss [5] is a standard W-GAN (Arjovsky et al., 2017) loss for training the first generator-critic pair, $(g_\theta_1, f_{w_1})$, of which $g_\theta_1$ learns to generate the independent realization of the noise mimicking the patches in $N = \{n^{(j)}\}_{j=1}^{N}$, taking the random vector $r \sim N(0, I)$ as input. The second loss [6] links the two generators, $g_\theta_1$ and $g_\theta_2$, with the second critic, $f_{w_2}$. The second generator $g_\theta_2$ is intended to generate the estimate of the underlying clean patch for $Z$, i.e., coarsely denoise $Z$, and the critic $f_{w_2}$ determines how close the distribution of the generated noisy image, $g_\theta_2(Z) + g_\theta_1(r)$, is to the that of $Z$.

Our intuition is, if $g_\theta_1$ can realistically simulate the noise, then enforcing $g_\theta_2(Z) + g_\theta_1(r)$ to mimic $Z$ would result in learning a reasonable initial denoiser $g_\theta_2$. One important detail regarding $g_\theta_2$ is its final activation must be the sigmoid function for stable training. The third loss [7], which

---

1We assume $g_\theta_2$ implicitly has the cropping step for $Z$ such that the dimension of $g_\theta_2(Z)$ and $g_\theta_1(r)$ match.
resembles the cycle loss in (Zhu et al., 2017), imposes the encoder-decoder structure between \( g_\theta \),
and \( g_\theta \), hence, helps \( g_\theta \) to compress the most redundant part of \( Z \), i.e., the noise, and carry out
the initial denoising. Once the losses are defined, training the generators and critics are done in an
alternating manner, as in the training of W-GAN (Arjovsky et al., 2017), to approximately solve
\[
\min_{\theta_1, \theta_2, \theta_3} \max_{w_1, w_2} \left[ \alpha L_n(\theta_1, w_1) + \beta L_Z(\theta_1, \theta_2, w_2) + \gamma L_{\text{cyc}}(\theta_2, \theta_3) \right],
\] (8)
in which \((\alpha, \beta, \gamma)\) are hyperparameters to control the trade-offs between the loss functions.
The pseudo algorithm for training a generative model is given in Algorithm 1. There are a couple of
subtle points for training with the overall objective (8), and we describe the full details on model
architectures and hyperparameters in the S.M.

Algorithm 1 Training a generative model, all experiments in this paper used the defaults values,
\( n_{\text{critic}} = 5, n_{\text{epoch}} = 30, m = 64, \alpha_g = 4 e^{-4}, \alpha_{\text{critic}} = 5 e^{-5}, \alpha = 5, \beta = 1, \gamma = 10 \)

1: Require \( \mathcal{D}, \lambda \)
2: \( \mathcal{N} \leftarrow \text{NoisePatchExtraction}(\mathcal{D}, \lambda) \)
3: for \( c\text{PGAN} \leftarrow 1, n_{\text{epoch}} \) do
4: \( \text{Sample } \{n(i)\}_{i=1}^m \sim \mathcal{N}, \{r(i)\}_{i=1}^m \sim N(0, I), \{Z(i)\}_{i=1}^m \sim \mathcal{D} \)
5: for \( c_{\text{critic}} \leftarrow 1, n_{\text{critic}} \) do
6: \( g_1 \leftarrow \nabla_{\theta_1, \theta_2, \theta_3} \alpha L_n(\theta_1, w_1) + \beta L_Z(\theta_1, \theta_2, w_2) + \gamma L_{\text{cyc}}(\theta_2, \theta_3) \)
7: \( \theta_1 \leftarrow \theta_1 - \alpha_g \text{Adam}(\theta_1, g_1) \), \( \theta_2 \leftarrow \theta_2 - \alpha_g \text{Adam}(\theta_2, g_2) \), \( \theta_3 \leftarrow \theta_3 - \alpha_{\text{critic}} \text{Adam}(\theta_3, g_3) \)
8: end for
9: \( \theta_1, \theta_2 \leftarrow \nabla_{\theta_3} \alpha L_n(\theta_1, w_1) + \beta L_Z(\theta_1, \theta_2, w_2) + \gamma L_{\text{cyc}}(\theta_2, \theta_3) \)
10: \( \theta_1, \theta_2 \leftarrow \nabla_{\theta_3} \alpha L_n(\theta_1, w_1) + \beta L_Z(\theta_1, \theta_2, w_2) + \gamma L_{\text{cyc}}(\theta_2, \theta_3) \)
11: end for
12: return \( \theta_1, \theta_2 \)

4.3 Iterative GAN2GAN training of a denoiser

With our generative model, we then carry out the iterative Noisy N2N training as described in Section
3 with the generated noisy images. Namely, given each \( Z(i) \in \mathcal{D} \), we generate the pair
\[
(Z_{11}(i), Z_{12}(i)) \triangleq (g_{\theta_1}(Z(i)) + g_{\theta_2}(r_{11}(i), g_{\theta_2}(Z(i)) + g_{\theta_2}(r_{12}(i))),
\] (9)
in which \( r_{11}(i), r_{12}(i) \in \mathbb{R}^{128} \) are i.i.d. \( \sim N(0, I) \). In contrast to the ideal case in Section 3 each generated image in (9) is a noise-corrupted version of \( g_{\theta_2}(Z(i)) \), in which the corruption is done by the
simulated noise \( g_{\theta_1}(r) \). Denoting the set of such pairs as \( \mathcal{D}_1 = \{Z(i)_{11}, Z(i)_{12}\}_{i=1}^n \), a denoiser \( \hat{X}_\phi(Z) \) is trained by minimizing \( L_{\text{G2G}}(\phi, \mathcal{D}_1) \triangleq \frac{1}{n} \sum_{i=1}^n (Z(i)_{11} - \hat{X}_\phi(Z(i)_{12}))^2 \). In \( \text{G2G(C)} \), we only use the generated noisy images and do not use the actual observed \( Z(i) \), hence, we dubbed our training as GAN2GAN (G2G) training. Now, denoting the learned denoiser as \( G2G_1 \) (with parameter \( \phi_1 \)), we can iterate the G2G training. For the \( j \)-th iteration (with \( j \geq 2 \)), we generate
\[
(Z_{j1}(i), Z_{j2}(i)) \triangleq (\hat{X}_{\phi_{j-1}}(Z(i)) + g_{\theta_2}(r_{j1}(i), \hat{X}_{\phi_{j-1}}(Z(i)) + g_{\theta_2}(r_{j2}(i))),
\] (10)
for each \( Z(i) \) and denote the resulting set of the pairs as \( \mathcal{D}_j \). Note in (10), we update the noisy estimate of the clean image with the output of \( G2G_{j-1} \). Then, the new denoiser \( G2G_{j} \) is obtained by computing \( \phi_j \triangleq \arg \min_{\phi} L_{\text{G2G}}(\phi, \mathcal{D}_j) \), where the minimization is done via warm-starting from \( \phi_{j-1} \). In our experiments, we show the sequence, \( G2G_{j \geq 1} \), successively refines the denoising quality and significantly improves the initial noisy estimate, similarly as in Figure 4. Moreover, we identify the benefit of the iterative G2G training becomes greater when noise is more sophisticated; i.e.,
for synthetic noise, the performance of \( G2G_{j \geq 1} \) converges after \( 1 \sim 3 \) iterations, whereas for the
real-world microscopy noise, the performance keeps increasing until larger number of iterations.

5 Experimental results

5.1 Data and experimental settings

Data & training details In synthetic noise experiments, we always used the noisy training images
from BSD400 (Martin et al., 2001). For evaluation, we used the standard BSD68 (Roth & Black...
Table 3: Results on BSD68/Mixture & Correlated noise. The boldface and colored texts are as before.
10% $\sim$ Unif$[-s, s]$ which means the random variable that is uniformly distributed between $[-s, s]$ with $s = 15, 25$. For case B, we tested with larger variances, i.e., 70% Gaussian $N(0, 15^2)$, 20% Gaussian $N(0, 25^2)$, and 10% Uniform $[-s, s]$ with $s = 30, 50$. For correlated noise, we generated the following noise for each $\ell$-th pixel,

$$N_\ell = \eta M_\ell + (1 - \eta)\left(\frac{1}{\sqrt{|NB_\ell|}} \sum_{m \in NB_\ell} M_m\right), \quad \ell = 1, 2, \ldots$$

in which $\{M_\ell\}$ are white Gaussian $N(0, \sigma^2)$, $NB_\ell$ is the $k \times k$ neighborhood patch except for the pixel $\ell$, and $\eta$ is a mixture parameter. We set $\eta = 1/\sqrt{2}$ such that the marginal distribution of $N_\ell$ is also $\mathcal{N}(0, \sigma^2)$ and set $k = 16$. Note in this case, $N_\ell$ has a spatial correlation, and we tested with $\sigma = 15, 25$. From the table, we first note that DnCNN-B and N2N suffer from serious performance degradation for both mixture and correlated noises due to noise mismatch, and the conventional BM3D outperforms them for some cases (e.g., Case A for mixture noise). However, we note our G2G can still denoise very well after just two iterations and outperforms all the baselines for all noise types. Note N2V seriously suffers and is not comparable to ours. Finally, N2C(Eq. (4)) is a sound upper bound for all noise types, confirming the correctness of the extraction rule (4).

### 5.3 Denoising results on real noise

We also test our method on the real-world noise. While some popular real noise is known to have source-dependent characteristics, there are also cases in which the noise is source-independent and pixel-wise correlated, which satisfies the assumption of our method. We tested on two such datasets, the Wide-Focal (WF) set in the microscopy image dataset [Zhang et al. 2019] and a Reconstructed CT dataset. A more detailed description and analysis on these two datasets are in S.M. The WF and Reconstructed CT data has 5 sets (Avg = 1, 2, 4, 8, 16) and 4 sets (Dose=25, 50, 75, 100) with different noise levels, respectively. We did not exploit the fact that the images are multiple noisy measurements of a clean image, which enables employing N2N, but treated them as noisy images of distinct clean images. Figure 4(a) and 4(b) shows the PSNR of all methods for each dataset, respectively, averaged over all sets. The baselines were DnCNN-B, BM3D and N2V. Note for G2G, we iterated until $G2G_3$ and N2C(Eq. (4)) was an upper bound for each set. We clearly observe that the performance of G2G, significantly improves (over $g_{th}$) as the iteration continues. In results, G2G3 becomes significantly better than DnCNN-B and N2V as well as BM3D, still one of the strongest baselines for real-world noise denoising when no clean target images are available, for both datasets. We report more detailed experimental results (including SSIM) on both datasets in S.M. Moreover, the inference time for BM3D is about 4.5~5.0 seconds per image since a noise estimation has to be done for each image separately, whereas that for G2G, is only 4 ms (on GPU), which is another significant advantage of our method. Figure 4(c) shows the visualizations on the WF, and we give more examples in the S.M.

### 5.4 Ablation study

**Noise patch extraction** Here, we evaluate the effect of the noisy patch extraction rules (3) and (4) in the final denoising performance. Figure 5 compares the PSNR of N2C(GCBD Eq.(3)), a re-implementation of [Chen et al. 2018], N2C(Ours Eq. (4)) and the best G2G, for each dataset.
We note neither source code nor training data of (Chen et al., 2018) is publicly available, and the PSNR in (Chen et al., 2018) could not be reproduced (with the exact same η and γ as in (Chen et al., 2018)). From the figure, we clearly observe the significant gap between N2C(Our Eq.(4)) and N2C(GCBD Eq.(3)), particularly when the noise is not white Gaussian. Moreover, our pure unsupervised G2G with (4) even outperforms N2C(GCBD Eq.(3)) that utilizes the clean target images, confirming the quality difference shown in Figure 2(b) significantly affects learning noise and a denoiser.

**Generative model and iterative G2G training** Figure 6(a) shows the PSNRs of $g_{θ_2}$ on BSD68/Gaussian($σ = 25$) trained with three variations; “No $L_Z$” for no $f_{w_2}$, “No $L_{cyc}$” for no (7) and $g_{θ_1}$, and “No sigmoid” for no sigmoid activation at the output layer of $g_{θ_2}$. We confirm that our proposed architecture achieves the highest PSNR for $g_{θ_2}$, the sigmoid activation and $f_{w_2}$ are essential, and the cycle loss (7) is also important. Achieving a decent PSNR for $g_{θ_2}$ is beneficial for saving the number of G2G iterations and achieving high final PSNR. More detailed analyses on the generative model architecture are in the S.M. Figure 6(b) and 6(c) show the effect of the quality of the initial estimate for the iterative G2G training. From Figure 1, one may ask whether $g_{θ_2}$ is indeed necessary, since even when $c_0 ≈ σ$, the iterating the Noisy N2N can mostly achieve the upper bound. Hence, for samples of synthetic and real microscopy data, we evaluate how G2G performs when the iteration simply starts with $Z$. Figure 6(b) shows a somewhat surprising result that for synthetic noises, starting from $Z$ achieves essentially the same performance as starting from $g_{θ_2}$ with a couple more G2G iterations. However, for real microscopy noise case in Figure 6(c) WF(Avg = 1) in which starting from $Z$ achieves far lower performance than starting from $g_{θ_2}$, justifying our generative model for attaining the initial noisy estimate.

6  **Concluding Remark**

Motivated by a novel observation on Noisy N2N, we proposed a novel GAN2GAN method, which can tackle the challenging blind image denoising problem solely with single noisy images. Our method showed impressive denoising performance that even sometimes outperform the methods with more information as well as VST+BM3D for real noise denoising. As a future work, we plan to extend our framework to more explicitly handle the source-dependent real-world noise.

**Acknowledgment**

This work was supported in part by NRF Mid-Career Research Program [NRF-2021R1A2C2007884] and IITP grant [No.2019- 0-01396, Development of framework for analyzing, detecting, mitigating of bias in AI model and training data], funded by the Korean government (MSIT).
REFERENCES


