Test-Time Adaptation for Detecting Image Inpainting Forgeries

Long Sun, Guopu Zhu, *Senior Member, IEEE*, Hongli Zhang, Xinpeng Zhang, *Senior Member, IEEE*, Yicong Zhou, *Senior Member, IEEE*, and Ligang Wu, *Fellow, IEEE*

***Abstract*—The rapid development of deep learning-based image inpainting poses serious challenges to image authenticity. As inpainting methods continue to evolve, the inpainted images exhibit extremely high visual fidelity, presenting recognition difficulties to the forgery detection model due to differences in operational mode and forgery traces among methods. In particular, the detection performance tends to drop significantly in the testing phase when the test samples differ from the training data. To address this issue, we propose a test-time adaptive detection framework for image inpainting forgeries. First, we propose an image gradient-based metric that quantifies model uncertainty and orchestrates the entire adaptation process. Integrating this metric with sample-specific batch normalization (BN) statistics enhances the ability of pretrained models in the inference stage. Second, we introduce a cross-attention module as a side-tuning module, enabling the model to adapt dynamically to reliable test samples without altering the backbone network. To validate the effectiveness of the proposed method, we construct a dataset comprising synthetic images of multiple inpainting methods and design experiments under two scenarios of distributional bias. The results demonstrate that our proposed framework outperforms the existing baseline method, enhancing the adaptability and detection performance of the forgery detection model in dynamic environments.**

***Index Terms*—Batch normalization, forgery detection, image inpainting, side tuning, test-time adaptation.**

I. Introduction

IMAGE inpainting [1] refers to the technique used to restore damaged images, fill missing areas, or remove unwanted elements, ensuring visual coherence and esthetic integrity. As deep learning advances, inpainting methods have matured and can produce images as vivid and realistic as authentic ones. Furthermore, the growing accessibility of inpainting tools—along with the frequent release of new models and variants—enables users to alter critical visual information effortlessly. In real-world scenarios, this rapid iteration of inpainting methods poses a significant and evolving threat to static image forgery detectors trained on fixed distributions. Crucially, the inpainted images encountered during testing often exhibit substantial distributional shifts from the training data, reflecting differences in image sources, manipulation algorithms, and tampered regions. The test-time environment itself may also change dynamically over time. As a result, there is an urgent need for detection models that can adaptively adjust at inference time to remain robust against emerging and previously unseen inpainting forgeries.

Current inpainting forgery detection methods primarily focus on designing neural networks to identify inpainting traces in forged images. For example, Zhang et al. [2] developed artifact enhancement modules to detect inpainting forgeries in the frequency domain. Wu and Zhou [3] designed the methods to adapt neural network architectures to various inpainting methods. Additional architectural enhancements include the integration of advanced components, such as feature pyramids [4] or self-attention mechanisms [5]. Although these methods are effective, their performance can degrade substantially when encountering images generated with inpainting methods, sources, or regions not seen during training, as shown in Fig. 1(a). This inability to adapt to the test-time distribution shift is precisely the challenge addressed by continual test-time adaptation (CTTA) [6].

CTTA has emerged as a promising paradigm for improving model robustness against unforeseen distribution shifts during deployment. However, most existing CTTA methods are designed for general-purpose tasks, such as image classification or semantic segmentation, and may be suboptimal for inpainting forgery detection. For example, EATA [7] updates the model by focusing on samples with higher class entropy. However, in inpainting forgery tasks, the model outputs are masks, which are difficult to leverage for adaptation. In addition, pseudo-labeling based on data augmentation [6] often introduces prediction inconsistencies, which can lead to error accumulation during CTTA. Moreover, while conventional CTTA methods typically address environmental corruptions (e.g., weather changes, sensor noise, or blur),

Received 11 August 2025; revised 30 November 2025; accepted 15 December 2025. Date of publication 30 December 2025; date of current version 17 April 2026. This work was supported in part by the National Natural Science Foundation of China under Grant 62172402, Grant 62472128, Grant U22B2047, and Grant 62450067; and in part by the Fundamental Research Funds for the Central Universities under Grant FRFCU5710011322. This article was recommended by Associate Editor S. Rong. *(Corresponding author: Guopu Zhu.)*

Long Sun, Guopu Zhu, and Hongli Zhang are with the School of Cyberspace Science, Harbin Institute of Technology, Harbin 150001, China (e-mail: 23b903089@stu.hit.edu.cn; guopu.zhu@hit.edu.cn; zhanghongli@hit.edu.cn).

Xinpeng Zhang is with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: zhangxinpeng@fudan.edu.cn).

Yicong Zhou is with the Department of Computer and Information Science, University of Macau, Macau, China (e-mail: yicongzhou@um.edu.mo).

Ligang Wu is with the Department of Control Science and Engineering, Harbin Institute of Technology, Harbin 150001, China (e-mail: ligangwu@hit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2025.3647640.

Digital Object Identifier 10.1109/TCYB.2025.3647640

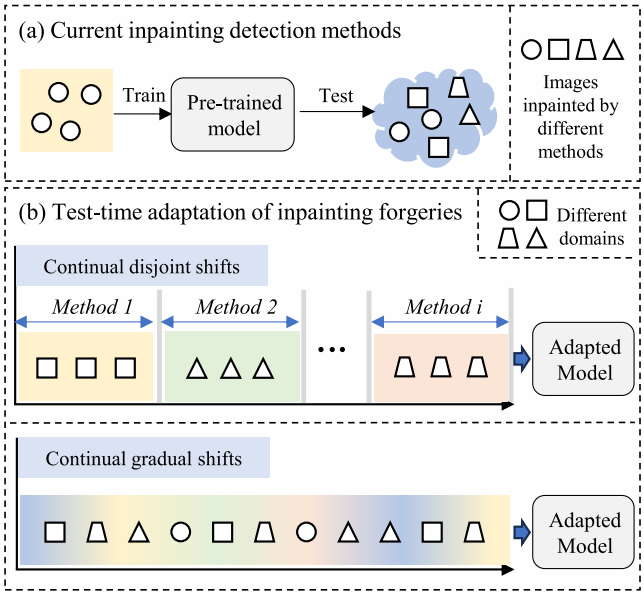

Fig. 1. Challenges in continuous inpainting forgery detection. (a) Current inpainted forgery detection methods do not account for the continuous adaptation of the model to test samples. In testing scenarios, they may perform poorly when faced with inpainting methods not seen during training. (b) This article aims to adapt the model for learning forged images in the test set. We consider continuous disjoint shift and continuous progressive shift scenarios, with an offset of the inpainting methods, to simulate the actual testing process.

inpainting detection faces "methodological shifts" driven by the rapid evolution of generative algorithms. As inpainting models evolve, the underlying statistical traces and artifact patterns change drastically. Static models trained on legacy algorithms often fail to generalize to these emerging, unseen generative mechanisms.

To address these limitations, we propose a test-time adaptation (TTA) framework for inpainting forgery detection (IDTTA). The core idea is to strategically differentiate adaptation strategies based on the model's intrinsic certainty for each test sample. Specifically, we introduce a novel metric that quantifies model certainty by measuring the sensitivity of predictions to input pixels. This score is then used to guide updates to the parameters. Since most inpainting forgery detection models employ an encoder–decoder architecture, we propose a side-tuning module based on this structure. It leverages the selected sample to adaptively learn from test samples and enhance the knowledge of the pretrained model. For high-certainty samples, we selectively update the running mean and variance of the batch normalization (BN) statistics and the side-tuning module using exponential moving average (EMA). These updates help calibrate feature distribution statistics, which are critical for mitigating domain shift. For low-certainty samples, we perform inference using the test sample's own BN statistics and the side-tuning module, thereby avoiding harmful updates caused by ambiguous or adversarial inputs.

To rigorously evaluate forensic detectors in the dynamic landscape, we define two distinct testing scenarios that differ significantly from prior adaptation settings (as illustrated in Fig. 1(b)) as follows.

1) *Technological Iteration (Continual Disjoint Shifts)*: This scenario simulates the chronological evolution of technology. The domain shift here is abrupt and distinct, requiring the model to adapt to a completely new generative mechanism without forgetting previous ones.
2) *Real-World Mixed Stream (Continual Gradual Shifts)*: This scenario simulates a chaotic deployment environment. The incoming stream is characterized by the stochastic influx of forgeries from heterogeneous sources, creating a nonstationary and high-entropy environment.

The results across two scenarios demonstrate that our method outperforms the existing methods in terms of average performance.

In this work, our major contributions are as follows.

1) We investigate, for the first time, the problem of TTA for inpainting forgery detection from the perspective of different inpainting methods, proposing IDTTA, a simple yet effective approach to adapt models to test samples from diverse domains.
2) We propose a metric to dynamically assess the reliability of the model's output on each test sample. Based on this certainty, we introduce instance-specific BN statistics and a side-tuning module to learn useful information while mitigating error accumulation.
3) We introduce two benchmarks based on different inpainting methods and construct a diverse-inpainting test dataset. Extensive experiments demonstrate the effectiveness and robustness of the proposed framework under varying distribution shifts.

The rest of this article is structured as follows. Section II provides a brief introduction to image inpainting, inpainting forensics, and TTA methods. Section III details the process of the proposed framework. Section IV describes the experimental setup and analyzes the results. Section V presents the conclusions of this study.

II. Related Work

A. Image Inpainting

The development of deep learning enables inpainting methods to generate new content and perform large-area restorations. Convolutional neural networks (CNNs) [8] and generative adversarial networks (GANs) [9] are now mainstream methods in image inpainting. Pathak et al. [10] introduced the first deep image inpainting method, combining an encoder–decoder network with the GAN. Many existing inpainting methods use a single-shot framework [11], [12], [13], incorporating methods like spatial regionwise normalization [11]. Attention mechanisms have been incorporated to enhance detail and relevance in image inpainting [14], [15]. For example, Zeng et al. [16] designed a pyramid context encoder network that uses attention shifts to refine image information from high level to low level. Furthermore, the two-stage inpainting process has become popular, enhancing both visual and semantic coherence [17], [18]. This process typically begins with a rough initial patch, followed by detailed restoration. There are two representative methods: gated convolution [19], which differentiates effective areas from holes,

and EdgeConnect [20], a two-stage adversarial model guided by generated edges. The growing popularity of transformers has led to their incorporation in image inpainting [21], [22], [23], enhancing the processing of high-resolution images by combining the strengths of transformers and convolutions [21]. Additionally, some methods employ diffusion models to restore image regions [24] or generate videos [25]. These methods outperform the previously mentioned models in terms of image quality and detail retention.

B. Inpainting Forensics

The principal objective of early inpainting forgery detection methods is to design appropriate hand-crafted features that can reveal inpainting artifacts. Wu et al. [26] introduced the first specific method for detecting inpainting images, employing zero-connected labeling and fuzzy membership to identify suspicious areas and inpainting blocks, respectively. Furthermore, Li et al. [27] utilized Laplacian transformations to construct intrachannel and interchannel variance features between inpainted and unpainted areas. However, these traditional manual feature-based inpainting detection algorithms are typically only effective for specific methods and often lack robustness and accuracy.

More recently, the focus of inpainting forgery detection has shifted to developing complex neural networks and more effective feature extraction mechanisms. Li and Huang [28] pioneered the detection of inpainting images using neural networks. They introduced high-pass residuals into a fully CNN to detect inpainting images by bilinear kernels in transposed convolution layers. Recently, Wu and Zhou [3] developed a detection algorithm that automatically selects neural network blocks and employs local–global attention mechanisms to improve performance. Additionally, Zhang et al. [29] designed a primary–secondary network approach to identify Photoshop-altered image areas. The primary network detects subtle inpainting clues using convolutional networks, while the secondary network amplifies co-occurring feature weights to capture overlooked features. Li et al. [30] proposed a transformer architecture that enhances local features for inpainting detection by learning pixel dependencies and statistical behaviors of inpainted and real areas. Zhang et al. [4] developed a feature pyramid network that captures traces from diffusion-based inpaintings using an improved U-shaped net for feature extraction and concatenation with a stagewise weighted cross-entropy loss to improve the prediction rate. Yao et al. [31] propose a dense feature interaction-based network, combined with a feature pyramid architecture to capture and enhance multiscale representations at each stage, while leveraging edge and shape information, like [32], to refine the localization of forged regions. However, these methods cannot adapt during testing. With the evolution of image inpainting, retraining the model each time to maintain detection performance would require significant effort.

C. Continual TTA

CTTA has been widely explored and applied, especially in the field of computer vision. The primary goal of continuous learning is to enable the model to adjust to test data in a given test environment adaptively. Tent [33] introduced an entropy minimization loss function for TTA, fixing model parameters and updating only the BN parameters to handle test data. Subsequent methods have been proposed to enhance detection performance by correcting BN parameters or their statistics. For example, Song et al. [34] proposed collecting knowledge from multiple representative domains to perform TTA based on composite domain knowledge. Additionally, Cotta [6] addressed the issue of continuous TTA by updating the pretrained source model to accommodate changing test data. This was accomplished by constructing a consistency loss between the enhanced and original images. BECotta [35] proposed combining mixed-domain low-rank experts with Cotta's approach, selectively capturing domain adaptation knowledge via multiple domain routers. In addition to these objectives, methods have been developed to assist model prediction through the design of image prompts. For example, VPTTA [36] proposed using low-frequency prompts in the frequency domain, combined with a prompt library to select similar image prompts, and using a mixture of source and target statistics as a loss function to update the prompts.

III. Proposed Method

In this section, we first formally define the task of TTA for inpainting forgery detection (IDTTA). Subsequently, we detail the process of our proposed method.

A. Notation and Preliminary

1) Notation: To facilitate representation, we formalize the setting of IDTTA. The continuously changing test dataset I_t for each task t continually arrives, as formulated to a task sequence $I_t = \{I_t^1, I_t^2, \ldots, I_t^i, \ldots, \}$. $I_t^i \in \mathbb{R}^{C\times H\times W}$ is the ith unlabeled test image with C channels and size of $H \times W$. Hence, the primary focus of IDTTA is to determine how to adapt the pretrained model to deal with the testing sequence.

2) Preliminary: BN [37] is extensively utilized in deep neural networks (DNNs) to standardize data within each mini-batch. During training, BN processes an input feature map $\mathbf{f} \in \mathbb{R}^{B\times C\times H\times W}$, where B is the batch size, C is the number of channels, and H and W are the height and width of the feature map, respectively. BN calculates the mean $\hat{\mu}$ and variance $\hat{\sigma}^2$ of the feature map $\mathbf{f}$ to ascertain the batch's central tendency and distribution. The formulas for calculating the mean and variance are as follows:

$$\hat{\mu} = \frac{1}{\mathrm{BHW}} \sum_{b\in\mathfrak{B}} \sum_{i\in\mathfrak{I}} \mathbf{f}_{b,c,i} \tag{1}$$

and

$$\hat{\sigma}^2 = \frac{1}{\mathrm{BHW}} \sum_{b\in\mathfrak{B}} \sum_{i\in\mathfrak{I}} \left(\mathbf{f}_{b,c,i} - \hat{\mu}\right)^2 \tag{2}$$

where $\hat{\mu}, \hat{\sigma}^2 \in \mathbb{R}^C$, each feature $\mathbf{f}_{b,c,i}$ is normalized over the batch dimensions $b \in \mathfrak{B} = \{1, 2, \ldots, B\}$ and spatial dimensions $i \in \mathfrak{I} = \{1, 2, \ldots, H \cdot W\}$, respectively.

Moreover, during training, the running mean μ and variance σ^2 are updated with the mean and variance through an EMA method as follows:

$$\mu_k = (1 - \eta) \cdot \mu_{k-1} + \eta \cdot \hat{\mu} \tag{3}$$

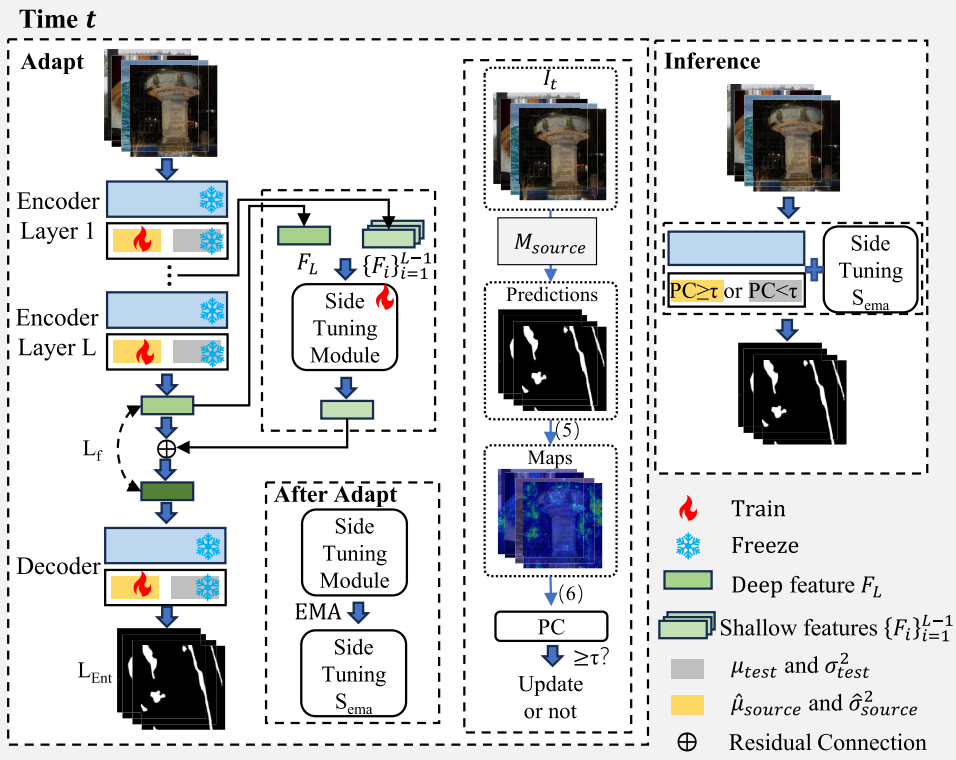

Fig. 2. Overview of the algorithm structure. Upon receiving a test sample, the adaptation process is first triggered, selectively fine-tuning the BN statistics and the side-tuning module based on the PC score. After adaptation, the EMA version of the side-tuning module S_{ema} is updated using the EMA method. Finally, the prediction is generated using the adapted BN and S_{ema}.

and

$$\sigma_k^2 = (1-\eta)\cdot\sigma_{k-1}^2 + \eta\cdot\hat{\sigma}^2 \tag{4}$$

where $\eta \in [0,1]$. During the testing phase, BN employs μ and σ^2 to normalize the features.

B. Dual-Mode BN

In the IDTTA setting, test data may originate from a distribution that differs from the training data, and the pretrained model may perform well on in-domain samples but poorly on out-of-domain ones. The standard BN layer normalizes features using the mean μ_{source} and variance σ^2_{source} estimated from the source domain. When the test distribution diverges from the training distribution, using either fixed training statistics or online-updated global test statistics for normalization may introduce bias, leading to feature distortion and performance degradation.

To address this, we propose a dual-mode normalization strategy, as depicted in Fig. 2: for in-domain samples, we apply the updated source-domain ($\hat{\mu}_{source}$ and $\hat{\sigma}^2_{source}$), while for out-of-domain samples, we compute normalization statistics from the test sample itself (μ_{test} and σ^2_{test}). This design prevents samples with small and large distribution shifts from interfering with each other during normalization. Samples with minimal shifts preserve the model's performance on known domains, while those with significant shifts encourage adaptation to unseen domains. It collaborates with the sample selection mechanism to dynamically choose between source statistics (for high-certainty samples) and instance statistics (for low-certainty samples), thereby balancing stability and adaptability.

C. Sample Selection

Previous TTA methods typically select samples for active learning using the entropy minimization principle or perturbation error rate. These strategies primarily focus on image classification tasks, aiming to improve recognition performance and accuracy at test time by selecting more reliable samples. However, for inpainting detection tasks, the model typically focuses only on the forged regions, and entropy minimization can mislead the model into learning incorrect labels with excessive confidence. Therefore, we propose to measure the model's reliability on a test sample by evaluating the overlap between the predicted region and the predefined sensitive region, enabling the selection of more trustworthy samples for model updating.

Inspired by [38], analyzing the influence of each pixel in x_t on the model's prediction p reveals the pixel sensitivity. Sensitive areas are defined as pixels crucial to the outcomes of inpainting detection. Therefore, if the model's predictions significantly differ from the sensitive areas, we can consider the model to be "inconsistent" with the current sample. Based on this perspective, we calculate the IOU score between the sensitive area and the prediction result to assess the model's "prediction consistency" (PC score) for the current test sample.

Initially, the input image undergoes forward propagation to detect forgery areas, denoted by $p_f, p_a \in \mathbb{R}^{B\times H\times W}$ for forgery and authentic region prediction, respectively. We compute

gradients for each pixel and assess pixel sensitivity based on the absolute differences in gradients, like [38]

$$\text{Map}_{x_t} = \max\left(\nabla_{x_t}\left|p_f - p_a\right|\right) \tag{5}$$

where the function $|\cdot|$ obtains the absolute value between p_f and p_a, and the function $\max(\cdot)$ calculates the maximum value along the channel axis. Next, we calculate the IOU score between the obtained Map_{x_t} and the forged region prediction p_f by:

$$\text{PC}_{x_t} = \text{IOU}\left(\text{Map}_{x_t}, p_f\right) \tag{6}$$

to obtain the PC_{x_t} score like Algorithm 2. A larger gradient value of Map_{x_t} indicates its contribution to the prediction result. Assuming that the model can effectively detect the inpainting image, its map must be highly consistent with the prediction result, and the PC_{x_t} score must be high. Otherwise, it indicates that the model's detection ability for the current sample is insufficient. After obtaining the sample performance evaluation metric PC_{x_t}, we apply a threshold τ to select samples for learning during the test process, using those with values above the threshold to update the model.

D. Cross Attention Side Tuning Module

Deep features typically encode high-level semantic information but may lose subtle local cues essential for forgery detection—such as edge inconsistencies, texture anomalies, and compression artifacts—which are often better preserved in shallow features. In addition, a pretrained network provides stable and generalized feature representations. However, directly updating the full or partial parameters risks distorting the pretrained feature representations or introducing unexpected noise.

To address this, we propose a side-tuning module that fuses shallow and deep features from the test data to facilitate test-time learning. By combining sample selection, noisy samples are effectively filtered out, allowing the side branches to focus on improving the representation of forged regions. Adapting cross-attention from [39], our module serves as a complementary branch with shallow features to refine deep feature instead of acting as a substitute. During the TTA process, only the components introduced in Section III-B and the side-tuning module are updated, while the rest of the network remains frozen as described in Fig. 2.

Specifically, given hierarchical features $\{\mathbf{F}_i\}_{i=1}^{L}$, where $L = 4$ is the number of layers. First, all features are projected to a common embedding space with dimension D_e

$$\mathbf{P}_i = \text{Downsample}_i\left(\mathbf{F}_i\right), \quad i = 1, \ldots, L \tag{7}$$

where $P_i \in \mathbb{R}^{B\times D_e\times H_L\times W_L}$ and $\text{Downsample}_i(\cdot)$ consists of a 1×1 convolution followed by a BN and an average pooling layer.

For each shallow features $\{\mathbf{F}_i\}_{i=1}^{L-1}$, we compute cross-attention between the following:

1) *Query*: Query-augmented deep features

$$\mathbf{Q}_i = \text{Flat}\left(\mathbf{P}_L\right) \in \mathbb{R}^{B\times(H_L W_L)\times D_e}. \tag{8}$$

2) *Key/Value*: Pooled shallow features

$$\mathbf{K}_i = \mathbf{V}_i = \text{Flat}\left(\mathbf{P}_i\right) \in \mathbb{R}^{B\times(H_L W_L)\times D_e}. \tag{9}$$

The attention weights are computed with identity augmentation

$$\mathbf{A}_i = \text{softmax}\left(\frac{\mathbf{Q}_i\mathbf{K}_i^T}{\sqrt{D_e}}\right) + \mathbf{I} \tag{10}$$

where $\mathbf{I}$ is an identity matrix.

The attended features are then computed by $\hat{\mathbf{V}}_i = \mathbf{A}_i\mathbf{V}_i$. Finally, all attended features are summed and projected by

$$\hat{\mathbf{F}_L} = \alpha \cdot \mathbf{F}_{mix} + \mathbf{F}_L \tag{11}$$

$$\mathbf{F}_{mix} = \text{Up}\left(\text{ReLU}\left(\text{Conv}_{1\times1}\left(\sum_{i=1}^{L-1}\hat{\mathbf{V}}_i + \mathbf{P}_L\right)\right)\right) \tag{12}$$

where α is a scale factor.

E. Loss Functions

1) Distillation Loss: Distillation loss is commonly used to mitigate the forgetting of previously learned knowledge in a model. To avoid the accumulation of erroneous information, we compute the distillation loss for deep features

$$L_f = \text{MSE}\left(\hat{\mathbf{F}_L}, \mathbf{F}_L\right) \tag{13}$$

where $\text{MSE}(\cdot)$ is the MSE loss and $\mathbf{F}_L$ is from pretrained model.

2) Entropy Minimization Loss: The model M_{test} uses entropy minimization loss to learn directly from the prediction

$$L_{Ent} = -P_{x_t}\log P_{x_t}, P_{x_t} = \text{Sigmoid}\left(p_{ij}\right) \tag{14}$$

where p_{ij} is the probability of being forged at coordinate (i, j).

3) Total Loss: Thus, the total loss function L is defined as follows:

$$L = L_{Ent} + \lambda L_f \tag{15}$$

where λ is the weight for L_f.

F. Overall Method

First, replace the BatchNorm layers in the pretrained model with those described in Section III-B. Then, integrate the side-tuning module in Section III-D, freeze all other parameters, and conduct TTA with the replaced model M_{test} following the procedure outlined in Algorithm 1. For each test sample x_t, prediction consistency PC_{x_t} is first computed. The model's BatchNorm statistic is selected according to a predefined threshold τ. If PC_{x_t} is greater than τ, the parameters of the side-tuning module S are updated, and S_{ema} is updated using exponential moving average (EMA). Otherwise, the prediction is made using the test sample's statistics, and the side-tuning module remains unchanged.

IV. Experiments

A. Experimental Setup

1) Datasets: Table I lists several representative inpainting methods, including three traditional methods (i.e., NS [40], TE [41], and Exemplar [42]) and 12 deep-learning-based methods (i.e., EC [20], RN [11], AOT [12], RFR [14], ICT [22], MAT [21], Lama [43], CMT [23], DALLE-2, GLIDE, SD2, and SDXL). Besides CNN-based methods, we include recent

Algorithm 1 IDTTA Workflow

Initialize:
- Model M_{test}, side-tuning module S and its copy S_{ema}
- threshold τ, momentum m

For each test sample x**:**
1. Compute prediction consistency PC_{x_t}
2. **If** $PC_{x_t} \geq \tau$**:**
   a. Update S with loss L (Eq. 15)
   b. Update S_{ema} with momentum m
   c. Use S_{ema} and updated $\hat{\mu}_{source}$ and $\hat{\sigma}^2_{source}$ for final prediction
3. **Else:**
   a. Use current S_{ema} and test batch statistics μ_{test} and σ^2_{test} for final prediction without updating

Return: Predictions

Algorithm 2 Prediction Consistency Calculation

Input: Model M_{source}, input x_t
Output: PC_{x_t}
1. Compute model prediction p_f and p_a
2. Calculate Map_{x_t} using Eq. 5
3. Binarize Map_{x_t} and p_f
4. Compute PC_{x_t} with binarized Map_{x_t} and p_f by Eq. 6

Return: PC_{x_t}, Map_{x_t}

TABLE I
DATASET DESCRIPTION

Datasets	Source	Data Numbers	Inpainting Method
NS	Places365	3000	NS
TE	MSCOCO	3000	TE
RN	Places365	3000	RN
EC	Places365	3000	EC
AOT	Places365	3000	AOT
RFR	MSCOCO	3000	RFR
ICT	MSCOCO	3000	ICT
MAT	Places365	3000	MAT
Lama	MSCOCO	3000	Lama
CMT	Places365	3000	CMT
DEFACTO	MSCOCO	3000	Exemplar
AutoSplice	Visual News	3621	DALLE-2
CocoGlide	MSCOCO	512	GLIDE
TGIF	MSCOCO	4116	SD2, SDXL
Total	-	41249	-

transformer-based methods (such as MAT, ICT, and CMT) and diffusion-model-based methods (such as DALLE-2 and GLIDE). Except for the DEFACTO [42], AutoSplice [44], CocoGlide [45], and TGIF [46], which are the existing datasets (sampled from the test dataset), the others are generated by us. Masks are categorized into three types: 1) randomly sampled from [47]; 2) randomly generated circular, rectangular, and sector shapes; and 3) large irregular areas generated by [21]. Each mask type comprises one-third of the total dataset and can appear at any position within an image. Original images are sourced from random samples of places [48] and MSCOCO [49]. Table I provides statistical data for the dataset.

2) Implementation Details: We implement the proposed method with the PyTorch framework, employing the latest inpainting detection model DeFI-Net [31] as the backbone network. Experiments are conducted on a server equipped with one GTX 3090 GPU. The SGD optimizer with default parameters is used. We set the batch size to 1 for online continual test time adaptation. The threshold τ and the momentum m are set to 0.1 and 0.999, respectively. We train the network end-to-end with an initial learning rate of $2.5e^{-4}$. The weight λ for L_f is set to 0.01. The thresholds for calculating IOU are 0.01 (Map_{x_t}) and 0.5 (prediction), respectively. All images used during the training and inference phases are resized to a size of 352×352. In addition, we randomly generated 1000 images inpainted by GC for initializing side tuning parameters.

3) Evaluation Metrics: This experiment aims to detect and locate the inpainting areas in the images to be inspected. We measure the network's detection performance using the $F1$ score and the area under the receiver operating characteristic curve area under the curve (AUC). The calculation formula for the $F1$ score is as follows:

$$F_1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

where Precision = (TP)/(TP + FP), Recall = (TP)/(TP + FN), TP is the number of true positive pixels, FP is the number of false positive pixels, and FN is the number of false negative pixels. The $F1$ score is the harmonic mean of precision and recall, with a maximum value of 1 and a minimum value of 0. Area under the curve (AUC) measures the overall performance of a binary classification model across all classification thresholds, formulated as

$$\text{AUC} = \int_0^1 \text{TPR}\,(\text{FPR})\, d\text{FPR} \tag{17}$$

where TPR = (TP)/(TP + FN) and FPR = (FP)/(FP + TN). The higher $F1$ score and AUC indicate better detection results. In the following tables, $F1$ and AUC scores are presented as percentages, e.g., "92.58/99.00" denotes an $F1$ score of 92.58% and an AUC of 99.00%.

B. Comparison With Existing TTA Methods

We evaluate our proposed method against several TTA methods as follows.

Source [31]: Test results of pretrained models on testing datasets.

Norm-EMA: A method that only updates the batchnorm statistic.

Tent [33]: A method to minimize the loss according to entropy and only update the BN parameters.

Cotta [6]: An algorithm for updating models based on consistency loss and momentum for image enhancement.

VPTTA [36]: An algorithm for adding prompts in the frequency domain and storing prompt memories.

Grata [50]: A method to improve gradient direction and learning rate based on gradient alignment.

TABLE II
QUANTITATIVE COMPARISONS OF DIFFERENT TTA METHODS OVER SOURCE, NORM-EMA, TENT, COTTA, VPTTA, GRATA, AND THE PROPOSED FRAMEWORK WITH CONTINUAL DISJOINT SHIFTS

Datasets	Source	Norm-EMA	Tent	Cotta	VPTTA	Grata	IDTTA
NS	92.58 / 99.00	92.31 / 98.93	87.99 / 97.24	87.53 / 97.13	92.51 / 99.00	88.10 / 97.37	93.06/98.78
TE	96.74 / 99.91	96.61 / 99.91	91.80 / 99.17	91.02 / 99.17	96.87 / 99.92	89.53 / 99.00	96.30/99.87
RN	40.52 / 48.22	44.76 / 55.62	46.78 / 58.42	47.02 / 62.19	40.53 / 48.46	48.77 / 59.98	48.54/58.47
EC	40.71 / 49.50	44.26 / 55.28	45.89 / 56.24	45.01 / 57.61	40.73 / 49.67	48.13 / 57.90	47.46/55.57
AOT	68.19 / 91.65	76.47 / 92.49	75.11 / 90.46	73.63 / 89.34	68.74 / 91.74	73.92 / 90.44	80.67/93.33
RFR	47.15 / 50.31	48.95 / 56.34	50.18 / 57.40	48.97 / 54.01	47.17 / 50.39	49.68 / 56.15	51.62/60.55
ICT	82.84 / 97.98	88.38 / 98.49	79.46 / 95.28	74.77 / 93.36	83.46 / 98.02	76.61 / 94.83	88.34/98.24
MAT	93.65 / 98.97	93.42 / 98.77	90.09 / 98.11	85.46 / 97.05	93.62 / 98.97	88.38 / 97.66	92.42/98.25
Lama	94.60 / 99.82	94.72 / 99.74	88.77 / 97.19	81.38 / 95.30	94.74 / 99.82	84.77 / 96.74	95.32/99.70
CMT	94.59 / 99.07	94.02 / 98.98	91.13 / 98.22	87.88 / 97.61	94.57 / 99.08	85.82 / 96.55	93.57/98.64
DEFACTO	43.57 / 56.52	34.11 / 60.36	43.01 / 64.14	42.04 / 64.20	43.46 / 56.82	41.48 / 61.20	40.02/63.35
AutoSplice	67.83 / 85.69	71.68 / 85.62	64.74 / 77.77	69.49 / 83.69	68.19 / 85.54	61.02 / 71.81	74.42/87.55
CocoGlide	57.11 / 71.76	51.77 / 64.36	60.41 / 71.17	44.68 / 45.46	57.89 / 72.21	57.97 / 69.63	53.73/62.32
TGIF	42.44 / 54.27	33.27 / 38.74	42.36 / 46.47	37.54 / 29.92	42.65 / 54.21	45.02 / 50.60	41.79/46.46
Average	65.85 / 75.82	64.70 / 74.37	65.41 / 75.44	62.24 / 71.18	66.03 / 75.89	65.62/ 76.70	**69.27/77.84**

1) Continual Disjoint Shifts Scenes: In real-world forensics, new inpainting tools emerge sequentially. A detector trained on legacy methods must adapt to these new distinct domains one by one. This distinguishes our work from [35], as we focus on the interalgorithm domain gap rather than continuous environmental variations. First, we tested the scenario where the inpainting images come in sequence (i.e., continual disjoint shifts) and compared our proposed method with various test-time strategies. The detailed results are shown in Table II. It can be seen that the performance of our proposed IDTTA is superior to that of the compared methods. Tent only uses test data to update the mean and variance of the BN layers during inference, failing to effectively leverage the BN statistics of the pretrained model. This leads to a performance drop on inpainting methods, where the source model originally performs well. In contrast, Cotta relies on image augmentation consistency losses to adapt the model, which may introduce prediction errors and cause gradual forgetting of previously learned knowledge.

Overall, the compared TTA methods each have their strengths and limitations. In contrast, our proposed method not only maintains detection performance in domains where the source model is originally effective, but also shows improvements in more challenging domains. Overall, IDTTA achieves the highest average performance compared to other methods, with $F1$ and AUC of 69.27% and 77.84%. Specifically, our method improves the $F1$ scores on AOT, ICT, and AutoSplice (where the source model exhibits suboptimal performance) by 12.48%, 5.50%, and 6.59% against the source model, respectively. Meanwhile, it also boosts the $F1$ scores on RN, EC, and RFR (where the source model performs poorly) by 8.02%, 6.75%, and 4.47%.

2) Continual Gradual Shifts Scenes: Next, we evaluated a scenario involving out-of-order image inpainting, referred to as continual gradual shifts, and compared our proposed method with several TTA strategies. This scenario simulates a more chaotic deployment environment, such as inspecting images uploaded to a social media platform. It rigorously evaluates the model's robustness against "catastrophic forgetting" when the distribution of forgery traces changes unpredictably at every time step. The detailed results are presented in Table III. The table shows that under the disorder scenario, Tent, Norm-EMA, VPTTA, and Grata all exhibit improved performance. We hypothesize that the performance improvement of these methods may stem from the shuffled nature of the scenario, which results in a more gradual domain shift compared to the disjoint scenario, where the shift is more abrupt and continuous. Our proposed method continues to demonstrate superior detection performance, achieving $F1$ and AUC scores of 71.85% and 81.33%, respectively. Specifically, the $F1$ scores for AOT, ICT, and AutoSplice are improved by 8.29%, 6.99%, and 9.28% against the source model, respectively. Meanwhile, improvements of 10.09%, 8.93%, and 5.02% are observed in RN, EC, and RFR, where the source model exhibits poor performance.

C. Ablation Study

We conduct ablation experiments on various modules to demonstrate the effectiveness of each module, as described in Tables IV and V. In the experiments, "DeFI-Net" is the pretrained model, "DualBN" denotes the dual-mode BN in Section III-B, and "side tuning" indicates the side tuning module in Section III-D. The "+" symbol indicates the inclusion of the mentioned methods in training, e.g., "DeFI-Net+DualBN" representing the combined use of the pretrained model and dual-mode BN. Additionally, "Var#0," "Var#1," "Var#2," and "Var#3" correspond to "DeFI-Net," "DeFI-Net+DualBN," "DeFI-Net+DualBN+Side tuning," and "DeFI-Net+DualBN+Side tuning with EMA update," respectively.

1) Impact of DualBN: In the disjoint shift scenario, introducing DualBN improves the $F1$ score from 65.85% to 67.54% (+1.69%) and the AUC from 75.82% to 77.03% (+1.21%). The improvement is more pronounced in the

TABLE III
QUANTITATIVE COMPARISONS OF DIFFERENT TTA METHODS OVER SOURCE, NORM-EMA, TENT, COTTA, VPTTA, GRATA, AND THE PROPOSED FRAMEWORK WITH CONTINUAL GRADUAL SHIFTS

Datasets	Source	Norm-EMA	Tent	Cotta	VPTTA	Grata	IDTTA
NS	92.58 / 99.00	93.13 / 98.97	88.08 / 97.27	76.11 / 91.45	92.52 / 98.96	87.35 / 97.03	89.13/97.33
TE	96.74 / 99.91	96.20 / 99.89	91.70 / 99.18	80.96 / 95.32	96.62 / 99.91	88.89 / 98.92	95.10/99.69
RN	40.52 / 48.22	40.66 / 46.65	47.00 / 58.49	42.36 / 53.00	40.56 / 47.98	47.31 / 58.24	50.61/61.00
EC	40.71 / 49.50	40.83 / 49.23	46.05 / 56.28	42.26 / 51.76	40.71 / 49.02	46.11 / 55.32	49.64/58.85
AOT	68.19 / 91.65	66.62 / 88.62	75.07 / 90.48	57.40 / 71.40	67.86 / 91.40	75.12 / 90.89	76.48/92.71
RFR	47.15 / 50.31	47.26 / 50.33	50.20 / 57.47	47.60 / 51.91	47.16 / 49.42	49.83 / 56.50	52.17/61.11
ICT	82.84 / 97.98	89.68 / 98.77	79.25 / 95.19	67.34 / 84.05	81.80 / 97.54	79.56 / 95.81	89.83/98.61
MAT	93.65 / 98.97	92.69 / 98.65	90.06 / 98.10	70.83 / 86.91	93.28 / 98.88	88.58 / 97.73	91.57/98.24
Lama	94.60 / 99.82	94.47 / 99.74	88.70 / 97.16	68.70 / 84.36	93.58 / 99.73	87.85 / 97.58	94.83/99.67
CMT	94.59 / 99.07	93.30 / 98.80	91.10 / 98.22	68.60 / 83.33	94.30 / 98.99	90.55 / 98.12	92.71/98.38
DEFACTO	43.57 / 56.52	33.99 / 54.54	42.50 / 63.97	48.28 / 55.70	45.30 / 57.11	42.36 / 64.33	41.33/61.18
AutoSplice	67.83 / 85.69	67.29 / 90.63	65.80 / 78.49	39.34 / 42.97	66.41 / 83.60	66.64 / 79.35	77.06/92.29
CocoGlide	57.11 / 71.76	52.15 / 72.66	59.93 / 70.81	60.74 / 74.01	59.21 / 73.79	57.27 / 67.29	53.37/63.51
TGIF	42.44 / 54.27	34.75 / 49.21	44.55 / 47.28	50.58 / 56.51	44.82 / 57.32	43.99 / 45.35	52.07/56.07
Average	65.85 / 75.82	67.36 / 78.33	68.57 / 79.17	58.65 / 70.19	68.87 / 78.83	67.96 / 78.75	**71.85/81.33**

TABLE IV
ABLATION STUDY OF THE PROPOSED METHOD WITH DISJOINT SHIFTS

Datasets	Var#0	Var#1	Var#2	Var#3
NS	92.58 / 99.00	92.15/98.88	94.46/99.18	93.06/98.78
TE	96.74 / 99.91	96.49/99.89	96.25/99.88	96.30/99.87
RN	40.52 / 48.22	47.16/58.60	46.79/56.87	48.54/58.47
EC	40.71 / 49.50	46.22/56.35	45.78/54.42	47.46/55.57
AOT	68.19 / 91.65	76.62/92.51	79.08/91.89	80.67/93.33
RFR	47.15 / 50.31	50.23/57.54	52.30/62.32	51.62/60.55
ICT	82.84 / 97.98	85.90/97.31	88.47/98.13	88.34/98.24
MAT	93.65 / 98.97	93.21/98.71	92.91/98.22	92.42/98.25
Lama	94.60 / 99.82	94.20/99.53	95.45/99.70	95.32/99.70
CMT	94.59 / 99.07	93.76/98.93	93.74/98.62	93.57/98.64
DEFACTO	43.57 / 56.52	37.52/61.46	37.72/61.14	40.02/63.35
AutoSplice	67.83 / 85.69	71.83/85.71	78.71/89.97	74.42/87.55
CocoGlide	57.11 / 71.76	54.17/66.61	50.42/61.49	53.73/62.32
TGIF	42.44 / 54.27	36.79/41.70	37.65/44.80	41.79/46.46
Average	65.85 / 75.82	67.54/77.03	68.49/77.43	69.27/77.84

TABLE V
ABLATION STUDY OF THE PROPOSED METHOD WITH GRADUAL SHIFTS

Datasets	Var#0	Var#1	Var#2	Var#3
NS	92.58 / 99.00	92.66/98.91	91.48/98.77	89.13/97.33
TE	96.74 / 99.91	96.53/99.88	94.35/99.80	95.10/99.69
RN	40.52 / 48.22	47.17/58.57	41.49/51.03	50.61/61.00
EC	40.71 / 49.50	46.20/56.34	41.96/55.22	49.64/58.85
AOT	68.19 / 91.65	70.85/91.24	74.06/92.49	76.48/92.71
RFR	47.15 / 50.31	50.23/57.54	48.02/56.17	52.17/61.11
ICT	82.84 / 97.98	86.60/97.54	88.55/98.76	89.83/98.61
MAT	93.65 / 98.97	92.57/98.70	91.66/98.70	91.57/98.24
Lama	94.60 / 99.82	94.44/99.58	94.95/99.80	94.83/99.67
CMT	94.59 / 99.07	93.50/98.91	93.10/98.95	92.71/98.38
DEFACTO	43.57 / 56.52	38.13/58.34	41.55/59.61	41.33/61.18
AutoSplice	67.83 / 85.69	71.73/91.07	74.57/90.36	77.06/92.29
CocoGlide	57.11 / 71.76	56.02/71.75	55.91/69.50	53.37/63.51
TGIF	42.44 / 54.27	38.77/47.97	41.44/46.58	52.07/56.07
Average	65.85 / 75.82	69.67/80.45	69.51/79.70	71.85/81.33

gradual shift scenario, with an $F1$ score increase of 3.82% points and an AUC improvement of 4.63% points. These results indicate that DualBN is better suited for progressive distribution shifts, consistent with its original design objective: to address varying degrees of domain shift by dynamically selecting normalization strategies based on source domain or test sample statistics. Moreover, when DualBN is combined with "side tuning," the model achieves optimal performance in both scenarios. This suggests that DualBN provides a more stable feature representation for downstream adaptation modules (e.g., side tuning) by mitigating feature distribution shifts, thereby enabling complementary enhancement. In addition, we evaluated the impact of varying threshold values as displayed in Table VI. Based on the results, we selected the threshold τ of 0.1, which yields the best performance in terms of both $F1$ and AUC. In addition, we analyzed the sample ratio of the pretrained model across different datasets, as shown in Fig. 3. As shown in the figure, for datasets with large distribution shifts (e.g., RN and EC), the PC_{x_t} tends to reject most samples. In contrast, for datasets with smaller shifts (e.g., Lama and CMT), it effectively filters out samples with poor performance.

TABLE VI
IMPACT OF VARYING THRESHOLD VALUES

Values	Shifts	Setup	Average F1	Average AUC
0.05	Disjoint	Var#1	67.55	76.96
0.1	Disjoint	Var#1	67.54	77.03
0.15	Disjoint	Var#1	67.35	77.08
0.2	Disjoint	Var#1	67.13	77.05

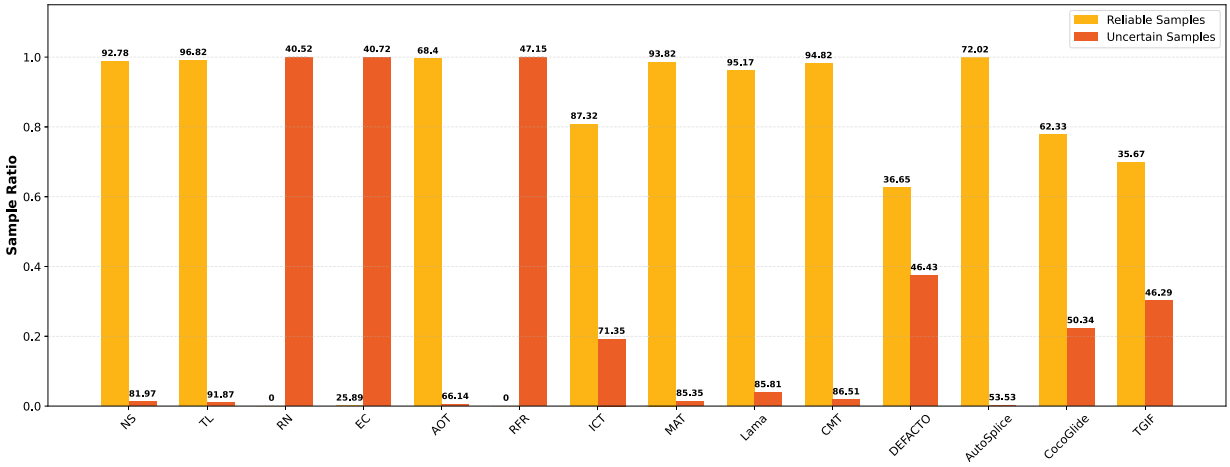

Fig. 3. Proportion of test samples selected according to PC_{x_t}. The horizontal axis in the figure represents the datasets, and the vertical axis represents the proportion of $PC_{x_t} >= \tau$ and $PC_{x_t} < \tau$ samples in the total dataset. The average $F1$ score for both certain and uncertainty samples from the pretrained model is displayed at the top.

TABLE VII
IMPACT OF MODEL PARAMETER INITIALIZATION

Initialization	Shifts	Setup	Average F1	Average AUC
Random	Disjoint	Var#3	69.12	77.76
GC	Disjoint	Var#3	69.27	77.84
Random	Gradual	Var#3	71.12	80.64
GC	Gradual	Var#3	71.85	81.33

TABLE VIII
EFFECT OF VARYING SCALE FACTORS α

Values	Shifts	Setup	Average F1	Average AUC
0.05	Disjoint	Var#3	69.05	77.44
0.1	Disjoint	Var#3	69.27	77.84
0.2	Disjoint	Var#3	69.19	77.80
0.3	Disjoint	Var#3	68.85	77.59

2) Effectiveness of Side Tuning: Our research also explores the impact of the side tuning module. We compare the side-tuning module with and without the EMA update. In the disjoint shift scenario, the average performance improves from 67.54%/77.03% (Var#1) to 68.49%/77.43% (Var#2). However, in the gradual shift scenario, side tuning alone results in a sharp performance drop due to error accumulation, whereas incorporating EMA updates yields the best performance. These results suggest that adding the side-tuning module enables the model to learn more effectively and improves detection performance on test samples with high prediction consistency, such as AOT and ICT. The EMA update mechanism demonstrates greater adaptability in the gradual shift scenario, yielding average improvements of 3.36% and 3.90% in $F1$ and AUC, respectively. In particular, EMA provides notable improvements on datasets, such as RN, EC, AOT, AutoSplice, and TGIF. Across both scenarios, the complete method (Var#3) achieves the highest average performance. These findings demonstrate that combining side-tuning with EMA updates effectively enhances the model's adaptability to distribution shifts, particularly under the conditions of gradual shift. In Table VII, we also investigate the impact of different initialization strategies (random vs. GC) on the IDTTA framework. Moreover, we evaluated the effect of different fixed alpha values in Table VIII. The results indicate that setting α to our method yields the best performance.

3) Generalizability Across Inpainting Detection Models: To demonstrate the architectural generalizability of the proposed IDTTA framework, we evaluated its performance on DeFI-Net [31], PSCC-Net [51], and SparseViT [52], all trained on the IID dataset [3]. DeFI-Net demonstrates superior performance on the generated initial GC dataset, justifying its selection as the backbone network. For SparseViT, the standard IDTTA configuration was applied. In the case of PSCC-Net, however, we observed that its shallow feature extractors exhibited sensitivity to the high variance. We froze the standard BN layers in these stages to preserve low-level feature stability, applying dual-mode BN exclusively to the deeper stages. As shown in Table IX, IDTTA significantly enhances the detection performance of both PSCC-Net and SparseViT across the evaluated scenarios. These findings align with the improvements observed in DeFI-Net, demonstrating that the proposed strategies can be effectively integrated into these forgery detection networks.

D. Discussion

1) Challenges From Specific Inpainting Mechanisms: The performance variations across datasets, detailed in Tables II and III, highlight two distinct categories of challenges for IDTTA as follows.

Severe Domain Shifts (e.g., RN and EC): In these scenarios, the distribution of forgery artifacts deviates fundamentally from that of the training set. As analyzed in Section IV-C, the proposed PC metric effectively filters out a substantial majority of these samples [see Fig. 4(b)].

TABLE IX
IMPACT OF INPAINTING DETECTION MODELS

Model	Shifts	Average F1	Average AUC
DeFI-Net	GC	95.23	99.05
DeFI-Net	NS-TGIF	65.85	75.82
DeFI-Net+IDTTA	Disjoint	69.27	77.84
DeFI-Net+IDTTA	Gradual	71.85	81.33
PSCC-Net	GC	94.31	99.18
PSCC-Net	NS-TGIF	65.37	70.81
PSCC-Net+IDTTA	Disjoint	67.04	74.66
PSCC-Net+IDTTA	Gradual	68.20	76.27
SparseVIT	GC	94.49	97.54
SparseVIT	NS-TGIF	70.00	73.86
SparseVIT+IDTTA	Disjoint	70.00	75.50
SparseVIT+IDTTA	Gradual	71.32	76.76

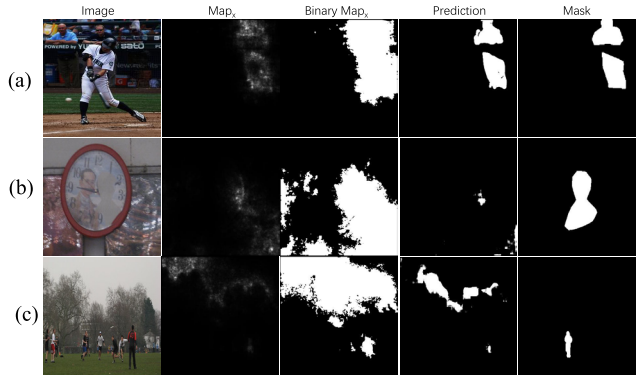

Fig. 4. Comparison of cases where $PC_{x_t} >= \tau$ and $PC_{x_t} < \tau$ (with the testing threshold τ set to 0.1). As shown in the first row, when $PC_{x_t} >= \tau$, the prediction exhibits significant overlap with Map_{x_t}, indicating robust detection performance. Conversely, when $PC_{x_t} < \tau$, the overlap is minimal, showing the suboptimal detection performance. Notably, the figure also highlights an anomaly, where the PC score exceeds the threshold despite poor prediction performance. (a) PC = 0.33. (b) PC = 0.02. (c) PC = 0.15.

High-Fidelity Synthesis (e.g., TGIF): Diffusion-based generators synthesize images with realistic high-frequency details that obscure traditional forgery traces. Consequently, the gradient-based sensitivity map (Map_{x_t}) struggles to localize these subtle anomalies, yielding erratic PC scores that fail to accurately reflect the model's true uncertainty [Fig. 4(c)].

2) Behavioral Analysis of PC_{x_t}: To elucidate the intuitive interpretation of the PC score, we analyze its behavior across various types of inpainting forgeries. Specifically, test samples are categorized into three distinct regimes based on the distribution presented in Fig. 4.

High-PC Group (Reliable Adaptation): For datasets, such as Lama [Fig. 4(a)], the pretrained model exhibits a high sample selection ratio (high PC_{x_t}). Consequently, the pixel sensitivity map, Map_{x_t}, demonstrates strong alignment with the predicted regions, yielding high IoU scores. These samples provide high-quality pseudo-supervision, rendering them ideal for side-tuning parameter updates.

Low-PC Group (Uncertainty Avoidance): Conversely, for datasets exhibiting severe domain shifts, such as DEFACTO [Fig. 4(b)], the model generates low PC scores. As visualized, the sensitivity maps for these samples are often diffuse, focusing on irrelevant background noise rather than the tampered regions. Instead of imposing parameter updates—which would introduce noise—the system adaptively switches to instance-specific BN statistics.

High-PC Group (False Positives): While PC_{x_t} effectively identifies reliable samples, certain instances may erroneously yield high scores, potentially leading to incorrect updates [Fig. 4(c)]. However, the proposed loss function, L_f, combined with EMA-based side-tuning, mitigates potential feature shifts.

V. CONCLUSION

This article tackles the growing challenge of detecting image inpainting forgeries, especially when test data diverge from the training distribution in terms of inpainting methods, sources, and tampered regions. We pioneer the application of TTA to inpainting forgery detection and propose a robust framework that enables models to select reliable samples and adapt from unlabeled test streams. Specifically, we introduce dual-mode BN to leverage pretrained models under varying conditions, selectively updating statistics based on the consistency of predictions. We further propose a cross-attention side-tuning module to incorporate test-time knowledge into the pretrained model, while reducing error accumulation via EMA updates. Finally, we conduct extensive experiments using 15 inpainting algorithms under both disjoint and gradual shift settings to validate the effectiveness of our approach. Our method consistently outperforms the state-of-the-art TTA approaches in terms of average $F1$ score and AUC, demonstrating adaptability to complex tampering patterns.

While our framework marks a significant step forward in TTA for inpainting forensics, real-world deployment introduces further challenges. Future work will focus on enhancing robustness to compound domain shifts, such as inpainting combined with image processing operations (e.g., compression, resizing) to improve the practical applicability of the model.

REFERENCES

[1] H. Xiang, Q. Zou, M. A. Nawaz, X. Huang, F. Zhang, and H. Yu, "Deep learning for image inpainting: A survey," *Pattern Recognit.*, vol. 134, Feb. 2023, Art. no. 109046.
[2] Y. Zhang, Z. Fu, S. Qi, M. Xue, Z. Hua, and Y. Xiang, "Localization of inpainting forgery with feature enhancement network," *IEEE Trans. Big Data*, vol. 9, no. 3, pp. 936–948, Jun. 2023.
[3] H. Wu and J. Zhou, "IID-Net: Image inpainting detection network via neural architecture search and attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 3, pp. 1172–1185, Mar. 2022.
[4] Y. Zhang, F. Ding, S. Kwong, and G. Zhu, "Feature pyramid network for diffusion-based image inpainting detection," *Inf. Sci.*, vol. 572, pp. 29–42, Sep. 2021.
[5] X. Fu, G. Zhu, H. Zhang, X. Zhang, A. T. S. Ho, and S. Kwong, "Multi-level feature fusion network for shadow removal detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 35, no. 7, pp. 6508–6521, Jul. 2025.
[6] Q. Wang, O. Fink, L. Van Gool, and D. Dai, "Continual test-time domain adaptation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 7201–7211.
[7] S. Niu et al., "Efficient test-time model adaptation without forgetting," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 16888–16905.

[8] Y. LeCun et al., "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.
[9] R. Labaca-Castro, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2023, pp. 73–76.
[10] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2016, pp. 2536–2544.
[11] T. Yu et al., "Region normalization for image inpainting," in *Proc. AAAI Conf. Arti. Intell.*, 2020, pp. 12733–12740.
[12] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Aggregated contextual transformations for high-resolution image inpainting," *IEEE Trans. Vis. Comput. Graphics*, vol. 29, no. 7, pp. 3266–3280, Jul. 2023.
[13] H. Li, G. Li, L. Lin, H. Yu, and Y. Yu, "Context-aware semantic inpainting," *IEEE Trans. Cybern.*, vol. 49, no. 12, pp. 4398–4411, Dec. 2019.
[14] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2020, pp. 7757–7765.
[15] D. Zhan, J. Wu, X. Luo, and Z. Jin, "Learning from text: A multimodal face inpainting network for irregular holes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 8, pp. 7484–7497, Aug. 2024.
[16] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2019, pp. 1486–1494.
[17] Y. Ma et al., "Regionwise generative adversarial image inpainting for large missing areas," *IEEE Trans. Cybern.*, vol. 53, no. 8, pp. 5226–5239, Aug. 2023.
[18] Y. He and G. K. Atia, "Coarse to fine two-stage approach to robust tensor completion of visual data," *IEEE Trans. Cybern.*, vol. 54, no. 1, pp. 136–149, Jan. 2024.
[19] J. Yu, Z. Lin, S. Yan, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4470–4479.
[20] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "EdgeConnect: Structure guided image inpainting using edge prediction," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 3265–3274.
[21] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "MAT: Mask-aware transformer for large hole image inpainting," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. (CVPR)*, Jun. 2022, pp. 10748–10758.
[22] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 4672–4681.
[23] K. Ko and C.-S. Kim, "Continuously masked transformer for image inpainting," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2023, pp. 13123–13132.
[24] A. Nichol et al., "GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 16784–16804.
[25] D.-M. Liu et al., "NADM: Noise-aware diffusion model for landscape painting video generation," *IEEE Trans. Cybern.*, vol. 55, no. 8, pp. 3686–3698, Aug. 2025.
[26] Q. Wu, S. Sun, W. Zhu, G. Li, and D. Tu, "Detection of digital doctoring in exemplar-based inpainted images," in *Proc. Int. Conf. Mach. Learn. Cybern.*, 2008, pp. 1222–1226.
[27] H. Li, W. Luo, and J. Huang, "Localization of diffusion-based inpainting in digital images," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 12, pp. 3050–3064, Dec. 2017.
[28] H. Li and J. Huang, "Localization of deep inpainting using high-pass fully convolutional network," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 8300–8309.
[29] Y. Zhang, Z. Fu, S. Qi, M. Xue, X. Cao, and Y. Xiang, "PS-Net: A learning strategy for accurately exposing the professional photoshop inpainting," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 35, no. 10, pp. 13874–13886, Oct. 2024.
[30] Y. Li et al., "Transformer-based image inpainting detection via label decoupling and constrained adversarial training," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 34, no. 3, pp. 1857–1872, Mar. 2024.
[31] Y. Yao, T. Han, S. Jia, and S. Lyu, "Dense feature interaction network for image inpainting localization," *IEEE Trans. Inf. Forensics Security*, vol. 20, pp. 1636–1648, 2025.
[32] Y. Zhang, G. Zhu, L. Wu, S. Kwong, H. Zhang, and Y. Zhou, "Multi-task SE-network for image splicing localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4828–4840, Jul. 2022.
[33] D. Wang, E. Shelhamer, S. Liu, B. A. Olshausen, and T. Darrell, "Tent: Fully test-time adaptation by entropy minimization," 2020, *arXiv:2006.10726*.
[34] J. Song, K. Park, I. Shin, S. Woo, C. Zhang, and I. S. Kweon, "Test-time adaptation in the dynamic world with compound domain knowledge management," *IEEE Robot. Autom. Lett.*, vol. 8, no. 11, pp. 7583–7590, Nov. 2023.
[35] D. Lee, J. Yoon, and S. Ju Hwang, "BECoTTA: Input-dependent online blending of experts for continual test-time adaptation," 2024, *arXiv:2402.08712*.
[36] Z. Chen, Y. Pan, Y. Ye, M. Lu, and Y. Xia, "Each test image deserves a specific prompt: Continual test-time adaptation for 2D medical image segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 11184–11193.
[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[38] C. Wang and W. Deng, "Representative forgery mining for fake face detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14923–14932.
[39] H. Diao, B. Wan, Y. Zhang, X. Jia, H. Lu, and L. Chen, "UniPT: Universal parallel tuning for transfer learning with efficient parameter and memory," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2024, pp. 28729–28740.
[40] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Dec. 2001, pp. 355–362.
[41] A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, Jan. 2004.
[42] G. Mahfoudi, B. Tajini, F. Retraint, F. Morain-Nicolier, J. L. Dugelay, and M. Pic, "DEFACTO: Image and face manipulation dataset," in *Proc. Eur. Signal Process. Conf.*, 2019, pp. 1–5.
[43] R. Suvorov et al., "Resolution-robust large mask inpainting with Fourier convolutions," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2149–2159.
[44] S. Jia, M. Huang, Z. Zhou, Y. Ju, J. Cai, and S. Lyu, "AutoSplice: A text-prompt manipulated image dataset for media forensics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recogn. Workshop*, Jun. 2023, pp. 893–903.
[45] F. Guillaro, D. Cozzolino, A. Sud, N. Dufour, and L. Verdoliva, "TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2023, pp. 20606–20615.
[46] H. Mareen, D. Karageorgiou, G. V. Wallendael, P. Lambert, and S. Papadopoulos, "TGIF: Text-guided inpainting forgery dataset," in *Proc. IEEE Int. Workshop Inf. Forensics Secur.*, Dec. 2024, pp. 1–6.
[47] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 89–105.
[48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, Jun. 2018.
[49] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[50] Z. Chen, Y. Ye, Y. Pan, and Y. Xia, "Gradient alignment improves test-time adaptation for medical image segmentation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 39, 2025, pp. 2429–2437.
[51] X. Liu, Y. Liu, J. Chen, and X. Liu, "PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 11, pp. 7505–7517, Nov. 2022.
[52] L. Su, X. Ma, X. Zhu, C. Niu, Z. Lei, and J.-Z. Zhou, "Can we get rid of handcrafted feature extractors? SparseViT: Nonsemantics-centered, parameter-efficient image manipulation localization through spare-coding transformer," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2025, pp. 7024–7032.

Long Sun received the B.S. and M.S. degrees in computer science from Harbin Institute of Technology, Harbin, China, in 2021 and 2023, respectively, where he is currently pursuing the Ph.D. degree with the School of Cyberspace Science.

His primary research interests include multimedia security and image processing.

Guopu Zhu (Senior Member, IEEE) received the B.S. degree in transportation from Jilin University, Changchun, China, in 2002, and the M.S. and Ph.D. degrees in control science and engineering from Harbin Institute of Technology, Harbin, China, in 2004 and 2007, respectively.

He is currently a Professor with Harbin Institute of Technology. He has authored or co-authored more than 70 papers in peer-reviewed international journals. His main research areas are multimedia security, image processing, and control theory.

Dr. Zhu serves as an Associate Editor for several journals, including IEEE TRANSACTIONS ON CYBERNETICS, IEEE SYSTEMS JOURNAL, the *Journal of Information Security and Applications*, and *Electronics Letters*.

Hongli Zhang received the B.Sc. degree in computer science from Sichuan University, Chengdu, China, in 1994, and the Ph.D. degree in computer science from Harbin Institute of Technology, Harbin, China, in 1999.

She is currently a Professor with the School of Cyberspace Science, Harbin Institute of Technology, and the Dean of the School. Her research interests include network and computer security, network modeling, and parallel processing.

Xinpeng Zhang (Senior Member, IEEE) received the B.S. degree in computational mathematics from Jilin University, Changchun, China, in 1995, and the M.E. and Ph.D. degrees in communication and information system from Shanghai University, Shanghai, China, in 2001 and 2004, respectively.

Since 2004, he was a Faculty Member with the School of Communication and Information Engineering, Shanghai University, where he is currently a Professor. He is also a Faculty Member with the School of Computer Science, Fudan University, Shanghai. He was with The State University of New York at Binghamton, Binghamton, NY, USA, as a Visiting Scholar from 2010 to 2011, and also with Konstanz University, Konstanz, Germany, as an Experienced Researcher, sponsored by the Alexander von Humboldt Foundation from 2011 to 2012. He has published over 300 papers in these areas. His research interests include multimedia security, AI security, and image processing.

Dr. Zhang was an Associate Editor of IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY from 2014 to 2017.

Yicong Zhou (Senior Member, IEEE) received the B.S. degree in electrical engineering from Hunan University, Changsha, China, in 1992, and the M.S. and Ph.D. degrees in electrical engineering from Tufts University, Medford, MA, USA, in 2006 and 2010, respectively.

He is a Professor with the Department of Computer and Information Science, University of Macau, Macau, China. His research interests include image processing, computer vision, machine learning, and multimedia security.

Dr. Zhou is a fellow of the Society of Photo-Optical Instrumentation Engineers (SPIE) and was recognized as one of "Highly Cited Researchers" in 2020, 2021, 2023, and 2024. He serves as a Senior Associate Editor for IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and an Associate Editor for IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, and IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING.

Ligang Wu (Fellow, IEEE) received the B.S. degree in automation from Harbin University of Science and Technology, Harbin, China, in 2001, and the M.E. degree in navigation guidance and control and the Ph.D. degree in control theory and control engineering from Harbin Institute of Technology, Harbin, in 2003 and 2006, respectively.

He was a Research Associate/a Senior Research Associate with The University of Hong Kong, Hong Kong, the City University of Hong Kong, Hong Kong, and the Imperial College London, London, U.K. He is currently a Professor with Harbin Institute of Technology. He has published seven research monographs and more than 200 research articles in internationally referred journals. His current research interests include analysis and design for cyber-physical systems, robotic and autonomous systems, intelligent systems, and power electronic systems.

Dr. Wu's awards and recognitions include the National Science Fund for Distinguished Young Scholar, the China Young Five-Four Medal, the Distinguished Professor of Chang Jiang Scholar, and the Highly Cited Researcher since 2015. He also serves as an Associate Editor for a number of journals, including IEEE TRANSACTIONS ON AUTOMATIC CONTROL, IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, IEEE/ASME TRANSACTIONS ON MECHATRONICS, and *IET Control Theory and Applications*.