Early Stopping for Deep Image Prior

Hengkang Wang *wang9881@umn.edu*
University of Minnesota, Twin Cities

Taihui Li *lixx5027@umn.edu*
University of Minnesota, Twin Cities

Zhong Zhuang *zhuan143@umn.edu*
University of Minnesota, Twin Cities

Tiancong Chen *chen6271@umn.edu*
University of Minnesota, Twin Cities

Hengyue Liang *liang656@umn.edu*
University of Minnesota, Twin Cities

Ju Sun *jusun@umn.edu*
University of Minnesota, Twin Cities

Reviewed on OpenReview: *https://openreview.net/forum?id=231ZzrLC8X*

Abstract

Deep image prior (DIP) and its variants have shown remarkable potential to solve inverse problems in computational imaging, *needing no separate training data.* Practical DIP models are often substantially overparameterized. During the learning process, these models first learn the desired visual content and then pick up potential modeling and observational noise, i.e., performing early learning then overfitting (ELTO). Thus, the practicality of DIP hinges on early stopping (ES) that can capture the transition period. In this regard, most previous DIP works for computational imaging tasks only demonstrate the potential of the models, reporting peak performance against ground truth, but providing no clue about how to operationally obtain near-peak performance *without access to ground truth.* In this paper, we set out to break this practicality barrier of DIP and propose an effective ES strategy that consistently detects near-peak performance in various computational imaging tasks and DIP variants. Simply based on the running variance of DIP intermediate reconstructions, our ES method not only outpaces the existing ones—which only work in very narrow regimes, but also remains effective when combined with methods that try to mitigate overfitting. The code to reproduce our experimental results is available at `https://github.com/sun-umn/Early_Stopping_for_DIP`.

1 Introduction

Inverse problems (IPs) are prevalent in computational imaging, ranging from basic image denoising, super-resolution, and deblurring, to advanced 3D reconstruction and major tasks in scientific and medical imaging (Szeliski, 2022). Despite the disparate settings, all these problems take the form of recovering a visual object $\boldsymbol{x}$ from $\boldsymbol{y} = f(\boldsymbol{x})$, where f models the forward physical process to obtain the observation $\boldsymbol{y}$. Typically, these visual IPs are not determined in a unique way: $\boldsymbol{x}$ cannot be determined uniquely from $\boldsymbol{y}$. This is exacerbated by potential modeling (e.g., linear f to approximate a nonlinear process) and observational (e.g., Gaussian or shot) noise, i.e., $\boldsymbol{y} \approx f(\boldsymbol{x})$. To overcome nonuniqueness and improve noise stability, researchers often encode a variety of problem-specific priors on $\boldsymbol{x}$ when formulating IPs.

Traditionally, IPs are phrased as regularized data fitting problems:

$$\min_{\boldsymbol{x}} \ \ell(\boldsymbol{y}, f(\boldsymbol{x})) + \lambda R(\boldsymbol{x}) \qquad \ell(\boldsymbol{y}, f(\boldsymbol{x})) : \text{data-fitting loss}, \ R(\boldsymbol{x}) : \text{regularizer} \tag{1}$$

where λ is the regularization parameter. Here, the loss ℓ is often chosen according to the noise model, and the regularizer R encodes priors on $\boldsymbol{x}$. The advent of deep learning has revolutionized the way IPs are solved. On the radical side, deep neural networks (DNNs) are trained to directly map any given $\boldsymbol{y}$ to an $\boldsymbol{x}$; on the mild side, pre-trained or trainable deep learning models are taken to replace certain nonlinear mappings in iterative numerical algorithms for solving Eq. (1) (e.g. plug-and-play and algorithm unrolling); see recent surveys Ongie et al. (2020); Janai et al. (2020) on these developments. All of these deep-learning-based methods rely on large training sets to adequately represent the underlying priors and/or noise distributions. **This paper concerns another family of striking ideas that do not require separate training data**.

Deep image prior (DIP) Ulyanov et al. (2018) proposes parameterizing $\boldsymbol{x}$ as $\boldsymbol{x} = G_{\boldsymbol{\theta}}(\boldsymbol{z})$, where $G_{\boldsymbol{\theta}}$ is a trainable DNN parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{z}$ is a frozen or trainable random seed. **No separate training data other than $\boldsymbol{y}$ are used!** Plugging the reparametrization into Eq. (1), we obtain

$$\min_{\boldsymbol{\theta}} \ \ell(\boldsymbol{y}, f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})) + \lambda R \circ G_{\boldsymbol{\theta}}(\boldsymbol{z}). \tag{2}$$

$G_{\boldsymbol{\theta}}$ is often "overparameterized"—containing substantially more parameters than the size of $\boldsymbol{x}$, and "structured"—e.g., consisting of convolution networks to encode structural priors in natural visual objects. The resulting optimization problem is solved using standard first-order methods (e.g., (adaptive) gradient descent). When $\boldsymbol{x}$ has multiple components with different physical meanings, one can naturally parametrize $\boldsymbol{x}$ using multiple DNNs. This simple idea has led to surprisingly competitive results in numerous visual IPs, from low-level image denoising, super-resolution, inpainting (Ulyanov et al., 2018; Heckel & Hand, 2019; Liu et al., 2019) and blind deconvolution (Ren et al., 2020; Wang et al., 2019; Asim et al., 2020; Tran et al., 2021; Zhuang et al., 2022a), to mid-level image decomposition and fusion (Gandelsman et al., 2019; Ma et al., 2021), and to advanced computational imaging problems (Darestani & Heckel, 2021; Hand et al., 2018; Williams et al., 2019; Yoo et al., 2021; Baguer et al., 2020; Cascarano et al., 2021; Hashimoto & Ote, 2021; Gong et al., 2022; Veen et al., 2018; Tayal et al., 2021; Zhuang et al., 2022b); see the survey Qayyum et al. (2021).

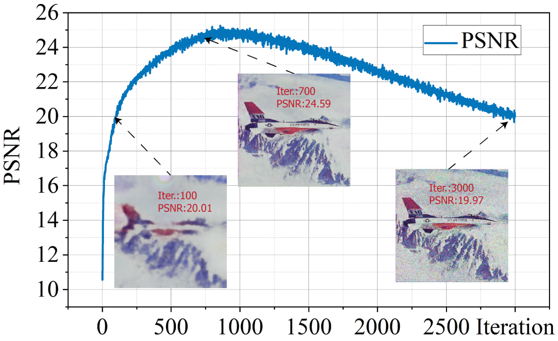


Figure 1: The "early-learning-then-overfitting" (ELTO) phenomenon in DIP for image denoising. The quality of the estimated image climbs first to a peak and then drops once the noise is picked up by the model $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ also.

Overfitting issue in DIP A critical detail that we have glossed over is **overfitting**. Since $G_{\boldsymbol{\theta}}$ is often substantially overparameterized, $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ can represent arbitrary elements in the $\boldsymbol{x}$ domain. Global optimization of equation 2 would normally lead to $\boldsymbol{y} = f \circ G_{\boldsymbol{\theta}}(\boldsymbol{z})$, but $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ may not reproduce $\boldsymbol{x}$, e.g., when f is non-injective, or $\boldsymbol{y} \approx f(\boldsymbol{x})$ so that $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ also accounts for the modeling and observational noise. Fortunately, DIP models and first-order optimization methods together offer a blessing: in practice, $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ has a bias towards the desired visual content and learns it much faster than learning noise. Therefore, the quality of reconstruction climbs to a peak before the potential degradation due to noise; see Fig. 1. This "early-learning-then-overfitting" (ELTO) phenomenon has been repeatedly reported in previous work and is also supported by theories on simple $G_{\boldsymbol{\theta}}$ and linear f (Heckel & Soltanolkotabi, 2020b;a). **The successes of the DIP models claimed above are on the premise that appropriate early stopping (ES) around performance peaks can be made**.

Is ES for DIP trivial? Natural ideas trying to perform ES can fail quickly. **(1) Visual inspection**: This subjective approach is fine for small-scale tasks involving few problem instances, but quickly becomes infeasible for many scenarios, such as (a) large-scale batch processing, (b) recovery of visual contents tricky to visualize and/or examine by eyes (e.g. 3D or 4D visual objects), and (c) scientific imaging of unfamiliar objects (e.g., MRI imaging of rare tumors and microscopic imaging of new virus species); **(2) Tracking full-reference/no-reference image quality metrics (FR/NR-IQMs) or fitting loss**: Without the ground truth $\boldsymbol{x}$, computing any FR-IQM and thereby tracking its trajectory (e.g., the PNSR curve in Fig. 1) is out of the question. We consider the tracking of NR-IQMs as a family of baseline methods in Sec. 3.1; the performance is much worse than ours. We also explore the possibility of using the loss curve for ES here, but are unable to find correlations between the trend of the loss and that of the PSNR curve, shown in Fig. 15; **(3) Tuning the iteration number**: This ad-hoc solution is taken in most previous work. But since the peak iterations of DIP vary considerably across images and tasks (see, e.g., Figs. 4 and 29 and Appendices A.7.3 and A.7.5), this could entail numerous trial-and-error steps and lead to suboptimal stopping points; **(4) Validation-based ES**: ES easily reminds us of validation-based ES in supervised learning. The DIP approach to IPs, as summarized in Eq. (2) **does not belong to** supervised learning, as it only deals with a single instance $\boldsymbol{y}$, without separate $(\boldsymbol{x}, \boldsymbol{y})$ pairs as training data. There are recent ideas (Yaman et al., 2021; Ding et al., 2022) that hold part of the observation $\boldsymbol{y}$ out as a validation set to emulate validation-based ES in supervised learning, but they quickly become problematic for nonlinear IPs due to the significant violation of the underlying i.i.d. assumption; see Sec. 3.4.

Prior work addressing the overfitting There are three main approaches to counteracting the overfitting of DIP models. **(1) Regularization**: Heckel & Hand (2019) mitigates overfitting by restricting the size of $G_{\boldsymbol{\theta}}$ to the underparametrization regime. Metzler et al. (2018); Shi et al. (2022); Jo et al. (2021); Cheng et al. (2019) control the network capacity by regularizing the layer-wise weights or the network Jacobian. Liu et al. (2019); Mataev et al. (2019); Sun (2020); Cascarano et al. (2021) use additional regularizer(s) $R(G_{\boldsymbol{\theta}}(\boldsymbol{z}))$, such as the total-variation norm or trained denoisers. These methods require the right regularization level—which depends on the noise type and level—to avoid overfitting; with an improper regularization level, they can still lead to overfitting (see Fig. 4 and Sec. 3.1). Moreover, when they do succeed, the performance peak is postponed to the last iterations, often increasing the computational cost by several folds. **(2) Noise modeling**: You et al. (2020) models sparse additive noise as an explicit term in their optimization objective. Jo et al. (2021) designs regularizers and ES criteria specific to Gaussian and shot noise. Ding et al. (2021) explores subgradient methods with diminishing step size schedules for impulse noise with the ℓ_1 loss, with preliminary success. These methods do not work beyond the types and levels of noise they target, whereas our knowledge of the noise in a given visual IP is typically limited. **(3) Early stopping (ES)**: Shi et al. (2022) tracks progress based on a ratio of no-reference blurriness and sharpness, but the criterion only works for their modified DIP models, as acknowledged by the authors. Jo et al. (2021) provides noise-specific regularizer and ES criterion, but it is not clear how to extend the method to unknown types and levels of noise. Li et al. (2021) proposes monitoring DIP reconstruction by training a coupled autoencoder. Although its performance is similar to ours, the extra autoencoder training slows down the whole process dramatically; see Sec. 3. Yaman et al. (2021); Ding et al. (2022) emulate validation-based ES in supervised learning by splitting elements of $\boldsymbol{y}$ into "training" and "validation" sets so that validation-based ES can be performed. But in IPs, especially nonlinear ones (e.g., in blind image deblurring (BID), $\boldsymbol{y} \approx \boldsymbol{k} * \boldsymbol{x}$ where $*$ is the linear convolution), elements of $\boldsymbol{y}$ can be far from being i.i.d., and so validation may not work well. Moreover, holding out part of the observation in $\boldsymbol{y}$ can substantially reduce the peak performance; see Sec. 3.4.

Our contribution We advocate the ES approach—**the iteration process stops once a good ES point is detected**, as (1) the regularization and noise modeling approaches, even if effective, often do not improve peak performance but push it until the last iterations; there could be $\geq 10\times$ more iterations spent than climbing to the peak in the original DIP models; (2) both need deep knowledge about the noise type/level, which is practically unknown for most applications. If their key models and hyperparameters are not set appropriately, overfitting probably remains, and ES is still needed. **In this paper, we build a novel ES criterion for various DIP models simply by monitoring the trend of the running variance of the reconstruction sequence**. Our ES method is **(1) Effective**: The gap between our detected and the peak performance, i.e., the detection gap, is typically very small, as measured by standard visual quality metrics

Table 1: Summary of performance of our DIP+ES-WMV and competing methods on image denoising and blind image deblurring (BID). ✓: working reasonably well (PSNR $\geq 2dB$ less of the original DIP peak); -: not working well (PSNR $\leq 2dB$ less of the original DIP peak): N/A: not applicable (i.e., we do not perform comparison due to certain reasons). Note that DF-STE, DOP, and SB are based on modified DIP models.

	Image denoising								BID	
	Gaussian		Impulse		Speckle		Shot		Real world	
	Low	High	Low	High	Low	High	Low	High	Low	High
DIP+ES-WMV (Ours)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
DIP+NR-IQMs	-	-	-	-	-	-	-	-	N/A	N/A
DIP+SV-ES	✓	✓	✓	✓	✓	✓	✓	✓	N/A	N/A
DIP+VAL	✓	✓	✓	✓	✓	✓	✓	✓	-	-
DF-STE	✓	✓	N/A	N/A	N/A	N/A	✓	✓	N/A	N/A
DOP	N/A	N/A	✓	✓	N/A	N/A	N/A	N/A	N/A	N/A
SB	✓	✓	N/A	N/A	N/A	N/A	N/A	N/A	N/A	N/A

(PSNR and SSIM). Our method works well for DIP and its variants, including sinusoidal representation networks (Sitzmann et al., 2020, SIREN) and deep decoder (Heckel & Hand, 2019), on different noisy types/levels and in 5 visual IPs, including both linear and non-linear ones. Furthermore, our method can help several regularization-based methods, e.g., Gaussian process-DIP (Cheng et al., 2019, GP-DIP), DIP with total variation regularization (Liu et al., 2019; Cascarano et al., 2021, DIP-TV) to perform reasonable ES when they fail to prevent overfitting; **(2) Efficient**: The per-iteration overhead is a fraction—for the standard version in Algorithm 1, or negligible—for the variant in Algorithm 2, relative to the per-iteration cost of Eq. (2); **(3) Robust**: Our method is relatively insensitive to the two hyperparameters, i.e. window size and patience number. We keep the same hyperparameters for all experiments Secs. 2 and 3 except for the ablation study. In contrast, the hyperparameters of most of the methods reviewed above are sensitive to the noise type/level. We summarize the performance of our DIP+ES method against competing methods for image denoising and BID in Tab. 1; we present the detailed results in Sec. 3.

Remarks on diffusion models for IPs Recently, diffusion-based models (DBMs) have shown great promise in solving linear IPs Wang et al. (2022); Zhu et al. (2023). However, we note three things about these ideas: (1) their performance seems to be sensitive to the match of noise type and level between the training of the diffusion models and those in the actual IPs. Mismatch can lead to miserable results, as we demonstrate in Tabs. 5 and 9; (2) DBMs in solving IPs can suffer from overfitting issues similar to that in DIP also (see Appendix A.8); (3) there has been limited success in tackling nonlinear IPs by DBMs so far, see, e.g., the very recent attempt Chung et al. (2023). It remains to be seen how effective these ideas can be on general nonlinear IPs.

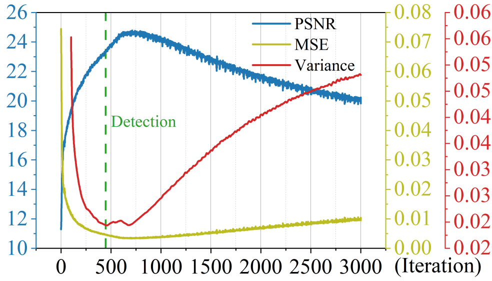


Figure 2: Relationship between the PSNR, MSE, and VAR curves. Our method relies on the VAR curve, whose valley is often well aligned with the MSE valley—that corresponds to the PSNR peak.

2 Our Early-Stopping (ES) Method

Intuition for our method We assume that $\boldsymbol{x}$ is the unknown groundtruth visual object of size N, $\{\boldsymbol{\theta}^t\}_{t\geq 1}$ is the iterate sequence and $\{\boldsymbol{x}^t\}_{t\geq 1}$ the reconstruction sequence where $\boldsymbol{x}^t \doteq G_{\boldsymbol{\theta}^t}(\boldsymbol{z})$. Since we do not know $\boldsymbol{x}$, we cannot compute the PNSR or any FR-IQM curve. But we observe from Fig. 2 that the MSE (resp. PSNR; recall $\text{PSNR}(\boldsymbol{x}^t) = 10 \log_{10} \|\boldsymbol{x}\|_\infty^2 / \text{MSE}(\boldsymbol{x}^t)$) curve follows a U (resp. bell) shape: $\|\boldsymbol{x}^t - \boldsymbol{x}\|_F^2$ initially drops rapidly to a low level and then climbs back due to the noise effect, i.e., the ELTO phenomenon in Sec. 1; we hope to detect the valley of this U-shaped MSE curve.

Then how to gauge the MSE curve **without knowing $\boldsymbol{x}$**? We consider the running variance (VAR):

$$\mathrm{VAR}(t) \doteq \frac{1}{W}\sum_{w=0}^{W-1} \|\boldsymbol{x}^{t+w} - 1/W \cdot \sum_{i=0}^{W-1} \boldsymbol{x}^{t+i}\|_F^2. \tag{3}$$

Initially, the models quickly learn the desired visual content, resulting in a monotonic and rapidly decreasing MSE curve (see Fig. 2). So we expect the running variance of $\{\boldsymbol{x}^t\}_{t\geq 1}$ to also drop quickly, as shown in Fig. 2. When the iteration is near the MSE valley, all $\boldsymbol{x}^t$' s are near, but scattered around $\boldsymbol{x}$. So $\frac{1}{W}\sum_{i=0}^{W-1}\boldsymbol{x}^{t+i} \approx \boldsymbol{x}$ and $\mathrm{VAR}(t) \approx \frac{1}{W}\sum_{w=0}^{W-1}\|\boldsymbol{x}^{t+w} - \boldsymbol{x}\|_F^2$. Afterward, the noise effect kicks in and the MSE curve bounces back, leading to a similar bounce back in the VAR curve as the $\boldsymbol{x}^t$ sequence gradually moves away from $\boldsymbol{x}$.

This argument suggests a U-shaped VAR curve and the curve should follow the trend of the MSE curve, with approximately aligned valleys, which in turn are aligned with the PSNR peak. To quickly verify this, we randomly sample 1024 images from the RGB track of the NTIRE 2020 Real Image Denoising Challenge (Abdelhamed et al., 2020), and perform DIP-based image denoising (i.e. min $\ell(\boldsymbol{y}, G_{\boldsymbol{\theta}}(\boldsymbol{z}))$ where $\boldsymbol{y}$ denotes the noisy image). Tab. 2 reports the average detected PSNR/SSIM and the average detection gaps based on our ES method (see Algorithm 1) that tries to detect the valley of the VAR curve. On average, the detection gaps are ≤ 0.95 in PSNR and ≤ 0.02 in SSIM, and the difference in visual qualities is typically barely noticeable by eyes! Furthermore, we provide histograms of the PSNR and SSIM gaps in Fig. 25. For more than 95% of the images, our ES method obtains a PSNR gap less than $2dB$.

Table 2: ES-WMV (our method) on real-world image denoising for **1024 images**: mean and (std) on the images. (**D**: detected)

ℓ (loss)	PSNR (**D**)	PSNR Gap	SSIM (**D**)	SSIM Gap
MSE	34.04 (3.68)	0.92 (0.83)	0.92 (0.07)	0.02 (0.04)
ℓ_1	33.92 (4.34)	0.92 (0.59)	0.93 (0.05)	0.02 (0.02)
Huber	33.72 (3.86)	0.95 (0.73)	0.92 (0.06)	0.02 (0.03)

Detecting transition by running variance Our lightweight method only involves computing the VAR curve and numerically detecting its valley—**the iteration stops once the valley is detected**. To obtain the curve, we set a window size parameter W and compute the windowed moving variance (WMV). To robustly detect the valley, we introduce a patience number P to tolerate up to P consecutive steps of variance stagnation. Obviously, the cost is dominated by the calculation of variance per step, which is $O(WN)$ (N is the size of the visual object). In comparison, a typical gradient update step for solving Eq. (2) costs at least $\Omega(|\boldsymbol{\theta}|N)$, where $|\boldsymbol{\theta}|$ is the number of parameters in the DNN $G_{\boldsymbol{\theta}}$. Since $|\boldsymbol{\theta}|$ is typically much larger than W (default: 100), our running VAR and detection incur very little computational overhead. Our entire algorithmic pipeline is summarized in Algorithm 1. To confirm the effectiveness, we provide qualitative samples in Figs. 3 and 4, with more quantitative results included in the experiment part (Sec. 3; see also Tab. 2). Fig. 3 shows that for image denoising with different noise types/levels, our ES method can detect ES points that achieve near-peak performance. Similarly, our method remains effective in several popular DIP variants, as shown in Fig. 4. Note that although our detection for DIP-TV in Fig. 4 is a bit far from the peak in terms of iteration count (as the VAR curve is almost flat after the peak), the detection gap is still small ($\sim$ 1.29dB).

Algorithm 1 DIP with ES–WMV

```
Input: random seed $\boldsymbol{z}$, randomly-initialized $\boldsymbol{\theta}^0$, window size $W$, patience $P$, empty queue $\mathcal{Q}$, iteration counter $k = 0$, $\mathrm{VAR}_{\min} = \infty$
Output: reconstruction $\boldsymbol{x}^*$
while not stopped do
    update $\boldsymbol{\theta}$ via Eq. (2) to obtain $\boldsymbol{\theta}^{k+1}$ and $\boldsymbol{x}^{k+1}$
    push $\boldsymbol{x}^{k+1}$ to $\mathcal{Q}$, pop queue if $|\mathcal{Q}| > W$
    if $|\mathcal{Q}| = W$ then
        compute VAR of elements in $\mathcal{Q}$ via Eq. (3)
        if $\mathrm{VAR} < \mathrm{VAR}_{\min}$ then
            $\mathrm{VAR}_{\min} \leftarrow \mathrm{VAR}$, $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{k+1}$
        end if
        if $\mathrm{VAR}_{\min}$ stagnates for $P$ iterations then
            stop and return $\boldsymbol{x}^*$
        end if
    end if
    $k = k + 1$
end while
```

Seemingly similar ideas Our running variance and its U-shaped curve are reminiscent of the classical U-shaped bias-variance tradeoff curve and therefore validation-based ES (Geman et al., 1992; Yang et al., 2020). But there are crucial differences: (1) our learning setting is not supervised; (2) the variance in supervised

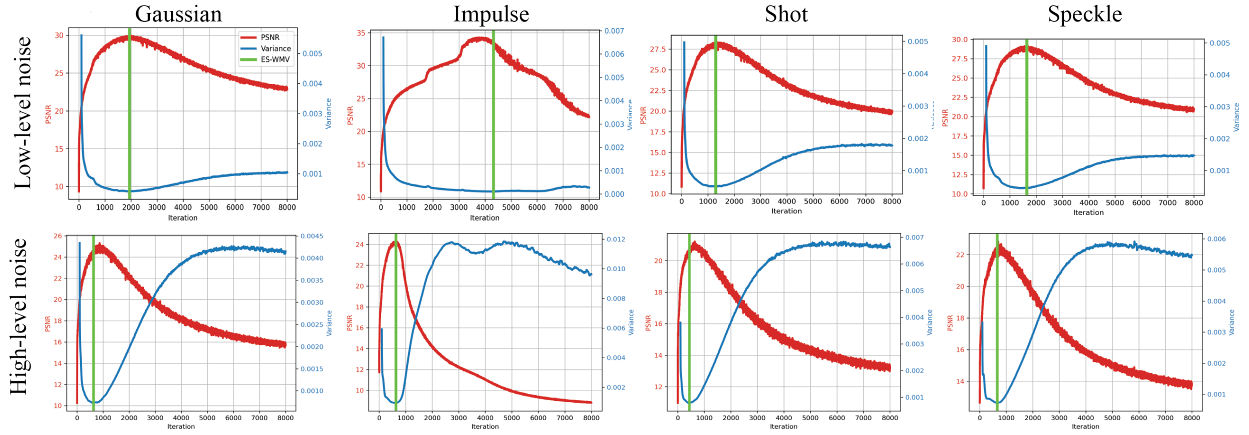


Figure 3: Our ES-WMV method on DIP for denoising “F16" with different noise types and levels (top: low-level noise; bottom: high-level noise). Red curves are PSNR curves, and blue curves are VAR curves. The green bars indicate the detected ES point.

learning is with respect to the sample distribution, while our variance here pertains to the $\{\boldsymbol{x}^t\}_{t\geq 1}$ sequence. As discussed in Sec. 1, we cannot directly apply validation-based ES, although it is possible to heuristically emulate it by splitting the elements in $\boldsymbol{y}$ (Yaman et al., 2021; Ding et al., 2022)—which might be problematic for nonlinear IPs. Another line of related ideas is the detection of variance-based online change points in time series analysis (Aminikhanghahi & Cook, 2017), where the running variance is often used to detect shifts in means under the assumption that the means are piecewise constant. Here, the piecewise constancy assumption does not hold for our $\{\boldsymbol{x}^t\}_{t\geq 1}$.

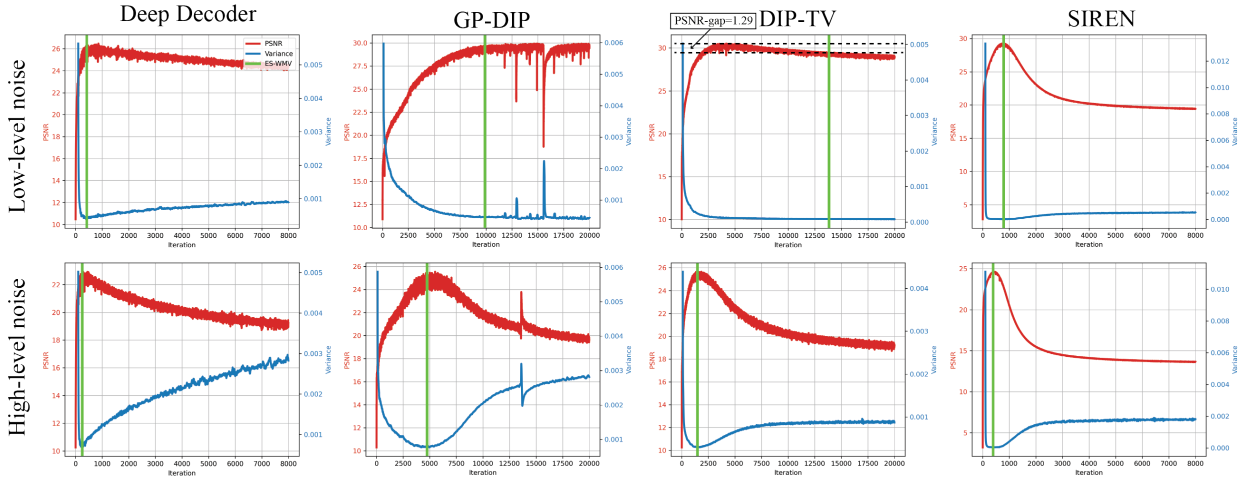


Figure 4: ES-WMV on deep decoder, GP-DIP, DIP-TV, and SIREN for denoising “F16” with different levels of Gaussian noise (top: low-level noise; bottom: high-level noise). Red curves are PSNR curves, and blue curves are VAR curves. The green bars indicate the detected ES point. (We sketch the details of the above DIP variants in Appendix A.5)

Theoretical justification We can make our heuristic argument in Sec. 2 more rigorous by restricting ourselves to additive denoising, that is, $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where the noise $\boldsymbol{n} \sim \mathcal{N}\left(\boldsymbol{0}, \xi^2/n \cdot \boldsymbol{I}\right)$, and appealing to the popular linearization strategy (i.e. neural tangent kernel Jacot et al. (2018); Heckel & Soltanolkotabi (2020b)) in understanding DNN. The idea is based on the assumption that during DNN training $\boldsymbol{\theta}$ does not move much away from initialization $\boldsymbol{\theta}^0$, so that the learning dynamic can be approximated by that of a linearized model, i.e. suppose that we take the MSE loss,

$$\|\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z})\|_2^2 \approx \left\|\boldsymbol{y} - G_{\boldsymbol{\theta}^0}(\boldsymbol{z}) - \boldsymbol{J}_G\left(\boldsymbol{\theta}^0\right)\left(\boldsymbol{\theta} - \boldsymbol{\theta}^0\right)\right\|_2^2 \doteq \widehat{f}(\boldsymbol{\theta}), \tag{4}$$

where $\boldsymbol{J}_G\left(\boldsymbol{\theta}^0\right)$ is the Jacobian of G with respect to $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^0$, and $G_{\boldsymbol{\theta}^0}(\boldsymbol{z}) + \boldsymbol{J}_G\left(\boldsymbol{\theta}^0\right)\left(\boldsymbol{\theta} - \boldsymbol{\theta}^0\right)$ is the first-order Taylor approximation to $G_{\boldsymbol{\theta}}(\boldsymbol{z})$ around $\boldsymbol{\theta}^0$. $\widehat{f}(\boldsymbol{\theta})$ is simply a linear least-squares objective. We can directly calculate the running variance based on the linear model, as shown below.

Theorem 2.1. *Let σ_i's and $\boldsymbol{w}_i$'s be the singular values and left singular vectors of $\boldsymbol{J}_G(\boldsymbol{\theta}^0)$, and suppose that we run a gradient descent with step size η on the linearized objective $\widehat{f}(\boldsymbol{\theta})$ to obtain $\{\boldsymbol{\theta}^t\}$ and $\{\boldsymbol{x}^t\}$ with $\boldsymbol{x}^t \doteq G_{\boldsymbol{\theta}^0}(\boldsymbol{z}) + \boldsymbol{J}_G(\boldsymbol{\theta}^0)(\boldsymbol{\theta}^t - \boldsymbol{\theta}^0)$. Then, provided that $\eta \leq 1/\max_i(\sigma_i^2)$,*

$$\mathrm{VAR}(t) = \sum_i C_{W,\eta,\sigma_i} \left\langle \boldsymbol{w}_i, \widehat{\boldsymbol{y}}\right\rangle^2 \left(1 - \eta\sigma_i^2\right)^{2t}, \tag{5}$$

where $\widehat{\boldsymbol{y}} = \boldsymbol{y} - G_{\boldsymbol{\theta}^0}(\boldsymbol{z})$, and $C_{W,\eta,\sigma_i} \geq 0$ depend only on W, η, and σ_i for all i.

The proof can be found in Appendix A.2. Theorem 2.1 shows that if the learning rate (LR) η is sufficiently small, the WMV of $\{\boldsymbol{x}^t\}$ decreases monotonically. We can develop a complementary upper bound for the WMV that has a U shape. To this end, we make use of Theorem 1 of Heckel & Soltanolkotabi (2020b), which can be summarized (some technical details omitted; precise statement is reproduced in Appendix A.3) as follows: consider the two-layer model $G_{\boldsymbol{C}}(\boldsymbol{B}) = \mathrm{ReLU}(\boldsymbol{UBC})\boldsymbol{v}$, where $\boldsymbol{C} \in \mathbb{R}^{n\times k}$ models 1×1 trainable convolutions, $\boldsymbol{v} \in \mathbb{R}^{k\times 1}$ contains fixed weights, $\boldsymbol{U}$ is an upsampling operation, and $\boldsymbol{B}$ is the fixed random seed. Let $\boldsymbol{J}$ be a reference Jacobian matrix solely determined by the upsampling operation $\boldsymbol{U}$, and σ_i's and $\boldsymbol{w}_i$'s the singular values and left singular vectors of $\boldsymbol{J}$. Assume that $\boldsymbol{x} \in \operatorname{span}\{\boldsymbol{w}_1, \dots, \boldsymbol{w}_p\}$. Then, when η is sufficiently small, with high probability,

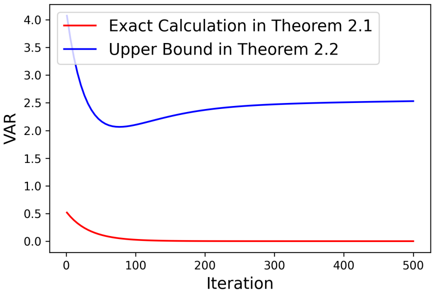


Figure 5: The exact and upper bounds predicted by Theorems 2.1 and 2.2.

$$\|G_{\boldsymbol{C}^t}(\boldsymbol{B}) - \boldsymbol{x}\|_2 \leq \left(1 - \eta\sigma_p^2\right)^t \|\boldsymbol{x}\|_2 + E(\boldsymbol{n}) + \varepsilon\|\boldsymbol{y}\|_2, \tag{6}$$

where $\varepsilon > 0$ is a small scalar related to the structure of the network and $E(\boldsymbol{n})$ is the error introduced by noise: $E^2(\boldsymbol{n}) \doteq \sum_{j=1}^n ((1-\eta\sigma_j^2)^t - 1)^2 \langle \boldsymbol{w}_j, \boldsymbol{n}\rangle^2$. So, if the gap $\sigma_p/\sigma_{p+1} > 1$, $\|G_{\boldsymbol{C}^t}(\boldsymbol{B}) - \boldsymbol{x}\|_2$ is dominated by $\left(1 - \eta\sigma_p^2\right)^t \|\boldsymbol{x}\|_2$ when t is small and then by $E(\boldsymbol{n})$ when t is large. However, since the former decreases and the latter increases as t grows, the upper bound has a U shape with respect to t. On the basis of this result, we have the following.

Theorem 2.2. *Assume the same setting as Theorem 2 of Heckel & Soltanolkotabi (2020b). With high probability, our WMV is upper bounded by*

$$\frac{12}{W}\|\boldsymbol{x}\|_2^2 \frac{\left(1 - \eta\sigma_p^2\right)^{2t}}{1 - (1 - \eta\sigma_p^2)^2} + 12\sum_{i=1}^n \left(\left(1 - \eta\sigma_i^2\right)^{t+W-1} - 1\right)^2 (\boldsymbol{w}_i^\intercal \boldsymbol{n})^2 + 12\varepsilon^2\|\boldsymbol{y}\|_2^2. \tag{7}$$

The exact statement and proof can be found in Appendix A.3. Using a reasoning process similar to that above, we can conclude that the upper bound in Theorem 2.2 also has a U shape. To interpret the results, Fig. 5 shows the curves (as functions of t) predicted by Theorems 2.1 and 2.2. The actual VAR curve should be between the two curves. These results are primitive and limited, similar to the situations for many deep learning theories that provide loose upper and lower bounds; we leave a complete theoretical justification for future work.

A memory-efficient variant While Algorithm 1 is already lightweight and effective in practice, we can modify it slightly to avoid maintaining $\mathcal{Q}$ and therefore saving memory. The trick is to use exponential moving variance (EMV), together with exponential moving average (EMA), shown in Appendix A.4. The hard window size parameter W is now replaced by the soft forgetting factor α: the larger the α, the smaller the impact of the history, and hence a smaller effective window. We systematically compare ES-WMV with ES-EMV in Appendix A.7.13 for image denoising tasks. The latter has slightly better detection due to the strong smoothing effect ($\alpha = 0.1$). For this paper, we prefer to remain simple and leave systematic evaluations of ES-EMV on other IPs for future work.

3 Experiments

We test ES-WMV for DIP in **image denoising, inpainting, demosaicing, super-resolution, MRI reconstruction, and blind image deblurring**, spanning both linear and nonlinear IPs. For image denoising, we also systematically evaluate ES-WMV for main DIP variants, including deep decoder (Heckel & Hand, 2019), DIP-TV (Cascarano et al., 2021), GP-DIP (Cheng et al., 2019), and demonstrate ES-WMV as a reliable helper to detect good ES points. Details of the DIP variants are discussed in Appendix A.5. We also compare ES-WMV with the main competing methods, including DF-STE (Jo et al., 2021), SV-ES (Li et al., 2021), DOP (You et al., 2020), SB (Shi et al., 2022), and VAL (Yaman et al., 2021; Ding et al., 2022). Details of the main ES-based methods can be found in Appendix A.6. We use both PSNR and SSIM to assess reconstruction quality and report PSNR and SSIM gaps (the difference between our detected and peak numbers) as indicators of our detection performance. **Common acronyms, pointers to external codes, detailed experiment settings, real-world denoising, image inpainting, and image demosaicing are in Appendices A.1, A.7.1, A.7.2, A.7.7, A.7.8 and A.7.10, respectively.**

3.1 Image denoising

Prior work dealing with DIP overfitting mostly focuses on image denoising and typically only evaluates their methods on one or two kinds of noise with low noise levels, e.g., low-level Gaussian noise. To stretch our evaluation, we consider 4 types of noise: Gaussian, shot, impulse, and speckle. We take the classical 9-image dataset (Dabov et al., 2008), and for each noise type, generate two noise levels, low and high, i.e., level 2 and 4 of Hendrycks & Dietterich (2019), respectively. In Tab. 2 and Appendix A.7.7, we also report the performance of our ES-WMV on real-world denoising evaluated on **large-scale datasets**. In addition, we also compare DIP-based denoising with a state-of-the-art diffusion-model-based denoising in Tab. 9.

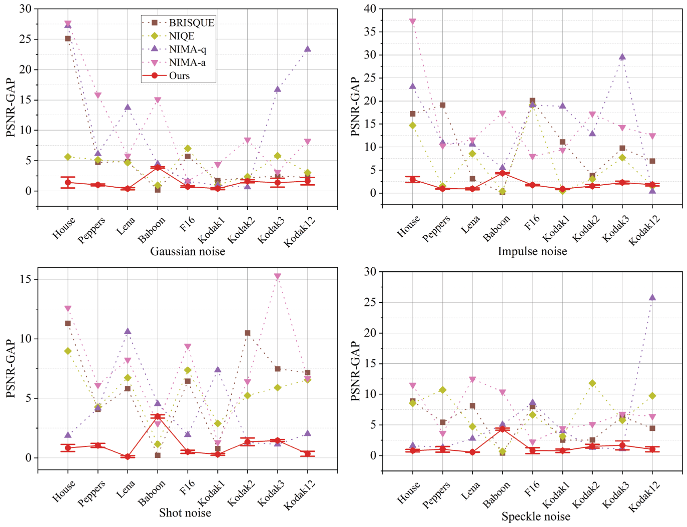


Figure 6: Baseline ES vs our ES-WMV on denoising with **low-level noise**. For NIMA, we report both technical quality assessment (NIMA-q) and aesthetic assessment (NIMA-a). Smaller PSNR gaps are better.

Comparison with baseline ES methods It is natural to expect that NR-IQMs, such as the classical BRISQUE (Mittal et al., 2012), NIQE (Mittal et al., 2013), and modern DNN-based NIMA (Esfandarani & Milanfar, 2018), can be used to monitor the quality of intermediate reconstructions and hence induce natural ES criteria. Therefore, we set 3 baseline methods using BRISQUE, NIQE, and NIMA, respectively, and seek the optimal $\boldsymbol{x}^t$ using these metrics. Fig. 6 presents the comparison (in terms of PSNR gaps) of these 3 methods with our ES-WMV on denoising with low-level noise by using DIP; results on high-level noise and also as measured by SSIM are included in Appendix A.7.4. Visual comparisons between our ES-WMV and the baseline methods are shown in Figs. 7 and 16. While **our method enjoys favorable detection gaps** (≤ 2) for most tested noise types/levels (except for Baboon, Kodak1, Kodak2 for certain noise types/levels; DIP itself is suboptimal in terms of denoising such images with substantial high-frequency components), **the baseline methods can see huge detection gaps up to** 10**.**

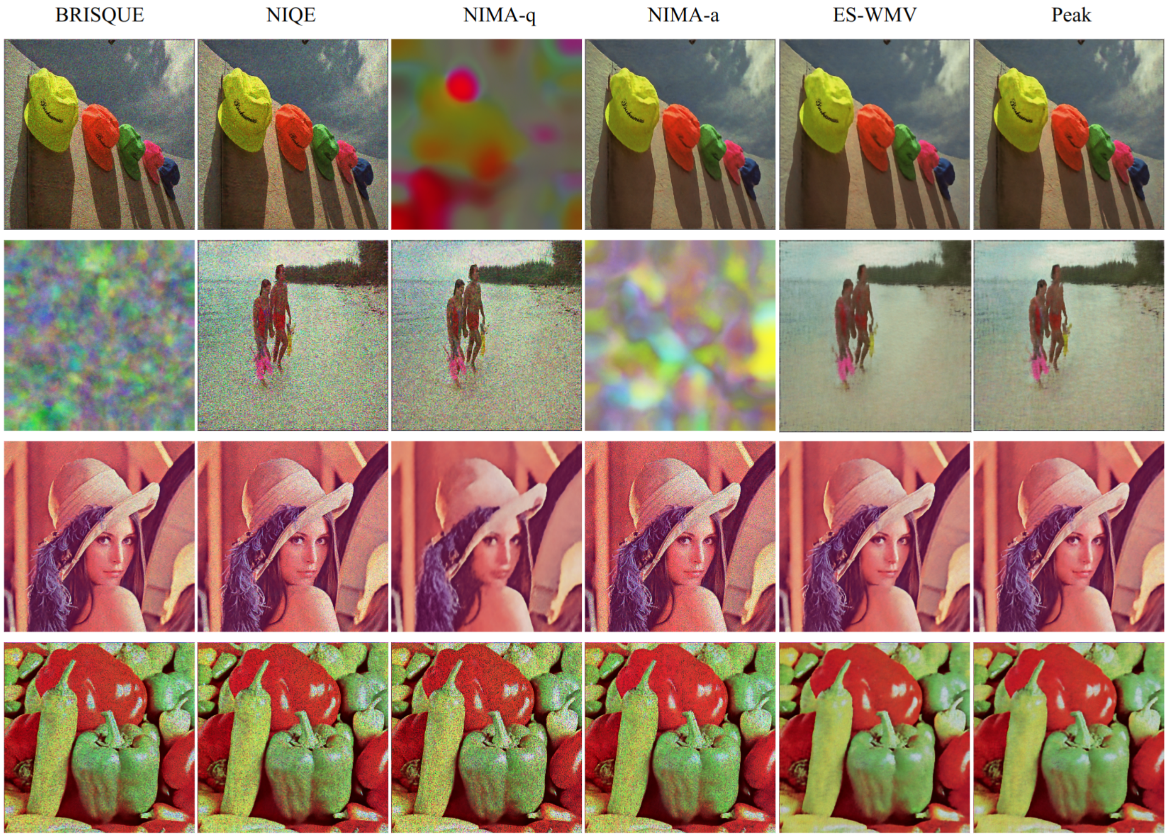


Figure 7: Visual comparisons of the detected images by NR-IQMs and ES-WMV. From top to bottom: shot noise (low), shot noise (high), speckle noise (low), speckle noise (high).

Competing methods DF-STE (Jo et al., 2021) is specific for Gaussian and Poisson denoising, and noise variance is needed for their tuning parameters. Fig. 8 presents the comparison of our method with DF-STE in terms of PSNR; SSIM results are in Appendix A.7.5. Here, we directly report the final PSNRs obtained by both methods. For low-level noise, there is no clear winner. **For high-level noise, ES-WMV outperforms DF-STE by considerable margins.** Although the right variance level is provided to DF-STE in order to tune their regularization parameters, DF-STE stops after only very few epochs, leading to very low performance and almost zero standard deviations—since they return almost the noisy input. However, we do not perform any parameter tuning for ES-WMV. Furthermore, we compare the two methods on the CBSD68 dataset in Appendix A.7.5 that leads to a similar conclusion.

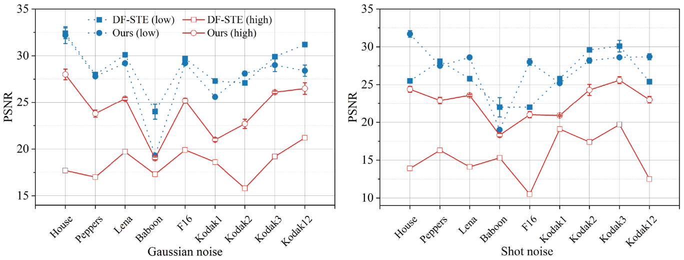


Figure 8: Comparison of DF-STE and ES-WMV for Gaussian and shot noise in terms of PSNR.

We report the results of SV-ES in Appendix A.7.5 since ES-WMV performs largely comparable to SV-ES. However, ES-WMV is much faster in wall-clock time, as reported in Tab. 3: for each epoch, the overhead of our ES-WMV is less than 3/4 of the DIP update itself, while SV-ES is around 25× of that.

There is no surprise: while our method only needs to update the running variance of $\{\boldsymbol{x}^t\}_{t\geq 1}$ each time, **SV-ES needs to train a coupled autoencoder which is extremely expensive.**

DOP is **designed specifically just for impulse noise**, so we compare ES-WMV with DOP on impulse noise (see Appendix A.7.5). The loss is changed to ℓ_1 to account for the sparse noise. In terms of the final PSNRs, DOP outperforms DIP with ES-WMV by a small gap, but even the peak PSNR of DIP with ℓ_1 lags behind DOP by about 2dB for high noise levels.

Table 3: Wall-clock time (secs) of DIP and three ES methods per epoch on *NVIDIA Tesla K40 GPU*: mean and (std). The total wall clock time should contain both DIP and a certain ES method.

	DIP	SV-ES	ES-WMV	ES-EMV
Time	0.448 (0.030)	**13.027 (3.872)**	0.301 (0.016)	0.003 (0.003)

Table 4: Comparison between ES-WMV and SB for image denoising on the CBSD68 dataset with varying noise level σ. The higher PSNR detected and earlier detection are better, which are in red: mean and (std).

	$\sigma = 15$		$\sigma = 25$		$\sigma = 50$	
	PSNR	Epoch	PSNR	Epoch	PSNR	Epoch
WMV	28.7(3.2)	3962(2506)	27.4(2.6)	3068(2150)	24.2(2.3)	1548(1939)
SB	29.0(3.1)	4908(1757)	27.3(2.2)	5099(1776)	23.0(1.0)	5765(1346)

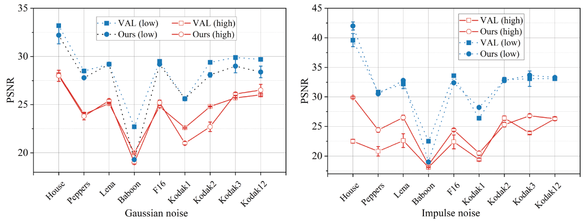
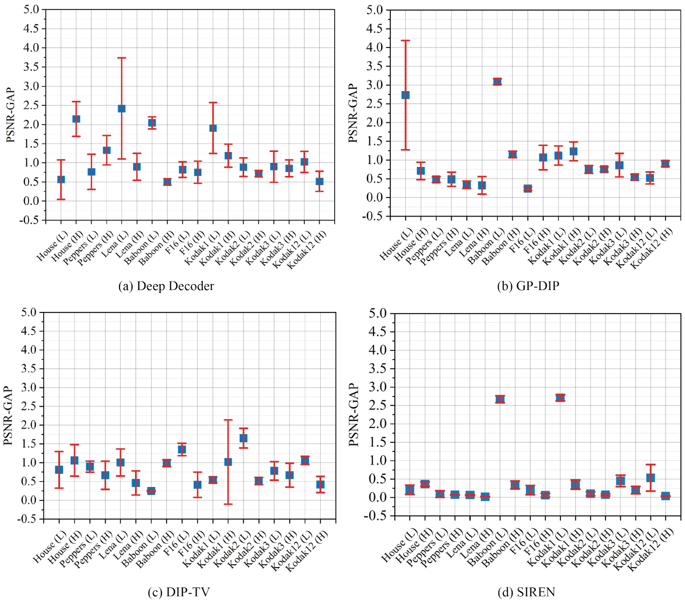


Figure 9: Comparison of VAL and ES-WMV for Gaussian and impulse noise in terms of PSNR.

The ES method in SB is acknowledged by its authors to fail for vanilla DIP (Shi et al., 2022). Moreover, their modified model still suffers from the overfitting issue beyond very low noise levels, as shown in Fig. 22. Their ES method fails to stop at appropriate places when the noise level is high. Hence, we test both ES-WMV and SB on their modified DIP model in (Shi et al., 2022), based on the two datasets they test: the classic 9-image dataset (Dabov et al., 2008) and the CBSD68 dataset (Martin et al., 2001). Qualitative results on 9 images are shown in Appendix A.7.5; detected PSNR and stop epochs on the CBSD68 dataset are reported in Tab. 4. For SB, the detection threshold parameter is set to 0.01. It is evident that both methods have similar detection performance for low noise levels, but ES-WMV outperforms SB when the noise level is high. Also, ES-WMV tends to stop much earlier than SB, saving computational cost.



(a) Deep Decoder

(b) GP-DIP

(c) DIP-TV

(d) SIREN

Figure 10: Performance of ES-WMV on deep decoder, GP-DIP, DIP-TV, and SIREN for Gaussian denoising in terms of PSNR gaps. L: low noise level; H: high noise level.

We compare VAL with our ES-WMV on the 9-image dataset with low-/high-level Gaussian and impulse noise. Since Ding et al. (2022) takes 90% pixels to train DIP and this usually decreases the peak performance, we report the final PSNRs detected by both methods (see Fig. 9). The two ES methods **perform very comparably in image denoising**, probably due to a mild violation of the i.i.d. assumption only, and also to a relatively low degree of information loss due to data splitting. **The more complex nonlinear BID in Sec. 3.4 reveals their gap.**

ES-WMV as a helper for DIP variants Deep decoder, DIP-TV, and GP-DIP represent different regularization strategies to control overfitting. However, a critical issue is setting the right hyperparameters for them so that overfitting is removed while peak-level performance is preserved. Therefore, practically, these methods are not free from overfitting, especially when the noise level is high. Thus, instead of treating them as competitors, we test whether ES-WMV can reliably detect good ES points for them. We focus on Gaussian denoising and report the results in Fig. 10 (a)-(c) and Appendix A.7.6. **ES-WMV is able to attain ≤ 1 PNSR gap for most cases**, with few outliers; we provide a detailed analysis about some of the outliers in Appendix A.9.

ES-WMV as a helper for implicit neural representations (INRs) INRs, such as Tancik et al. (2020) and Sitzmann et al. (2020), use multilayer perceptrons to represent highly nonlinear functions in low-dimensional problem domains and have achieved superior results in complex 3D visual tasks. We further extend our ES-WMV to help the INR family and take SIREN (Sitzmann et al., 2020) as an example. SIREN parameterizes $\boldsymbol{x}$ as the discretization of a continuous function: this function takes in spatial coordinates and returns the corresponding function values. Here, we test SIREN, which is reviewed in Appendix A.5, as a replacement for DIP models for Gaussian denoising and summarize the results in Fig. 10 and Fig. 24. **ES-WMV is again able to detect near-peak performance for most images.**

3.2 Image Super-Resolution

In this task, we try to recover a clean image $\boldsymbol{x}_0$ from a noisy downsampled version $\boldsymbol{y} = \mathcal{D}_t(\boldsymbol{x}_0) + \boldsymbol{\epsilon}$, where $\mathcal{D}_t(\cdot) : [0,1]^{3\times tH\times tW} \to [0,1]^{3\times H\times W}$ is a downsampling operator that resizes an image by the factor t and $\boldsymbol{\epsilon}$ models extra additive noise. We consider the following DIP-reparametrized formulation $\min_{\boldsymbol{\theta}} \ \ell(\boldsymbol{\theta}) \doteq \|\mathcal{D}_t(G_{\boldsymbol{\theta}}(\boldsymbol{z})) - \boldsymbol{y}\|_F^2$, where $G_{\boldsymbol{\theta}}$ is a trainable DNN parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{z}$ is a frozen random seed. Then we conduct experiments for 2× super-resolution with low-level Gaussian and impulse noise. We test our ES-WMV for DIP and a state-of-the-art zero-shot method based on pre-trained diffusion model—DDNM+ Wang et al. (2022) on the standard super-resolution dataset Set14 Zeyde et al. (2012), as shown in Tab. 5, Fig. 11, and Appendix A.7.9. We note that **DDNM+ relies on pre-trained models from large external training datasets, while DIP does not.** We observe that (1) **Our ES-WMV is again able to detect near-peak performance for most images**: the average PSNR gap is ≤ 1.50 and the average SSIM gap is ≤ 0.07; (2) DDNM+ is sensitive to the noise type and level: from Tab. 5, DDNM+ trained assuming Gaussian noise level $\sigma_y = 0.12$ outperforms DIP and DIP+ES-WMV when there is Gaussian measurement noise at the level $\sigma_y = 0.12$, **which is unrealistic in practice, as the noise level is often unknown beforehand**. When the noise level is not set correctly, e.g., as $\sigma_y = 0$ in the DDNM+ ($\sigma_y = .00$) row of Tab. 5, the performance of DDNM+ is much worse than that of DIP and DIP+ES-WMV. Also, for super-resolution with impulse noise, DIP is also a clear winner that leads DDNM+ by a large margin; and (3) in Appendix A.8, we show that DDNM+ may also suffer from the overfitting issue and our ES-WMV can help DDNM+ to stop around the performance peak as well.

Table 5: Comparison of ES-WMV for DIP and DDNM+ Wang et al. (2022) for **2× image super-resolution with low-level Gaussian and impulse noise**: mean and (std). The highest PSNR and SSIM for each task are in red. In particular, we set the best hyperparameter for DDNM+ ($\sigma_y = 0.12$), **which is unfair for the DIP + ES-WMV combination as we fix its hyperparameter setting**.

	PSNR		SSIM	
	Gaussian	Impulse	Gaussian	Impulse
DIP (peak)	22.88 (1.58)	28.28 (2.73)	0.61 (0.09)	0.88 (0.06)
DIP + ES-WMV	22.11 (1.90)	26.77 (3.76)	0.54 (0.11)	0.86 (0.06)
DDNM+ ($\sigma_y = .12$)	25.37 (2.00)	18.50 (0.68)	0.74 (0.11)	0.50 (0.08)
DDNM+ ($\sigma_y = .00$)	16.91 (0.42)	16.59 (0.34)	0.31 (0.09)	0.49 (0.06)

3.3 MRI reconstruction

We further test ES-WMV on MRI reconstruction, a classical linear IP with a nontrivial forward mapping: $\boldsymbol{y} \approx \mathcal{F}(x)$, where $\mathcal{F}$ is the subsampled Fourier operator, and we use $\approx$ to indicate that the noise encountered in practical MRI imaging may be hybrid (e.g., additive, shot) and uncertain. Here, we take the 8-fold undersampling and

Table 6: ConvDecoder on MRI reconstruction for **30 cases**: mean and (std). (**D**: Detected)

PSNR(**D**)	PSNR Gap	SSIM(**D**)	SSIM Gap
32.63 (2.36)	0.23 (0.32)	0.81 (0.09)	0.01 (0.01)

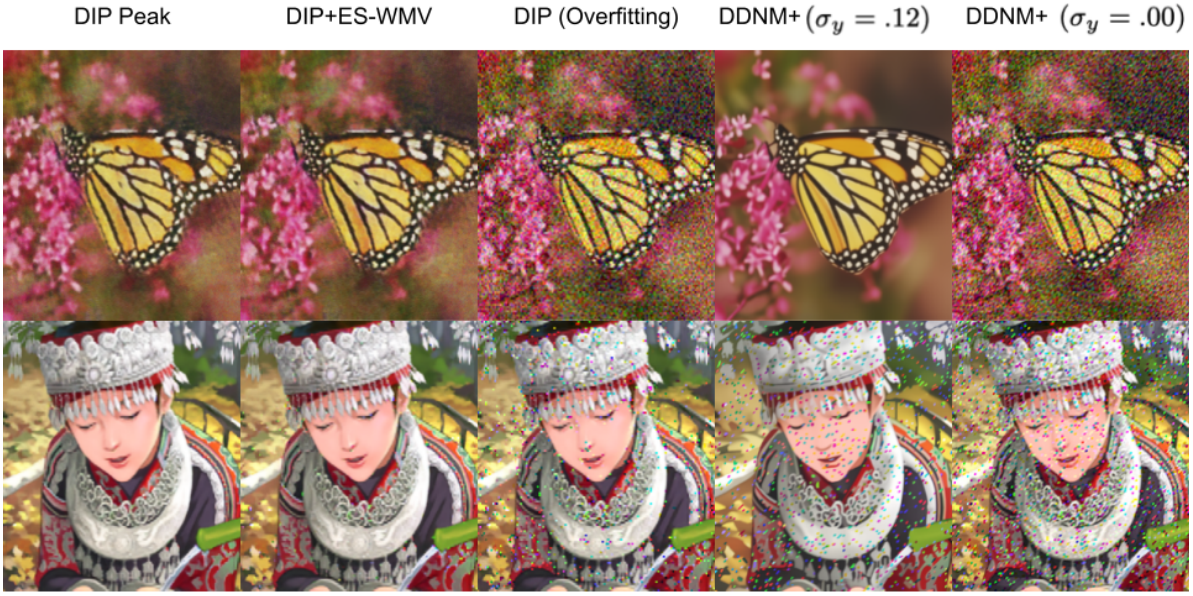


Figure 11: Visual comparisons for 2× image super-resolution task with additional noise. Top row: with additional low-level Gaussian noise; Bottom row: with additional low-level impulse noise.

parameterize $\boldsymbol{x}$ using "Conv-Decoder" (Darestani & Heckel, 2021), a variant of deep decoder. Due to the heavy over-parameterization, overfitting occurs and ES is needed. Darestani & Heckel (2021) directly sets the stopping point at the 2500-th epoch, and we run our ES-WMV. We visualize the performance on two random cases (C1: 1001339 and C2: 1000190 sampled from Darestani & Heckel (2021), part of the fastMRI datatset (Zbontar et al., 2018)) in Fig. 29 (quality measured in SSIM, consistent with Darestani & Heckel (2021)). It is clear that ES-WMV detects near-peak performance for both cases and is adaptive enough to yield comparable or better ES points than heuristically fixed ES points. Furthermore, we test our ES-WMV on ConvDecoder for **30 cases** from the fastMRI dataset (see Tab. 6), which **shows the precise and stable detection of ES-WMV**.

3.4 Blind image deblurring (BID)

In BID, a blurry and noisy image is given, and the goal is to recover a sharp and clean image. The blur is mostly caused by motion and/or optical non-ideality in the camera, and the forward process is often modeled as $\boldsymbol{y} = \boldsymbol{k} * \boldsymbol{x} + \boldsymbol{n}$, where $\boldsymbol{k}$ is the blur kernel, $\boldsymbol{n}$ models additive sensory noise, and $*$ is linear convolution to model the spatial uniformity of the blur effect (Szeliski, 2022). BID is a very challenging visual IP due to bilinearity: $(\boldsymbol{k}, \boldsymbol{x}) \mapsto \boldsymbol{k} * \boldsymbol{x}$. Recently, Ren et al. (2020); Wang et al. (2019); Asim et al. (2020); Tran et al. (2021) have tried to use DIP models to solve BID by modeling $\boldsymbol{k}$ and $\boldsymbol{x}$ as two separate DNNs, i.e., $\min_{\boldsymbol{\theta}_k, \boldsymbol{\theta}_x} \|\boldsymbol{y} - G_{\boldsymbol{\theta}_k}(\boldsymbol{z}_k) * G_{\boldsymbol{\theta}_x}(\boldsymbol{z}_x)\|_2^2 + \lambda \|\nabla G_{\boldsymbol{\theta}_x}(\boldsymbol{z}_x)\|_1 / \|\nabla G_{\boldsymbol{\theta}_x}(\boldsymbol{z}_x)\|_2$, where the regularizer Li et al. (2023b) is to promote sparsity in the gradient domain for the reconstruction of $\boldsymbol{x}$, as standard in BID. We follow Ren et al. (2020) and choose

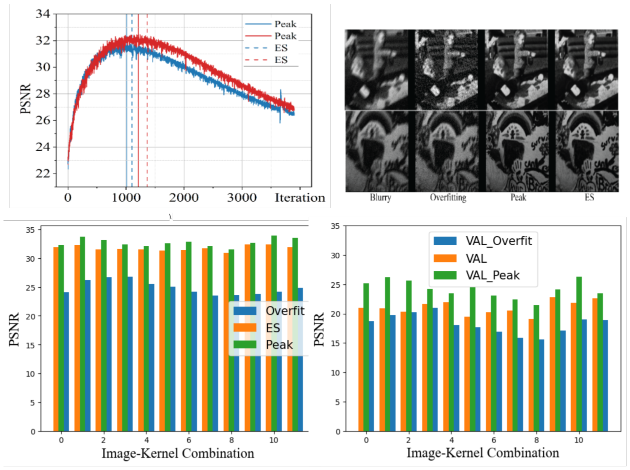


Figure 12: Top left: ES-WMV in BID; top right: visual results of ES-WMV; bottom: quantitative results of ES-WMV and VAL, respectively.

multilayer perceptron (MLP) with softmax activation for $G_{\boldsymbol{\theta}_k}$, and the canonical DIP model (CNN-based encoder-decoder architecture) for $G_{\boldsymbol{\theta}_x}(\boldsymbol{z}_x)$. We change their regularizer from the original $\|\nabla G_{\boldsymbol{\theta}_x}(\boldsymbol{z}_x)\|_1$ to the current, as their original formulation is tested only at a very low noise level $\sigma = 10^{-5}$ and no overfitting is observed. We set the test with a higher noise level $\sigma = 10^{-3}$, and find that its original formulation does not work. The benefit of the modified regularizer on BID is discussed in Krishnan et al. (2011).

First, we take 4 images and 3 kernels from the standard Levin dataset (Levin et al., 2011), resulting in 12 image-kernel combinations. The high noise level leads to substantial overfitting, as shown in Fig. 12 (top left). However, ES-WMV can reliably detect good ES points and lead to impressive visual reconstructions (see Fig. 12 (top right)). We systematically compare VAL and our ES-WMV on this difficult nonlinear IP, as we suspect that nonlinearity can break down VAL as discussed in Sec. 1, and subsampling the observation $\boldsymbol{y}$ for training-validation splitting may be unwise. Our results (Fig. 12 (bottom left/right)) confirm these predictions: **the peak performance detected by VAL is much worse after** 10% **of the elements in** $\boldsymbol{y}$ **are removed for validation**. In contrast, our ES-WMV returns quantitatively near-peak performance, much better than leaving the process to overfit. In Tab. 13, we further test both low- and high-level noise on the entire Levin dataset for completeness.

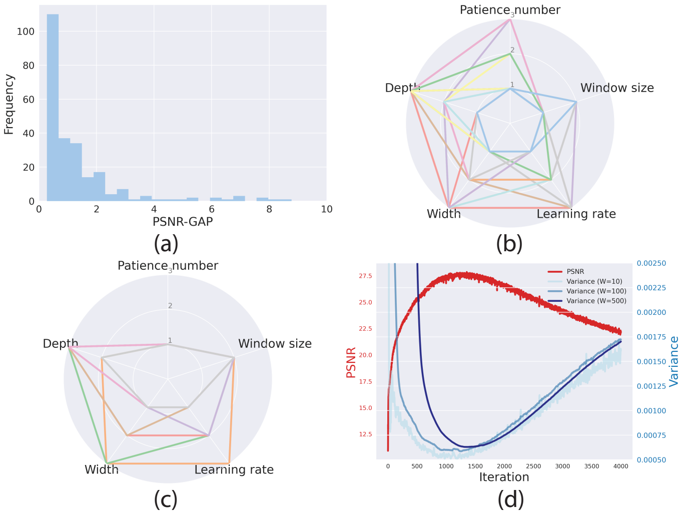


Figure 13: (a) Histogram of PSNR gaps for 243 different hyperparamter combinations; (b) radar plot of the cases whose PSNR gaps are larger than 2 dB; (c) radar plot of the cases whose PSNR gaps are larger than 2 dB and window size = 100; (d) an example case to visualize VAR curves with different window sizes. (For radar plots, there are three grid values for each hyperparameter. The farther away from the center, the higher the value.)

3.5 Ablation study

The window size W (default: 100) and the patience number P (default: 1000) are the only hyperparameters for our ES-WMV. Moreover, in this abalation study, we also include the key DIP hyperparameters, which obviously can also affect our ES performance—in our experiments above, we have used the default published DIP hyperparameters for each IP, as our ES method works under the condition that DIP performs reasonably well for the IP under consideration. To this end, we select the learning rate, which typically determines the learning pace and peak performance of DIP, and the depth/width of the network, which rules the network capacity.

Our base task is Gaussian denoising on the classic 9-image dataset (Dabov et al., 2008) with medium-level noise. We take the same default U-Net backbone model as the experiments in Fig. 3, and perform experiments on the following hyperparameter grid: window size $\{500, 100, 10\}$, patience number $\{5000, 1000, 100\}$, DIP learning rate $\{0.01, 0.001, 0.0001\}$, DIP model width $\{256, 128, 64\}$, and DIP model depth $\{7, 5, 3\}$, resulting in a total of 243 hyperparameter combinations. For each combination, we calculate the mean PSNR gap, on which our subsequent analysis is based. First of all, we see from Fig. 13(a) that for most ($\geq$ 83%) hyperparameter combinations, the mean PSNR gap falls below 2 dB. For cases larger than 2dB, we use the radar plot Fig. 13(b) to explore the deciding factors and find that most of these cases tend to have small (10) or medium (100) window sizes. This is not surprising, as a small window size can lead to a very fluctuating VAR curve, as shown in Fig. 13(d). To further explore other deciding factors, we focus on the subset with mean PSNR gap $\geq$ 2dB and window size = 100 and plot their settings in Fig. 13(c). We find that these cases invariably have a small patience number (100), which can trap our valley detection algorithm into a local fluctuation. So, overall, it seems that our window size and patience number are deciding factors for failures, relative to DIP hyperparameters such as learning rate and network capacity.

Hence, we next look closely at the combined effect of window size (W) and patience number (P) on ES performance. For this, we plot 9 histograms for the 9 different (W, P) combinations in Fig. 14 (i.e., each histogram is over the 27 DIP hyperparameters). The trend is clear: the larger the patience number, the smaller the PSNR gaps; the larger the window size, the smaller the detected PSNR gaps. The average PSNR gap of our default hyperparameter combinations is below 1 dB (the center one). If we further increase the patience number and the window size, the average PSNR gap can even be lower than 0.5 dB (top left corner). Overall, this confirms again our above observation that window size and patience number are the deciding factors for the detection performance of our ES method. Also, this suggests that our ES method can operate well with small PSNR gaps over a wide range of combinations (W, P), unless both are very small.

4 Discussion

We have proposed a simple yet effective ES detection method (ES-WMV, and the ES-EMV variant) that works robustly on multiple visual IPs and DIP variants. In comparison, most competing ES methods are noise- or DIP-model-specific and only work for limited scenarios; Li et al. (2021) has comparable performance

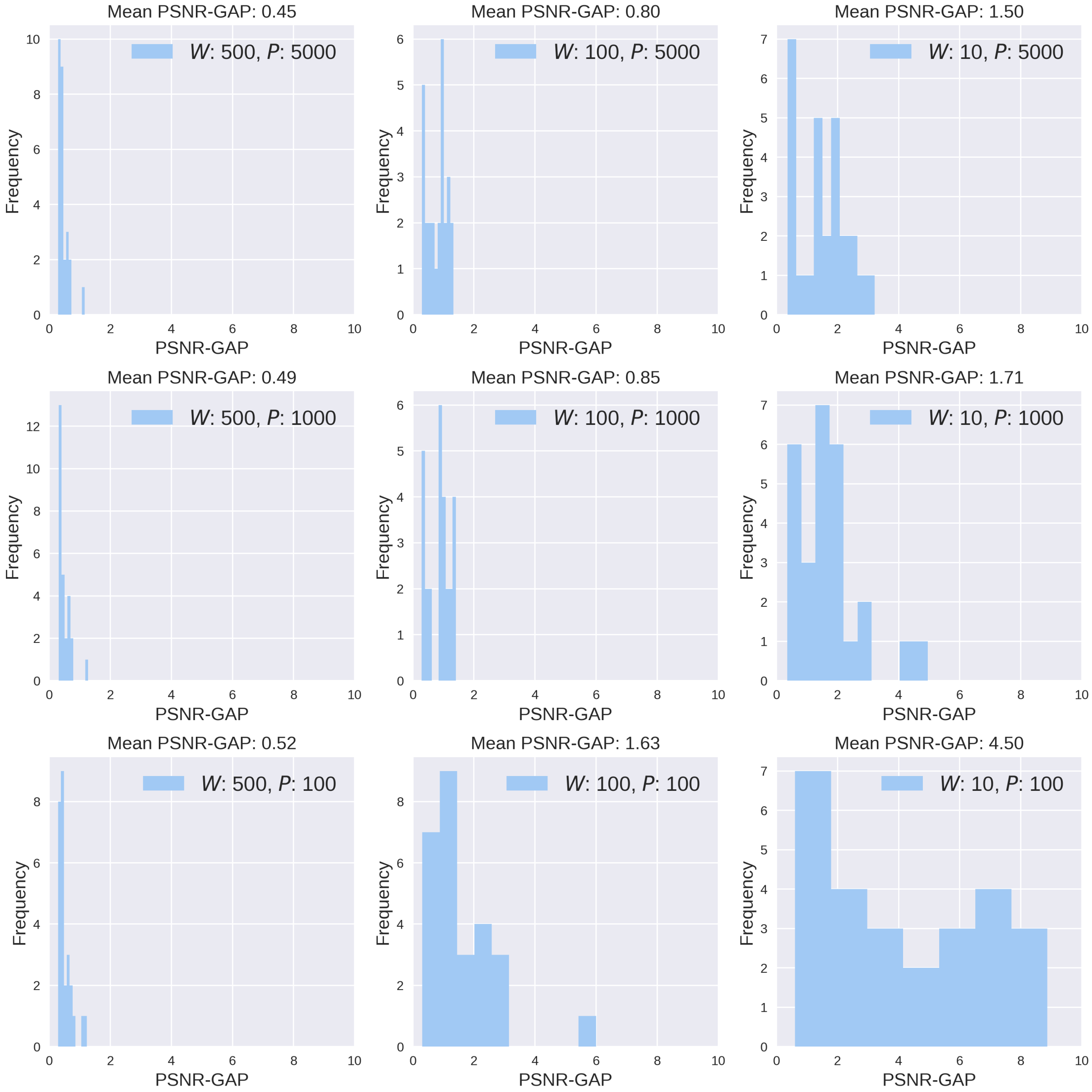


Figure 14: 9 histograms for 9 different hyperparameter combinations of our ES method. In each histogram, we show the frequencies of PSNR-GAPs for 27 various DIP hyperparameter possibilities. W: window size; P: patience number.

but slows down the running speed too much; validation-based ES (Ding et al., 2022) works well for the simple denoising task while significantly lags behind our ES method on nonlinear IPs, e.g. BID.

As for limitations, our theoretical justification is only partial, sharing the same difficulty of analyzing DNNs in general; our ES method struggles with images with substantial high-frequency components; our detection is sometimes off the peak in terms of iteration numbers when helping certain DIP variants, e.g. DIP-TV with low-level Gaussian noise (Fig. 4), but the detected PSNR gap is still small. DIP variants typically do not improve peak performance and also do not necessarily avoid overfitting, especially for high-level noise. We recommend the original DIP with our ES method for visual IPs discussed in this paper for the best performance and overall speed. Besides ES, there are other major technical barriers to making DIP models practical and competitive for visual IPs. A major one is efficiency: one needs to train a DNN using iterative methods for every instance; our recent works Li et al. (2023c;d) have made progress on this issue.

Acknowledgements

Zhong Zhuang, Hengkang Wang, Tiancong Chen and Ju Sun are partly supported by NSF CMMI 2038403. We thank the anonymous reviewers and the associate editor for their insightful comments that have substantially helped us to improve the presentation of this paper. The authors acknowledge the Minnesota Supercomputing Institute (MSI) at the University of Minnesota for providing resources that contributed to the research results reported within this paper.

References

Abdelrahman Abdelhamed, Mahmoud Afifi, Radu Timofte, Michael S. Brown, Yue Cao, Zhilu Zhang, Wangmeng Zuo, Xiaoling Zhang, Jiye Liu, Wendong Chen, Changyuan Wen, Meng Liu, Shuailin Lv, Yunchao Zhang, Zhihong Pan, Baopu Li, Teng Xi, Yanwen Fan, Xiyu Yu, Gang Zhang, Jingtuo Liu, Junyu Han, Errui Ding, Songhyun Yu, Bumjun Park, Jechang Jeong, Shuai Liu, Ziyao Zong, Nan Nan, Chenghua Li, Zengli Yang, Long Bao, Shuangquan Wang, Dongwoon Bai, Jungwon Lee, Youngjung Kim, Kyeongha Rho, Changyeop Shin, Sungho Kim, Pengliang Tang, Yiyun Zhao, Yuqian Zhou, Yuchen Fan, Thomas S. Huang, Zhihao Li, Nisarg A. Shah, Wei Liu, Qiong Yan, Yuzhi Zhao, Marcin Mozejko, Tomasz Latkowski, Lukasz Treszczotko, Michal Szafraniuk, Krzysztof Trojanowski, Yanhong Wu, Pablo Navarrete Michelini, Fengshuo Hu, Yunhua Lu, Sujin Kim, Wonjin Kim, Jaayeon Lee, Jang-Hwan Choi, Magauiya Zhussip, Azamat Khassenov, Jong Hyun Kim, Hwechul Cho, Priya Kansal, Sabari Nathan, Zhangyu Ye, Xiwen Lu, Yaqi Wu, Jiangxin Yang, Yanlong Cao, Siliang Tang, Yanpeng Cao, Matteo Maggioni, Ioannis Marras, Thomas Tanay, Gregory G. Slabaugh, Youliang Yan, Myungjoo Kang, Han-Soo Choi, Kyungmin Song, Shusong Xu, Xiaomu Lu, Tingniao Wang, Chunxia Lei, Bin Liu, Rajat Gupta, and Vineet Kumar. NTIRE 2020 challenge on real image denoising: Dataset, methods and results. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020, Seattle, WA, USA, June 14-19, 2020*, pp. 2077–2088. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPRW50498.2020.00256.

Samaneh Aminikhanghahi and Diane J. Cook. A survey of methods for time series change point detection. *Knowl. Inf. Syst.*, 51(2):339–367, 2017. doi: 10.1007/s10115-016-0987-z.

Muhammad Asim, Fahad Shamshad, and Ali Ahmed. Blind image deconvolution using deep generative priors. *IEEE Trans. Computational Imaging*, 6:1493–1506, 2020. doi: 10.1109/TCI.2020.3032671.

Daniel Otero Baguer, Johannes Leuschner, and Maximilian Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *CoRR*, abs/2003.04989, 2020.

Khosro Bahrami and A. C. Kot. A fast approach for no-reference image sharpness assessment based on maximum local variation. *IEEE Signal Process. Lett.*, 21(6):751–755, 2014. doi: 10.1109/LSP.2014.2314487.

Pasquale Cascarano, Andrea Sebastiani, Maria Colomba Comes, Giorgia Franchini, and Federica Porta. Combining weighted total variation and deep image prior for natural and medical image restoration via ADMM. In *2021 21st International Conference on Computational Science and Its Applications (ICCSA),* *Cagliari, Italy, September 13-16, 2021 - Workshops*, pp. 39–46. IEEE, 2021. doi: 10.1109/ICCSA54496.2021.00016.

Zezhou Cheng, Matheus Gadelha, Subhransu Maji, and Daniel Sheldon. A bayesian perspective on the deep image prior. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 5443–5451. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00559.

Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=OnD9zGAGT0k`.

Frederique Crete, Thierry Dolmiere, Patricia Ladret, and Marina Nicolas. The blur effect: perception and estimation with a new no-reference perceptual blur metric. In Bernice E. Rogowitz, Thrasyvoulos N. Pappas, and Scott J. Daly (eds.), *Human Vision and Electronic Imaging XII, San Jose, CA, USA, January 29 - February 1, 2007*, volume 6492 of *SPIE Proceedings*, pp. 64920I. SPIE, 2007. doi: 10.1117/12.702790.

Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen O. Egiazarian. Image restoration by sparse 3d transform-domain collaborative filtering. In Jaakko Astola, Karen O. Egiazarian, and Edward R. Dougherty (eds.), *Image Processing: Algorithms and Systems VI, San Jose, California, USA, January 28-29, 2008*, volume 6812 of *SPIE Proceedings*, pp. 681207. SPIE, 2008. doi: 10.1117/12.766355.

Mohammad Zalbagi Darestani and Reinhard Heckel. Accelerated MRI with un-trained neural networks. *IEEE Trans. Computational Imaging*, 7:724–733, 2021. doi: 10.1109/TCI.2021.3097596.

Lijun Ding, Liwei Jiang, Yudong Chen, Qing Qu, and Zhihui Zhu. Rank overspecified robust matrix recovery: Subgradient method and exact recovery. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 26767–26778, 2021.

Lijun Ding, Zhen Qin, Liwei Jiang, Jinxin Zhou, and Zhihui Zhu. A validation approach to over-parameterized matrix and image recovery. *CoRR*, abs/2209.10675, 2022. doi: 10.48550/arXiv.2209.10675.

Hossein Talebi Esfandarani and Peyman Milanfar. NIMA: neural image assessment. *IEEE Trans. Image Process.*, 27(8):3998–4011, 2018. doi: 10.1109/TIP.2018.2831899.

Yossi Gandelsman, Assaf Shocher, and Michal Irani. "double-dip": Unsupervised image decomposition via coupled deep-image-priors. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 11026–11035. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01128.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Comput.*, 4(1):1–58, 1992. doi: 10.1162/neco.1992.4.1.1.

Kuang Gong, Ciprian Catana, Jinyi Qi, and Quanzheng Li. Direct reconstruction of linear parametric images from dynamic PET using nonlocal deep image prior. *IEEE Trans. Medical Imaging*, 41(3):680–689, 2022. doi: 10.1109/TMI.2021.3120913.

Paul Hand, Oscar Leong, and Vladislav Voroninski. Phase retrieval under a generative prior. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 9154–9164, 2018.

Fumio Hashimoto and Kibo Ote. Direct PET image reconstruction incorporating deep image prior and a forward projection model. *CoRR*, abs/2109.00768, 2021.

Reinhard Heckel and Paul Hand. Deep decoder: Concise image representations from untrained non-convolutional networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.* OpenReview.net, 2019.

Reinhard Heckel and Mahdi Soltanolkotabi. Compressive sensing with un-trained neural networks: Gradient descent finds a smooth approximation. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4149–4158. PMLR, 2020a.

Reinhard Heckel and Mahdi Soltanolkotabi. Denoising and regularization via exploiting the structural bias of convolutional generators. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020b.

Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Arthur Jacot, Clément Hongler, and Franck Gabriel. Neural tangent kernel: Convergence and generalization in neural networks. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pp. 8580–8589, 2018.

Joel Janai, Fatma Güney, Aseem Behl, and Andreas Geiger. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Found. Trends Comput. Graph. Vis.*, 12(1-3):1–308, 2020. doi: 10.1561/0600000079.

Yeonsik Jo, Se Young Chun, and Jonghyun Choi. Rethinking deep image prior for denoising. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pp. 5067–5076. IEEE, 2021. doi: 10.1109/ICCV48922.2021.00504.

Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *The 24th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2011, Colorado Springs, CO, USA, 20-25 June 2011*, pp. 233–240. IEEE Computer Society, 2011. doi: 10.1109/CVPR.2011.5995521.

Anat Levin, Yair Weiss, Frédo Durand, and William T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2354–2367, 2011. doi: 10.1109/TPAMI.2011.148.

Taihui Li, Zhong Zhuang, Hengyue Liang, Le Peng, Hengkang Wang, and Ju Sun. Self-validation: Early stopping for single-instance deep generative priors. In *32nd British Machine Vision Conference 2021, BMVC 2021, Online, November 22-25, 2021*, pp. 108. BMVA Press, 2021.

Taihui Li, Anish Lahiri, Yutong Dai, and Owen Mayer. Joint demosaicing and denoising with double deep image priors. *CoRR*, abs/2309.09426, 2023a. doi: 10.48550/arXiv.2309.09426. URL `https://doi.org/10.48550/arXiv.2309.09426`.

Taihui Li, Hengkang Wang, Le Peng, Xian'e Tang, and Ju Sun. Robust Autoencoders for Collective Corruption Removal. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, June 2023b. doi: 10.1109/ICASSP49357.2023.10095099. URL `https://ieeexplore.ieee.org/abstract/document/10095099`. ISSN: 2379-190X.

Taihui Li, Hengkang Wang, Zhong Zhuang, and Ju Sun. Deep Random Projector: Accelerated Deep Image Prior. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 18176–18185, Vancouver, BC, Canada, June 2023c. IEEE. ISBN 9798350301298. doi: 10.1109/CVPR52729.2023.01743. URL `https://ieeexplore.ieee.org/document/10205276/`.

Taihui Li, Zhong Zhuang, Hengkang Wang, and Ju Sun. Random Projector: Efficient Deep Image Prior. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5, Rhodes Island, Greece, June 2023d. IEEE. ISBN 978-1-72816-327-7. doi: 10.1109/ICASSP49357.2023.10097088. URL `https://ieeexplore.ieee.org/document/10097088/`.

Jiaming Liu, Yu Sun, Xiaojian Xu, and Ulugbek S. Kamilov. Image restoration using total variation regularized deep image prior. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019*, pp. 7715–7719. IEEE, 2019. doi: 10.1109/ICASSP.2019.8682856.

Xudong Ma, Alin Achim, and Paul R. Hill. Unsupervised image fusion using deep image priors. *CoRR*, abs/2110.09490, 2021.

David R. Martin, Charless C. Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the Eighth International Conference On Computer Vision (ICCV-01), Vancouver, British Columbia, Canada, July 7-14, 2001 - Volume 2*, pp. 416–425. IEEE Computer Society, 2001. doi: 10.1109/ICCV.2001.937655.

Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

Christopher A. Metzler, Ali Mousavi, Reinhard Heckel, and Richard G. Baraniuk. Unsupervised learning with stein's unbiased risk estimator. *CoRR*, abs/1805.10531, 2018.

Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.*, 21(12):4695–4708, 2012. doi: 10.1109/TIP.2012.2214050.

Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal Process. Lett.*, 20(3):209–212, 2013. doi: 10.1109/LSP.2012.2227726.

Gregory Ongie, Ajil Jalal, Christopher A. Metzler, Richard G. Baraniuk, Alexandros G. Dimakis, and Rebecca Willett. Deep learning techniques for inverse problems in imaging. *IEEE J. Sel. Areas Inf. Theory*, 1(1):39–56, 2020. doi: 10.1109/jsait.2020.2991563.

Adnan Qayyum, Inaam Ilahi, Fahad Shamshad, Farid Boussaid, Mohammed Bennamoun, and Junaid Qadir. Untrained neural network priors for inverse imaging problems: A survey. *TechRxiv*, mar 2021. doi: 10.36227/techrxiv.14208215.v1.

Dongwei Ren, Kai Zhang, Qilong Wang, Qinghua Hu, and Wangmeng Zuo. Neural blind deconvolution using deep priors. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pp. 3338–3347. Computer Vision Foundation / IEEE, 2020. doi: 10.1109/CVPR42600.2020.00340.

Zenglin Shi, Pascal Mettes, Subhransu Maji, and Cees G. M. Snoek. On measuring and controlling the spectral bias of the deep image prior. *Int. J. Comput. Vis.*, 130(4):885–908, 2022. doi: 10.1007/s11263-021-01572-7.

Vincent Sitzmann, Julien N. P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Zhaodong Sun. Solving inverse problems with hybrid deep image priors: the challenge of preventing overfitting. *CoRR*, abs/2011.01748, 2020.

Richard Szeliski. *Computer Vision - Algorithms and Applications, Second Edition.* Texts in Computer Science. Springer, 2022. ISBN 978-3-030-34371-2. doi: 10.1007/978-3-030-34372-9.

Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33:*

Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, 2020.

Kshitij Tayal, Raunak Manekar, Zhong Zhuang, David Yang, Vipin Kumar, Felix Hofmann, and Ju Sun. Phase retrieval using single-instance deep generative prior. *CoRR*, abs/2106.04812, 2021.

Phong Tran, Anh Tuan Tran, Quynh Phung, and Minh Hoai. Explore image deblurring via encoded blur kernel space. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 11956–11965. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01178.

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Deep image prior. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 9446–9454. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00984.

Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 2968–2979, 2019.

David Van Veen, Ajil Jalal, Eric Price, Sriram Vishwanath, and Alexandros G. Dimakis. Compressed sensing with deep image prior and learned regularization. *CoRR*, abs/1806.06438, 2018.

Yinhuai Wang, Jiwen Yu, and Jian Zhang. Zero-Shot Image Restoration Using Denoising Diffusion Null-Space Model, December 2022. URL `http://arxiv.org/abs/2212.00490`. arXiv:2212.00490 [cs].

Zhunxuan Wang, Zipei Wang, Qiqi Li, and Hakan Bilen. Image deconvolution with deep image and kernel priors. In *2019 IEEE/CVF International Conference on Computer Vision Workshops, ICCV Workshops 2019, Seoul, Korea (South), October 27-28, 2019*, pp. 980–989. IEEE, 2019.

Francis Williams, Teseo Schneider, Cláudio T. Silva, Denis Zorin, Joan Bruna, and Daniele Panozzo. Deep geometric prior for surface reconstruction. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 10130–10139. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.01037.

Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *CoRR*, abs/1804.02603, 2018.

Burhaneddin Yaman, Seyed Amir Hossein Hosseini, and Mehmet Akcakaya. Zero-shot physics-guided deep learning for subject-specific MRI reconstruction. In *NeurIPS 2021 Workshop on Deep Learning and Inverse Problems*, 2021.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10767–10777. PMLR, 2020.

Jaejun Yoo, Kyong Hwan Jin, Harshit Gupta, Jérôme Yerly, Matthias Stuber, and Michael Unser. Time-dependent deep image prior for dynamic MRI. *IEEE Trans. Medical Imaging*, 40(12):3337–3348, 2021. doi: 10.1109/TMI.2021.3084288.

Chong You, Zhihui Zhu, Qing Qu, and Yi Ma. Robust recovery via implicit bias of discrepant learning rates for double over-parameterization. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Jure Zbontar, Florian Knoll, Anuroop Sriram, Matthew J. Muckley, Mary Bruno, Aaron Defazio, Marc Parente, Krzysztof J. Geras, Joe Katsnelson, Hersh Chandarana, Zizhao Zhang, Michal Drozdzal, Adriana Romero, Michael G. Rabbat, Pascal Vincent, James Pinkerton, Duo Wang, Nafissa Yakubova, Erich Owens, C. Lawrence Zitnick, Michael P. Recht, Daniel K. Sodickson, and Yvonne W. Lui. fastmri: An open dataset and benchmarks for accelerated MRI. *CoRR*, abs/1811.08839, 2018.

Roman Zeyde, Michael Elad, and Matan Protter. On Single Image Scale-Up Using Sparse-Representations. In Jean-Daniel Boissonnat, Patrick Chenin, Albert Cohen, Christian Gout, Tom Lyche, Marie-Laurence Mazure, and Larry Schumaker (eds.), *Curves and Surfaces*, volume 6920, pp. 711–730. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012. ISBN 978-3-642-27412-1 978-3-642-27413-8. doi: 10.1007/978-3-642-27413-8_47. URL `http://link.springer.com/10.1007/978-3-642-27413-8_47`. Series Title: Lecture Notes in Computer Science.

Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising Diffusion Models for Plug-and-Play Image Restoration, May 2023. URL `http://arxiv.org/abs/2305.08995`. arXiv:2305.08995 [cs, eess].

Zhong Zhuang, Taihui Li, Hengkang Wang, and Ju Sun. Blind image deblurring with unknown kernel size and substantial noise. *CoRR*, abs/2208.09483, 2022a. doi: 10.48550/arXiv.2208.09483.

Zhong Zhuang, David Yang, Felix Hofmann, David Barmherzig, and Ju Sun. Practical phase retrieval using double deep image priors. *arXiv preprint arXiv:2211.00799*, 2022b.

A Appendix

A.1 Acronyms

List of Common Acronyms (in alphabetic order)

CNN	convolutional neural network
DIP	deep image prior
DIP-TV	DIP with total variation regularization
DNN	deep neural network
ELTO	early-learning-then-overfitting
ES	early stopping
EMA	exponential moving average
EMV	exponential moving variance
FR-IQM	full-reference image quality metric
GP-DIP	Gaussian process DIP
INR	implicit neural representations
IP	inverse problem
MSE	mean squared error
NR-IQM	no-reference image quality metric
PSNR	peak signal-to-noise ratio
SIREN	sinusoidal representation networks
VAR	variance
WMV	windowed moving variance

A.2 Proof of 2.1

Proof. To simplify the notation, we write $\widehat{\boldsymbol{y}} \doteq \boldsymbol{y} - G_{\boldsymbol{\theta}^0}(\boldsymbol{z})$, $\boldsymbol{J} \doteq \boldsymbol{J}_G\big(\boldsymbol{\theta}^0\big)$, and $\boldsymbol{c} \doteq \boldsymbol{\theta} - \boldsymbol{\theta}^0$. So, the least-squares objective in Eq. (4) is equivalent to

$$\|\widehat{\boldsymbol{y}} - \boldsymbol{J}\boldsymbol{c}\|_2^2 \tag{8}$$

and the gradient update reads

$$\boldsymbol{c}^t = \boldsymbol{c}^{t-1} - \eta \boldsymbol{J}^\intercal \left(\boldsymbol{J}\boldsymbol{c}^{k-1} - \widehat{\boldsymbol{y}}\right), \tag{9}$$

where $\boldsymbol{c}^0 = \boldsymbol{0}$ and $\boldsymbol{x}^t = \boldsymbol{J}\boldsymbol{c}^t + G_{\boldsymbol{\theta}^0}(\boldsymbol{z})$. The residual at time t can be computed as

$$\begin{aligned}
\boldsymbol{r}^t &\doteq \widehat{\boldsymbol{y}} - \boldsymbol{J}\boldsymbol{c}^t \qquad (10)\\
&= \widehat{\boldsymbol{y}} - \boldsymbol{J}\left(\boldsymbol{c}^{t-1} - \eta \boldsymbol{J}^\intercal \left(\boldsymbol{J}\boldsymbol{\theta}^{t-1} - \widehat{\boldsymbol{y}}\right)\right) \qquad (11)\\
&= \left(\boldsymbol{I} - \eta \boldsymbol{J}\boldsymbol{J}^\intercal\right)\left(\widehat{\boldsymbol{y}} - \boldsymbol{J}\boldsymbol{c}^{t-1}\right) \qquad (12)\\
&= \left(\boldsymbol{I} - \eta \boldsymbol{J}\boldsymbol{J}^\intercal\right)^2\left(\widehat{\boldsymbol{y}} - \boldsymbol{J}\boldsymbol{c}^{t-2}\right) = \dots \qquad (13)\\
&= \left(\boldsymbol{I} - \eta \boldsymbol{J}\boldsymbol{J}^\intercal\right)^t\left(\widehat{\boldsymbol{y}} - \boldsymbol{J}\boldsymbol{c}^{0}\right) \quad (\text{using } \boldsymbol{c}^0 = \boldsymbol{0}) \qquad (14)\\
&= \left(\boldsymbol{I} - \eta \boldsymbol{J}\boldsymbol{J}^\intercal\right)^t \widehat{\boldsymbol{y}}. \qquad (15)
\end{aligned}$$

Assume that the SVD of $\boldsymbol{J}$ is as $\boldsymbol{J} = \boldsymbol{W}\boldsymbol{\Sigma}\boldsymbol{V}^\intercal$. Then

$$\boldsymbol{r}^t = \left(\boldsymbol{I} - \eta \boldsymbol{W}\boldsymbol{\Sigma}^2\boldsymbol{W}^\intercal\right)^t \widehat{\boldsymbol{y}} = \sum_i \left(1 - \eta\sigma_i^2\right)^t \boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}} \boldsymbol{w}_i \tag{16}$$

and so

$$\boldsymbol{J}\boldsymbol{c}^t = \widehat{\boldsymbol{y}} - \boldsymbol{r}^t = \sum_i \left(1 - \left(1 - \eta\sigma_i^2\right)^t\right) \boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}} \boldsymbol{w}_i. \tag{17}$$

Consider a set of W vectors $\mathcal{V} = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_W\}$. We have the empirical variance.

$$\mathrm{VAR}(\mathcal{V}) = \frac{1}{W}\sum_{w=1}^{W}\left\|\boldsymbol{v}_w - \frac{1}{W}\sum_{j=1}^{W}\boldsymbol{v}_j\right\|_2^2 = \frac{1}{W}\sum_{w=1}^{W}\|\boldsymbol{v}_w\|_2^2 - \left\|\frac{1}{W}\sum_{w=1}^{W}\boldsymbol{v}_w\right\|_2^2. \tag{18}$$

Therefore, the variance of the set $\left\{\boldsymbol{x}^t, \boldsymbol{x}^{t+1}, \dots, \boldsymbol{x}^{t+W-1}\right\}$, same as the variance of the set $\left\{\boldsymbol{J}\boldsymbol{c}^t, \boldsymbol{J}\boldsymbol{c}^{t+1}, \dots, \boldsymbol{J}\boldsymbol{c}^{t+W-1}\right\}$, can be calculated as

$$\frac{1}{W}\sum_{w=0}^{W-1}\sum_i \left(\boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}}\right)^2 \left(1 - \left(1 - \eta\sigma_i^2\right)^{t+w}\right)^2 - \frac{1}{W^2}\sum_i \left(\boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}}\right)^2 \left(\sum_{w=0}^{W-1} 1 - \left(1 - \eta\sigma_i^2\right)^{t+w}\right)^2 \tag{19}$$

$$= \frac{1}{W^2}\sum_i \left(\boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}}\right)^2 \left[W\sum_{w=0}^{W-1}\left(1 - \left(1 - \eta\sigma_i^2\right)^{t+w}\right)^2 - \left(\sum_{w=0}^{W-1} 1 - \left(1 - \eta\sigma_i^2\right)^{t+w}\right)^2\right] \tag{20}$$

$$\begin{aligned}
= \frac{1}{W^2}\sum_i \left(\boldsymbol{w}_i^\intercal \widehat{\boldsymbol{y}}\right)^2 \Bigg[&\left(W^2 + W\frac{(1-\eta\sigma_i^2)^{2t}(1-(1-\eta\sigma_i^2)^{2W})}{1-(1-\eta\sigma_i^2)^2} - 2W\frac{(1-\eta\sigma_i^2)^{t}(1-(1-\eta\sigma_i^2)^{W})}{\eta\sigma_i^2}\right)\\
&- \left(W^2 - 2W\frac{(1-\eta\sigma_i^2)^{t}(1-(1-\eta\sigma_i^2)^{W})}{\eta\sigma_i^2} + \frac{\left(1-\eta\sigma_i^2\right)^{2t}\left(1-\left(1-\eta\sigma_i^2\right)^{W}\right)^2}{\eta^2\sigma_i^4}\right)\Bigg]
\end{aligned} \tag{21}$$

$$= \frac{1}{W^2}\sum_i \left\langle \boldsymbol{w}_i, \widehat{\boldsymbol{y}}\right\rangle^2 \frac{(1-\eta\sigma_i^2)^{2t}}{\eta\sigma_i^2}\left[W\frac{1-(1-\eta\sigma_i^2)^{2W}}{2-\eta\sigma_i^2} - \frac{(1-(1-\eta\sigma_i^2)^W)^2}{\eta\sigma_i^2}\right]. \tag{22}$$

So the constants C_{W,η,σ_i}'s are defined as

$$C_{W,\eta,\sigma_i} \doteq \frac{1}{W^2\eta\sigma_i^2}\left[W\frac{1-(1-\eta\sigma_i^2)^{2W}}{2-\eta\sigma_i^2} - \frac{(1-(1-\eta\sigma_i^2)^W)^2}{\eta\sigma_i^2}\right]. \tag{23}$$

To see they are nonnegative, it is sufficient to show that

$$W\frac{1-(1-\eta\sigma_i^2)^{2W}}{2-\eta\sigma_i^2}-\frac{(1-(1-\eta\sigma_i^2)^W)^2}{\eta\sigma_i^2}\geq 0$$
$$\iff \eta\sigma_i^2 W\big(1-(1-\eta\sigma_i^2)^{2W}\big)-\big(2-\eta\sigma_i^2\big)(1-(1-\eta\sigma_i^2)^W)^2\geq 0. \quad (24)$$

Now consider the function.

$$h(\xi,W)=\xi W\big(1-(1-\xi)^{2W}\big)-(2-\xi)(1-(1-\xi)^W)^2 \quad \xi\in[0,1], W\geq 1. \quad (25)$$

First, one can easily check that $\partial_W h(\xi,W)\geq 0$ for all $W\geq 1$ and all $\xi\in[0,1]$, that is, $h(\xi,W)$ increases monotonically with respect to W. Thus, to prove $C_{W,\eta,\sigma_i}\geq 0$, it suffices to show that $h(\xi,1)\geq 0$. Now

$$h(\xi,1)=\xi\big(1-(1-\xi)^2\big)-(2-\xi)\xi^2=0, \quad (26)$$

completing the proof. □

A.3 Proof of 2.2

We first re-state Theorem 2 in Heckel & Soltanolkotabi (2020b).

Theorem A.1 (Heckel & Soltanolkotabi (2020b))**.** *Let $\boldsymbol{x}\in\mathbb{R}^n$ be a signal in the span of the first p trigonometric basis functions, and consider a noisy observation $\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{n}$, where the noise $\boldsymbol{n}\sim\mathcal{N}\left(\boldsymbol{0},\xi^2/n\cdot\boldsymbol{I}\right)$. To denoise this signal, we fit a two-layer generator network $G_{\boldsymbol{C}}(\boldsymbol{B})=\mathrm{ReLU}(\boldsymbol{UBC})\boldsymbol{v}$, where $\boldsymbol{v}=[1,\ldots,1,-1,\ldots,-1]/\sqrt{k}$, and $\boldsymbol{B}\sim_{iid}\mathcal{N}(0,1)$, and $\boldsymbol{U}$ is an upsampling operator that implements circular convolution with a given kernel $\boldsymbol{u}$. Denote $\boldsymbol{\sigma}\doteq\|\boldsymbol{u}\|_2|\boldsymbol{F}g(\boldsymbol{u}\circledast\boldsymbol{u}/\|\boldsymbol{u}\|_2^2)|^{1/2}$ where $g(t)=(1-\cos^{-1}(t)/\pi)t$ and $\circledast$ denote the circular convolution. Fix any $\varepsilon\in(0,\sigma_p/\sigma_1]$, and suppose that $k\geq C_{\boldsymbol{u}}n/\varepsilon^8$, where $C_{\boldsymbol{u}}>0$ is a constant depending only on $\boldsymbol{u}$. Consider gradient descent with step size $\eta\leq\|\boldsymbol{Fu}\|_\infty^{-2}$($\boldsymbol{Fu}$ is the Fourier transform of $\boldsymbol{u}$) starting from $\boldsymbol{C}_0\sim_{iid}\mathcal{N}\left(0,\omega^2\right)$, entries $\omega\propto\frac{\|\boldsymbol{y}\|_2}{\sqrt{n}}$. Then, for all iterations t obeying $t\leq\frac{100}{\eta\sigma_p^2}$, the reconstruction error obeys*

$$\|G_{\boldsymbol{C}^t}(\boldsymbol{B})-\boldsymbol{x}\|_2\leq\left(1-\eta\sigma_p^2\right)^t\|\boldsymbol{x}\|_2+\sqrt{\sum_{i=1}^n((1-\eta\sigma_i^2)^t-1)^2(\boldsymbol{w}_i^\intercal\boldsymbol{n})^2}+\varepsilon\|\boldsymbol{y}\|_2$$

with probability at least $1-\exp\left(-k^2\right)-n^{-2}$.

Note that since $\boldsymbol{B}\sim_{iid}\mathcal{N}(0,1)$ and hence is full-rank with probability one, the original Theorem 1 & 2 of Heckel & Soltanolkotabi (2020b) rename $\boldsymbol{BC}$ to $\boldsymbol{C}'$ and state the result directly on $\boldsymbol{C}'$, that is, assume that the model is $\mathrm{ReLU}(\boldsymbol{UC}')\boldsymbol{v}$. It is easy to see that the original theorems imply the version stated here.

With this, we can obtain our Theorem 2.2, stated in full technical form here:

Theorem A.2. *Let $\boldsymbol{x}\in\mathbb{R}^n$ be a signal in the span of the first p trigonometric basis functions, and consider a noisy observation $\boldsymbol{y}=\boldsymbol{x}+\boldsymbol{n}$, where the noise $\boldsymbol{n}\sim\mathcal{N}\left(\boldsymbol{0},\xi^2/n\cdot\boldsymbol{I}\right)$. To denoise this signal, we fit a two-layer generator network $G_{\boldsymbol{C}}(\boldsymbol{B})=\mathrm{ReLU}(\boldsymbol{UBC})\boldsymbol{v}$, where $\boldsymbol{v}=[1,\ldots,1,-1,\ldots,-1]/\sqrt{k}$, and $\boldsymbol{B}\sim_{iid}\mathcal{N}(0,1)$, and $\boldsymbol{U}$ is an upsampling operator that implements circular convolution with a given kernel $\boldsymbol{u}$. Denote $\boldsymbol{\sigma}\doteq\|\boldsymbol{u}\|_2|\boldsymbol{F}g(\boldsymbol{u}\circledast\boldsymbol{u}/\|\boldsymbol{u}\|_2^2)|^{1/2}$ where $g(t)=(1-\cos^{-1}(t)/\pi)t$ and $\circledast$ denotes the circular convolution. Fix any $\varepsilon\in(0,\sigma_p/\sigma_1]$, and suppose $k\geq C_{\boldsymbol{u}}n/\varepsilon^8$, where $C_{\boldsymbol{u}}>0$ is a constant only depending on $\boldsymbol{u}$. Consider gradient descent with step size $\eta\leq\|\boldsymbol{Fu}\|_\infty^{-2}$($\boldsymbol{Fu}$ is the Fourier transform of $\boldsymbol{u}$) starting from $\boldsymbol{C}_0\sim_{iid}\mathcal{N}\left(0,\omega^2\right)$, entries $\omega\propto\frac{\|\boldsymbol{y}\|_2}{\sqrt{n}}$. Then, for all iterates t obeying $t\leq\frac{100}{\eta\sigma_p^2}$, our WMV obeys*

$$\mathrm{WMV}\leq\frac{12}{W}\|\boldsymbol{x}\|_2^2\frac{\left(1-\eta\sigma_p^2\right)^{2t}}{1-(1-\eta\sigma_p^2)^2}+12\sum_{i=1}^n\left(\left(1-\eta\sigma_i^2\right)^{t+W-1}-1\right)^2(\boldsymbol{w}_i^\intercal\boldsymbol{n})^2+12\varepsilon^2\|\boldsymbol{y}\|_2^2 \quad (27)$$

with probability at least $1-\exp\left(-k^2\right)-n^{-2}$.

Proof. We make use of the basic inequality: $\|\boldsymbol{a}-\boldsymbol{b}\|_2^2 \le 2\|\boldsymbol{a}\|_2^2 + 2\|\boldsymbol{b}\|_2^2$ for any two vectors $\boldsymbol{a}, \boldsymbol{b}$ of compatible dimension. We have

$$\frac{1}{W}\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \frac{1}{W}\sum_{j=0}^{W-1} G_{\boldsymbol{C}^{t+j}}(\boldsymbol{B})\|_2^2 \tag{28}$$

$$= \frac{1}{W}\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x} + \boldsymbol{x} - \frac{1}{W}\sum_{j=0}^{W-1} G_{\boldsymbol{C}^{t+j}}(\boldsymbol{B})\|_2^2 \tag{29}$$

$$\le \left(\frac{2}{W}\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2\right) + 2\|\boldsymbol{x} - \frac{1}{W}\sum_{j=0}^{W-1} G_{\boldsymbol{C}^{t+j}}(\boldsymbol{B})\|_2^2 \tag{30}$$

$$\le \frac{2}{W}\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2 + \frac{2}{W}\sum_{j=0}^{W-1} \|G_{\boldsymbol{C}^{t+j}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2 \tag{31}$$

$$(\boldsymbol{z} \mapsto \|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \text{ convex and Jensen's inequality})$$

$$= \frac{4}{W}\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2. \tag{32}$$

In view of Theorem A.1,

$$\|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2 \le 3\left(1-\eta\sigma_p^2\right)^{2t+2w}\|\boldsymbol{x}\|_2^2 + 3\sum_{i=1}^{n}\left(\left(1-\eta\sigma_j^2\right)^{t+w} - 1\right)^2 (\boldsymbol{w}_i^\intercal \boldsymbol{n})^2 + 3\varepsilon^2\|\boldsymbol{y}\|_2^2. \tag{33}$$

Thus,

$$\sum_{w=0}^{W-1} \|G_{\boldsymbol{C}^{t+w}}(\boldsymbol{B}) - \boldsymbol{x}\|_2^2$$

$$\le 3\|\boldsymbol{x}\|_2^2 \sum_{w=0}^{W-1}\left(1-\eta\sigma_p^2\right)^{2t+2w} + 3\sum_{w=0}^{W-1}\sum_{i=1}^{n}\left(\left(1-\eta\sigma_i^2\right)^{t+w} - 1\right)^2 (\boldsymbol{w}_i^\intercal \boldsymbol{n})^2 + 3W\varepsilon^2\|\boldsymbol{y}\|_2^2 \tag{34}$$

$$\le 3\|\boldsymbol{x}\|_2^2 \frac{\left(1-\eta\sigma_p^2\right)^{2t}(1-(1-\eta\sigma_p^2)^{2W})}{1-(1-\eta\sigma_p^2)^2} + 3W\sum_{i=1}^{n}\left(\left(1-\eta\sigma_i^2\right)^{t+W-1} - 1\right)^2 (\boldsymbol{w}_i^\intercal \boldsymbol{n})^2 + 3W\varepsilon^2\|\boldsymbol{y}\|_2^2 \tag{35}$$

$$\le 3\|\boldsymbol{x}\|_2^2 \frac{\left(1-\eta\sigma_p^2\right)^{2t}}{1-(1-\eta\sigma_p^2)^2} + 3W\sum_{i=1}^{n}\left(\left(1-\eta\sigma_i^2\right)^{t+W-1} - 1\right)^2 (\boldsymbol{w}_i^\intercal \boldsymbol{n})^2 + 3W\varepsilon^2\|\boldsymbol{y}\|_2^2, \tag{36}$$

completing the proof. □

A.4 ES-EMV algorithm

The exponential moving variance version of our method is summarized in Algorithm 2.

A.5 More details on major DIP variants

Deep Decoder (Heckel & Hand, 2019) differs from DIP mainly in terms of network architecture: It is typically a *under-parameterized* network consisting mainly of 1 × 1 convolutions, upsampling, ReLU and channel-wise normalization layers, while DIP uses an *over-parameterized*, U-net like convolutional network.

GP-DIP (Cheng et al., 2019) uses the original DIP (Ulyanov et al., 2018) network and formulation, but replaces stochastic gradient descent (SGD) by stochastic gradient Langevin dynamics (SGLD) in the gradient update step. i.e., for the generic gradient step for optimizing Eq. (2) reads:

$$\boldsymbol{\theta}^+ = \boldsymbol{\theta} - t\nabla_{\boldsymbol{\theta}}[\ell(\boldsymbol{y}, f(G_{\boldsymbol{\theta}}(\boldsymbol{z}))) + \lambda R(G_{\boldsymbol{\theta}}(\boldsymbol{z}))] + \eta \tag{37}$$

where η is zero-mean Gaussian with an isotropic variance level t.

Algorithm 2 DIP with ES–EMV

```
Input: random seed $\boldsymbol{z}$, randomly-initialized $G_{\boldsymbol{\theta}}$, forgetting factor $\alpha \in (0,1)$, patience number $P$, iteration
    counter $k = 0$, $\mathrm{EMA}^0 = 0$, $\mathrm{EMV}^0 = 0$, $\mathrm{EMV}_{\min} = \infty$
Output: reconstruction $\boldsymbol{x}^*$
while not stopped do
    update $\boldsymbol{\theta}$ via Eq. (2) to obtain $\boldsymbol{\theta}^{k+1}$ and $\boldsymbol{x}^{k+1}$
    $\mathrm{EMA}^{k+1} = (1-\alpha)\mathrm{EMA}^k + \alpha \boldsymbol{x}^{k+1}$
    $\mathrm{EMV}^{k+1} = (1-\alpha)\mathrm{EMV}^k + \alpha(1-\alpha)\|\boldsymbol{x}^{k+1} - \mathrm{EMA}^k\|_2^2$
    if $\mathrm{EMV}^{\mathrm{k+1}} < \mathrm{EMV}_{\min}$ then
        $\mathrm{EMV}_{\min} \leftarrow \mathrm{EMV}^{\mathrm{k+1}}$, $\boldsymbol{x}^* \leftarrow \boldsymbol{x}^{k+1}$
    end if
    if $\mathrm{EMV}_{\min}$ stagnates for $P$ iterations then
        stop and return $\boldsymbol{x}^*$
    end if
    $k = k + 1$
end while
```

DIP-TV (Cascarano et al., 2021) uses the original DIP (Ulyanov et al., 2018) network, with a Total Variation (TV) regularizer added. Then, the proposed objective is solved with the Alternating Direction Method of Multipliers (ADMM) framework.

SIREN (Sitzmann et al., 2020) treats the object directly as a continuous function on $\mathbb{R}^2$ or $\mathbb{R}^3$ (or higher-dimensional spaces depending on the application) and hence parameterizes it as a multi-layer perceptron (MLP): 1) the input to SIREN is the 2D/3D coordinate of each pixel instead of random values, and 2) the network uses a sinusoidal activation function instead of the commonly used ReLU. When substituting the DIP network with SIREN and solve Eq. (2) problems, similar overfitting issue is still observed.

A.6 More details on major ES methods

Here, we provide more details on the main competing methods.

Spectral Bias (SB) Shi et al. (2022) operates on deep decoder models and proposes two modifications to change the spectral bias: (1) controlling the operator norm of the weight $\boldsymbol{w}$ for each convolutional layer by normalization

$$\boldsymbol{w}' = \frac{\boldsymbol{w}}{\max\left(1, \|\boldsymbol{w}\|_{\mathrm{op}}/\lambda\right)}, \tag{38}$$

ensuring that $\|\boldsymbol{w}'\|_{\mathrm{op}} \leq \lambda$, which in turn controls the Fourier spectrum of the underlying function represented by the layer; (2) performing Gaussian upsampling instead of the typical bilinear upsampling to suppress the smoothness effect of the latter. These two modifications with appropriate parameter setting (λ, and σ in Gaussian filtering) can improve the learning of the high-frequency components by deep decoder, and allow the blurriness-over-sharpness stopping criterion.

$$\Delta r\left(\boldsymbol{x}^t\right) = \frac{1}{W}\left|\sum_{w=1}^{W} r\left(\boldsymbol{x}^{t-w}\right) - \sum_{w=1}^{W} r\left(\boldsymbol{x}^{t-W-w}\right)\right|, \tag{39}$$

where $r(\boldsymbol{x}') = B(\boldsymbol{x}')/S(\boldsymbol{x}')$, and $B(\cdot)$ and $S(\cdot)$ are the blurriness and sharpness metrics in Crete et al. (2007) and Bahrami & Kot (2014), respectively. In other words, the criterion in Eq. (39) measures the change in the average blurriness-over-sharpness ratios in consecutive windows of size W, and small changes indicate good ES points. But, as mentioned, this criterion only works for modified DD models and not for other DIP variants, as acknowledged by the authors in Shi et al. (2022) and confirmed in our experiment (see Sec. 3.1).

DF-STE Jo et al. (2021) targets Gaussian denoising with known noise levels (i.e. $\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n}$, where n is the i.i.d. Gaussian noise) and considers the objective.

$$\min_{\boldsymbol{\theta}} \frac{1}{n^2} \|\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{y})\|_F^2 + \frac{\sigma^2}{n^2} \operatorname{tr} \boldsymbol{J}_{G_{\boldsymbol{\theta}}}(\boldsymbol{y}), \tag{40}$$

where $\operatorname{tr} \boldsymbol{J}_{G_{\boldsymbol{\theta}}}(\boldsymbol{y})$ is the trace of the network Jacobian with respect to the input, that is, the divergence term in Jo et al. (2021). The divergence term is a proxy for controlling the capacity of the network. The paper then proposes a heuristic zero-crossing stopping criterion that stops the iteration when the loss starts to cross zero into negative values. Although the idea works reasonably well on Gaussian denoising with low and known noise level (the variance level σ^2 is explicitly needed in the regularization parameter ahead of the divergence term), it starts to break down when the noise level increases even if the right noise level is provided; see Sec. 3.1. Also, although the paper has extended the formulation to handle Poisson noise, it is unclear how to generalize the idea for handling other types of noise, as well as how to move beyond simple additive denoising problems.

SV-ES Li et al. (2021) proposes training an autoencoder online using the reconstruction sequence $\{\boldsymbol{x}^t\}_{t \geq 1}$:

$$\min_{\boldsymbol{w}, \boldsymbol{v}} \sum_{t \geq 1} \ell_{\mathrm{AE}}\left(\boldsymbol{x}^t, D_{\boldsymbol{w}} \circ E_{\boldsymbol{v}}\left(\boldsymbol{x}^t\right)\right). \tag{41}$$

Any new $\boldsymbol{x}^t$ passes through the current autoencoder and the reconstruction error ℓ_{AE} is recorded. They observe that the error curve typically follows a U-shaped shape and that the valley of the curve is approximately aligned with the peak of the PNSR curve. Therefore, they design an ES method by detecting the valley of the error curve. This method works reasonably well for different IPs and different DIP variants. A major drawback is efficiency: the overhead caused by the online training of the autoencoder is on an order of magnitude higher than the cost of the DIP update itself, as shown in Tab. 3.

DOP You et al. (2020) considers only additive sparse noise (e.g., salt and pepper noise) and proposes modeling the clean image and noise explicitly in the objective:

$$\min_{\boldsymbol{\theta}, \boldsymbol{g}, \boldsymbol{h}} \|\boldsymbol{y} - G_{\boldsymbol{\theta}}(\boldsymbol{z}) - (\boldsymbol{g} \circ \boldsymbol{g} - \boldsymbol{h} \circ \boldsymbol{h})\|_F^2, \tag{42}$$

where the overparameterized term $\boldsymbol{g} \circ \boldsymbol{g} - \boldsymbol{h} \circ \boldsymbol{h}$ ($\circ$ denotes the Hadamard product) is meant to capture sparse noise, where a similar idea has been shown to be effective for sparse recovery in Vaskevicius et al. (2019). Different properly tuned learning rates for the clean image and sparse noise terms are necessary for success. The downside includes the prolonged running time, as it pushes the peak reconstruction to the very last iteration, and the difficulty to extend the idea to other types of noise.

A.7 Additional experimental details & results

A.7.1 External codes

- DIP: `https://github.com/DmitryUlyanov/deep-image-prior`
- Deep decoder: `https://github.com/reinhardh/supplement_deep_decoder`
- DIP-TV: `https://github.com/sedaboni/ADMM-DIPTV`
- GP-DIP: `https://people.cs.umass.edu/~zezhoucheng/gp-dip/`
- DF-STE: `https://github.com/gistvision/dip-denosing`
- SV-ES: `https://github.com/sun-umn/Self-Validation`
- DOP: `https://github.com/ChongYou/robust-image-recovery`
- SB: `https://github.com/shizenglin/Measure-and-Control-Spectral-Bias`
- CBSD68: `https://github.com/clausmichele/CBSD68-dataset`

A.7.2 Experiment Settings

Our default setup for all experiments is as follows. Our DIP model is the original from Ulyanov et al. (2018); the optimizer is ADAM with a learning rate 0.01. For all other models, we use their default architectures, optimizers, and hyperparameters. For ES-WMV, the default window size $W = 100$, and the patience number $P = 1000$. We use both PSNR and SSIM to access the reconstruction quality and report PSNR and SSIM gaps (the difference between our detected and peak numbers) as an indicator of our detection performance. **For most experiments, we repeat the experiments 3 times to report the mean and standard deviation**; when not, we explain why.

Noise generation Following the noise generation rules of Hendrycks & Dietterich (2019)[1], we simulate four types of noise and three intensity levels for each type of noise. The detailed information is as follows. **Gaussian noise:** 0 mean additive Gaussian noise with variance 0.12, 0.18 and 0.26 for low, medium and high noise levels, respectively; **Impulse noise:** also known as salt-and-pepper noise, replacing each pixel with probability $p \in [0, 1]$ in a white or black pixel with half chance each. Low, medium and high noise levels correspond to $p = 0.3, 0.5, 0.7$, respectively; **Speckle noise:** for each pixel $x \in [0, 1]$, the noisy pixel is $x(1 + \varepsilon)$, where ε is zero-mean Gaussian with a variance level 0.20, 0.35, 0.45 for low, medium, and high noise levels, respectively; **Shot noise:** also known as Poisson noise. For each pixel, $x \in [0, 1]$, the noisy pixel is Poisson distributed with the rate λx, where λ is $25, 12, 5$ for low, medium, and high noise levels, respectively.

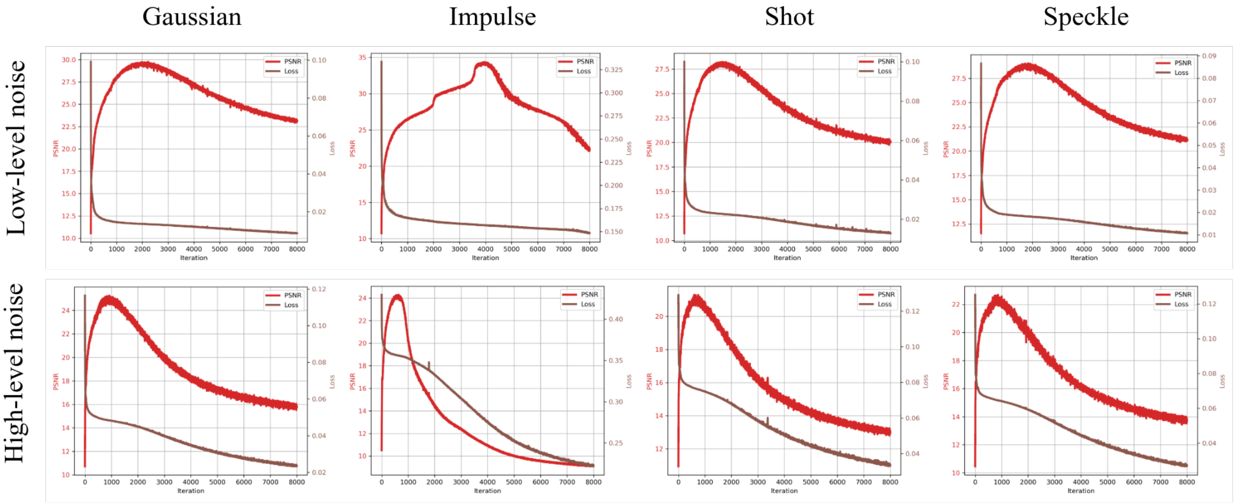


Figure 15: Our ES-WMV method on DIP for denoising "F16" with different noise types and levels (top: low-level noise; bottom: high-level noise). Red curves are PSNR curves, and brown curves are loss curves.

A.7.3 Denoising examples

We explore the possibility of using the fitting loss for ES here, but we are unable to find correlations between the trend of the loss and that of the PSNR curve, shown in Fig. 15

A.7.4 Comparison with baseline methods

To further compare with baseline methods, we report the PSNR gaps in high-level noise cases and the SSIM gaps in low- and high-level noise cases in Fig. 17,Fig. 18 and Fig. 19, respectively, which show a trend similar to the results of PSNR gaps. The detection gaps of our method are very marginal (< 0.02) for most types and levels of noise (except Baboon and Kodak1 for certain types / levels of noise), while the baseline methods

[1] https://github.com/hendrycks/robustness

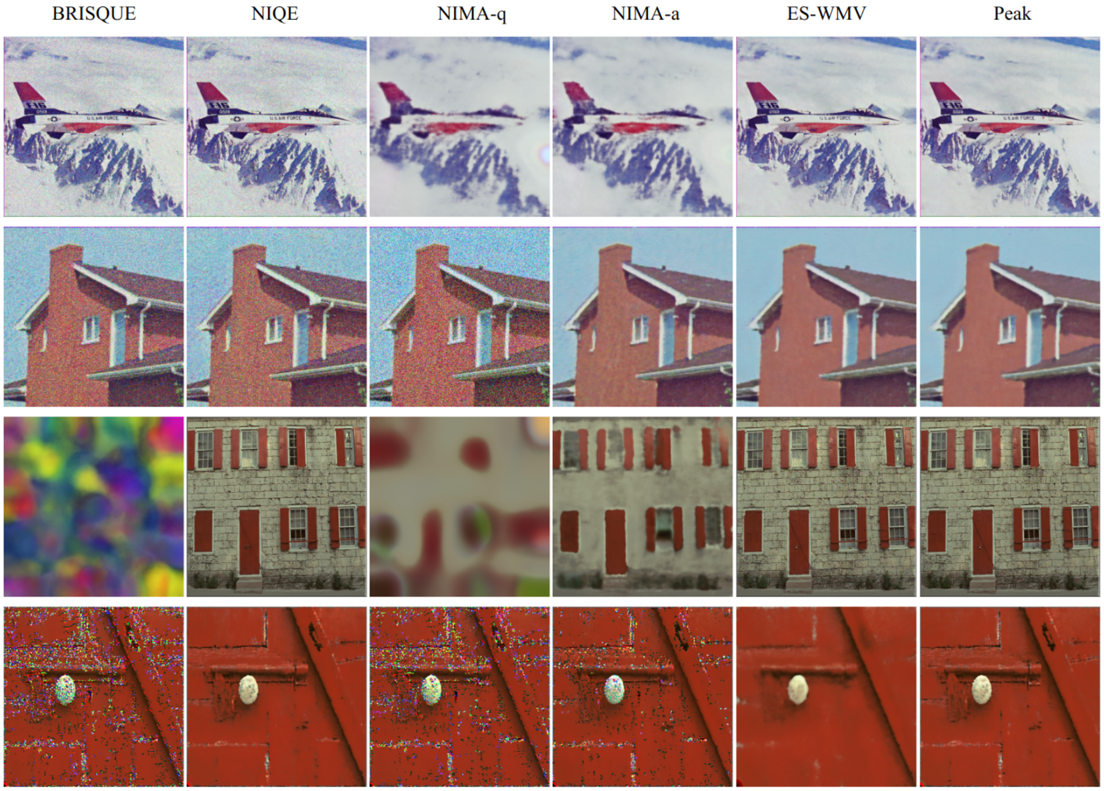


Figure 16: Visual comparisons of NR-IQMs and ES-WMV. From top to bottom: Gaussian noise (low), Gaussian noise (high), impulse noise (low), impulse noise (high).

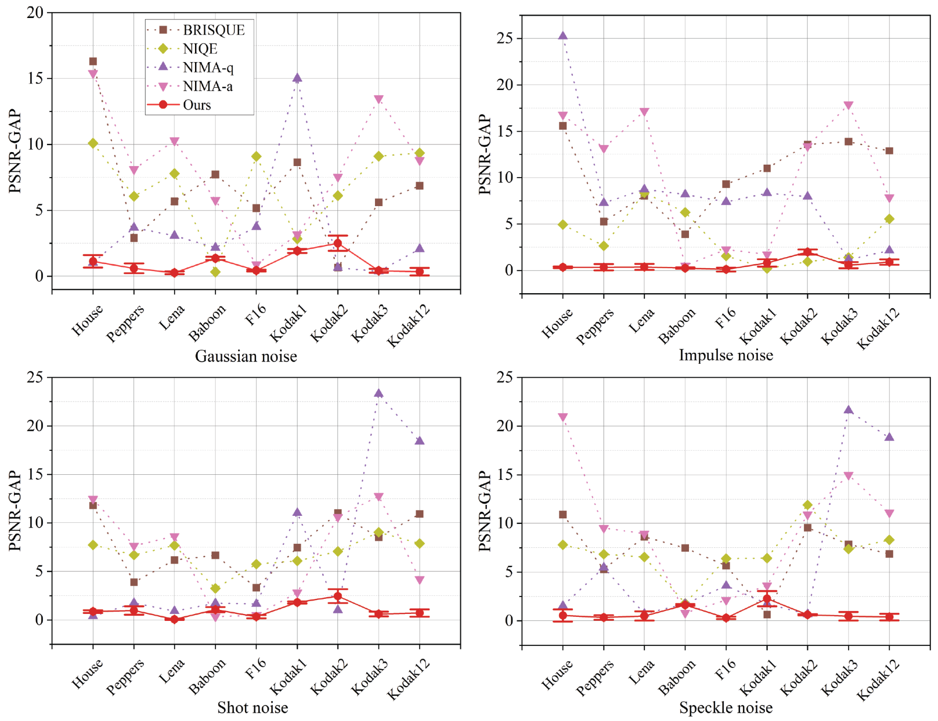


Figure 17: **High-level noise** detection performance in terms of PSNR gaps. For NIMA, we report both technical quality assessment (NIMA-q) and aesthetic assessment (NIMA-a). Smaller PSNR gaps are better.

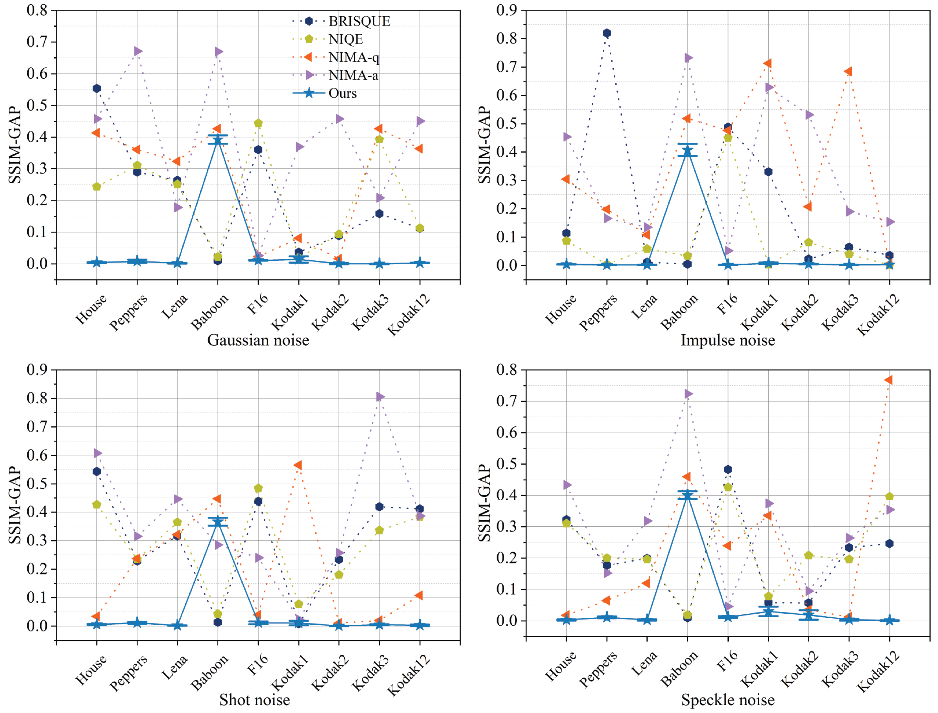


Figure 18: **Low-level noise** detection performance in terms of SSIM gaps. For NIMA, we report both technical quality assessment (NIMA-q) and aesthetic assessment (NIMA-a). Smaller SSIM gaps are better.

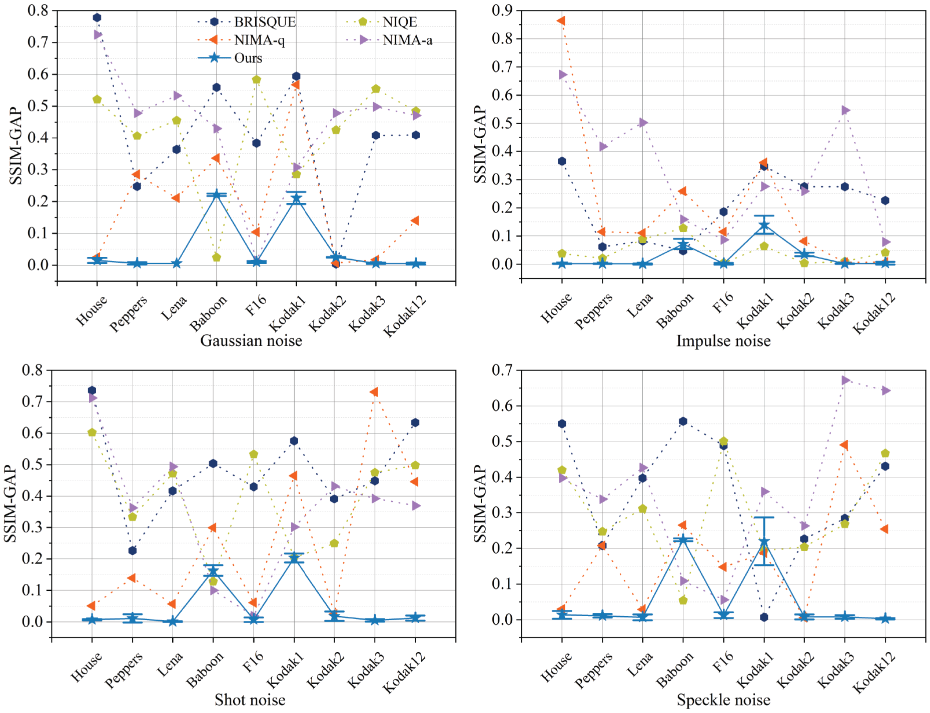


Figure 19: **High-level noise** detection performance in terms of SSIM gaps. For NIMA, we report both technical quality assessment (NIMA-q) and aesthetic assessment (NIMA-a). Smaller SSIM gaps are better.

can easily exceed 0.1 for most cases. In addition, we provide some visual detection results in Figs. 7 and 16. Our ES-WMV significantly outperforms the four baseline methods visually.

A.7.5 Comparison with competing methods

Comparison between ES-WMV with DF-STE for Gaussian and shot noise on the 9 image dataset in terms of SSIM is reported in Fig. 20. Furthermore, we also test our ES-WMV and DF-STE on CBSD68 in Tab. 7. Our ES-WMV wins in high-level noise cases but lags behind DF-STE in the low-level cases. The gaps between our ES-WMV and DF-STE for all noise levels mostly come from the peak performance between the original DIP and DF-STE—modifications in DF-STE have affected peak performance, positively for low-level cases and negatively for high-level cases, not much from our ES method, as evident from the uniformly small detection gaps reported in Tab. 7. Moreover, DF-STE can only handle Gaussian and Poisson noise for denoising, and the exact noise level is a required hyperparameter for their method to work.

Then we compare our ES-WMV and SV-ES in Fig. 21. The DIP results with ES-WMV versus DOP in impulse noise are shown in Tab. 8. For SB, part of the qualitative detection results on the 9 images[2] are reported in Fig. 22.

For reference, we compare DIP with the recent one-shot methods based on diffusion models for solving linear IPs—DDNM+ for image denoising, as shown in Tab. 9. Like for Tab. 5, we observe that (1) **Our ES-WMV is again able to detect near-peak performance for most images**: the average PSNR gap is ≤ 1.02 and the average SSIM gap is ≤ 0.08; (2) DDNM+ is sensitive to the noise type and level: from Tab. 9, DDNM+ outperforms DIP and DIP+ES-WMV when there is Gaussian noise, but this is when the noise level set for pretraining DDNM+ matches the true noise level, i.e., $\sigma_y = 0.18$, **which is unrealistic in practice as the noise level is not known beforehand**. When the noise level is not set correctly, e.g., as $\sigma_y = 0$ in the DDNM+ ($\sigma_y = .00$) row of Tab. 9, the performance of DDNM+ is much worse than that of DIP and DIP+ES-WMV. Also, for impluse noise denoising, DIP is also a clear winner that leads DDNM+

[2] http://www.cs.tut.fi/~foi/GCF-BM3D/index.html#ref_results

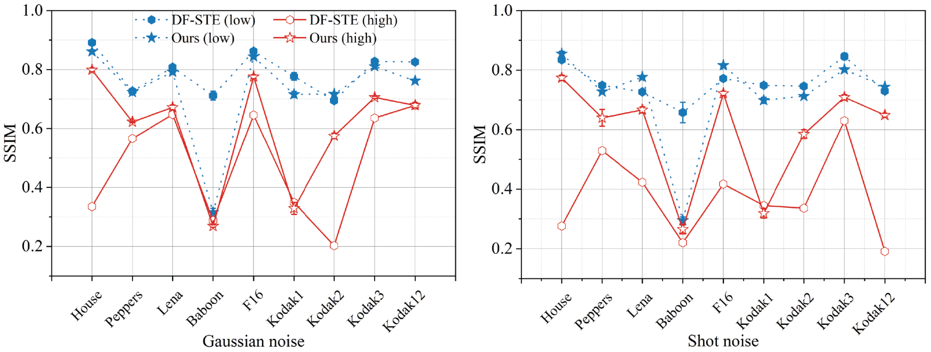


Figure 20: Comparison of DF-STE and ES-WMV for Gaussian and shot noise in terms of SSIM.

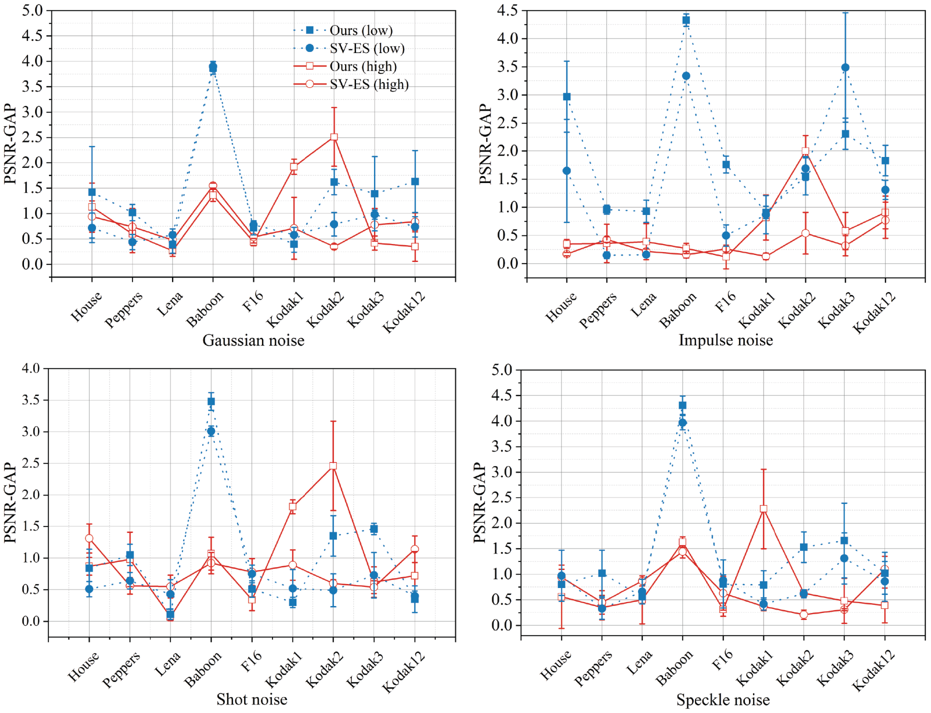


Figure 21: **Low- and high-level noise** detection performance of SV-ES and ours in terms of PSNR gaps.

by a large margin; and (3) in Appendix A.8, we show that DDNM+ may also suffer from the overfitting issue and our ES-WMV can help DDNM+ to stop around the performance peak as well.

Table 7: Comparison between ES-WMV and DF-STE for image denoising on the CBSD68 dataset with varying noise level σ: mean and (std). PSNR gaps below 1.0 are colored as red.

	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
ES-WMV	28.7(3.2)	27.4(2.6)	24.2(2.3)
DIP (Peak)	29.7(3.0)	28.0(2.4)	24.9(2.3)
PSNR Gap	1.0(0.7)	0.7(0.5)	0.7(0.5)
DF-STE	31.4(1.8)	28.4(2.2)	21.1(2.5)

Table 8: DIP with ES-WMV vs. DOP on impulse noise: mean and (std).

	Low Level		High Level	
	PSNR	SSIM	PSNR	SSIM
DIP-ES	31.64 (5.69)	0.85 (0.18)	24.74 (3.23)	0.67 (0.19)
DOP	32.12 (4.52)	0.92 (0.07)	27.34 (3.78)	0.86 (0.10)

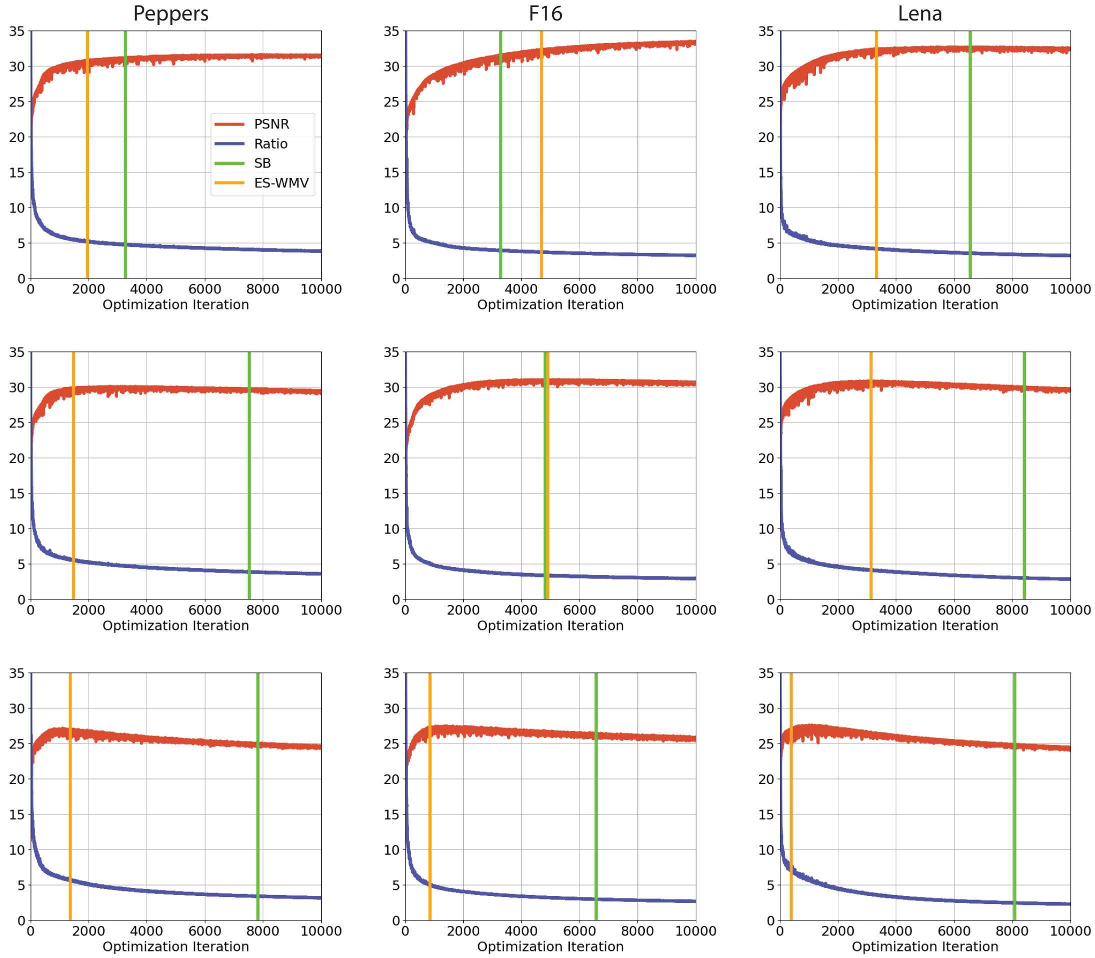


Figure 22: Comparison between ES-WMV and SB for image denoising (top: $\sigma = 15$; middle: $\sigma = 25$; bottom: $\sigma = 50$). The red and blue curves are the PNSR and the ratio metric curves. The orange and green bars indicate the ES points detected by our ES-WMV and SB, respectively.

Table 9: Comparison of ES-WMV for DIP and DDNM+ Wang et al. (2022) for **denoising images with medium-level Gaussian and impulse noise**: mean and (std). The highest PSNR and SSIM for each task are in red. In particular, we set the best hyperparameter for DDNM+ ($\sigma_y = 0.18$), **which is unfair for the DIP + ES-WMV combination as we fix its hyperparameter setting**.

	PSNR		SSIM	
	Gaussian	Impulse	Gaussian	Impulse
DIP (peak)	24.63 (2.06)	37.75 (3.32)	0.68 (0.06)	0.96 (0.10)
DIP + ES-WMV	23.61 (2.67)	36.87 (4.29)	0.60 (0.13)	0.96 (0.10)
DDNM+ ($\sigma_y = .18$)	26.93 (2.25)	22.29 (3.00)	0.78 (0.07)	0.62 (0.12)
DDNM+ ($\sigma_y = .00$)	15.66 (0.39)	15.52 (0.43)	0.25 (0.10)	0.30 (0.10)

A.7.6 ES-WMV as a helper

Performance of ES-WMV on DD, GP-DIP, DIP-TV, and SIREN for Gaussian denoising in terms of SSIM gaps (see Fig. 24).

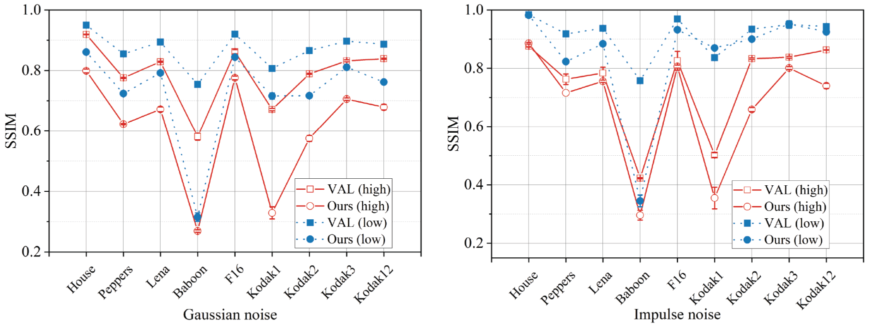
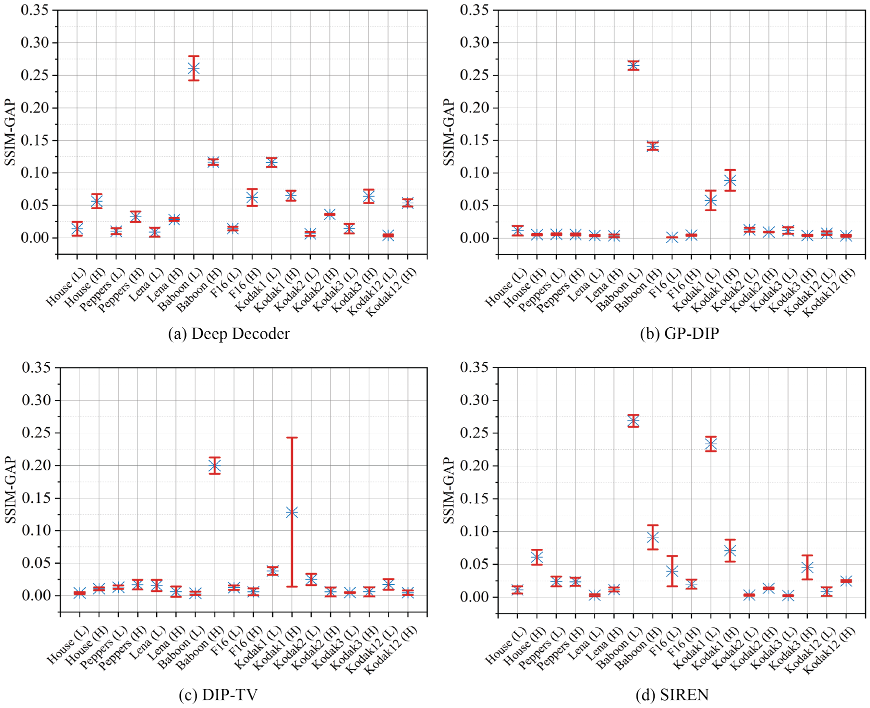


Figure 23: Comparison of VAL and ES-WMV for Gaussian and impulse noise in terms of SSIM.



(a) Deep Decoder

(b) GP-DIP

(c) DIP-TV

(d) SIREN

Figure 24: Performance of ES-WMV on DD, GP-DIP, DIP-TV, and SIREN for Gaussian denoising in terms of SSIM gaps. L: low noise level; H: high noise level

A.7.7 Performance on real-world denoising

We randomly sample 1024 images from the RGB track of the NTIRE 2020 Real Image Denoising Challenge (Abdelhamed et al., 2020), and perform DIP-based image denoising. Histograms of PSNR and SSIM gaps are shown in Fig. 25. For DIP with the three different losses, there are only 4.79%, 4.69% and 4.40% images, respectively, whose PSNR gaps are larger than $2dB$. As stated from the beginning, ES-WMV is designed with real-world IPs, targeting unknown noise types and levels. Given the encouraging performance above, we test it on a common real-world denoising dataset—PolyU Dataset Xu et al. (2018), which contains 100 cropped regions of 512×512 from 40 scenes. The results are reported in Tab. 10. We do not repeat

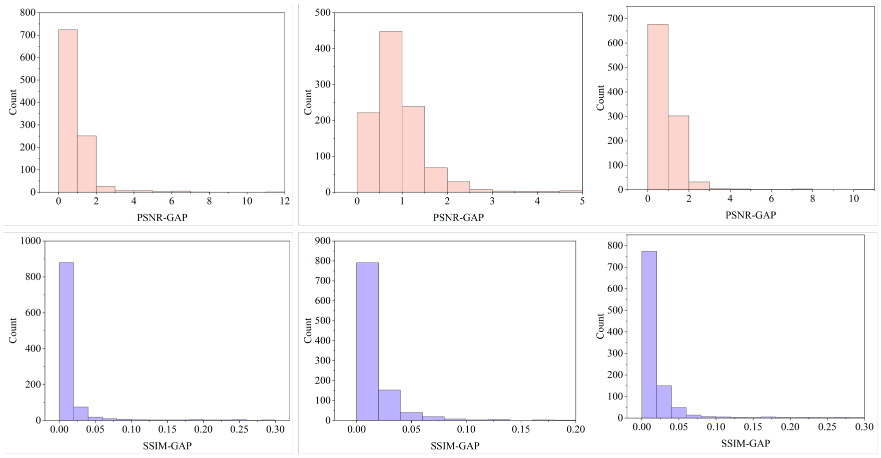


Figure 25: Image denoising of DIP + ES-WMV on the RGB track of the NTIRE 2020 Real Image Denoising Challenge. Top row: histograms of PSNR gaps for DIP (MSE), DIP (ℓ_1) and DIP (Huber), respectively; bottom row: histograms of SSIM gaps for DIP (MSE), DIP (ℓ_1) and DIP (Huber), respectively.

Table 10: DIP with ES-WMV on real image denoising on the PolyU Dataset: mean and (std). (**D**: Detected)

	PSNR(**D**)	PSNR Gap	SSIM(**D**)	SSIM Gap
DIP (MSE)	36.83 (3.07)	1.26 (1.22)	0.98 (0.02)	0.01 (0.01)
DIP (ℓ_1)	36.20 (2.81)	1.64 (1.58)	0.97 (0.02)	0.01 (0.01)
DIP (Huber)	36.76 (2.96)	1.28 (1.09)	0.98 (0.02)	0.01 (0.01)

the experiments here; the means and standard deviations are obtained over the 100 images of the PolyU dataset. On average, our detection gaps are ≤ 1.64 in PSNR and ≤ 0.01 in SSIM for this dataset across various losses. The absolute PNSR and SSIM detected are surprisingly high.

A.7.8 Image Inpainting

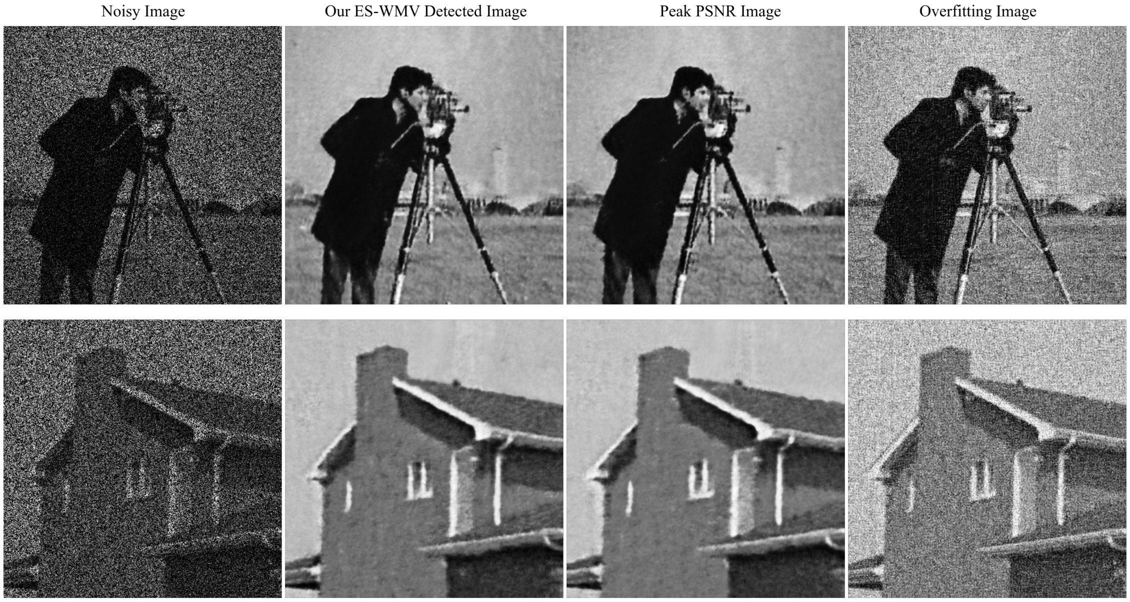


Figure 26: Visual detection performance of ES-WMV on image inpainting.

In this task, a clean image $\boldsymbol{x}_0 \in [0,1]^{H\times W}$ is contaminated by additive Gaussian noise ε, and then only partially observed to yield the observation $\boldsymbol{y} = (\boldsymbol{x}_0 + \boldsymbol{\varepsilon}) \odot \boldsymbol{m}$, where $\boldsymbol{m} \in \{0,1\}^{H\times W}$ is a binary mask and $\odot$ denotes the Hadamard product. Given $\boldsymbol{y}$ and $\boldsymbol{m}$, the goal is to reconstruct $\boldsymbol{x}_0$. We consider the formulation reparametrized by DIP, where $G_{\boldsymbol{\theta}}$ is a trainable DNN parametrized by $\boldsymbol{\theta}$ and $\boldsymbol{z}$ is a frozen random seed:

$$\ell(\boldsymbol{\theta}) = \|(G_{\boldsymbol{\theta}}(\boldsymbol{z}) - \boldsymbol{y}) \odot \boldsymbol{m}\|_F^2. \tag{43}$$

Mask $\boldsymbol{m}$ is generated according to an i.i.d. Bernoulli model with a rate of 50%, i.e., half of pixels not observed in expectation. The **noise ε is set to the medium level**, i.e., additive Gaussian with 0 mean and 0.18 variance. We test our ES-WMV for DIP on the inpainting dataset used in the original DIP paper Ulyanov et al. (2018). PSNR gaps are $\leq$ 1.00 and SSIM gaps are $\leq$ 0.05 for most cases (see Tab. 11). We also visualize two examples in Fig. 26.

Table 11: Detection performance of DIP with ES-WMV for image inpainting: mean and (std). PSNR gaps below 1.00 are colored as red; SSIM gaps below 0.05 are colored as blue. (**D**: Detected)

	PSNR(**D**)	PSNR Gap	SSIM(**D**)	SSIM Gap
Barbara	21.59 (0.03)	0.20 (0.03)	0.67 (0.00)	0.00 (0.00)
Boat	21.91 (0.10)	1.16 (0.18)	0.68 (0.00)	0.03 (0.01)
House	27.95 (0.33)	0.48 (0.10)	0.89 (0.01)	0.01 (0.00)
Lena	24.71 (0.30)	0.37 (0.18)	0.80 (0.00)	0.01 (0.00)
Peppers	25.86 (0.22)	0.23 (0.05)	0.84 (0.01)	0.02 (0.00)
C.man	25.26 (0.09)	0.23 (0.14)	0.82 (0.00)	0.01 (0.00)
Couple	21.40 (0.44)	1.21 (0.53)	0.63 (0.01)	0.04 (0.02)
Finger	20.87 (0.04)	0.24 (0.17)	0.77 (0.00)	0.01 (0.01)
Hill	23.54 (0.08)	0.25 (0.11)	0.70 (0.00)	0.00 (0.00)
Man	22.92 (0.25)	0.46 (0.11)	0.70 (0.01)	0.01 (0.00)
Montage	26.16 (0.33)	0.38 (0.26)	0.86 (0.01)	0.03 (0.01)

A.7.9 Image Super-resolution

Visual comparisons for 2× image super-resolution task with additional low-level Gaussian noise and impulse noise are shown in Figs. 27 and 28, respectively.

A.7.10 RAW images demosaicing and denoising

RAW images demosaicing and denoising are two essential procedures for modern digital cameras to produce high-quality full-color images (Li et al. (2023a)). Given a noisy RAW image $\widehat{\boldsymbol{x}}^{1ch} = \boldsymbol{x}^{1ch} + \boldsymbol{n}$, where H and W are the height and width of the image and $\boldsymbol{n}$ is the noise, the goal is to obtain a high quality full-color image $\boldsymbol{x}^{3ch} \in \mathbb{R}^{H\times W\times 3}$ from it. To achieve this goal, it is obvious that we need to fill in the missing pixels (*demosaicing*) and remove the noisy components $\boldsymbol{n}$ (*denoising*). In this section, we formulate this problem as an image-inpainting problem as Li et al. (2023a) and adopt DIP to reconstruct the desired full-color image. In addition, we plug our early stopping method into DIP and explore the effectiveness of our method on this low-level vision task. We conduct experiments on the Kodak dataset[3] and prepare it following the pipeline in Li et al. (2023a). We experiment with Poisson noise ($\lambda = 25$; a detailed description of the noise intensity could be found in Li et al. (2023a)), which is a very common noise under low light conditions. We report the experimental results in Tab. 12. It is evident that our method could effectively detect the near-peak points and produce reliable early stopping signals for DIP.

A.7.11 MRI reconstruction

[3] `https://r0k.us/graphics/kodak/`

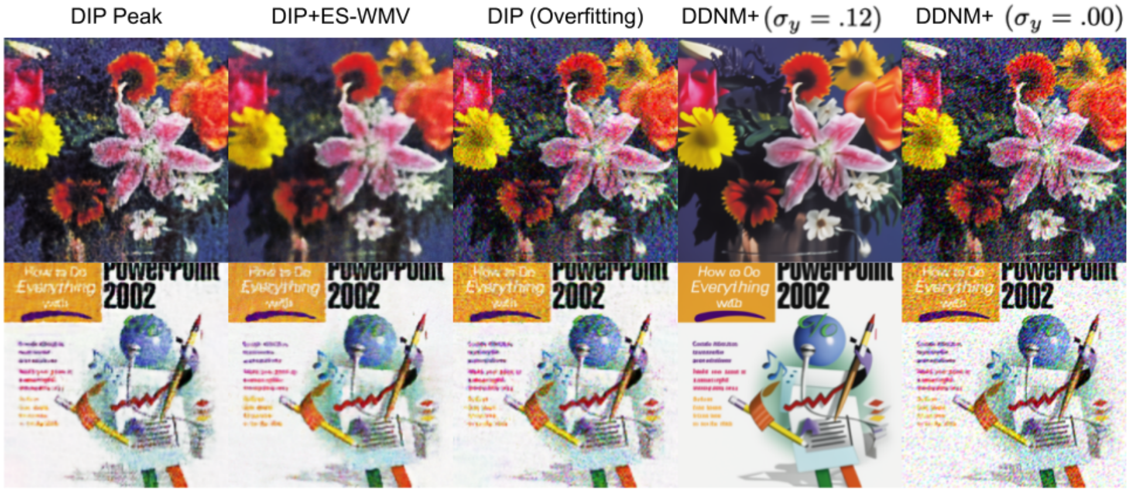


Figure 27: Visual comparisons for 2× image super-resolution task with additional low-level Gaussian noise.

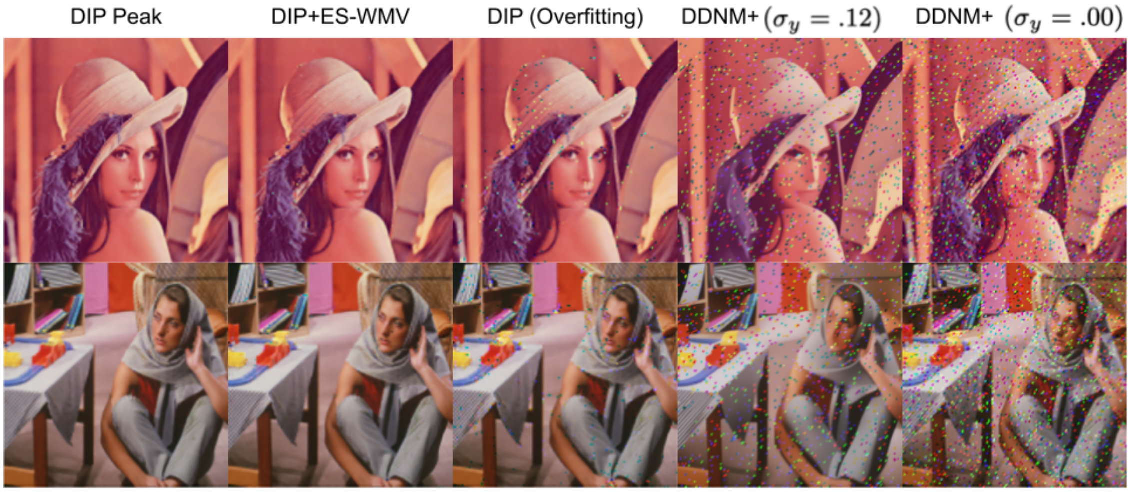


Figure 28: Visual comparisons for 2× image super-resolution task with additional low-level Impulse noise.

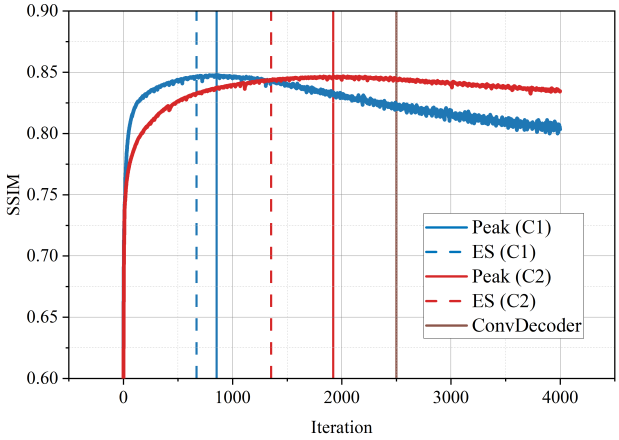


Figure 29: Detection on MRI reconstruction

We visualize the performance on two random cases (C1: 1001339 and C2: 1000190 sampled from Darestani & Heckel (2021), part of the fastMRI datatset (Zbontar et al., 2018)) in Fig. 29 (quality measured in SSIM, consistent with Darestani & Heckel (2021)).

Table 12: RAW images demosaicing and denoising on the Kodak dataset for **25 cases**: mean and (std). (**D**: Detected)

PSNR(**D**)	PSNR Gap	SSIM(**D**)	SSIM Gap
24.22 (2.49)	0.92 (0.87)	0.58 (0.14)	0.06 (0.08)

A.7.12 Blind image deblurring (BID)

In this section, we systematically test our ES-WMV and VAL on the entire standard Levin dataset for both low-level and high-level cases. We set the maximum number of iterations to 10,000 to ensure sufficient optimization. The detected images of our ES-WMV are substantially better than those of VAL, as shown in Tab. 13.

Table 13: BID detection comparison between ES-WMV and VAL on the Levin dataset for both low-level and high-level noise: mean and (std).Higher PSNR is in red and higher SSIM is in blue. (**D**: Detected)

	Low Level		High Level	
	PSNR(**D**)	SSIM(**D**)	PSNR(**D**)	SSIM(**D**)
WMV	28.54(0.61)	0.83(0.04)	26.41(0.67)	0.76(0.04)
VAL	18.87(1.44)	0.50(0.09)	16.69(1.39)	0.44(0.10)

A.7.13 ES-WMV vs. ES-EMV

We now consider our memory-efficient version (ES-EMV) as described in Algorithm 2, and compare it with ES-WMV, as shown in Fig. 30. Besides the memory benefit, ES-EMV runs around 100 times faster than ES-WMV, as reported in Tab. 3 and does seem to provide a consistent improvement on the detected PSNRs for image denoising tasks on NTIRE 2020 Real Image Denoising Challenge (Abdelhamed et al., 2020), PolyU dataset Xu et al. (2018) and the classic 9-image dataset (Dabov et al., 2008) (see Tabs. 14 and 15 and Fig. 30), due to the strong smoothing effect (we set $\alpha = 0.1$). In this paper, we prefer to keep it simple and leave systematic evaluations of these variants for future work.

A.7.14 Ablation study

We also notice that smaller learning rates can smooth out the VAR curves and mitigate the multi-valley phenomenon in Fig. 33. Therefore, we apply our ES-WMV to deep decoder and GP-DIP with smaller learning rates (both 0.001), as shown in Fig. 31. Compared to the results of deep decoder and GP-DIP with the default learning rates in Fig. 10, most of the PSNR gaps decrease.

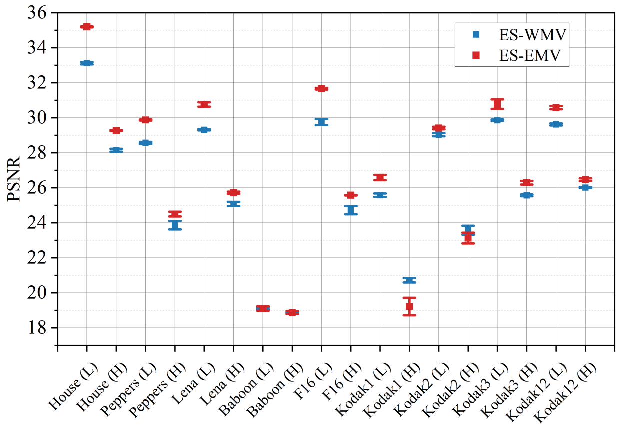


Figure 30: Detected PSNR comparison between DIP with ES-WMV and DIP with ES-EMV on the classic 9-image dataset (Dabov et al., 2008).

Table 14: Detection performance comparison between DIP with ES-WMV and DIP with ES-EMV for real image denoising on 1024 images from the RGB track of NTIRE 2020 Real Image Denoising Challenge (Abdelhamed et al., 2020): mean and (std). Higher PSNR and SSIM are in red. (**D**: Detected)

	PSNR(**D**)-WMV	PSNR(**D**)-EMV	SSIM(**D**)-WMV	SSIM(**D**)-EMV
DIP (MSE)	34.04 (3.68)	34.96 (3.80)	0.92 (0.07)	0.93 (0.07)
DIP (ℓ_1)	33.92 (4.34)	34.83 (4.35)	0.93 (0.05)	0.94 (0.05)
DIP (Huber)	33.72 (3.86)	34.72 (4.04)	0.92 (0.06)	0.93 (0.06)

Table 15: Detection performance comparison between DIP with ES-WMV and DIP with ES-EMV for real image denoising on the PolyU dataset Xu et al. (2018): mean and (std). Higher PSNR and SSIM are in red. (**D**: Detected)

	PSNR(**D**)-WMV	PSNR(**D**)-EMV	SSIM(**D**)-WMV	SSIM(**D**)-EMV
DIP (MSE)	36.83 (3.07)	37.32 (3.82)	0.98 (0.02)	0.98 (0.03)
DIP (ℓ_1)	36.20 (2.81)	36.43 (3.22)	0.97 (0.02)	0.97 (0.02)
DIP (Huber)	36.76 (2.96)	37.21 (3.19)	0.98 (0.02)	0.98 (0.02)

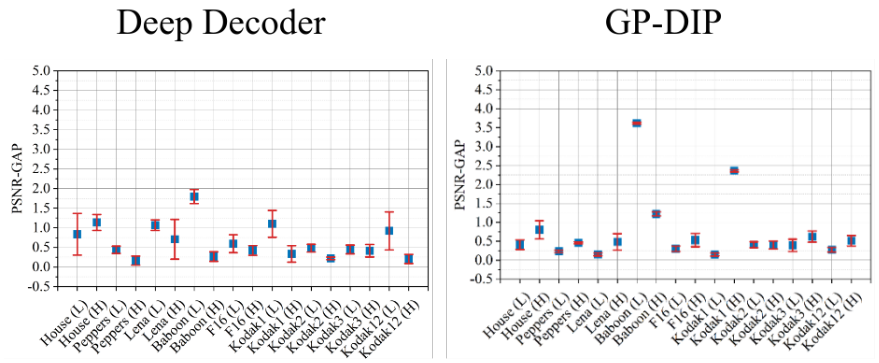


Figure 31: Performance of ES-WMV on deep decoder and GP-DIP with smaller learning rates for Gaussian denoising in terms of PSNR gaps. L: low noise level; H: high noise level.

A.8 Potential of ES-WMV for effective ES in zero-shot super-resolution with diffusion models

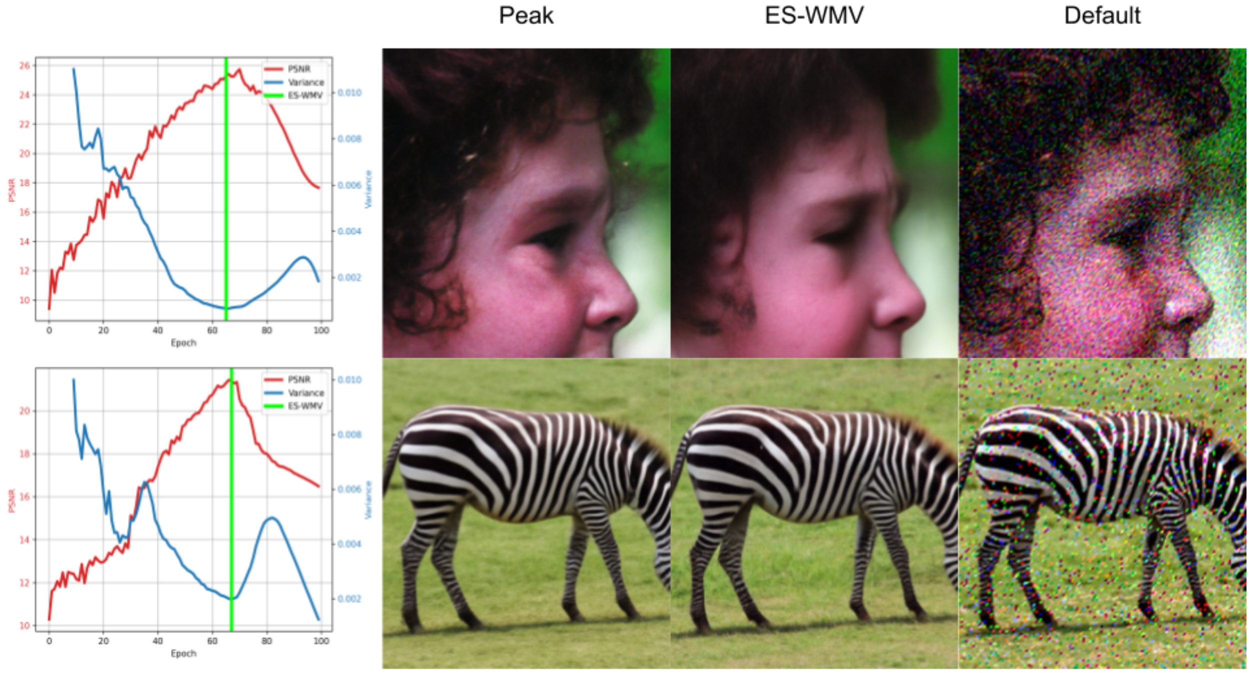


Figure 32: **Left**: PSNR/VAR curves of DDNM+ for 2× noisy image super-resolution (top: with Gaussian noise at $\sigma_y = 0.12$; bottom: with impulse noise). **Right**: visualization of the super-resolved results by DIP (PSNR peak), DIP with ES-WMV, and the default DDNM+. The DDNM+ paper Wang et al. (2022) fixes the iteration number as 1000 and stops the reverse diffusion process at the very last iteration by default.

Recently, zero-shot methods based on diffusion models have been proposed to solve linear image restoration tasks, e.g., DDNM+ Wang et al. (2022)[4]. However, these methods usually rely on pre-trained models from large external training datasets, while DIP does not need any training data or pre-trained models. In Fig. 32, we show that DDNM+ can also have overfitting issues similar to those in DIP methods, especially when the observation $\boldsymbol{y}$ is noisy but the noise type and/or level are not correctly specified to the diffusion models—very likely to happen in practice, as knowing the exact measurement noise type/level is often unrealistic. When DDNM+ is trained assuming no noise, i.e., $\sigma_y = 0.00$, but the downsampled image is contaminated by Gaussian noise at a level $\sigma_y = 0.12$, or by impluse noise, there is substantial overfitting to the noise, as is evident from both the PSNR plots (left of Fig. 32) and direct visualization of the super-resolved images (right of Fig. 32). Moreover, we observe that our ES-MWV method can also help DDNM+ detect near-peak performance! We stress that the experiment here is exploratory and preliminary and that tackling the overfitting issue in DDNM+ style methods for solving inverse problems is out of the scope of this paper. We leave a complete study for future work.

A.9 Limitations and analysis of failure cases

For limitations, our theoretical justification is only partial, sharing the same difficulty of analyzing DNNs in general; our ES method struggles with images with substantial high-frequency components; our detection is sometimes off the peak in terms of iteration numbers when helping certain DIP variants, e.g. DIP-TV with low-level Gaussian noise (Fig. 4), but the detected PSNR gap is still small.

Our ES method needs three things to succeed: (1) the U-shape of the VAR curve, (2) the VAR valley aligning with the PSNR peak, and (3) the successful numerical detection of the VAR valley. In this section, we discuss

[4] `https://github.com/wyhuai/DDNM/tree/main/hq_demo`

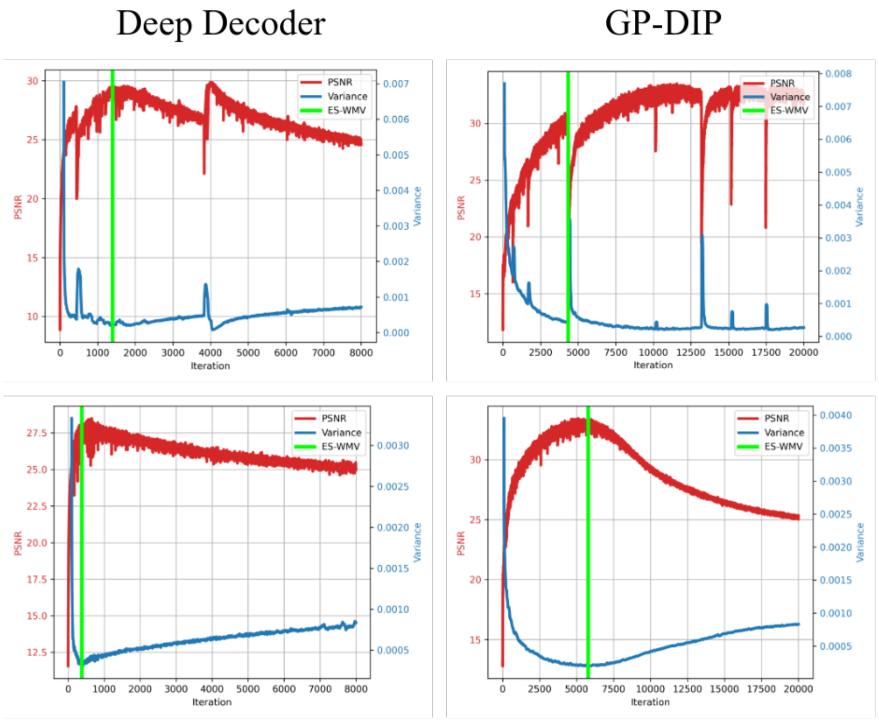


Figure 33: Top left: DD with the default learning rate for “Lena(L)”; top right: GP-DIP with the default learning rate for “House(L)”; bottom left: DD with learning rate = 0.001 for “Lena(L)”; bottom right: GP-DIP with learning rate = 0.001 for “House(L)”.

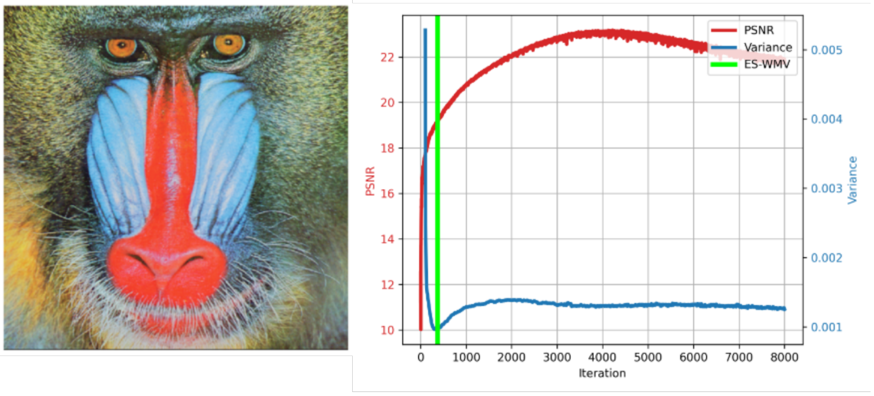


Figure 34: Our ES-WMV method on DIP for denoising “Baboon" with low-level Gaussian noise. Left: clean “Baboon"; right: the denoising process.

two major failure modes of our ES method: (I) the VAR valley aligns well with the PSNR peak, but the U-shape assumption is violated. A dominant pattern is the presence of multiple valleys, see, e.g., the top row of Fig. 33 that shows such examples with DIP variants, DD and GP-DIP (we do not observe the multi-valley phenomenon on DIP itself in Fig. 3). Since our numerical valley detection method aims to locate the first major valley, it may not locate the deepest valley among the multiple valleys. Fortunately, for these cases, we observe that using smaller learning rates can help to smooth out their curves and mitigate the multi-valley phenomenon, leading to much smaller detection gaps (see the bottom row of Fig. 33); (II) the VAR valley does not align well with the PSNR peak, which often happens on images with significant high-frequency components, e.g., Fig. 34. We suspect that this is because the initial VAR decrease tends to correlate with the early learning of low-frequency components in DIP. When there are substantial high-frequency components in an image, the PSNR curve takes more time to pick up the high-frequency components, after the VAR curve already reaches the first major valley; hence, the misalignment between the VAR valley and the PSNR peak occurs. In addition, our ES-WMV can fail for images with substantial high-frequency components, e.g. Fig. 34.