Bayesian Self-Attention with Intra-Token Modeling for Lightweight Denoising Transformers

Runyang He, Zuowei Shen, and Hui Ji

Department of Mathematics, National University of Singapore, 119076, Singapore
runyang_he@u.nus.edu {matzuows,matjh}@nus.edu.sg

Abstract. Transformer architectures have achieved strong performance in image denoising, but their computational and memory costs remain high. We revisit self-attention (SA) from a Bayesian perspective and show that standard SA mainly exploits first-order inter-patch statistics, resembling a learnable non-local averaging scheme. However, this formulation neglects second-order intra-patch statistics, which are important for capturing local pixel dependencies. To address this, we introduce a Bayesian SA formulation that jointly models first-order inter-patch and second-order intra-patch statistics. This leads to a lightweight denoising Transformer, termed NLformer, featuring a dual-branch attention design and an efficient feed-forward module. Experiments on several benchmarks show that NLformer outperforms existing lightweight denoising networks and substantially narrows the gap to full-size Transformer-based denoisers while maintaining low model complexity.

Keywords: Image denoising · Transformer · Bayesian inference

1 Introduction

As a basic task in image processing, image denoising aims at recovering an clean image $\mathbf{x}$ from its noisy measurement $\mathbf{y}$, typically modeled as $\mathbf{y} = \mathbf{x} + \mathbf{z}$, where $\mathbf{z}$ denotes random noise. Traditional methods rely on the priors of both the image and the noise to separate them. Typical image priors include sparsity prior of image gradients (*e.g.*, total variation [42], data-driven tight frame [9]) and self-similarity of image patches (*e.g.*, non-local means [6,8] and BM3D [17]). Among these approaches, the non-local approach shows strong performance. The key idea is to treat the image $\mathbf{x}$ as a set of image patches $\{p_k\}_k$, group similar patches across the image, and collaboratively estimate the clean patches p_ks from its associated patch group. This approach estimates each patch from multiple related noisy measurements, thereby improving performance.

Deep learning has been the prominent paradigm for image denoising [40, 53, 56, 58]. The neural networks (NNs) are trained on noisy-clean image pairs to learn a mapping from noisy inputs to clean ones. The main focus is on the design of NN architectures. One particular influential one is convolutional NNs(CNNs), including U-Net [41], DnCNN [56], FFDNet [58], complex-valued CNN [37], and

many others. These architectures combine convolutional operations, non-linear activations, and hierarchical up-down sampling.

Transformer [45] also shows strong performance on image denoising. The transformer-based models are commonly built on the Vision Transformer (ViT) [19], which treats an image as a set of image patches embedded into tokens. A series of SA layers is then applied to capture long-range patch dependencies, enhancing noise-resistant recovery by drawing from more information sources. By modeling non-local correlations of tokens, Transformers effectively exploit the self-similarity prior of image patches, offering advantages over CNNs, like traditional non-local methods outperformed local ones.

Challenges in denoising Transformer: Denoising Transformer needs to compute pairwise affinities with multi-head SA, which is computationally costly as SA incurs quadratic complexity in the number of patches. Consequently, efforts to adapt ViT architectures for image denoising have focused on reducing model complexity and computational cost. Notably, SwinIR [28] confines attention to local spatial windows through Window SA while Restormer [50] reorders input dimensions to apply attention along the channel axis, thereby reducing computational cost. Despite these efforts, Transformer-based denoisers still exhibit significantly large model sizes and high computational costs.

As shown in [15], standard strategies for reducing model size, such as decreasing the number of blocks or channels, lead to substantial performance degradation. Recently, there have been a few attempts to design lightweight Transformer architectures for image denoising [16, 62]. However, the performance of these models is far behind their full-sized counterparts. There is a practical need for developing lightweight denoising Transformers that can achieve competitive performance while maintaining computational efficiency.

1.1 Main idea

In this paper, we develop a lightweight denoising Transformer by revisiting SA through the lens of non-local Bayesian inference (NLB) [27]. Existing denoising Transformers mainly exploit first-order inter-patch statistics to capture non-local similarity of patches, while overlooking second-order intra-patch statistics that describe local pixel dependencies within each patch. To address this, we propose a Bayesian formulation that unifies non-local inter-patch aggregation with intra-patch second-order information, leading to a more effective SA mechanism and a lightweight architecture with strong denoising performance.

Revisiting SA from the lens of inter-patch statistics: Existing denoising Transformers typically follow the ViT design, i.e., an image $\mathbf{y}$ is split into N patches and embedded as tokens $Y = [Y_1, \ldots, Y_N]$. A single-head SA layer produces the value $\Phi(Y_i)$ for each patch Y_i by

$$\Phi(Y_i) = \sum_{j=1}^{N} w_{i,j}\phi^V(Y_j); \quad w_{i,j} \propto \exp(\langle \phi^Q(Y_i), \phi^K(Y_j)\rangle \cdot d^{-\frac{1}{2}}), \tag{1}$$

where $\phi^Q(\cdot)$, $\phi^K(\cdot)$, $\phi^V(\cdot)$ are learnable linear projections that map each token to a query, key, and value vector, respectively; $\langle\cdot,\cdot\rangle$ denotes the inner product, and $w_{i,j}$ is the normalized attention weight.

To understand SA from Bayesian inference, consider a simplified setting where the projection matrices Q, K, V are all identity. In this case, SA reduces to a weighted averaging of patches across the image, where the weights depend on the similarity between each pair of patches. Introducing learnable projections Q, K, V can then be interpreted as performing such averaging in a projected feature space that is more robust to noise and better captures truth similarity with noisy patches. In short, SA functions as a learnable extension of non-local means [8], leveraging inter-patch correlations through first-order statistics (mean), but differing in that both the similarity measure (query-key space) and the aggregation space (value space) are learned over the training data.

Bayesian estimator with intra-patch correlation: The weighted average over inter-patches ignores intra-patch correlations, which are strong in structured regions such as edges and textures. Neglecting such intra-patch correlations can lead to suboptimal estimation. NLB [27] addresses this limitation by approximating the linear minimum mean squared error (MMSE) estimator of the patch through exploiting local pixel correlations within each patch. Specifically, it models intra-patch dependencies using second-order statistics among patch entries, represented by the covariance terms $\{Y_j Y_j^\top\}_{j\in\Omega_i}$. More specifically, NLB estimates X_i by approximating Wiener estimator that minimizes the Bayesian risk:

$$\min_W \mathbb{E}_{p(X),p(Z)}[\|X_i - \widehat{X}_i\|^2] := \mathbb{E}_{p(X),p(Z)}[\|X_i - W(Y_i)\|^2]. \tag{2}$$

Let μ_i, Σ_{X_i} denote the mean and covariance matrix of X_i. Then, the Wiener estimation (optimal linear estimation) is

$$X_i^* = W^*(Y_i) = \mu_i + D^*(Y_i - \mu_i), \quad D^* = \Sigma_{X_i}\big(\Sigma_{X_i} + \sigma^2 I\big)^{-1}, \tag{3}$$

where σ^2 is the noise variance.

A new Bayesian estimator with inter- and intra-patch correlation: standard SA and NLB are two different approaches to exploiting the set of image patches. SA leverages inter-patch correlations through the statistics on $\{Y_j\}_{j\in\Omega_i}$, while NLB utilizes intra-patch correlations, i.e., dependencies among pixel values within each patch, through the statistics on $\{Y_j Y_j^\top\}_{j\in\Omega_i}$.

In this paper, we propose a new Bayesian estimator that exploits both inter- and intra-patch correlations. The proposed estimator can be expressed as

$$\widehat{X}_i = \sum_{j\in\Omega_i} w_{i,j} Y_j + H_i Y_i. \tag{4}$$

The optimal linear estimator H^* that minimize the risk is

$$H_i^* = \big(\mathbb{E}[X_i X_i^\top] - \sum_{j\in\Omega_i} w_{i,j}\mathbb{E}[Y_j Y_i^\top]\big)\big(\mathbb{E}[X_i X_i^\top] + \sigma^2 I\big)^{-1}. \tag{5}$$

NLformer: SA with joint inter- and intra-patch modeling: We propose a lightweight Transformer, named as *NLformer*, that not only captures inter-patch correlations, as in existing SAs, but also captures intra-patch correlations. Built on the Bayesian estimator $\widehat{X}_i$ defined by (4), NLformer leverages both linear and quadratic statistics in the value projection $\phi^V(\cdot)$. Specifically, it utilizes $\phi_1^V(Y_j)$ and $\phi_2^V(Y_j)\phi_2^V(Y_j)^\top$ to encode inter- and intra-patch correlations, which can be formulated as:

$$\Phi^V(Y_i) = [\Phi_1^V(Y_i), \Phi_2^V(Y_i)] = \sum_{j\in\Omega_i} w_{i,j}[\phi_1^V(Y_j), \phi_2^V(Y_j)\phi_2^V(Y_j)^\top]. \tag{6}$$

Directly incorporating full second-order statistics of $\Phi_2^V(Y_i)$ into SA would substantially increase the model size and computational cost. To maintain efficiency, $\Phi_2^V(Y_i)$ is therefore approximated by a low-rank representation that preserves the essential statistical information.

Inspired by the optimal Bayesian estimator (5), the FFN module computes the matrix inverse $H_i = \Phi_2^V(Y_i)^{-1}$. Thanks to the low-rank structure of $\Phi_2^V(Y_i)$, this inverse can be efficiently obtained via the Woodbury matrix identity [47]:

$$(H + UU^\top)^{-1} = H^{-1} - H^{-1}U(I + U^\top H^{-1}U)^{-1}U^\top H^{-1}, \tag{7}$$

where U is a low-rank matrix. FFN part then estimates the clean patches through $\Phi_1^V(Y_i)$, $\Phi_2^V(Y_i)$ and $\Phi_2^V(Y_i)^{-1}$. One head of NLformer can be expressed as:

$$\begin{cases} \text{SA:} \quad \Phi_1^V(Y_i) = \displaystyle\sum_{j\in\Omega_i} w_{i,j}\,\phi_1^V(Y_j), \qquad H = \frac{1}{N}\sum_{j=1}^{N} \phi_2^V(Y_j)\phi_2^V(Y_j)^\top; \\ \qquad\quad \Phi_2^V(Y_i) = H + \displaystyle\sum_{k=1}^{2} \phi_k(Y_i)\phi_k(Y_i)^\top; \\ \text{FFN:} \quad \widehat{X}_i = \Psi\big(Y_i; \Phi_1^V(Y_i), \Phi_2^V(Y_i), \Phi_2^V(Y_i)^{-1}\big). \end{cases} \tag{8}$$

The proposed NLformer architecture is then formed by using multi-head SA and FFN defined above, as illustrated in Figure 1.

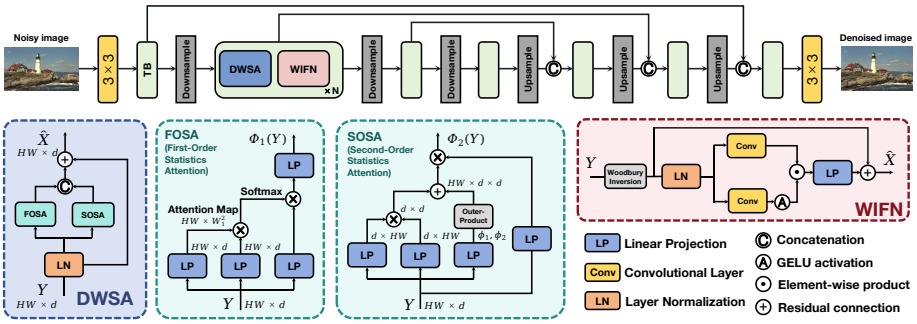

Fig. 1: Overall structure of NLformer.

1.2 Main contributions

The main contributions are summarized as follows:

- We reinterpret SA for image denoising from a Bayesian perspective, showing that standard self-attention mainly captures inter-patch correlations, while effective denoising also requires capturing intra-patch correlations.
- We propose a unified Bayesian estimator that combines non-local inter-patch aggregation with local intra-patch correlation modeling, thereby bridging first- and second-order patch statistics in a single formulation.
- Based on this formulation, we develop NLformer, a lightweight and interpretable denoising Transformer with dual-branch SA and an efficient Woodbury-based feed-forward design for modeling second-order information.

Extensive experiments on synthetic and real-world benchmarks show that NLformer outperforms existing lightweight denoising models and substantially narrows the gap to full-size Transformer-based denoisers.

2 Related Work

Non-local methods: Non-local methods exploit the recurrence of similar patches over an image [7]. Most works [8, 17, 21, 26, 27] adopt a grouping-and-denoising scheme, where patches are grouped by similarity or spatial proximity. The idea is to utilize multiple measurements of a target patch, which exhibit strong intensity correlation but uncorrelated noise. Non-local means [8] directly applies a weighted average of similar patches based on their disparity to the target patch. NL-Bayes [27] refines it by using an approximate Wiener estimator that includes pixel-wise correlations via Bayesian estimation. BM3D [17] performs a 3D Haar wavelet transform on patch groups to exploit both non-local similarity and gradient sparsity. Low-rank prior of patch groups are utilized in [21, 26] for image/video denoising.

Denoising Transformers: Originally developed for sequence modeling in natural language processing, Transformers [45] were introduced to computer vision in ViT [19], treating an image as a sequence of patches. They have been applied to many image restoration tasks, including image denoising [10, 18, 28, 38, 46, 50], leveraging the SA to capture non-local correlations among patches. However, two challenges remain in adapting plain Transformer for image restoration.

The first concerns token definition. Starting with IPT [10], many works adopt convolutional layers to extract linear patch features from the noisy image, treating each patch as a token [28,46]. Restormer [50] instead applies learned filters to extract multiple features, treating each as a token. Recent hybrid models combine both strategies [11, 12]. The second challenge is computational complexity, which is quadratic in the number of tokens for plain Transformer [45]. SwinIR [28] and Uformer [46] mitigate this by restricting attention within fixed-size non-overlapping windows. Restormer [50] uses transposed attention to keep attention on channel dimension, not spatial dimension. KGT [39] computes sparse attention over the top-k similar tokens.

Lightweight denoising Transformers: Despite the efforts to reduce computational complexity, full-size Transformer-based denoisers [28, 46, 50, 54] still incur higher model size and computational cost than CNN-based denoisers [43, 56, 58]. Choi *et al.* [15] shows that simply reducing blocks or channels leads to a notable performance drop. To address this, recent studies focus on optimizing Transformer architectures rather than simply reducing model size. LIDFormer [62] integrates discrete wavelet transforms with efficient multi-branch attention. RAMiT [16] introduces bi-dimensional self-attention within a hierarchical reciprocal structure. However, these models still exhibit notable performance gaps compared to their full-sized counterparts.

3 Method

In this section, we present the Bayesian estimator leveraging inter- and intra-patch statistics and connect it to SA. We then introduce the proposed NLformer.

3.1 Bayesian Estimation with Inter- and Intra-Patch Correlations

Bayesian estimator with inter-patch correlation: Let $\mathbf{y}$ be a noisy image, divided into N overlapping patches $\{Y_i\}_{i=1}^N$, where each $Y_i \in \mathbb{R}^d$ is a noisy observation of the clean patch X_i, modeled as

$$Y_i = X_i + Z_i, \quad Z_i \sim \mathcal{N}(0, \sigma^2 I).$$

For each patch Y_i, a set of patches $\{Y_j\}_{j\in\Omega_i}$ is identified based on similarity or spatial proximity. The goal of denoising with inter-patch correlation is to estimate X_i using both Y_i and its associated group $\{Y_j\}_{j\in\Omega_i}$.

In the ideal case, all patches Y_j are the noisy measurement of the target patch X_i. Then, the average of the noisy patches $\frac{1}{|\Omega_i|}\sum_{j\in\Omega_i} Y_j$ is the MMSE estimation of X_i. However, in practice, the patches Y_js are noisy measurements related to the clean patch X_i, rather than exact observations of it. Then, the clean patch X_i is estimated via the weighted average of the noisy patches $\{Y_j\}_{j\in\Omega_i}$:

$$\widehat{\mu}_i = \frac{\sum_{j\in\Omega_i} w_{i,j} Y_j}{\sum_{j\in\Omega_i} w_{i,j}}; \quad w_{i,j} = K(Y_i, Y_j), \tag{9}$$

where K denotes a kernel function that measures the patch-wise similarity. The common choice for K is Gaussian kernel $\exp(-\|Y_i - Y_j\|_2^2/h^2)$.

Under additive white Gaussian noise (AWGN), this estimator can be viewed as a *Nadaraya–Watson estimator* approximating the conditional mean $\mathbb{E}[X_i \mid Y_i]$:

$$\mathbb{E}[X_i \mid Y_i] = \frac{\int X_i\, p(Y_i \mid X_i) p_X(X_i)\, dX_i}{\int p(Y_i \mid X_i) p_X(X_i)\, dX_i} \quad \text{(Bayes' rule)} \tag{10}$$

Assuming clean patches $\{X_j\}_{j\in\Omega_i}$ are locally stationary and drawn from a neighborhood around X_i. $p_X(X)$ can be approximated by kernel density estimation:

$$p_X(X) \approx \frac{1}{|\Omega_i|} \sum_{j \in \Omega_i} \exp\left(-\frac{\|X - X_j\|_2^2}{h^2}\right). \tag{11}$$

Using (11) and replacing X_j by their noisy measurements Y_j, we obtain

$$\widehat{X}_i = \frac{\sum_{j \in \Omega_i} \exp\left(-\|Y_i - Y_j\|_2^2/h^2\right) Y_j}{\sum_{j \in \Omega_i} \exp(-\|Y_i - Y_j\|_2^2/h^2)}. \tag{12}$$

For normalized Y_i and Y_j with unit norm, we have

$$\widehat{X}_i = \frac{\sum_{j \in \Omega_i} \exp(\langle Y_i, Y_j \rangle \cdot 2/h^2) Y_j}{\sum_{j \in \Omega_i} \exp(\langle Y_i, Y_j \rangle \cdot 2/h^2)}. \tag{13}$$

Clearly, the estimator (13) closely resembles the SA with Q, K, V being identity.

Note that the estimator by (9) is not exactly $\mathbb{E}[X_i|Y_i]$, but rather its approximation. The main issue is that it does not explicitly model correlations among pixels within a patch, assuming nearby pixels are independent. This simplification is inaccurate for natural images. As a result, it requires larger search windows and more sophisticated inter-patch modeling to compensate for the missing intra-patch correlation, leading to higher computation and model complexity.

Bayesian estimator with intra-patch correlation: Instead, NLB is an Bayesian estimator focusing explicitly on exploiting intra-patch statistics to facilitate estimation. It assumes that the clean patches X_j, corresponding to noisy observations Y_j, are independent samples drawn from a common distribution with mean μ_i and covariance Σ_{X_i}. If the estimator is restricted to be a linear function of the noisy patch Y_i, it is known that the optimal linear estimator, known as *Wiener estimator*, minimizes the Bayesian risk:

$$\mathbb{E}_{p(X),\, p(Z)} \left[\|X_i - W(Y_i)\|^2\right].$$

The optimal Wiener estimator W^* is given by

$$W^*(Y_i) = \mu_i + D^*(Y_i - \mu_i),\ D^* = \Sigma_{X_i} \left(\Sigma_{X_i} + \sigma^2 I\right)^{-1}. \tag{14}$$

See [33] for more details. As both expectation μ_i and the covariance matrix Σ_{X_i} are unknown, one can approximate it from the noisy patches $\{Y_j\}_{j \in \Omega_i}$ as follows:

$$\mu_i \approx \frac{1}{|\Omega_i|} \sum_{j \in \Omega_i} Y_j, \quad \Sigma_{X_i} \approx \frac{1}{|\Omega_i|} \sum_{j \in \Omega_i} (Y_j - \mu_i)(Y_j - \mu_i)^\top - \sigma^2 I. \tag{15}$$

Discussion on Inter- and Intra-Patch correlations: Natural images exhibit both inter- and intra-patch correlations. Inter-patch correlation often arises between distant patches sharing similar textures or repetitive patterns, whereas intra-patch correlation is typically pronounced in locally structured regions, where neighboring pixels exhibit strong dependence. For illustration, we select five image patches with distinct patterns and visualize their first-order inter-patch and second-order intra-patch statistics. As shown in Figure 2, the first

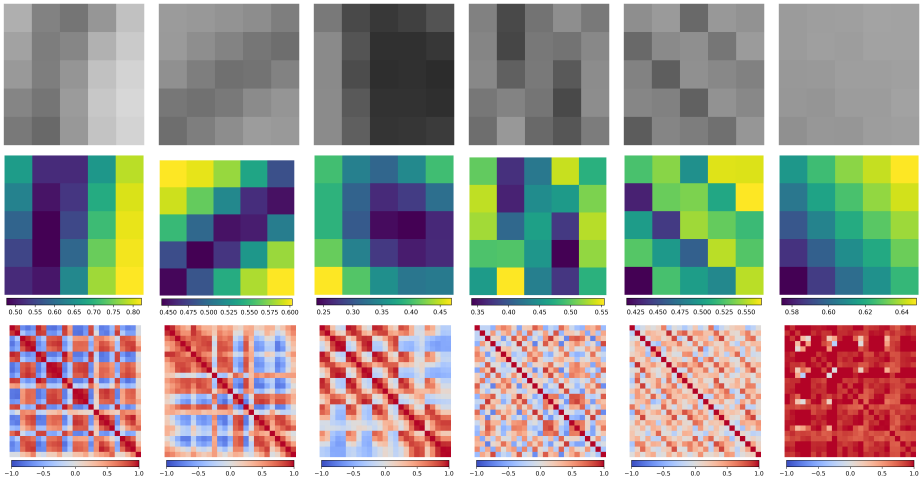

Fig. 2: Visualization of selected patches and the corresponding empirical statistics computed from patch groups. **First row**: original patches. **Second row**: first-order inter-patch statistics. **Third row**: second-order intra-patch statistics, where the diagonal entries are normalized to 1 for better visualization.

three patches exhibit distinct edges and thus high intra-patch correlations. In contrast, the last patch, sampled from a relatively flat region, shows strong local dependence, with nearly all pixels highly correlated with their neighbors.

New Bayesian estimation with both inter- and intra-patch correlation: For natural images, both inter- and intra-patch correlations inherently exist in them. Accordingly, the optimal Bayesian estimator should simultaneously leverage non-local inter-patch correlations and local intra-patch correlations to achieve more accurate estimation. Motivated by the complementary nature of inter- and intra-patch correlations, we propose a Bayesian estimator that exploits both inter- and intra-patch correlations. The idea is to correct the estimator (13) by a linear estimator that leverages the intra-patch correlation:

$$\widehat{X}_i = \underbrace{\sum_{j\in\Omega_i} w_{i,j} Y_j}_{\text{NLM prediction}} + \underbrace{H_i Y_i}_{\text{Linear correction}} \tag{16}$$

where H_i is a linear mapping. For a linear estimator, its Bayesian risk is

$$\min_{H_i} \mathbb{E}_{p(X),p(Z)}[\|X_i - \widehat{X}_i\|^2] := \mathbb{E}_{p(X),p(Z)}[\|X_i - \big(\sum_{j\in\Omega_i} w_{i,j} Y_j + H_i \cdot Y_i\big)\|^2]. \tag{17}$$

Proposition 1. *The optimizer to the problem* (17) *is*

$$H_i^* = \big(\mathbb{E}[X_i X_i^\top] - \sum_{j\in\Omega_i} w_{i,j}\mathbb{E}[Y_j Y_i^\top]\big)\big(\mathbb{E}[X_i X_i^\top] + \sigma^2 I\big)^{-1}, \tag{18}$$

where $\mathbb{E}[X_i X_i^\top]$ *denotes the second-order moment of* X_i.

Proof. See supplementary file for the proof.

3.2 Architecture of NLformer

Existing denoising Transformers employ SA based solely on inter-patch correlations through first-order statistics, ignoring intra-patch dependencies that are crucial in structured regions such as edges and textures. Consequently, they need many attention heads and large patch groups to gather sufficient information for good estimation. By including intra-patch correlations into the SA as shown in the proposed Bayesian estimator (18), we can use less attention heads and smaller windows to achieve comparable performance.

Building on the Bayesian estimator in (18), we incorporate both linear and quadratic statistics into the value projection $\phi^V(\cdot)$. One computational issue is that it is expensive to compute and average the outer product of the value vectors for each patch Y_j. To address this, we employ a low-rank approximation approach which is computationally efficient. In particular, we first average the outer product of all the value tokens to obtain a global estimation H:

$$H = \frac{1}{N}\sum_{j=1}^{N} \phi_2^V(Y_j)\phi_2^V(Y_j)^\top. \tag{19}$$

To have a more refined and patch-wise variant second-order statistics $\boldsymbol{\Phi}_2^V(Y_i)$ for each patch, we approximate the difference between $\boldsymbol{\Phi}_2^V(Y_i)$ and H by a low-rank matrix $\boldsymbol{\Delta} H_i$, which is expressed as the summation of r rank-1 matrices:

$$\boldsymbol{\Phi}_2^V(Y_i) \approx H + \boldsymbol{\Delta} H_i, \quad \boldsymbol{\Delta} H_i = \sum_{k=1}^{r} \phi_k(Y_i)\phi_k(Y_i)^\top, \tag{20}$$

where each $\phi_k(\cdot)$ is a learned linear projection. In our implementation, we choose $r = 2$ after evaluating its impact. Low-rank approximation enables SA to model intra-patch correlations with reduced computational cost while preserving key information. The dual-branch attention design then allows the model to capture both inter- and intra-patch correlations in a lightweight manner.

A lightweight FFN module first computes an explicit matrix inversion: $H_i = \boldsymbol{\Phi}_2^V(Y_i)^{-1}$ through the Woodbury matrix identity:

$$(H + U_i U_i^\top)^{-1} = H^{-1} - H^{-1}U_i(I + U_i^\top H^{-1} U_i)^{-1} U_i^\top H^{-1}, \tag{21}$$

where each $U_i = [\phi_1(Y_i), \phi_2(Y_i)] \in \mathbb{R}^{d\times 2}$ is a rank-2 matrix. FFN module then estimates the clean patches using $\boldsymbol{\Phi}_1^V(Y_i)$, $\boldsymbol{\Phi}_2^V(Y_i)$ and $\boldsymbol{\Phi}_2^V(Y_i)^{-1}$. A Transformer block with single attention head in NLformer is formulated as:

$$\begin{cases} \text{SA:} \quad \boldsymbol{\Phi}_1^V(Y_i) = \displaystyle\sum_{j\in\Omega_i} w_{i,j}\, \phi_1^V(Y_j), \qquad H = \frac{1}{N}\sum_{j=1}^{N} \phi_2^V(Y_j)\phi_2^V(Y_j)^\top; \\ \qquad\quad \boldsymbol{\Phi}_2^V(Y_i) = H + \displaystyle\sum_{k=1}^{2} \phi_k(Y_i)\phi_k(Y_i)^\top; \\ \text{FFN:} \quad \widehat{X}_i = \Psi\big(Y_i; \boldsymbol{\Phi}_1^V(Y_i), \boldsymbol{\Phi}_2^V(Y_i), \boldsymbol{\Phi}_2^V(Y_i)^{-1}\big)\,. \end{cases} \tag{22}$$

The proposed NLformer architecture is then formed by using multi-head SA and FFN defined above. The overall structure of NLformer is shown in Figure 1. Specifically, the proposed NLformer consists of three main components: (a) A 3×3 convolution is first applied to the input noisy image $\mathbf{y}$ to extract overlapping patches $\{Y_i\}_{i=1}^N$; (b) A U-shaped stack of m Transformer blocks processes these patches to estimate the clean counterparts $\{\hat{X}_i\}_{i=1}^N$; (c) A final 3×3 convolution aggregates the restored patches into the denoised image $\mathbf{x}$.

Each Transformer block contains a **D**ual-Branch **W**indow **S**elf-**A**ttention (DWSA) module followed by a **W**oodbury **I**nversion **F**eed-Forward **N**etwork (WIFN). DWSA first performs a patch-wise aggregation to estimate first-order statistics in its primal branch: $\sum_{j\in\Omega_i} w_{i,j}\cdot\phi_1^V(Y_{i,j})$, where $\phi_1^V(\cdot)$ is a linear projection. The attention weights $\{w_{i,j}\}$ are computed via softmax-normalized inner products between query and key matrices, as in standard Transformers. In the dual branch of DWSA, a global estimation of second-order statistics H is computed by averaging outer products of all the value tokens: $\frac{1}{N}\sum_{j=1}^N \phi_2^V(Y_j)\phi_2^V(Y_j)^\top$, where $\phi_2^V(\cdot)$ is a linear projection. Consequently, an incremental rank-2 approximation $\Delta H_i = \phi_1(Y_i)\phi_1(Y_i)^\top + \phi_2(Y_i)\phi_2(Y_i)^\top$ is computed separately for each patch Y_i. At last, DWSA merges $\Phi_V^1(Y_i)$ and $\Phi_V^2(Y_i)\cdot Y_i$ by a linear layer. The WIFN is implemented as a composition of the Woodbury matrix inversion 21 and a gated layer [50] with GELU activation [23].

Moreover, NLformer is implemented with a multi-scale structure [50] using pixel-unshuffle and pixel-shuffle operations for down- and up-sampling, respectively. This design helps preserve fine textures and structural details. Furthermore, we adopt an overlapped attention window [13], where each patch attends to a larger region with more samples. This improves the receptive field and mitigates redundancy caused by shifting non-overlapping windows [28, 46].

4 Experiments

4.1 Experimental Settings

Network settings: NLformer employs a 4-level encoder-decoder structure. For each level, the number of Transformer blocks is set as $[2, 2, 2, 2]$, the number of attention heads is set as $[1, 2, 4, 8]$, and the feature dimension is set as $[20, 40, 80, 160]$. For all DWSA layers, window size $W_1 = 8$. For all FFN layers, hidden dimension is set as $2.5\times$ input dimension.

Training settings: If not specified, our model is trained with L_1 loss on AdamW [31] optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$, weight decay $1e^{-2}$) for 500K iterations. Learning rate is initialized as $5e^{-4}$ and gradually reduced to $1e^{-6}$ with the cosine annealing [30]. We use mini-batches of size $[(64, 128^2)]$, and apply random flips and rotations for data augmentation. Training and evaluation are performed using PyTorch [35] on NVIDIA RTX 3090 GPUs. The code of the proposed method is available on Github[1].

Dataset for training and evaluation: The proposed method is evaluated on two types of noises: synthetic AWGN and real-world noise. For AWGN, following

[1] `https://github.com/Ryann2001/NLformer_Lightweight_Denoising.git`

Table 1: PSNR (dB) and SSIM results on real-world datasets. Models marked with * are trained with additional data. **Bold** and underline indicate the best and second-best performance among lightweight NNs, respectively.

	Category	Method	SIDD [1]	DND [36]	Params/M	FLOPs/G
	Non-learning	KSVD [3]	26.88/.842	36.49/.924	-	-
		BM3D [17]	25.65/.685	34.51/.851	-	-
		WNNM [21]	25.78/.809	34.67/.865	-	-
Large	CNN	MPRNet [53]	39.71/.958	39.82/.954	15.74	588
		MIRNet [52]	39.72/.959	39.88/.956	31.79	785
	Transformer	HWformer [43]	39.72/.958	-	40.06	170
		AKDT [5]	39.70/.961	39.90/.955	11.43	59.69
		Uformer32 [46]	39.77/.959	39.93/.955	20.63	43.86
		Restormer [50]	40.02/.960	40.03/.956	26.11	155
		ART [54]	39.96/.960	40.05/.956	25.7	782
Lightweight	CNN	DnCNN [56]	38.56/.910	32.43/.790	0.56	36.63
		CBDNet* [22]	30.78/.801	38.06/.942	4.34	34.02
		RIDNet [4]	38.71/.951	39.26/.953	1.49	197
		VDN [48]	39.28/.956	39.38/.952	7.81	99.00
		InvDN [29]	39.28/.955	39.57/.953	2.64	47.80
		DANet [49]	39.30/.916	39.47/.955	9.15	14.85
		DeamNet* [40]	39.47/.955	39.63/.953	2.23	146
		Thunder [61]	39.50/.956	39.57/.953	2.68	18.81
		CycleISP* [51]	39.52/.957	39.56/.956	2.84	184
	Transformer	Uformer16 [46]	39.57/.957	39.66/.952	5.29	11.39
		SwinIR-lw [28]	39.48/.957	39.61/.952	2.25	146
		Restormer-lw [50]	39.53/.957	39.70/.953	2.60	18.61
		RAMiT [16]	39.52/.957	39.70/.953	2.45	112
		LIDFormer [62]	39.63/.958	39.76/.956	2.72	2.83
		NLformer (Ours)	**39.88**/**.959**	**39.92**/**.956**	2.44	22.19

most existing works, training datasets are combination of DIV2K (800), Flickr2K (2650), BSD (400), and WED (4744) [2,32,34,44]. Noisy images are synthesized by adding AWGN with noise variance randomly sampled from $\sigma \in [10, 55]$. Testing datasets include Set12, (C)BSD68, Kodak24, McMaster, and Urban100. For real-world denoising, we follow [50], training on SIDD Medium and evaluating on the testset of SIDD and DND [1,36]. The PSNR and SSIM metrics are used to evaluate the performance. Computational cost is measured in terms FLOPs on an sRGB image of size 256×256, and total inference time on SIDD test set.

4.2 Performance evaluation

Real-world datasets: To assess the performance of NLformer on real-world image denoising, we compare it with a comprehensive set of lightweight denoising networks, including CBDNet [22], RIDNet [4], DeamNet [40], InvDN [29], DANet [49], VDN [48], Uformer16 [46], SwinIR-lw [28] (short for lightweight), Restormer-lw [50], LIDFormer [62] and RAMiT [16]. We also report results from full-size models to illustrate the performance gap. As shown in Table 1, NLformer

Table 2: PSNR (dB) results on synthesized dataset of color images with AWGN. The lightweight models with the best and second best score are marked as **bold** and underline, respectively. † denotes the model is non-blind to noise level.

Category		Method	CBSD68 [34]			Kodak24 [20]			McMaster [59]			Urban100 [25]		
			σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50	σ=15	σ=25	σ=50
Non-learning		NL-Bayes	33.47	30.68	27.32	34.38	32.27	28.45	34.45	31.88	28.67	-	-	-
		CBM3D	33.52	30.71	27.38	34.28	32.15	28.46	34.06	31.66	28.51	33.93	31.36	27.93
		WNNM	32.92	30.25	26.97	33.94	31.37	28.02	-	-	-	-	-	-
Large	CNN	RDN†	-	-	28.31	-	-	29.66	-	-	-	-	-	29.38
		DRUNet	34.30	31.69	28.51	35.31	32.89	29.86	35.40	33.14	30.08	34.81	32.60	29.61
	Transformer	SwinIR†	34.42	31.78	28.56	35.34	32.89	29.79	35.61	33.20	30.22	35.13	32.90	29.82
		Restormer	34.39	31.78	28.59	35.44	33.02	30.00	35.55	33.31	30.29	35.06	32.91	30.02
		ART†	34.46	31.84	28.63	35.39	32.95	29.87	35.68	33.41	30.31	35.29	33.14	30.19
		DSCAFormer	34.35	31.72	28.48	35.42	32.87	29.67	35.58	33.28	30.21	-	-	-
Lightweight	CNN	DnCNN	33.90	31.24	27.95	34.60	32.14	28.95	33.45	31.52	28.62	32.98	30.81	27.59
		IRCNN	33.86	31.16	27.86	34.69	32.18	28.93	34.58	32.18	28.91	33.78	31.20	27.70
		FFDNet	33.87	31.21	27.96	34.63	32.13	28.98	34.66	32.35	29.18	33.83	31.40	28.05
		BRDNet†	34.10	31.43	28.16	34.88	32.41	29.22	35.08	32.75	29.52	34.42	31.99	28.56
	Transformer	SwinIR-lw	34.16	31.50	28.22	35.18	32.69	29.54	35.23	32.90	29.71	34.59	32.23	28.89
		Restormer-lw	33.99	31.33	28.04	34.86	32.38	29.19	34.69	32.44	29.31	34.00	31.60	28.30
		CAT-lw	34.01	31.37	28.11	34.90	32.43	29.29	34.83	32.58	29.48	34.12	31.75	28.46
		ART-lw	34.08	31.40	28.08	35.00	32.49	29.27	35.10	32.74	29.48	34.44	32.03	28.62
		NLformer	**34.33**	**31.71**	**28.51**	**35.34**	**32.92**	**29.88**	**35.42**	**33.17**	**30.12**	**34.87**	**32.67**	**29.68**

outperforms all compared lightweight methods on the SIDD and DND benchmarks with a notable margin, while maintaining a compact model size and low computational cost. Compared to large models, NLformer achieves a comparable performance with highly reduced model size and computational complexity.

Synthetic datasets: We compare our NLformer with a comprehensive set of lightweight denoising models, including DnCNN [56], IRCNN [57], FFDNet [58], BRDNet [43], SwinIR-lw [28], Restormer-lw [50], CAT-lw [14] and ART-lw [54]. To highlight the performance gap, we also include several full-size denoising networks for reference, such as RDN [60], DRUNet [55], SwinIR [28], Restormer [50], ART [54] and DSCAFormer [24]. As shown in Table 2 for color images, NLformer achieves the best performance among all lightweight models. Moreover, Figure 3 and Figure 4 show that our NLformer delivers superior visual quality.

More settings and other tasks: We also evaluate our NLformer on other synthetic denoising settings (*e.g.* grayscale images, mixed Poisson-Gaussian noise) and other restoration tasks (*e.g.* plug-and-play deblurring, super-resolution). See supplementary file for the studies. Same as the case of color images, our NLformer remains the top performer among lightweight models.

Computational overhead: To verify the computational efficiency of NLformer, we compare it with representative Transformer-based denoisers. As shown in Table 3, our proposed model achieves the lowest inference time on the SIDD test set. Notably, compared to large-scale models, NLformer delivers comparable denoising performance with substantially reduced computational cost.

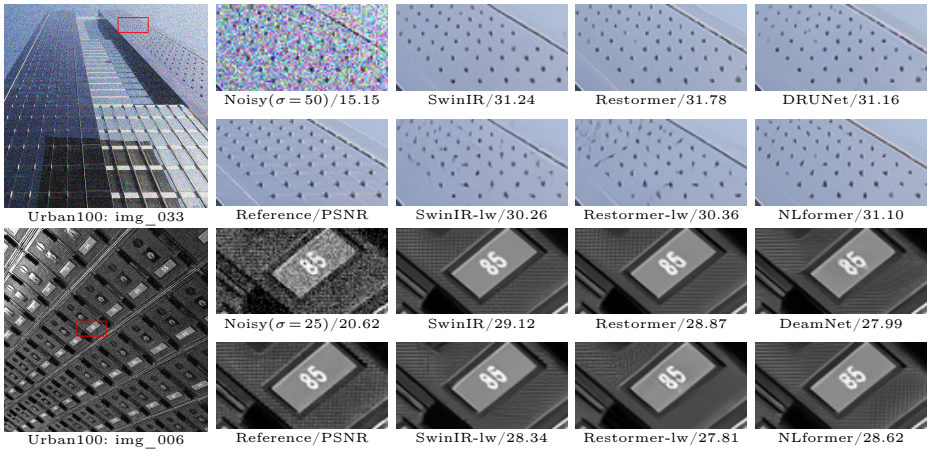

Fig. 3: Visual comparisons on Gaussian color and grayscale image denoising.

Table 3: Comparison on inference time, memory cost, model size and FLOPs cost.

Model	Large		Lightweight			
	Uformer32	Restormer	Uformer16	Restormer-lw	RAMiT	NLformer
Inference time (s)	38.4	400.6	30.8	21.5	119.1	19.5
Memory cost (G)	2.24	2.41	1.94	2.03	2.35	2.09
Params (M)	20.63	26.11	5.29	2.6	2.45	2.44
FLOPs (G)	43.86	155	11.39	18.61	112	22.19
PSNR/SSIM	39.77/.959	40.02/.960	39.57/.957	39.53/.957	39.52/.957	39.88/.959

4.3 Ablation study

Impact of model size: We evaluate NLformer's performance across model sizes, from lightweight to large. Table 4 shows the results on Gaussian and real-world denoising. Increasing the model size from lightweight($\sim$ 2M) to large($\sim$ 25M) yields significant gains, both on Gaussian and real-world noise. Notably, the benefit of second-order statistics remains evident as model capacity increases.

Impact of different component choice in SA: We study the impact of three components in the implementation of SA, summarized in Table 5. To be fair, all variants of the model in the study has comparable model size. (a) *Overlapped attention window*: Replacing it with non-overlapped ones will lead to the unbalanced attention range for patches at the corner of a window, which results in a sub-optimal performance on real-world noise. (b) *Dual-branch attention*: Dual branches in SA incorporate both first-order inter-patch and second-order intra-patch statistics. Removing either branch discards essential statistical information, leading to a noticeable degradation in performance.

Impact of different component choice in FFN: We study the following FFN designs, as summarized in Table 5: (a) *Plain linear FFN* composed of two convolutional layers learns a unified linear transform for all patches, ignor-

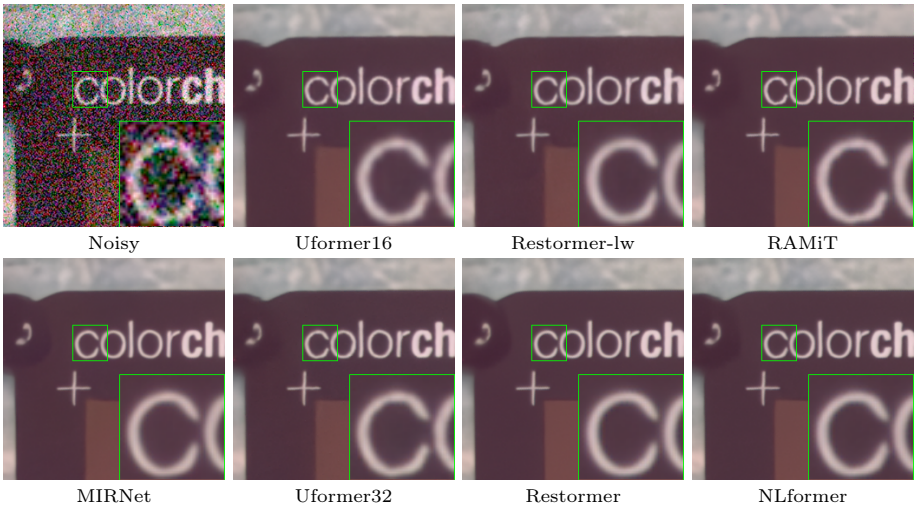

Fig. 4: Visual comparison on real-world image denoising selected from SIDD.

Table 4: Denoising performances under different model sizes.

Size	Gaussian (Urban 100 dataset)			Real-world		Params	FLOPs
	$\sigma=15$	$\sigma=25$	$\sigma=50$	SIDD	DND	(M)	(G)
Lightweight	34.87/.951	32.67/.928	29.68/.883	39.88/.959	39.92/.956	2.44	22.19
Large	35.10/.952	32.96/.931	30.04/.889	40.11/.960	40.08/.956	26.66	136

ing variant second-order statistics and resulting in sub-optimal performance. (b) *Standard gated FFN* [50] lacks an explicit matrix inversion and is therefore unable to accurately approximate the optimal estimator. (c) *Our proposed WIFN* incorporates an explicit matrix inversion motivated by the optimal Bayesian estimator 18. This design fully exploits second-order intra-patch information, thereby delivering the best performance at a lightweight design.

Impact of low-rank approximation: This ablation examines the trade-off between modeling accuracy and computational efficiency introduced by the low-rank approximation by comparing performance across different ranks r. As shown in Table 6, increasing r from 0 to 1 and from 1 to 2 yields clear improvements, while the difference becomes marginal at $r = 3$.

Impact of token embedding: This ablation examines the token-space implementation of NLformer, where each token is generated from a raw patch by convolutional embedding. Table 7 shows that the raw-patch variant also performs well, while the learned embedding gives further gains.

4.4 More experiments and details in the supplementary file

– The proof of Proposition 1.

Table 5: Ablation experiments for components of the NLformer structure.

Module	Variant	SIDD	DND	Params (M)	FLOPs (G)
Self-Attention	w/o Overlapped window	39.73/.958	39.87/.955	2.37	19.29
	w/o First-order SA	39.63/.958	39.56/.952	1.99	17.72
	w/o Second-order SA	39.55/.952	39.70/.953	2.25	18.90
FFN	Plain Linear	39.59/.957	39.69/.953	2.17	19.40
	w/o Woodbury inversion	39.74/.959	39.69/.953	2.44	22.02
NLformer (DWSA+WIFN)		**39.88**/**.959**	**39.92**/**.956**	2.44	22.19

Table 6: Ablation studies on rank of ΔH_i

Rank	$r=0$	$r=1$	$r=2$	$r=3$
SIDD	39.67/.958	39.77/.958	39.88/.959	39.89/.959
DND	39.55/.952	39.71/.953	39.92/.956	39.85/.955
Params/M	2.17	2.35	2.44	2.54
FLOPs/G	19.78	21.27	22.19	22.77

Table 7: Ablation on raw-pixel patches vs. token embedding.

Space	SIDD
Raw-pixel patches	39.68/0.958
Token embedding	39.88/0.959

– Detailed description on datasets involved.
– Additional results on gray-scale images and other types of noise.
– Experimental results on other image restoration tasks.
– Ablation study on the proposed Bayesian estimator (18).
– Visual comparisons on more examples for Gaussian and real-world denoising.
– Implementation details of NLformer.
– Detailed experimental setting for other compared lightweight Transformers.

5 Conclusion

We presented a lightweight denoising Transformer derived from a Bayesian view of self-attention. By combining first-order inter-patch and second-order intra-patch statistics, it captures both non-local similarity and local pixel dependencies. This yields a lightweight architecture with strong denoising performance and low complexity. Experiments on synthetic and real-world benchmarks show that NLformer outperforms existing lightweight denoising networks and narrows the gap to full-size Transformer-based denoisers.

Acknowledgements

This work was supported in part by the Singapore Ministry of Education Academic Research Fund Tier 2 under Grant No. A-8004364-00-00, and by the NUSRI-SZ research grant SZ2025-03.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1692–1700 (2018)
2. Agustsson, E., Timofte, R.: Ntire 2017 challenge on single image super-resolution: Dataset and study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 126–135 (2017)
3. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Transactions on Signal Processing **54**(11), 4311–4322 (2006)
4. Anwar, S., Barnes, N.: Real image denoising with feature attention. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3155–3164 (2019)
5. Brateanu, A., Balmez, R., Avram, A., Orhei, C.: Akdt: Adaptive kernel dilation transformer for effective image denoising. In: Proceedings of the 20th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications. vol. 3, pp. 418–425 (2025)
6. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. vol. 2, pp. 60–65 (2005)
7. Buades, A., Coll, B., Morel, J.M.: A review of image denoising algorithms, with a new one. Multiscale modeling & simulation **4**(2), 490–530 (2005)
8. Buades, A., Coll, B., Morel, J.M.: Non-local means denoising. Image Processing On Line **1**, 208–212 (2011)
9. Cai, J.F., Ji, H., Shen, Z., Ye, G.B.: Data-driven tight frame construction and image denoising. Applied and Computational Harmonic Analysis **37**(1), 89–105 (2014)
10. Chen, H., Wang, Y., Guo, T., Xu, C., Deng, Y., Liu, Z., Ma, S., Xu, C., Xu, C., Gao, W.: Pre-trained image processing transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12299–12310 (2021)
11. Chen, S., Ye, T., Liu, Y., Chen, E.: Dual-former: Hybrid self-attention transformer for efficient image restoration. Digital Signal Processing **149**, 104485 (2024)
12. Chen, X., Wang, X., Zhang, W., Kong, X., Qiao, Y., Zhou, J., Dong, C.: Hat: Hybrid attention transformer for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence **48**(3), 2676–2694 (2026)
13. Chen, X., Wang, X., Zhou, J., Qiao, Y., Dong, C.: Activating more pixels in image super-resolution transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 22367–22377 (2023)
14. Chen, Z., Zhang, Y., Gu, J., Kong, L., Yuan, X., et al.: Cross aggregation transformer for image restoration. In: Advances in Neural Information Processing Systems. vol. 35, pp. 25478–25490. Curran Associates, Inc. (2022)
15. Choi, H., Na, C., Kim, J., Yang, J.: Exploration of lightweight single image denoising with transformers and truly fair training. In: Proceedings of the 2023 ACM International Conference on Multimedia Retrieval. pp. 452–461 (2023)
16. Choi, H., Na, C., Oh, J., Lee, S., Kim, J., Choe, S., Lee, J., Kim, T., Yang, J.: Reciprocal attention mixing transformer for lightweight image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 5992–6002 (2024)

17. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Transactions on Image Processing **16**(8), 2080–2095 (2007)
18. Dai, T., Cai, J., Zhang, Y., Xia, S.T., Zhang, L.: Second-order attention network for single image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11065–11074 (2019)
19. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021)
20. Franzen, R.: Kodak lossless true color image suite (1999)
21. Gu, S., Zhang, L., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2862–2869 (2014)
22. Guo, S., Yan, Z., Zhang, K., Zuo, W., Zhang, L.: Toward convolutional blind denoising of real photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1712–1722 (2019)
23. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
24. Hu, Y., Cheng, D., Huang, Z., Chen, B., Lin, S., Zhang, S.: DSCA-former: Dual-stem cross-attentive transformer for image denoising. Neurocomputing **671**, 132656 (2026)
25. Huang, J.B., Singh, A., Ahuja, N.: Single image super-resolution from transformed self-exemplars. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5197–5206 (2015)
26. Ji, H., Liu, C., Shen, Z., Xu, Y.: Robust video denoising using low rank matrix completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1791–1798 (2010)
27. Lebrun, M., Buades, A., Morel, J.M.: A nonlocal bayesian image denoising algorithm. SIAM Journal on Imaging Sciences **6**(3), 1665–1688 (2013)
28. Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., Timofte, R.: Swinir: Image restoration using swin transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops. pp. 1833–1844 (2021)
29. Liu, Y., Qin, Z., Anwar, S., Ji, P., Kim, D., Caldwell, S., Gedeon, T.: Invertible denoising network: A light solution for real noise removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13365–13374 (2021)
30. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2017)
31. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019)
32. Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing **26**(2), 1004–1016 (2016)
33. Mallat, S.: A wavelet tour of signal processing. Elsevier (1999)
34. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. vol. 2, pp. 416–423 (2001)

35. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
36. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1586–1595 (2017)
37. Quan, Y., Chen, Y., Shao, Y., Teng, H., Xu, Y., Ji, H.: Image denoising using complex-valued deep cnn. Pattern Recognition **111**, 107639 (2021)
38. Quan, Y., Zheng, T., Ji, H.: Pseudo-siamese blind-spot transformers for self-supervised real-world denoising. In: Advances in Neural Information Processing Systems. vol. 37, pp. 13794–13817. Curran Associates, Inc. (2024)
39. Ren, B., Li, Y., Liang, J., Ranjan, R., Liu, M., Cucchiara, R., Van Gool, L., Sebe, N.: Key-graph transformer for image restoration. arXiv preprint arXiv:2402.02634 (2024)
40. Ren, C., He, X., Wang, C., Zhao, Z.: Adaptive consistency prior based deep network for image denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 8596–8606 (2021)
41. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 234–241. Springer (2015)
42. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D: nonlinear phenomena **60**(1-4), 259–268 (1992)
43. Tian, C., Xu, Y., Zuo, W.: Image denoising using deep cnn with batch renormalization. Neural Networks **121**, 461–473 (2020)
44. Timofte, R., Agustsson, E., Van Gool, L., Yang, M.H., Zhang, L.: Ntire 2017 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 114–125 (2017)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017)
46. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17683–17693 (2022)
47. Woodbury, M.A.: Inverting modified matrices. Department of Statistics, Princeton University (1950)
48. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational denoising network: Toward blind noise modeling and removal. In: Advances in Neural Information Processing Systems. vol. 32. Curran Associates, Inc. (2019)
49. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: Toward real-world noise removal and noise generation. In: Proceedings of European Conference on Computer Vision. pp. 41–58. Springer (2020)
50. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5728–5739 (2022)
51. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Cycleisp: Real image restoration via improved data synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2696–2705 (2020)

52. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Learning enriched features for real image restoration and enhancement. In: Proceedings of European Conference on Computer Vision. pp. 492–511. Springer (2020)
53. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14821–14831 (2021)
54. Zhang, J., Zhang, Y., Gu, J., Zhang, Y., Kong, L., Yuan, X.: Accurate image restoration with attention retractable transformer. In: International Conference on Learning Representations (2023)
55. Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., Timofte, R.: Plug-and-play image restoration with deep denoiser prior. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(10), 6360–6376 (2021)
56. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
57. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep cnn denoiser prior for image restoration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3929–3938 (2017)
58. Zhang, K., Zuo, W., Zhang, L.: FFDNet: Toward a fast and flexible solution for cnn-based image denoising. IEEE Transactions on Image Processing **27**(9), 4608–4622 (2018)
59. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic Imaging **20**(2), 023016–023016 (2011)
60. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence **43**(7), 2480–2495 (2020)
61. Zhou, Y., Xu, X., Liu, S., Wang, G., Lu, H., Shen, H.T.: Thunder: Thumbnail based fast lightweight image denoising network. arXiv preprint arXiv:2205.11823 (2022)
62. Zhou, Y., Lin, J., Ye, F., Qu, Y., Xie, Y.: Efficient lightweight image denoising with triple attention transformer. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 7704–7712 (2024)