FreqDebias: Towards Generalizable Deepfake Detection via Consistency-Driven Frequency Debiasing

Supplementary Material

1. Overview

This supplementary material provides detailed technical information, experimental analyses, and visualizations that support the main paper. Section 2 explains the Forgery Mixup (Fo-Mixup) augmentation, outlining Fourier transformation steps and the integration of dominant frequency components designed to improve generalization in forgery detection. Section 3 introduces attention consistency regularization, describing how instance normalization applied to highlighted class activation maps makes our model's attention consistent. Section 4 provides additional experimental evaluations to demonstrate the robustness, adaptability, and generalization of our proposed method. Section 5 presents an ablation study assessing the influence of different components, such as the number of binary masks and the perturbation parameter, on model performance. Section 6 provides visualizations, including spectral analyses and frequency heatmaps, illustrating the dominant frequency components identified by Fo-Mixup across various forgery types. Finally, Section 7 derives the Kullback-Leibler (KL) divergence between von Mises-Fisher (vMF) distributions for our hyperspherical consistency regularization.

2. Forgery Mixup Augmentation

In the Fo-Mixup augmentation, we linearly interpolate the dominant frequency components in the amplitude spectrum of two different forgery images with each other. Given an input forgery image x with $M \times N$ resolution, we first apply the Fourier transformation $\mathcal{F}(x)$ to different RGB channels to compute its amplitude $\mathcal{A}(x)$ and phase $\mathcal{P}(x)$ components [13]. The Fourier transformation for an input image x is computed as follows:

$$\mathcal{F}(x)(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n)\, e^{-2\pi i \left(\frac{mu}{M} + \frac{nv}{N}\right)}, \quad (1)$$

where u and v represent coordinates in frequency domain. From the computed Fourier transformation $\mathcal{F}(x)$, the amplitude and phase components can be extracted. The amplitude $\mathcal{A}(x)$ is calculated as follows:

$$\mathcal{A}(x)(u,v) = \sqrt{R^2(x)(u,v) + I^2(x)(u,v)}, \quad (2)$$

$$\mathcal{P}(x)(u,v) = arctan\left(\frac{I(x)(u,v)}{R(x)(u,v)}\right), \quad (3)$$

where $R(x)$ and $I(x)$ represent the real and imaginary components of the Fourier transformation $\mathcal{F}(x)$, respectively. To compute $R(x)$ and $I(x)$ components in RGB images, we apply Eqs. 1, 2, and 3 to each channel separately. Then, Fo-Mixup divides the amplitude spectrum into angular segments and identifies the indices of the key clusters to generate the top t binary masks. The k cluster indices are then employed to construct k binary masks, which are subsequently used to filter the input image into k distinct filtered images. The selected masks serve as templates that guide the interpolation process between the amplitude spectra of the input and random forgery images.

By blending these components, Fo-Mixup synthesizes new forgery images with varied frequency bands, targeting the over-reliance of our detector on dominant frequency components by providing exposure to a broader range of frequency spectra.

3. Attention Consistency Regularization

In the attention consistency regularization, we normalize the highlighted class activation map at the last stage of our backbone network using instance normalization [18] to achieve a standard distribution. Instance normalization is constructed from a standardization operation and an affine transformation operation. Let $M^{high} \in \mathbb{R}^{B\times N\times H\times W}$ represent the highlighted region, where B, N, H, and W denote the batch size, class number, height, and width of the input highlighted class activation map. First, we compute the instance-specific mean and standard deviation for the input M^{high} as follows:

$$\mu_{n,b} = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} M^{high}_{n,b,h,w}, \quad (4)$$

Model	CDFv2	DFDC
MAT [21]	99.9	90.3
RECCE [3]	99.9	91.3
SFDG [19]	99.9	94.4
FreqDebias (Ours)	99.6	**94.7**

Table 1. In-domain evaluation. We utilize the CDFv2 [11] and DFDC [1] datasets for training and testing. The results are based on the **video-level** AUC metric.

$$\sigma_{n,b} = \sqrt{\frac{1}{HW}\sum_{h=1}^{H}\sum_{w=1}^{W}\left(M^{high}_{n,b,h,w} - \mu_{n,b}\right)^2 + \varepsilon}, \quad (5)$$

where $M^{high}_{n,b,h,w}$ denotes the element of M^{high} at height h and width w for the b-th sample and n-th class. Also, ε is a small bias. Using the mean and standard deviation for each channel in M^{high}, we employ instance normalization as follows:

$$M^{cn} = \gamma_{n,b}\left(\frac{M^{high} - \mu_{n,b}}{\sigma_{n,b}}\right) + \beta_{n,b}, \quad (6)$$

where $\gamma_{n,b}$ and $\beta_{n,b}$ are learnable affine parameters. Also, M^{cn} denotes the final class activation maps, which are class-wise normalized (denoted by *cn*) for the attention consistency regularization.

4. Additional Experimental Evaluations

This section provides additional experimental evaluations to further demonstrate the robustness, adaptability, and generalizability of the proposed FreqDebias framework.

In-Domain Evaluations using Video-level AUC. We conduct comprehensive evaluations on the CDFv2 [11] and DFDC [1] datasets using video-level AUC. Results shown in Table 1 indicate that the FreqDebias framework achieves superior in-domain performance compared to different state-of-the-art studies, demonstrating its effectiveness across datasets with diverse forgeries.

Cross-Domain Evaluation using Video-level AUC. We further extend cross-domain evaluations by assessing our method on CDFv2 and DFDC datasets using the FF++ (HQ) training dataset and video-level AUC metric. Table 2 highlights the clear advantage of our method in generalizing to unseen forgeries compared to state-of-the-art video-based deepfake detection methods. This improvement stems from the Fo-Mixup augmentation, which effectively diversifies the frequency spectrum of training samples, and the dual consistency regularization, which mitigates spectral bias by preventing over-reliance on dominant frequency components.

Cross-Domain Evaluation with Low-Quality Training. To assess the generalization of the proposed FreqDebias framework to lower-quality data, we train our detector on the low-quality subset of the FF++ dataset and evaluate it on CDFv2 using video-level AUC. As shown in Table 3, our framework significantly outperforms existing methods despite the degraded training quality. This demonstrates the effectiveness of our FreqDebias framework in enhancing the robustness against quality degradation and its capacity to generalize to more challenging conditions.

Model	CDFv2	DFDC
SeeABLE [10]	87.3	75.9
Style Latent Flows [5]	89.0	-
NACO [20]	89.5	76.7
FreqDebias (Ours)	**89.6**	**77.8**

Table 2. Cross-domain evaluation. We utilize the FF++ (HQ) dataset [14] for training and the CDFv2 [11] and DFDC [1] datasets for testing. The results are based on the **video-level** AUC metric.

Model	FF++	CDFv2
MAT [21]	96.4	72.5
SPSL [12]	96.9	76.9
IID [8]	96.8	82.0
FreqDebias (Ours)	**97.0**	**87.5**

Table 3. Cross-domain evaluation with training on the low-quality subset of FF++ and testing on the CDFv2 [11] test set. Results are based on the **video-level** AUC.

Backbone	DFDCP
Xception [6]	82.7
EfficientNet [15]	83.2
ResNet-34 [7]	**82.4**

Table 4. Cross-domain evaluation with different backbones. We utilize the FF++ (HQ) dataset [14] for training and the DFDCP [1] dataset for testing. The results are based on the **frame-level** AUC metric.

Evaluation with different Backbones. We conduct additional experiments using alternative backbones widely adopted in forgery detection literature, namely Xception [6] and EfficientNet [15], and evaluated using frame-level AUC on DFDCP. As detailed in Table 4, FreqDebias performs consistently well across different backbones, demonstrating its effectiveness independent of backbone choice. The results underscore the flexibility of our method and its potential for integration into existing forgery detection frameworks without backbone-specific tuning.

Fo-Mixup Effectiveness. To validate the effectiveness of Fo-Mixup, we integrate it with the state-of-the-art IID

Method	DFDCP
IID [8]	76.2
Fo-Mixup + IID [8]	**78.6**

Table 5. Cross-Domain effectiveness of Fo-Mixup augmentation. We utilize the FF++ (HQ) dataset [14] for training and the DFDCP [1] dataset for testing. The results are based on the **frame-level** AUC metric.

method [8] and evaluate its impact on cross-domain forgery detection. As reported in Table 5, the integration of Fo-Mixup leads to a notable improvement in cross-domain performance compared to IID alone, highlighting its effectiveness in enhancing feature diversity and reducing spectral bias. This demonstrates that Fo-Mixup is a complementary augmentation strategy that can improve the generalization capability of existing deepfake detection frameworks beyond our proposed method.

5. Ablation Study

In this section, we conduct ablation studies following a cross-domain evaluation setting, as in [4, 12]. For these experiments, we use the FF++ (HQ) [14] dataset as the training set and the CDFv2 [11] dataset as the test set, aiming to analyze the effectiveness of various components and configurations within the Fo-Mixup augmentation and our FreqDebias framework.

Effectiveness of the Number of Binary Masks. In this experiment, we investigate the impact of the number of binary masks (denoted by k) and the selection of the top t masks generated by our Fo-Mixup augmentation on the generalization performance of face forgery detection. Initially, we set the number of masked areas to 8 and conduct ablation experiments with t values of 1, 2, 3, and 4. As reported in Table 6, setting t to 3 yields better generalization performance compared to others. In another experiment, we set $t = 3$ and ablate the maximum number of masked areas. It is observed that compared to 8 binary masks ($k = 8$), setting k to 12 can achieve partially better results. However, it is noteworthy that in our experiments, when k exceeds eight ($k > 8$), the improvement in generalization performance becomes marginal, and this improvement is accompanied by a noticeable increase in computational complexity.

Effectiveness of Perturbation. We examine the impact of the perturbation factor $p_{\mathcal{A}}$ in the Fo-Mixup augmentation by deactivating it in our framework. The results, summarized in Table 6 under the FreqDebias-6 experiment, indicate a decline in generalization performance when $p_{\mathcal{A}}$ is removed. This finding highlights the importance of the perturbation factor in enhancing the generalization of our model to unseen forgeries, as it helps the FreqDebias framework resist overfitting to specific frequency distributions.

Model	Components				Fo-Mixup			AUC
	DA	L_{att}	L_{sphere}	CS	t	k	$p_{\mathcal{A}}$	
FreqDebias-1	✓	✓	✓	✓	1	8	✓	80.4
FreqDebias-2	✓	✓	✓	✓	2	8	✓	82.5
FreqDebias-3	✓	✓	✓	✓	3	8	✓	83.6
FreqDebias-4	✓	✓	✓	✓	4	8	✓	82.2
FreqDebias-5	✓	✓	✓	✓	3	12	✓	**83.7**
FreqDebias-6	✓	✓	✓	✓	3	8		80.8

Table 6. Ablation study investigating the impact of the number of binary masks (k), the selection of the top t masks, and the perturbation parameter $p_{\mathcal{A}}$ on generalization performance. All components of our proposed framework are retained in these experiments. CS refers to the confidence sampling strategy, and DA denotes standard data augmentation. We utilize the FF++ (HQ) dataset [14] for training and the CDFv2 dataset [11] for testing. The results are based on the **frame-level** AUC metric.

Effectiveness of the Hyperparameters in Overall Loss Function. We conduct an ablation study to thoroughly investigate the impact of hyperparameters η (for L_{CAM} loss), δ (for L_{att} loss), and ρ (for L_{sphere} loss) in the overall loss function. The primary aim of our loss function is to ensure our network makes consistent predictions for a diverse range of forgery samples. The ablation study, detailed in Table 7, explores various combinations of η, δ, and ρ to identify the optimal settings. We observe that tuning these hyperparameters significantly impacts the ability of our proposed FreqDebias framework to maintain consistency across diverse forgery domains, ultimately leading to improved generalization performance. By finding the right balance with η, δ, and ρ, our forgery detector demonstrates robustness against forgery perturbations and out-of-distribution forgery samples, thereby enhancing its effectiveness in deepfake detection.

Effectiveness of Hyperspherical Consistency Regularization. In this experiment, we investigate the efficacy of hyperspherical consistency regularization compared to L_2 consistency regularization within our FreqDebias framework. Specifically, we substitute our hyperspherical consistency regularization with L_2 regularization between $\boldsymbol{F}^s_{cat}$ and $\boldsymbol{F}^t_{cat}$ while maintaining the remaining components. As indicated in Table 8, when comparing our FreqDebias-12 framework with the FreqDebias-11 model, we observe a performance decline of approximately 1.7% upon the substitution of hyperspherical consistency regularization with L_2 regularization in our FreqDebias framework. This result underscores the superiority of the vMF distributions in capturing a global view of the embedding space for geometric consistent learning using hyperspherical consistency regularization compared to L_2 consistency regularization.

Model	η	δ	ρ	μ	AUC
FreqDebias-7	0.1	0.1	0.1	1	83.4
FreqDebias-8	0.5	0.1	0.1	1	**83.6**
FreqDebias-9	0.1	0.5	0.1	1	83.2
FreqDebias-10	0.1	0.1	0.5	1	82.1

Table 7. Ablation study on the hyperparameters used the overall loss function. Here, η, δ, ρ, and μ denote the weight hyperparameters for the L_{CAM}, L_{att}, L_{sphere}, and L_{cls_sphere} loss terms, respectively. We utilize the FF++ (HQ) dataset [14] for training and the CDFv2 dataset [11] for testing. The results are based on the **frame-level** AUC metric.

Model	DA	L_{att}	L_{sphere}	L_2	Fo-Mixup	CS	AUC
FreqDebias-11	✓	✓		✓	✓	✓	81.9
FreqDebias-12	✓	✓	✓		✓	✓	**83.6**

Table 8. Performance comparisons with L_2 consistency regularization. In FreqDebias-11, L_{cls_sphere} is retained, as it is used in conjunction with L_{sphere}. CS and DA denote the confidence sampling strategy and standard data augmentation, respectively. We utilize the FF++ (HQ) dataset [14] for training and the CDFv2 dataset [11] for testing. The results are based on the **frame-level** AUC metric.

6. Spectral Analysis and Visualizations

Fig. 1 depicts the heatmaps of the average dominant frequency components identified by the Fo-Mixup method when using a standard face forgery detector on 30,000 forgery images from the FF++ [14] dataset. For this illustration, we select 10,000 forgery images for each forgery type, including DeepFake [2], Face2Face [16], and NeuralTextures [17]. To ensure a fair comparison across different face forgery types, we choose forgery images with similar identities from the FF++ [14] dataset. The ranking of dominant frequency components in Fig. 1 provides a structured analysis of the spectral importance for forgery detection. Specifically, the first rank ($t = 1$) corresponds to the top dominant frequency components, the second rank ($t = 2$) represents the second-top dominant frequency components, and the third rank ($t = 3$) identifies the third-top dominant Frequency components. This ranking highlights how the vanilla deepfake detector over-relies on dominant frequency components, with different forgery types emphasizing distinct frequency bands.

In the first column of Fig. 1, the heatmaps reveal that, in DeepFake forgery images, the dominant frequency components with the first rank ($t = 1$) are concentrated in very low-frequency bands, highlighting their crucial role in DeepFake forgery detection. As we progress to the dominant frequency components with the second rank ($t = 2$), the low-frequency bands expand, and by the third rank ($t = 3$), even some mid to high-frequency bands become crucial. In relation to Face2Face forgery, unlike DeepFake

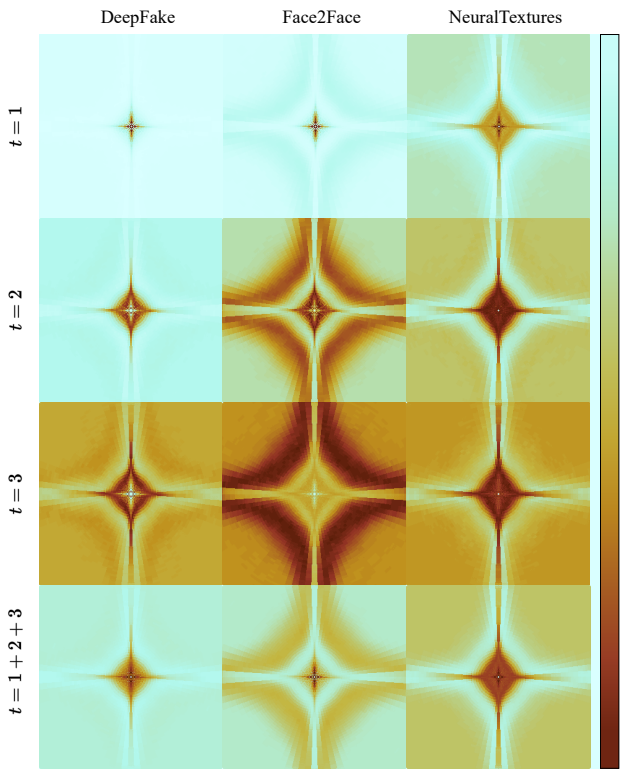

Figure 1. Heatmap of dominant frequency components generated by the top t masks from the perspective of a standard face forgery detector. For this illustration, we select a total of 30,000 forgery images, with 10,000 images chosen for each of the three forgery types in the FF++ dataset [14]: DeepFake [2], Face2Face [16], and NeuralTextures [17].

or NeuralTextures forgery images, mid-frequency bands are overly relied upon by the vanilla deepfake detector for Face2Face forgery detection, as demonstrated in the second rank ($t = 2$). In NeuralTextures forgery images, the first rank ($t = 1$) exhibits a broad range of low-frequency bands compared to DeepFake and Face2Face forgery images. In addition, the dominant frequency components within the first rank ($t = 1$) extend into higher frequency bands. This characteristic is not observed in Face2Face and NeuralTextures forgery images, where high-frequency bands are absent in the first rank ($t = 1$).

In Fig. 2, we present forgery samples of various types, along with the selected masks for the top three ranks and the corresponding filtered images using the Fo-Mixup masks. For DeepFake forgery images, the dominant frequency components at the first rank ($t = 1$) are concentrated in very low frequencies, evident in the corresponding $\mathcal{B}_1$ masks. Consequently, when the amplitude spectrum of the input forgery image is filtered with the $\mathcal{B}_1$ mask, high-frequency information is retained in the input forgery image (first row). In NeuralTextures forgery images, the domi-

Figure 2. Illustration of the top three selected masks ($t = 1, 2, 3$) and the corresponding filtered images using the proposed Fo-Mixup augmentation on the three forgery types in the FF++ dataset [14]: DeepFake [2], Face2Face [16], and NeuralTextures [17].

nant frequency components include high-frequency bands, resulting in filtered images that appear visually similar to the input forgery image from a human perspective. These observations highlight the capability of Fo-Mixup to effectively modulate dominant frequency components tailored to each input forgery. This mitigates over-reliance on certain frequency characteristics in training forgeries, referred to as spectral bias, and enhances the ability of our detector to generalize across a wide range of previously unseen forgery types, thereby improving detection robustness.

7. Computing KL Divergence for Distribution Matching Score

The Distribution Matching Score (DMS) relies on the KL divergence between two vMF distributions, corresponding to the synthesized (s) and training (t) domains. For two probability distributions $p(x)$ and $q(x)$, the KL divergence is defined as:

$$KL(p\|q) = \int p(x) \log\left(\frac{p(x)}{q(x)}\right) dx. \tag{7}$$

The KL divergence quantifies the difference between $p(x)$ and $q(x)$, measuring how one distribution diverges from the other. Consider two vMF distributions representing the synthesized (s) and training (t) domains. Let $\tilde{\boldsymbol{F}}_{cat} \in \mathbb{S}^{d-1}$ denote the d-dimensional unit vector representing the normalized facial feature representation. The probability density functions (PDFs) for the synthesized and training vMF distributions are given by:

$$p\left(\tilde{\boldsymbol{F}}_{cat} \mid \kappa_s, \tilde{\boldsymbol{\mu}}_s\right) = \mathcal{C}_d(\kappa_s) \exp\left(\kappa_s \tilde{\boldsymbol{F}}_{cat} \tilde{\boldsymbol{\mu}}_s^\top\right), \tag{8}$$

$$q\left(\tilde{\boldsymbol{F}}_{cat} \mid \kappa_t, \tilde{\boldsymbol{\mu}}_t\right) = \mathcal{C}_d(\kappa_t) \exp\left(\kappa_t \tilde{\boldsymbol{F}}_{cat} \tilde{\boldsymbol{\mu}}_t^\top\right), \tag{9}$$

where $\kappa_s, \kappa_t \in \mathbb{R}_{\geq 0}$ are the concentration parameters for the synthesized and training distributions, and $\tilde{\boldsymbol{\mu}}_s, \tilde{\boldsymbol{\mu}}_t \in \mathbb{S}^{d-1}$ are the orientation vectors. The normalization constant $\mathcal{C}_d(\kappa)$ ensures that the PDFs integrate to 1 over the unit hypersphere and is expressed as below:

$$\mathcal{C}_d(\kappa) = \frac{\kappa^{\frac{d}{2}-1}}{(2\pi)^{\frac{d}{2}} \cdot I_{\frac{d}{2}-1}(\kappa)}, \tag{10}$$

where $I_\beta(\kappa)$ is the modified Bessel function [9] of the first kind at order β.

To compute the KL divergence, we substitute Eqs. 8 and 9 into Eq. 7 as below:

$$KL(p\|q) = \int p\left(\tilde{\boldsymbol{F}}_{cat} \mid \kappa_s, \tilde{\boldsymbol{\mu}}_s\right) \log\left(\frac{\mathcal{C}_d(\kappa_s) \exp\left(\kappa_s \tilde{\boldsymbol{F}}_{cat} \tilde{\boldsymbol{\mu}}_s^\top\right)}{\mathcal{C}_d(\kappa_t) \exp\left(\kappa_t \tilde{\boldsymbol{F}}_{cat} \tilde{\boldsymbol{\mu}}_t^\top\right)}\right) d\tilde{\boldsymbol{F}}_{cat}. \tag{11}$$

Breaking down the integral in Eq. 11, we separate the terms as follows:

$$\begin{aligned} KL(p\|q) &= \log\left(\frac{\mathcal{C}_d(\kappa_s)}{\mathcal{C}_d(\kappa_t)}\right) \underbrace{\int p\left(\tilde{\boldsymbol{F}}_{cat} \mid \kappa_s, \tilde{\boldsymbol{\mu}}_s\right) d\tilde{\boldsymbol{F}}_{cat}}_{=1} \\ &\quad + \left(\kappa_s \tilde{\boldsymbol{\mu}}_s^\top - \kappa_t \tilde{\boldsymbol{\mu}}_t^\top\right) \underbrace{\int p\left(\tilde{\boldsymbol{F}}_{cat} \mid \kappa_s, \tilde{\boldsymbol{\mu}}_s\right) \tilde{\boldsymbol{F}}_{cat} d\tilde{\boldsymbol{F}}_{cat}}_{=\frac{I_{d/2}(\kappa_s)}{I_{d/2-1}(\kappa_s)} \tilde{\boldsymbol{\mu}}_s}, \end{aligned} \tag{12}$$

where the expectation $\mathbb{E}_p[\tilde{\boldsymbol{F}}_{cat}] = \frac{I_{d/2}(\kappa_s)}{I_{d/2-1}(\kappa_s)} \tilde{\boldsymbol{\mu}}_s$ represents the mean vector of the distribution p. Substituting the expectation into Eq. 12, we obtain the final expression of the KL divergence as follows:

$$\begin{aligned} &KL(p\|q) \\ &= \log\left(\frac{\mathcal{C}_d(\kappa_s)}{\mathcal{C}_d(\kappa_t)}\right) + \left(\kappa_s \tilde{\boldsymbol{\mu}}_s^\top - \kappa_t \tilde{\boldsymbol{\mu}}_t^\top\right) \left(\frac{I_{d/2}(\kappa_s)}{I_{d/2-1}(\kappa_s)} \tilde{\boldsymbol{\mu}}_s\right) \\ &= \log\left(\frac{\mathcal{C}_d(\kappa_s)}{\mathcal{C}_d(\kappa_t)}\right) + \kappa_s \frac{I_{d/2}(\kappa_s)}{I_{d/2-1}(\kappa_s)} - \kappa_t \frac{I_{d/2}(\kappa_s)}{I_{d/2-1}(\kappa_s)} \tilde{\boldsymbol{\mu}}_s \tilde{\boldsymbol{\mu}}_t^\top. \end{aligned} \tag{13}$$

This final expression shows the KL divergence between two vMF distributions in terms of their concentration parameters κ and the alignment of their mean directions $\tilde{\boldsymbol{\mu}}$. To summarize, the KL divergence between the synthesized and training vMF distributions is influenced by: 1) The ratio of their normalization constants $\mathcal{C}_d(\kappa_s)$ and $\mathcal{C}_d(\kappa_t)$. 2) The concentration parameters κ_s and κ_t, modulated by the modified Bessel functions. 3) The cosine of the angle between the mean directions $\tilde{\boldsymbol{\mu}}_s$ and $\tilde{\boldsymbol{\mu}}_t$, as represented by their dot product. A higher cosine similarity between the mean directions ($\tilde{\boldsymbol{\mu}}_s \tilde{\boldsymbol{\mu}}_t^\top$) reduces the divergence, indicating better overlap between the synthesized and training distributions. This also ensures that the DMS effectively measures the similarity between the two domains. In conclusion, these factors collectively determine the divergence between the two distributions, providing a quantitative measure of their overlap.

References

[1] Deepfake detection challenge. https://www.kaggle.com/competitions/deepfake-detection-challenge/data, 2019. 2, 3

[2] Deepfakes github. https://github.com/deepfakes/faceswap, 2019. 4, 5

[3] Junyi Cao, Chao Ma, Taiping Yao, Shen Chen, Shouhong Ding, and Xiaokang Yang. End-to-end reconstruction-classification learning for face forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4113–4122, 2022. 2

[4] Liang Chen, Yong Zhang, Yibing Song, Lingqiao Liu, and Jue Wang. Self-supervised learning of adversarial example: Towards good generalizations for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18710–18719, 2022. 3

[5] Jongwook Choi, Taehoon Kim, Yonghyun Jeong, Seungryul Baek, and Jongwon Choi. Exploiting style latent flows for generalizing deepfake video detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1133–1143, 2024. 2

[6] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1251–1258, 2017. 2

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[8] Baojin Huang, Zhongyuan Wang, Jifan Yang, Jiaxin Ai, Qin Zou, Qian Wang, and Dengpan Ye. Implicit identity driven deepfake face swapping detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2023. 2, 3

[9] John Kent. Some probabilistic properties of bessel functions. *The Annals of Probability*, pages 760–770, 1978. 6

[10] Nicolas Larue, Ngoc-Son Vu, Vitomir Struc, Peter Peer, and Vassilis Christophides. SeeABLE: Soft discrepancies and bounded contrastive learning for exposing deepfakes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21011–21021, 2023. 2

[11] Yuezun Li, Xin Yang, Pu Sun, Honggang Qi, and Siwei Lyu. Celeb-DF: A large-scale challenging dataset for deepfake forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3207–3216, 2020. 2, 3, 4

[12] Honggu Liu, Xiaodan Li, Wenbo Zhou, Yuefeng Chen, Yuan He, Hui Xue, Weiming Zhang, and Nenghai Yu. Spatial-phase shallow learning: rethinking face forgery detection in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 772–781, 2021. 2, 3

[13] Henri J Nussbaumer and Henri J Nussbaumer. *The fast Fourier transform*. Springer, 1982. 1

[14] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Niessner. FaceForensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1–11, 2019. 2, 3, 4, 5

[15] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 2

[16] Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2Face: Real-time face capture and reenactment of RGB videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2387–2395, 2016. 4, 5

[17] Justus Thies, Michael Zollhöfer, and Matthias Nießner. Deferred neural rendering: Image synthesis using neural textures. *ACM Transactions on Graphics*, 38(4):1–12, 2019. 4, 5

[18] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Improved texture networks: Maximizing quality and diversity in feed-forward stylization and texture synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6924–6932, 2017. 1

[19] Yuan Wang, Kun Yu, Chen Chen, Xiyuan Hu, and Silong Peng. Dynamic graph learning with content-guided spatial-frequency relation reasoning for deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7278–7287, 2023. 2

[20] Daichi Zhang, Zihao Xiao, Shikun Li, Fanzhao Lin, Jianmin Li, and Shiming Ge. Learning natural consistency representation for face forgery video detection. In *European Conference on Computer Vision*, pages 407–424, 2024. 2

[21] Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2185–2194, 2021. 2