Diffusion-Generated Fake Face Detection by Exploring Wavelet Domain Forgery Clues

Yufan Deng[1], Xin Deng[1*], Yiping Duan [2] Mai Xu[3]
[1] School of Cyber Science and Technology, Beihang University, Beijing, China
[2] Department of Electronic Engineering, Tsinghua University, Beijing, China
[3] School of Electronic and Information Engineering, Beihang University, Beijing, China
{yfdeng, cindydeng, maixu}@buaa.edu.cn, yipingduan@mail.tsinghua.edu.cn

***Abstract*—Recently, diffusion models have shown remarkable success in generating high-quality images, making them potentially more difficult to detect than GAN-generated images. In this paper, we address the emerging challenge of detecting face forgeries generated by diffusion models. We leverage insights from frequency artifacts in GAN-generated images and delve into the frequency domain characteristics of diffusion-generated images using both discrete Fourier transform (DFT) and Haar discrete wavelet transform (HDWT). Our investigation reveals that, while diffusion-generated images lack obvious DFT artifacts like those in GAN-generated images, they exhibit fewer high-frequency details compared to real images. Building upon these observations, our multi-scale network incorporates wavelet lifting and wavelet-spatial transformer blocks, enabling precise frequency decomposition and efficient feature fusion. Experimental results on two generated datasets demonstrate the superior robustness of our proposed method compared to state-of-the-art forgery detection methods, making it an effective solution for identifying face forgeries produced by diffusion models. The dataset and code are open source at https://github.com/eecoder-dyf/Diffusion-Detection-WCSP2023.**

***Index Terms*—Diffusion-generated images, Forgery detection, Wavelet domain analysis.**

I. Introduction

Recently, diffusion models [1]–[3] have achieved significant success in the field of image generation, overcoming many drawbacks of Generative Adversarial Network (GAN) based generation. Due to their ability to generate high-quality and realistic images of diffusion models, diffusion models could potentially enable attackers to develop new face forgery techniques based on them. Diffusion-based face forgery methods might be more challenging to distinguish, not only by humans but also by existing forgery detection methods [4]–[6], as the internal mechanism of diffusion models differ significantly from those of previous forgery methods based on GANs. Consequently, there is an urgent demand for the development of a detector capable of identifying diffusion-generated face images.

Previous forgery detection method usually focus on face manipulation detection or generated image detection. Face manipulation involves forging existing faces, potentially leading to spatial [7] or temporal inconsistencies [8]. In contrast to face manipulation, generated image detection typically targets faces synthesized by generation models like GANs, depicting individuals who do not exist in reality. Generated images usually exhibit less diversity than real images. For instance, GAN-generated images often exhibit frequency domain grid artifacts due to the up-sampling operations in the generator [9]. In essence, the key objective of forgery detection is to identify defects within fake images, a principle that also applies to the detection of diffusion-generated images.

In this paper, we draw inspiration from frequency artifacts observed in GAN-generated images and explore the frequency domain characteristics of diffusion-generated images. Initially, we visualize discrete Fourier transform (DFT) map of fake images and observe that the grid artifacts do not exist in diffusion-generated images. Subsequently, our investigation extends to wavelet domain, where we find that the diffusion-generated images contain less high-frequency information compared to real images. This finding leads us to design a network based on frequency decomposition methods. Specifically, we proposed a deepfake detection network for diffusion-generated images. The network is composed of 4 branches with different frequency decomposition levels. To achieve multi-scale learning across various frequency subbands, we introduce the wavelet lifting block, which is an adaptive frequency decomposition method. Furthermore, we design Wavelet-Spatial transformer block to fuse the decomposed wavelet domain features and spatial domain features effectively. These fused features are then fed into classification networks. The contribution of this paper can be summarized as follows:

- Our frequency domain analysis on these two datasets finds that diffusion-generated images contain less high frequency detail information.
- We design a multi-scale network with wavelet lifting blocks and wavelet-spatial transformer blocks to conduct better frequency decomposition and feature fusion.
- Experimental results on the two diffusion-generated datasets demonstrate that our proposed method exhibits superior robustness compared to other state-of-the-art forgery detection methods.

Corresponding author: Xin Deng.
This work is supported by supported by National Natural Science Foundation of China under Grants 62001016.

II. Related Works

A. Face Forgery Detection

Face forgery detection is a classic topic in computer vision. Typically, face forgery detection methods focus on detecting face manipulation forgery, including face swap and face reenactment [10]. Most existing face forgery methods [5], [6], [11] extract artifacts in spatial domain. For instance, Zhao *et al.* [5] designed a texture enhancement block to enhance local texture information and an attention module to extract global artifacts, leading to a local-global spatial domain forgery detection. Some methods also focus on frequency domain artifacts caused by face manipulation [7], [12], [13]. For example, Miao *et al.* [12] found that the diagonal high frequency component of 2D discrete wavelet transform (2D-DWT) exhibits artifacts in the context of fake images.

Recently, with the increasing realism of GAN-generated images, there has been a growing interest in the detection of images generated by GANs. Yu *et al.* [14] believe that every GAN model has unique fingerprints after training, and they extract these fingerprints from generated images to perform detection. Wang *et al.* [9] directly adopt pre-processing techniques such as JPEG compression and Gaussian blur during training to improve the robustness of detection. Since GAN-based generative models depend on up-scaling operations, the generated images contain unique frequency-level artifacts. Thus, frequency domain based methods are more effective in the aspect of GAN-generated image detection [4], [15].

B. Diffusion models

Inspired by nonequilibrium thermodynamics, Ho *et al.* [1] first introduced denoising diffusion probabilistic models (DDPMs) and achieves competitive performance compared to state-of-the-art (SOTA) GANs in image generation. The diffusion models gradually convert real data into noise and then learn to denoise the noisy image to obtain the generated image. Since then, many other forms of diffusion models have been proposed, such as Diffusion-GAN [3] and latent space diffusion [2]. To date, diffusion models have consistently outperformed GANs in image generation.

III. Motivation

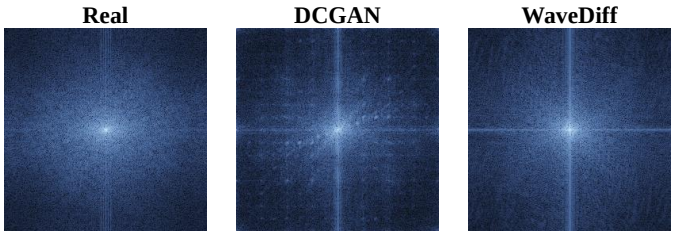

Fig. 1: Visualizations of the DFT spectrum of diffusion-generated fake images by WaveDiff [16], and GAN-generated images by DCGAN [17]. Brighter color represents larger value of DFT.

Previous works have primarily focused on generative adversarial network (GAN) to generate fake images. Thus, most fake detection methods are designed to distinguish the GAN-generated fake images. Wang *et al.* [9] and Frank *et al.* [15] visualized the spectrum of discrete Fourier transform (DFT) of GAN-generated and real images, revealing significant artifacts in the form of a regular grid. To investigate whether this phenomenon persists in diffusion-generated images, we also apply DFT on diffusion-generated images and visualize the results in Fig. 1.

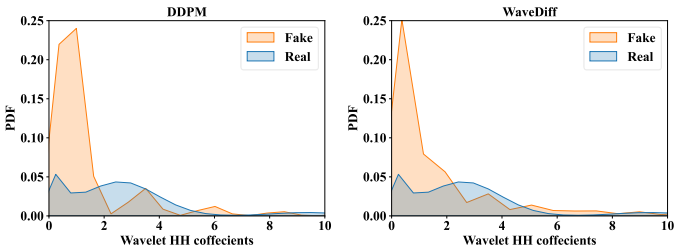

Fig. 2: Comparison of the probability density function (PDF) on fake images generated by diffusion-based methods DDPM [1] and WaveDiff [16], and real images on Celeba-HQ [18] dataset.

From Fig. 1, we can see that the diffusion-generated images contain no grid artifacts in the DFT domain. The reason behind this is that regular grids are mainly caused by the upsampling operations in the generator of GANs. GANs usually generate one image from a low dimensional vector. In contrast, diffusion networks usually generate one image from a sampled Gaussian noise image. Since it has the same resolution as the generated image, the upsampling operations are not needed. This indicates that diffusion-generated images are more challenging to detect than GAN-generated images.

Despite no grid artifact in DFT domain, we can still observe slight high-frequency artifacts in the DFT spectrum of diffusion-generated images. This observation leads us to focus more on the high frequency components of the fake images. Thus, we apply discrete wavelet transform (DWT) to all selected real and fake images and plot the probability density function (PDF) of the diagonal component (HH) wavelet coefficients of each single pixel in Fig. 2. As can be seen, the HH wavelet coefficients of fake images are more tightly clustered around 0. This indicates that the fake images exhibit fewer high frequency components than real images.

IV. Proposed Method

A. Framework

Fig. 3 shows the overall framework of our proposed method for diffusion-generated fake face detection. As illustrated in this figure, our framework is composed of four branches. The first branch is the spatial branch, while the other three branches are all frequency domain branches. As revealed in Section III, the fake images generated by diffusion models tend to lack high-frequency details. Specifically, after wavelet decomposition, the diagonal coefficients of the diffusion-generated

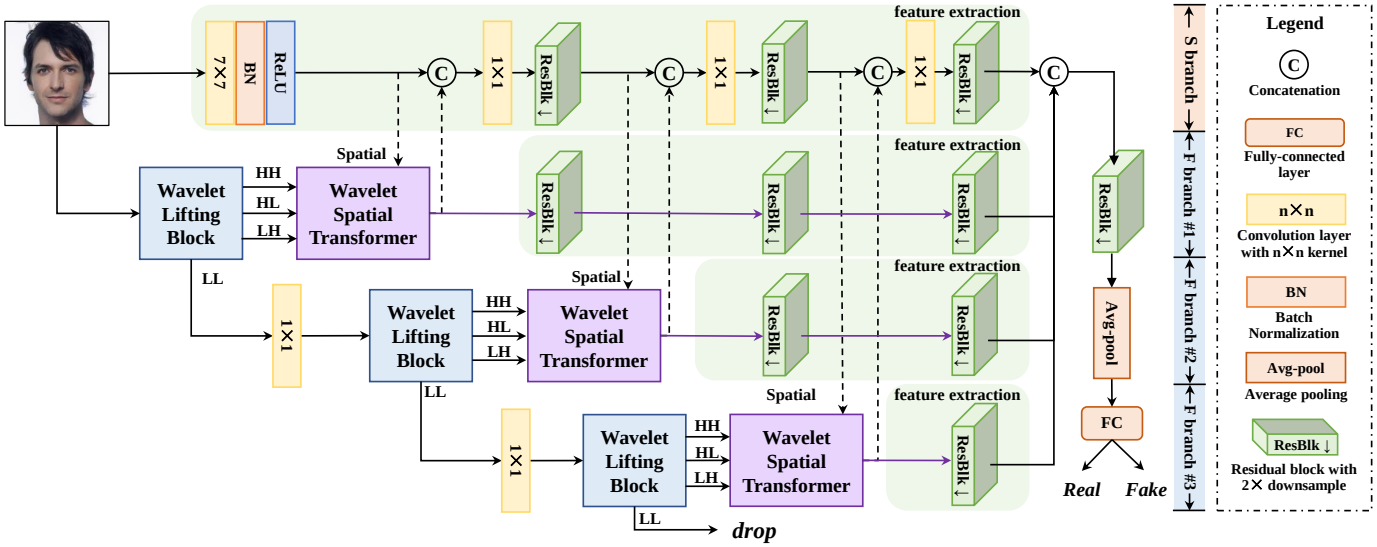

Fig. 3: The overall framework of our proposed network. Our network is composed of one spatial domain branch and three frequency domain branches. The "S branch" indicates spatial domain branch, and the "F branch #i" indicates the i-th frequency domain branch.

images are much closer to zero. This has prompted us to incorporate frequency departure schemes in our classification network architecture.

Inspired by the commonly used multi-level wavelet decomposition approach, we have incorporated wavelet lifting schemes based on [19] into our network to perform adaptive frequency decomposition. Each decomposition level contributes to the corresponding frequency domain branch, with the original facial image initially decomposed into four subband components. The low-frequency band undergoes the next decomposition level after channel alignment through a 1×1 convolution layer.

Due to wavelet lifting decomposition reducing the scale by a factor of 2, our network incorporates a multi-scale learning structure via the integration of three levels of wavelet lifting scheme blocks. In spatial domain branch, the facial image will be processed by a cascade of downsampling residual blocks to match up with the multi-level decomposition. For each level of decomposition, the three high-frequency band features and the feature with the same scale from the spatial domain branch are fused by the wavelet-spatial transformer block. Then, the fused features undergo the feature extraction stage based on downsampling residual blocks. Note that the fused features are also concatenated to the spatial domain branch to acquire an enhanced understanding of high-frequency features.

Finally, the output features from all branches are merged using a single residual block and then fed into a fully-connected layer for classification.

B. Wavelet Lifting Block (WLB)

To implement the 2D wavelet lifting scheme, we introduce a wavelet lifting block shown in Fig. 4, which decomposes the signal in two dimensions. The 2D wavelet lifting scheme initially decomposes signal $\boldsymbol{X} \in \mathbb{R}^{H \times W \times C}$ horizontally to

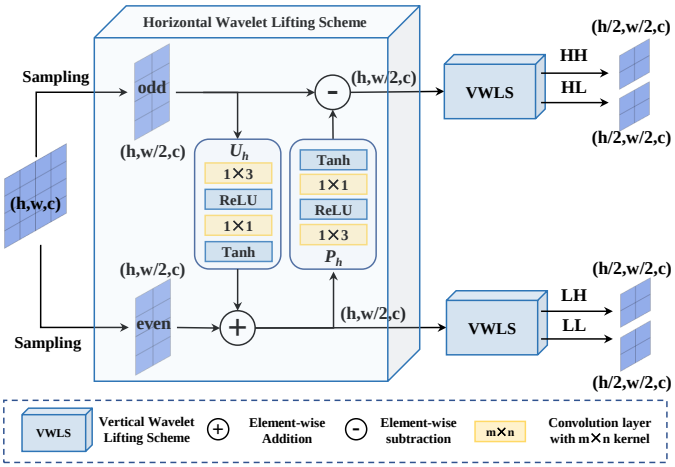

Fig. 4: The structure of our wavelet lifting block. We decompose input signal horizontally by horizontal wavelet lifting scheme (HWLS) and then vertically by vertical wavelet lifting scheme (VWLS).

$\boldsymbol{X}_H^h$ and $\boldsymbol{X}_L^h \in \mathbb{R}^{H \times \frac{W}{2} \times C}$ by horizontal wavelet lifting scheme (HWLS). It subsequently decomposes $\boldsymbol{X}_H^h$ and $\boldsymbol{X}_L^h$ respectively in vertical direction by vertical wavelet lifting scheme (VWLS). The entire wavelet lifting block can be formulated as follows:

$$\begin{aligned} \boldsymbol{X}_H^h, \boldsymbol{X}_L^h &= \mathrm{HWLS}(\boldsymbol{X}), \\ \boldsymbol{X}_{HH}, \boldsymbol{X}_{HL} &= \mathrm{VWLS}(\boldsymbol{X}_H^h), \\ \boldsymbol{X}_{LH}, \boldsymbol{X}_{LL} &= \mathrm{VWLS}(\boldsymbol{X}_L^h), \end{aligned} \tag{1}$$

where $\boldsymbol{X}_{HH}, \boldsymbol{X}_{HL}, \boldsymbol{X}_{LH}, \boldsymbol{X}_{LL} \in \mathbb{R}^{\frac{H}{2} \times \frac{W}{2} \times C}$ represent the diagonal high-frequency component, horizontal high-frequency component, vertical high-frequency component, and low-frequency component respectively.

The details of HWLS and VWLS. The details of HWLS are shown in Fig. 4. The lifting scheme is composed of three stages [20] as follows. In the first stage, the input signal $\boldsymbol{X}$ is divided into two non-overlapping partitions. We sample the odd-indexed and even-indexed elements of the horizontal direction to decomposed input $\boldsymbol{X} \in \mathbb{R}^{H\times W\times C}$ into two partitions $\boldsymbol{X}_{odd}, \boldsymbol{X}_{even} \in \mathbb{R}^{H\times \frac{W}{2}\times C}$. Then in the second stage, we conduct an updater function to update $\boldsymbol{X}_{odd}$ and add it with $\boldsymbol{X}_{even}$ to get the low-frequency component. This stage can be formulated as:

$$\boldsymbol{X}_L^h = \boldsymbol{X}_{even} + U_h(\boldsymbol{X}_{odd}), \tag{2}$$

where U_h represents the updater function, it can be any function. In the third stage, a predictor function is conducted to predict the redundant low-frequency information in $\boldsymbol{X}_{odd}$ from $\boldsymbol{X}_L^h$. Then we subtract the redundant information from $\boldsymbol{X}_{odd}$ to get the high-frequency component. This stage can be formulated as:

$$\boldsymbol{X}_H^h = \boldsymbol{X}_{odd} - P_h(\boldsymbol{X}_L^h) \tag{3}$$

where P_h represents the predictor function, it can also be any function. In this paper, we adopt a learnable convolutional neural network for U_h and P_h to achieve adaptive frequency decomposition. As shown in Fig. 4, for the input signal of U_h or P_h, reflection padding is first applied to prevent harmful border effect caused by convolution operation. Subsequently, a 2D convolution layer with 1×3 kernel size is applied, followed by the ReLU activation function. Next, a second convolutional layer with 1×1 kernel size is applied, followed by a tanh activation function. Note that U_h and P_h hold the same network structure but do not share network parameters.

For VWLS, the structure is almost the same as HWLS. The differences include: 1) Sampling odd and even terms of $\boldsymbol{X}$ in vertical direction; 2) In updater U_v and predictor P_v, 1×3 convolution layers are substituted by 3×1 convolution layers.

C. Wavelet-Spatial Transformer (WST) block

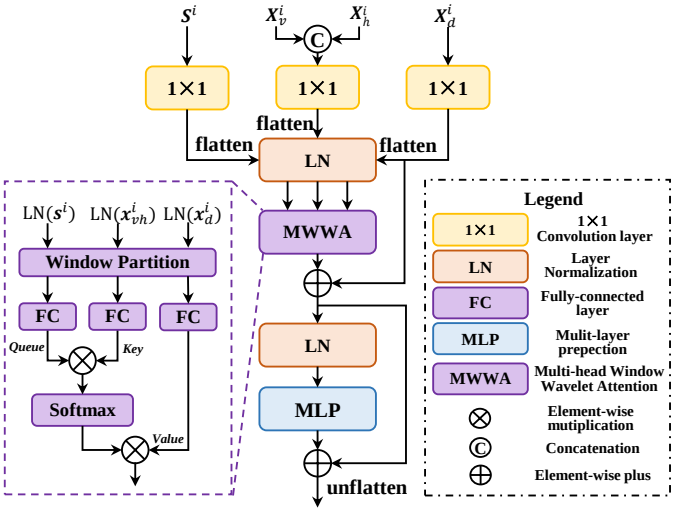

Fig. 5: The structure of wavelet-spatial transformer block.

In our network, we propose wavelet-spatial transformer to fuse the wavelet domain features and spatial domain features. As illustrated in Fig. 3, the inputs of wavelet-spatial transformer consist of three high frequency wavelet components, namely $\boldsymbol{X}_d^i$, $\boldsymbol{X}_h^i$ and $\boldsymbol{X}_v^i$, decomposed from wavelet lifting blocks, along with the corresponding spatial domain feature $\boldsymbol{S}^i$. Here, the subscript i denotes $i\ th$ wavelet domain branch, $\boldsymbol{X}_d^i$, $\boldsymbol{X}_h^i$ and $\boldsymbol{X}_v^i$ represent the decomposed diagonal, horizontal and vertical high frequency component, respectively.

The detailed structure of our proposed wavelet-spatial transformer (WST) is shown in Fig. 5. In WST, we first concatenate $\boldsymbol{X}_h^i$ and $\boldsymbol{X}_v^i$ as $\boldsymbol{X}_{vh}^i$, then a 1×1 convolution layer is applied to $\boldsymbol{S}^i$, $\boldsymbol{X}_d^i$ and $\boldsymbol{X}_{vh}^i$, respectively. Then these three features are flattened and layer normalized. Additionally, as we analyzed in Section III that the high frequency information might distinguish fake and real image, we regard $\boldsymbol{X}_d^i$ as the residual term in transformer to reserve more diagonal high frequency information. Then, the WST can be defined as follows,

$$\begin{aligned} \boldsymbol{x}_f &= \mathrm{MWWA}(\mathrm{LN}(\boldsymbol{s}^i), \mathrm{LN}(\boldsymbol{x}_d^i), \mathrm{LN}(\boldsymbol{x}_{vh}^i)) + \boldsymbol{x}_d^i, \\ \hat{\boldsymbol{x}}_f &= \mathrm{MLP}(\mathrm{LN}(\boldsymbol{x}_f)) + \boldsymbol{x}_f, \end{aligned} \tag{4}$$

where LN indicates layer normalization, MWWA indicates our proposed Multi-head Window Wavelet Attention layer, MLP indicates multi-layer perception, $\boldsymbol{s}^i$, $\boldsymbol{x}_d^i$ and $\boldsymbol{x}_{vh}^i$ are the flattened input features for the first LN layer. $\hat{\boldsymbol{x}}_f$ is the output of the transformer. Finally, $\hat{\boldsymbol{x}}_f$ is unflattened to $\hat{\boldsymbol{X}}_f$ as the output of WST.

The core fusion step in WST is performed by Multi-head Window Wavelet Attention (MWWA) layer. The MWWA takes $\mathrm{LN}(\boldsymbol{s}^i)$, $\mathrm{LN}(\boldsymbol{x}_d^i)$, and $\mathrm{LN}(\boldsymbol{x}_{vh}^i)$ as the input. In the MWWA layer, we draw inspiration from the window Attention scheme utilized in Swin Transformer [21] and incorporate it into our network. Thus, we first apply window partition for window attention. Then, the calculation of MWWA is as follows:

$$\begin{aligned} &\mathrm{MWWA}(\boldsymbol{y}_{vh}, \boldsymbol{y}_s, \boldsymbol{y}_d) = \mathrm{Softmax}(\frac{\boldsymbol{Q}_{vh}\boldsymbol{K}_s^T}{\sqrt{u}} + \boldsymbol{B})\boldsymbol{V}_d, \\ &\text{where } \boldsymbol{Q}_{vh} = \boldsymbol{W}^Q\boldsymbol{y}_{vh}, \boldsymbol{K}_s = \boldsymbol{W}^K\boldsymbol{y}_s, \boldsymbol{V}_d = \boldsymbol{W}^V\boldsymbol{y}_d. \end{aligned} \tag{5}$$

In Eq. (5), $\boldsymbol{y}_{vh}$, $\boldsymbol{y}_s$ and $\boldsymbol{y}_d$ are the windowed sequences of $\mathrm{LN}(\boldsymbol{x}_{vh}^i)$, $\mathrm{LN}(\boldsymbol{x}_s^i)$ and $\mathrm{LN}(\boldsymbol{x}_d^i)$, respectively. The $\boldsymbol{W}^Q$, $\boldsymbol{W}^K$ and $\boldsymbol{W}^V$ correspond to the weight matrices of query, key and value sequences, respectively.

Through the Wavelet-Spatial transformer block, we utilize spatial domain features along with horizontal and vertical wavelet domain features to guide the extraction of diagonal high frequency domain features. This guidance is accomplished through the residual of $\boldsymbol{x}_d^i$ and $\boldsymbol{y}_d$ as the value sequence in MWWA.

V. Experiments

A. Experiment Setup

Dataset. We choose CelebA-HQ-256 [18] as the real face dataset, which includes 30K face images with a resolution of 256×256. For the fake face dataset, we choose two diffusion

TABLE I: The robustness evaluation of accuracy (Acc) and AUC on WaveDiff [16] and DDPM [1] datasets. The best results are marked in bold and second bests are underlined.

Method	Dataset	Metric	Gaussian Blur			Gaussian Noise			JPEG Compression		
			σ=1	σ=2	σ=3	σ^2=3	σ^2=5	σ^2=10	QF=30	QF=50	QF=70
RFM [6]	WaveDiff	Acc	0.5000	0.5000	0.5000	0.6583	0.6110	0.5352	0.6738	0.9242	0.9897
Patch-fore [22]			0.5569	0.5534	0.5534	0.5880	0.5459	0.5242	0.5527	0.5982	0.6311
MesoNet [11]			0.5000	0.5000	0.5000	0.5885	0.5563	0.5173	0.8292	0.6248	0.9113
MAT [5]			0.5009	0.5000	0.5001	0.6712	0.6125	0.5549	**0.9008**	0.8994	0.9977
NAFID [23]			0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
Ours			**0.8998**	**0.7885**	**0.7627**	**0.9547**	**0.8927**	**0.6677**	0.7065	**0.9913**	**0.9987**
RFM [6]		AUC	0.8844	0.8402	0.8363	0.7899	0.6805	0.5590	0.8630	0.9801	0.9994
Patch-fore [22]			0.6207	0.6124	0.6133	0.6871	0.6416	0.5799	0.5789	0.6364	0.6768
MesoNet [11]			0.5237	0.5428	0.5555	0.6021	0.5643	0.5243	0.8947	0.7844	0.9686
MAT [5]			0.7469	0.7412	0.7398	0.7948	0.7097	0.5977	**0.9649**	0.9696	**1.0000**
NAFID [23]			0.4961	0.4961	0.4965	0.5014	0.5011	0.5029	0.5041	0.5025	0.5032
Ours			**0.9689**	**0.8995**	**0.8818**	**0.9912**	**0.9600**	**0.7553**	0.8332	**0.9995**	**1.0000**
RFM [6]	DDPM	Acc	0.5035	0.5013	0.5013	0.8608	0.8378	0.7477	0.6983	0.6177	0.5167
Patch-fore [22]			0.5717	0.5801	0.5762	**0.9075**	**0.8758**	**0.8060**	0.5129	0.5165	0.5113
MesoNet [11]			0.5547	0.5157	0.5137	0.8230	0.6935	0.5445	0.5893	0.5218	0.5548
MAT [5]			0.8365	0.7689	0.7540	0.6597	0.6537	0.7003	0.7011	0.8606	0.9436
NAFID [23]			0.9537	0.9532	0.9525	0.7240	0.6860	0.6273	**0.9647**	0.9733	0.9663
Ours			**0.9825**	**0.9765**	**0.9752**	0.8898	0.8482	0.7592	0.9600	**0.9898**	**0.9923**
RFM [6]		AUC	0.8721	0.7801	0.7676	**0.9918**	**0.9880**	**0.9666**	0.7791	0.9422	0.8256
Patch-fore [22]			0.8497	0.8435	0.8428	0.9125	0.8816	0.8075	0.6198	0.6491	0.6484
MesoNet [11]			0.6902	0.6273	0.6228	0.8589	0.7175	0.5632	0.7012	0.7118	0.7084
MAT [5]			0.9275	0.8610	0.8477	0.7366	0.7160	0.7580	0.7965	0.9313	0.9833
NAFID [23]			0.9950	0.9950	0.9950	0.7898	0.7700	0.7554	**0.9941**	0.9961	0.9947
Ours			**0.9976**	**0.9960**	**0.9957**	0.9518	0.9181	0.8287	0.9915	**0.9993**	**0.9995**

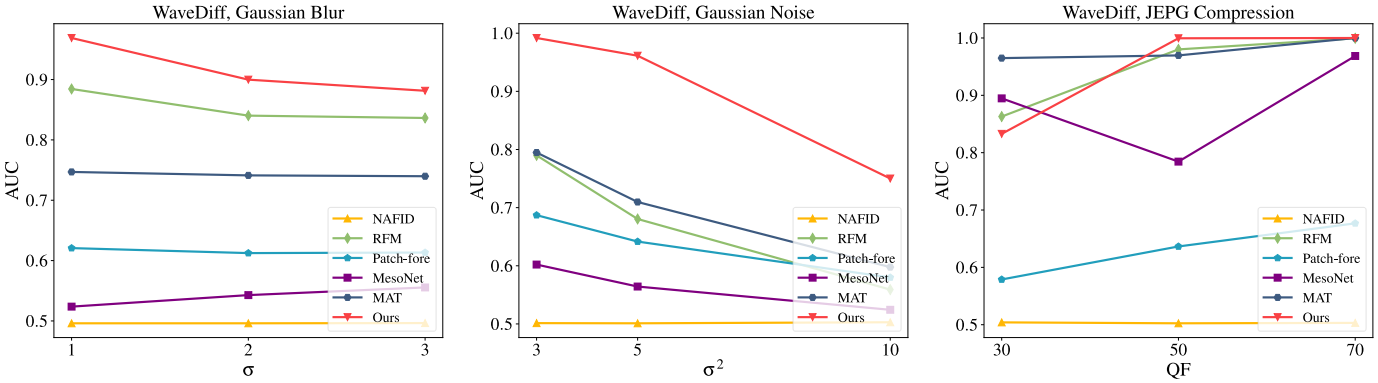

Fig. 6: The robustness evaluation on WaveDiff fake dataset with Gaussian blur, Gaussian noise and JPEG compression.

models, DDPM [1] and WaveDiff [16], and use their open-source pretrained model to generate an equivalent number of images (30K) as in the real data. We split the real data and fake data in an 8:1:1 ratio (8 for training and 1 for validation and test).

Implementation details. For each fake dataset, we mix it with the real dataset and train the detection model separately. During training, we randomly apply data augmentation methods including brightness, saturation and contrast transformation. For the residual blocks, we load the pretrained ResNet-18 [24] weights on ImageNet [25] dataset. The detection of generated fake images is a binary classification task, thus, we directly apply the cross entropy loss to train the entire network. We train the model for 50 epochs with the Adam optimizer. The batch size and learning rate are set to 64 and 1×10^{-4}, respectively. All the experiments are implemented by Pytorch and trained on one NVIDIA RTX 4090 GPU.

B. Performance Evaluation

We compare our method with five state-of-the-art methods, including MesoNet [11], Patch-fore [22], RFM [6], MAT [5], and NAFID [23]. For all the comparison methods, we retrain and evaluate them on our generated fake dataset. For deepfake detection, the robustness of the model is important. Thus, we apply three types of distortions to the original image, including Gaussian blur, Gaussian noise and JPEG compression for evaluation. For Gaussian blur, we apply 3×3 blur kernel with standard deviation σ=1,2,3. For Gaussian noise, we apply mean equals to zero and variance σ^2=3,5,10. For JPEG compression, we apply quality factor (QF)=30,50,70. Note that we do not apply any distortions while training for both all the comparison methods and our method.

Table I shows the quantitative results with different kinds and levels of distortions on WaveDiff and DDPM fake dataset, respectively. We use accuracy (Acc) and AUC as the metrics

TABLE II: The result of different ablation study cases on WaveDiff fake dataset.

Method	Metric	Gaussian Blur			Gaussian Noise			JPEG Compression		
		σ=1	σ=2	σ=3	σ^2=3	σ^2=5	σ^2=10	QF=30	QF=50	QF=70
No-WST	Acc	0.6213	0.6330	0.6393	0.9023	0.8138	0.6087	**0.8557**	0.9477	0.9892
Haar		0.5058	0.5023	0.5020	0.5820	0.5190	0.5008	0.5878	0.9248	0.9450
Ours		**0.8998**	**0.7885**	**0.7627**	**0.9547**	**0.8927**	**0.6677**	0.7065	**0.9913**	**0.9987**
No-WST	AUC	0.7654	0.7761	0.7823	0.9642	0.8951	0.6932	**0.9394**	0.9879	0.9991
Haar		0.6923	0.6646	0.6605	0.6905	0.6111	0.5424	0.7133	0.9772	0.9862
Ours		**0.9689**	**0.8998**	**0.8818**	**0.9912**	**0.9600**	**0.7553**	0.8332	**0.9995**	**1.0000**

to evaluate the performance of our model and other models. From the table, it can be observed that our method achieves the best or the second-best performance when facing most types and intensities of distortions. Especially on WaveDiff dataset, which is also visualized in Fig. 6. These results demonstrate the superiority of our proposed method of diffusion-generated image detection.

C. Ablation Study

In order to investigate the effectiveness of the proposed components of our network, we do several ablation experiments on WaveDiff dataset. **Wavelet Lifting block:** We replace WLB with Haar discrete wavelet transform, the results are marked as "Haar" in Table II. **Wavelet-Spatial Transformer:** We replace WST with simply concatenation and 3×3 convolution, the results are marked as "No-WST" in Table II. As can be seen, most of the results are worse than our method, indicating the effectiveness of the corresponding components.

VI. Conclusion

In this paper, aiming at diffusion-generated images, we applied discrete Fourier transform and Haar discrete wavelet transform (HDWT) and found that generated images exhibit fewer high frequency details. Based on these observations, we designed wavelet lifting blocks (WLB) in our network, which can achieve adaptive frequency decomposition. Moreover, we designed Wavelet-Spatial transformer (WST) block to fuse the spatial domain features and wavelet domain features. At last, a multi-scale network with WLB and WST is formed. The experimental results demonstrate the superior robustness of our proposed methods compared to state-of-the-art forgery detection techniques.

References

[1] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[2] R. Rombach, A. Blattmann, and et al., "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[3] Z. Wang, H. Zheng, P. He, W. Chen, and M. Zhou, "Diffusion-gan: Training gans with diffusion," in *The Eleventh International Conference on Learning Representations*, 2022.

[4] Y. Jeong, D. Kim, Y. Ro, and J. Choi, "Frepgan: robust deepfake detection using frequency-level perturbations," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 1, 2022, pp. 1060–1068.

[5] H. Zhao, W. Zhou, and et al., "Multi-attentional deepfake detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 2185–2194.

[6] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 14 923–14 932.

[7] Y. Qian, G. Yin, and et al., "Thinking in frequency: Face forgery detection by mining frequency-aware clues," in *European conference on computer vision*. Springer, 2020, pp. 86–103.

[8] C. Zhao, C. Wang, and et al., "Istvt: interpretable spatial-temporal video transformer for deepfake detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1335–1348, 2023.

[9] S.-Y. Wang, O. Wang, and et al., "Cnn-generated images are surprisingly easy to spot... for now," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 8695–8704.

[10] A. Rossler, D. Cozzolino, and et al., "Faceforensics++: Learning to detect manipulated facial images," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 1–11.

[11] D. Afchar, V. Nozick, and et al., "Mesonet: a compact facial video forgery detection network," in *2018 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2018, pp. 1–7.

[12] C. Miao, Z. Tan, and et al., "F^2 trans: High-frequency fine-grained transformer for face forgery detection," *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 1039–1051, 2023.

[13] J. Li, H. Xie, and et al., "Frequency-aware discriminative feature learning supervised by single-center loss for face forgery detection," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 6458–6467.

[14] N. Yu, L. S. Davis, and M. Fritz, "Attributing fake images to gans: Learning and analyzing gan fingerprints," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 7556–7566.

[15] J. Frank, T. Eisenhofer, and et al., "Leveraging frequency analysis for deep fake image recognition," in *International conference on machine learning*. PMLR, 2020, pp. 3247–3258.

[16] H. Phung, Q. Dao, and A. Tran, "Wavelet diffusion models are fast and scalable image generators," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 199–10 208.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[19] M. X. B. Rodriguez, A. Gruson, and et al., "Deep adaptive wavelet network," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 3111–3119.

[20] R. Claypoole, G. Davis, W. Sweldens, and R. Baraniuk, "Nonlinear wavelet transforms for image coding via lifting," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1449–1459, 2003.

[21] Z. Liu, Y. Lin, and et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.

[22] L. Chai, D. Bau, and et al., "What makes fake images detectable? understanding properties that generalize," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*. Springer, 2020, pp. 103–120.

[23] X. Deng, B. Zhao, Z. Guan, and M. Xu, "New finding and unified framework for fake image detection," *IEEE Signal Processing Letters*, vol. 30, pp. 90–94, 2023.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.