LFM: A Local Focus Mechanism for Generalized Deepfake Detection

Mingliang Li [1], Hanxi Li [1], Lin Yuanbo Wu [*2], Changhong Liu [1]

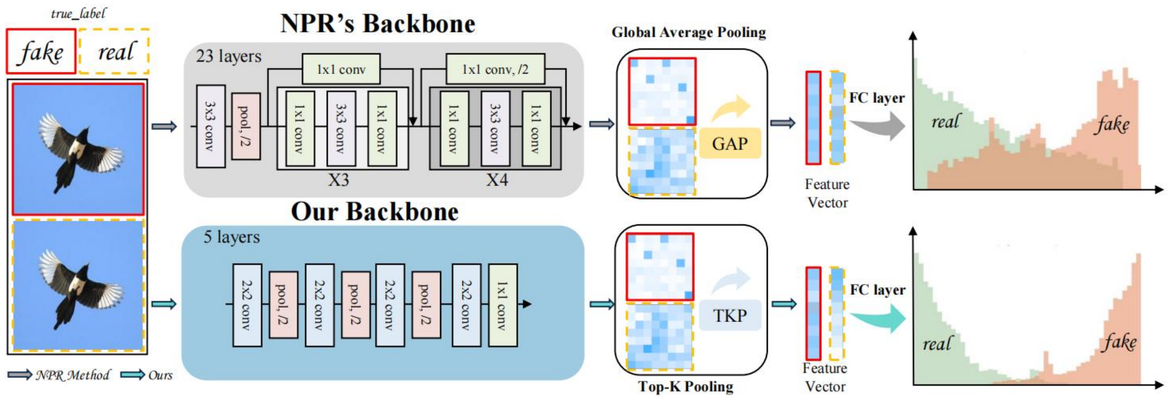
limingliang@jxnu.edu.cn, lihanxi2001@foxmail.com, Xiaoxian.wu9188@gmail.com, liuch@jxnu.edu.cn

[1] Jiangxi Normal University, China

[2] University of Warwick, Coventry, CV47AL, United Kingdom

Problem & Motivation

Current deepfake detectors, particularly those based on reconstruction learning, suffer from critical generalization limitations. They overfit to the semantic features of a specific training category (e.g., faces) and fail to transfer to others (e.g., cars). Furthermore, their performance degrades significantly across generation domains (e.g., from GANs to Diffusion Models). A primary cause is the widespread use of Global Average Pooling (GAP), which discards crucial local forgery artifacts by compressing spatial features into a single vector.



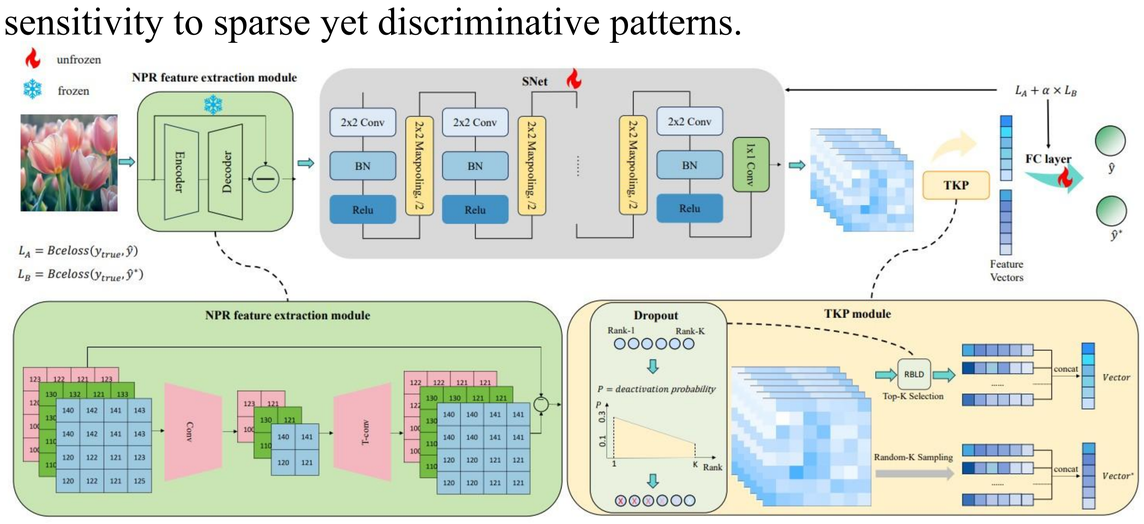
Proposed Method

Our approach introduces a Local Focus Mechanism to enhance generalization in deepfake detection by focusing on discriminative local forgery patterns. The framework consists of three core components:

Neighboring Pixel Relationships Feature Extraction: Input images are processed using a down-up-sampling reconstruction module to extract NPR features, which highlight subtle forgery artifacts by computing the residual.

Salience Network: A lightweight CNN with five convolutional layers and three max-pooling layers processes the absolute NPR values to generate salience maps, preserving critical local forgery cues while minimizing semantic overfitting.

Top-K Pooling with Regularization: Replaces Global Average Pooling by selecting the top-K most salient local features per channel, enhancing sensitivity to sparse yet discriminative patterns.



Regularization

Rank-Based Linear Dropout: Applies linearly increasing dropout probabilities to top-ranked activations to prevent over-reliance on dominant features.

Random-K Sampling: Introduces stochasticity by randomly sampling K features during training, improving robustness.

Algorithm 1 TKP with RBLD and RKS

Require: Feature maps **Maps** $\in \mathbb{R}^{H' \times W' \times 64}$, k, $p_{\min} = 0.1$, $p_{\max} = 0.3$.
Ensure: Feature **Vecter** $\in \mathbb{R}^{64k}$, **Vector*** $\in \mathbb{R}^{64k}$.

Initialize **Vector** $\leftarrow \mathbf{0}^{64\times k}$, **Buffer** $\leftarrow [\,]$ ▷ Initialisation
for $c \leftarrow 1$ to 64 **do** ▷ Process each channel
  Extract feature map: $\mathbf{M}_c \leftarrow$ **Maps**$[c,:,:] \in \mathbb{R}^{H' \times W'}$
  Sort values: $\{v_{(1)}, \ldots, v_{(H'*W')}\} \leftarrow \text{sort}(\mathbf{M}_c)$ ▷ Ascending order
  Select top-k: $\mathcal{T}_k \leftarrow \{v_{(H'*W'-k+1)}, \ldots, v_{(H'*W')}\}$
  for $i \leftarrow 1$ to k **do** ▷ Rank-based linear dropout
    $p_i \leftarrow p_{\min} + (p_{\max} - p_{\min}) \times \frac{i-1}{k-1}$ ▷ Linear prob by rank index
    $r \sim \mathcal{U}(0,1)$
    if $r \leq p_i$ **then**
      $v_{(H'*W'-k+i)} \leftarrow 0$ ▷ Apply dropout
    end if
  end for
  Vecter$[c,:] \leftarrow \mathcal{T}_k$ ▷ Store processed vector
  $\mathcal{R}_k \leftarrow \text{RandomSample}($**Maps**$[c,:,:], k)$ ▷ Random selection
  $\{v_{(1)}, \ldots, v_{(k)}\} \leftarrow \text{sort}(\mathcal{R}_k)$ ▷ Ascending sort
  Buffer.append$(\{v_{(1)}, \ldots, v_{(k)}\})$ ▷ Store sorted values
end for
Vecter $\leftarrow$ concat(**Vecter**) ▷ Flatten into 64k-dim vector
Vector* $\leftarrow$ concat(**Buffer**) ▷ Explicit concatenation
return Vecter, **Vector***

Code: https://github.com/lmlpy/LFM.git

Experimental Results

Evaluation of Cross-Domain Generalization -- GAN

Method	AttGAN		BEGAN		CGAN		IMGAN		MMDGAN		RelGAN		S3GAN		SNGAN		STGAN		mean	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
CNNDetection [43]	51.1	83.7	50.2	44.9	81.5	97.5	71.1	94.7	72.9	94.4	53.3	82.1	55.2	66.1	62.7	90.4	63.0	92.7	62.3	82.9
Frank [7]	65.0	74.4	39.4	39.9	31.0	36.0	41.1	41.0	38.4	40.5	69.2	96.2	69.7	81.9	48.4	47.9	25.4	34.0	47.5	54.7
Durall [6]	39.9	38.2	48.2	30.9	60.9	67.2	50.1	51.7	59.5	65.5	80.0	88.2	87.3	97.0	54.8	58.9	62.1	72.5	60.3	63.3
Patchfor [2]	68.0	92.9	97.1	100.0	97.8	99.9	93.6	98.2	97.9	100.0	99.6	100.0	66.8	68.1	97.6	99.8	92.7	99.8	90.1	95.4
F3Net [28]	85.2	94.8	87.1	97.5	89.5	99.8	67.1	83.1	73.7	99.6	98.8	100.0	65.4	70.0	51.6	93.6	60.3	99.9	75.4	93.1
SelfBland [38]	63.1	66.1	56.4	59.0	75.1	82.4	79.0	82.5	68.6	74.0	73.6	77.8	53.2	53.9	61.6	65.0	61.2	66.7	65.8	69.7
GANDetection [27]	57.4	75.1	67.9	100.0	67.8	99.7	67.6	92.4	67.7	99.3	60.9	86.2	69.6	83.5	66.7	90.6	69.6	97.2	66.1	91.6
LGrad [39]	68.6	93.8	69.9	89.2	50.3	54.0	71.1	82.0	57.5	67.3	89.1	99.1	78.5	86.0	78.0	87.4	54.8	68.0	68.6	80.8
Ojha [31]	78.5	98.3	72.0	98.9	77.6	99.8	77.6	98.9	77.6	99.7	78.2	98.7	85.2	98.1	77.6	98.7	74.2	97.8	77.6	98.8
NPR [40]	83.0	96.2	99.0	99.8	98.7	99.0	94.5	98.3	98.6	99.0	99.6	100.0	79.0	80.0	88.8	97.4	98.0	100.0	93.2	96.6
LFM (ours)	100.0	100.0	99.9	100.0	98.8	99.9	98.5	99.9	99.4	100.0	100.0	100.0	88.4	95.5	99.2	100.0	99.9	100.0	98.2	99.5

Method	progan		stylegan		stylegan2		biggan		cyclegan		Stargan		guagan		deepfake		mean	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
CNNDetection [12]	91.4	99.4	63.8	91.4	76.4	97.5	52.9	73.3	72.7	88.6	63.8	90.8	63.9	92.2	51.7	62.3	67.1	86.9
Frank [3]	90.3	85.2	74.5	72.0	73.1	71.4	88.7	86.0	75.5	71.2	99.5	99.5	69.2	77.4	60.7	49.1	78.9	76.5
Durall [2]	81.1	74.4	54.4	52.6	66.8	62.0	60.1	56.3	69.0	64.0	98.1	98.1	61.9	57.4	50.2	50.0	67.7	64.4
Patchfor [1]	97.8	100.0	82.6	93.1	83.6	98.5	64.7	69.5	74.5	87.2	100.0	100.0	57.2	55.4	85.0	93.2	80.7	87.1
F3Net [7]	99.4	100.0	92.6	99.7	88.0	99.8	65.3	69.9	76.4	84.3	100.0	100.0	58.1	56.7	63.5	78.8	80.4	86.2
SelfBland [9]	58.8	65.2	50.1	47.7	48.6	47.4	51.1	51.9	59.2	65.3	74.5	89.2	59.2	65.5	93.8	99.3	61.9	66.4
GANDetection [6]	82.7	95.1	74.4	92.9	69.9	87.9	76.3	89.9	85.2	95.5	68.8	99.7	61.4	75.8	60.0	83.9	72.3	90.1
BiHPF [4]	90.7	86.2	76.9	75.1	76.2	74.7	84.9	81.7	81.9	78.9	94.4	94.4	69.5	78.1	54.4	54.6	78.6	77.9
FrePGAN [5]	99.0	99.9	80.7	89.6	84.1	98.6	69.2	71.1	71.1	74.4	99.9	100.0	60.3	71.7	70.9	91.9	79.4	87.2
LGrad [10]	99.9	100.0	94.8	99.9	96.0	99.9	82.9	90.7	85.3	94.0	99.6	100.0	72.4	79.3	58.0	67.9	86.1	91.5
Ojha [8]	99.7	100.0	89.0	98.7	83.9	98.4	90.5	99.1	87.9	99.8	91.4	100.0	89.9	100.0	80.2	90.2	89.1	98.3
NPR [11]	99.8	100.0	96.3	99.8	97.3	100.0	87.5	94.5	95.0	99.5	99.7	100.0	86.6	88.8	77.4	86.2	92.5	96.1
LFM (Ours)	99.8	100.0	99.9	100.0	99.9	100.0	91.2	97.6	94.3	99.6	100.0	100.0	88.3	93.0	90.8	95.8	95.5	98.3

Evaluation of Cross-Domain Generalization -- Diffusion Model

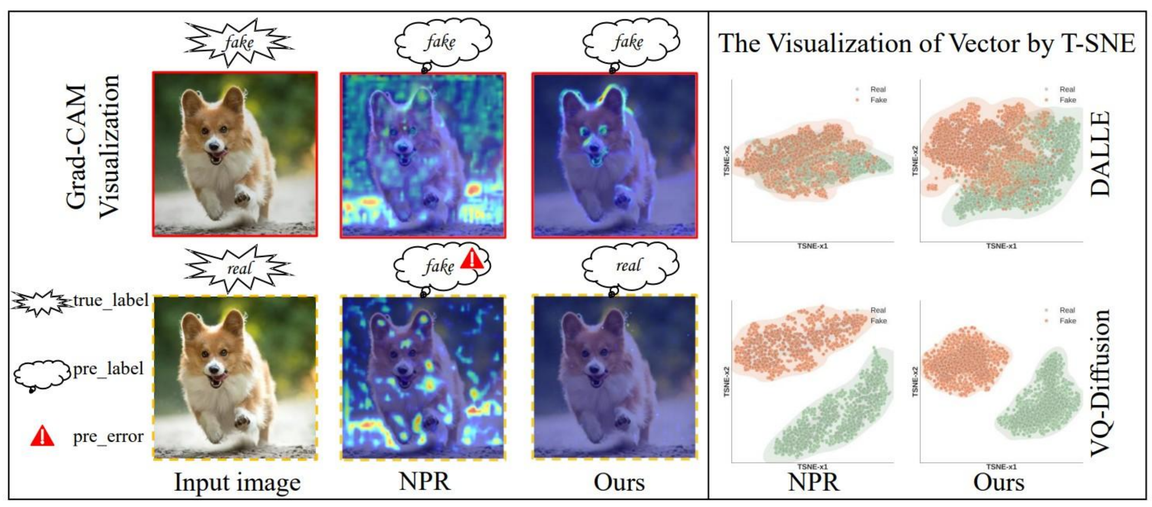
Method	ADM		DDPM		IDDPM		LDM		PNDM		VQ-Diffusion		SD-v1		SD-v2		mean	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
CNNDetection [12]	53.9	71.8	62.7	76.6	50.2	82.7	50.4	78.7	50.8	90.3	50.0	71.0	38.0	76.7	52.0	90.3	51.0	79.8
Frank [3]	58.9	65.9	37.0	27.6	51.4	65.0	51.7	48.5	44.0	38.2	51.7	66.7	32.8	52.3	40.8	37.5	46.0	50.2
Durall [2]	39.8	42.1	52.9	49.8	55.3	56.7	43.1	39.9	44.5	47.3	38.6	38.3	39.5	56.3	62.1	55.8	47.0	48.3
Patchfor [1]	77.5	93.9	62.3	97.1	50.0	91.6	99.5	100.0	50.2	99.9	100.0	100.0	90.7	99.8	94.8	100.0	78.1	97.8
F3Net [7]	80.9	96.9	84.7	99.4	74.7	98.9	100.0	100.0	72.8	99.5	100.0	100.0	73.4	97.2	99.8	100.0	85.8	99.0
SelfBland [9]	57.0	59.0	61.9	49.6	63.2	66.9	83.3	92.2	48.2	48.2	77.2	82.7	46.2	68.0	71.2	73.9	63.5	67.6
GANDetection [6]	51.1	53.1	62.3	46.4	50.2	63.0	51.6	48.1	50.6	79.0	51.1	51.2	39.8	65.6	50.1	36.9	50.8	55.4
LGrad [10]	86.4	97.5	99.9	100.0	66.1	92.8	99.7	100.0	69.5	98.5	96.2	100.0	90.4	99.4	97.1	100.0	88.2	98.5
Ojha [8]	78.4	92.1	72.9	78.8	75.0	92.8	82.2	97.1	75.3	92.5	83.5	97.7	56.4	90.4	71.5	92.4	74.4	91.7
NPR [11]	88.6	98.9	99.8	100.0	91.8	99.8	100.0	100.0	91.2	100.0	100.0	100.0	97.4	99.8	93.8	100.0	95.3	99.8
LFM (ours)	96.5	99.8	99.9	100.0	97.9	100.0	100.0	100.0	99.4	100.0	100.0	100.0	96.2	99.9	99.8	100.0	98.7	100.0

Method	DALLE		G_100_10		G_100_27		G_50_27		ADM		L_100		L_200		L_200_cfg		mean	
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP
CNNDetection [12]	51.8	61.3	53.3	72.9	53.0	71.3	54.2	76.0	54.9	66.6	51.9	63.7	52.0	64.5	51.6	63.1	52.8	67.4
Frank [3]	57.0	62.5	53.6	44.3	50.4	40.8	52.0	42.3	53.4	52.5	56.6	51.3	56.4	50.9	56.5	52.1	54.5	49.6
Durall [2]	55.9	58.0	54.9	52.3	48.9	46.9	51.7	49.9	40.6	42.3	62.0	62.6	61.7	61.7	58.4	58.5	54.3	54.0
Patchfor [1]	79.8	99.1	87.3	99.7	82.8	99.1	84.9	98.8	74.2	81.4	95.8	99.8	95.6	99.9	94.0	99.8	86.8	97.2
F3Net [7]	71.6	79.9	88.3	95.4	87.0	94.5	88.5	95.4	69.2	70.8	74.1	84.0	73.4	83.3	80.7	89.1	79.1	86.5
SelfBland [9]	52.4	51.6	58.8	63.2	59.4	64.1	64.2	68.3	58.3	63.4	53.0	54.0	52.6	51.9	51.9	52.6	56.3	58.7
GANDetection [6]	67.2	83.0	51.2	52.6	51.1	51.9	51.7	53.5	49.6	49.0	54.7	65.8	54.9	65.9	53.8	58.9	54.3	60.1
LGrad [10]	88.5	97.3	89.4	94.9	87.4	93.2	90.7	95.1	86.6	100.0	94.8	99.2	94.2	99.1	95.9	99.2	90.9	97.2
Ojha [8]	89.5	96.8	90.1	97.0	90.7	97.2	91.1	97.4	75.7	85.1	90.5	97.0	90.2	97.1	77.3	88.6	86.9	94.5
NPR [11]	94.5	99.5	98.2	99.8	97.8	99.7	98.2	99.8	75.8	81.0	99.3	99.9	99.1	99.9	99.0	99.8	95.2	97.4
LFM (ours)	78.6	94.7	99.8	100.0	99.5	100.0	99.7	100.0	86.9	92.4	99.7	100.0	99.7	100.0	99.5	99.9	95.4	98.4

Comparison with State-of-the-art Methods

Method	Test Set1 [43]		Test Set2 [40]		Test Set3 [44]		Test Set4 [31]		Test Set5 [40]		Mean		FLOPs	Params
	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP	ACC	AP		
LGrad [39]	86.1	91.5	68.6	80.8	88.2	98.5	90.9	97.2	74.3	64.2	80.5	88.5	50.95B	48.61M
Ojha [31]	89.1	98.3	77.6	98.8	74.1	91.7	86.6	94.5	66.4	73.9	79.8	91.4	51.90B	202.05M
NPR [40]	92.5	96.1	93.2	96.6	95.3	99.8	95.2	97.4	80.1	84.8	92.2	95.8	2.30B	1.44M
Fatformer [24]	98.3	99.7	98.9	100.0	93.8	98.5	72.5	82.2	66.8	85.4	87.8	93.7	577.25B	577.25M
SAFE [19]	96.2	98.8	99.0	99.8	95.7	99.1	91.2	99.0	81.9	91.8	93.8	98.2	2.30B	1.44M
LFM (ours)	95.5	98.3	98.2	99.5	98.7	100.0	95.4	98.4	88.9	95.8	95.9	98.6	1.61B	0.72M

Visualization



To qualitatively validate the efficacy of our Local Focus Mechanism, we employed two visualization techniques: Grad-CAM and T-SNE.

Localization with Grad-CAM: Grad-CAM visualizations confirm LFM's precise focus on localized forgery artifacts versus the NPR's attention to broader, less discriminative regions.

Feature Separability with T-SNE: T-SNE projections show TKP enables superior feature separability with enhanced intra-class compactness and inter-class separation across multiple datasets, validating its generalization strength.