Reinforced Multi-teacher Knowledge Distillation for Efficient General Image Forgery Detection and Localization

Zeqin Yu[1], Jiangqun Ni[2,3]*, Jian Zhang[1], Haoyi Deng[4], Yuzhen Lin[4]

[1]School of Computer Science and Engineering, Sun Yat-sen University
[2]School of Cyber Science and Technology, Sun Yat-sen University
[3]Department of New Networks, Peng Cheng Laboratory
[4]Guangdong Key Laboratory of Intelligent Information Processing, Shenzhen University

Abstract

Image forgery detection and localization (IFDL) is of vital importance as forged images can spread misinformation that poses potential threats to our daily lives. However, previous methods still struggled to effectively handle forged images processed with diverse forgery operations in real-world scenarios. In this paper, we propose a novel Reinforced Multi-teacher Knowledge Distillation (Re-MTKD) framework for the IFDL task, structured around an encoder-decoder **C**onvNeXt-**U**perNet along with **E**dge-Aware Module, named Cue-Net. First, three Cue-Net models are separately trained for the three main types of image forgeries, i.e., copy-move, splicing, and inpainting, which then serve as the multi-teacher models to train the target student model with Cue-Net through self-knowledge distillation. A Reinforced Dynamic Teacher Selection (Re-DTS) strategy is developed to dynamically assign weights to the involved teacher models, which facilitates specific knowledge transfer and enables the student model to effectively learn both the common and specific natures of diverse tampering traces. Extensive experiments demonstrate that, compared with other state-of-the-art methods, the proposed method achieves superior performance on several recently emerged datasets comprised of various kinds of image forgeries.

Introduction

Recent advances in image editing and generative models (Karras, Laine, and Aila 2019; Rombach et al. 2022) not only enhance the quality of image manipulation and synthesis but also simplify their process. However, tampered images generated by such techniques could be abused to deliver misinformation, posing significant threats to social security. Accordingly, it is urgent to explore effective image forensics methods to prevent the abuse of manipulated images.

In response to the above concern, the image forgery detection and localization (IFDL) task aims to identify forged images and localize their tampered regions. Generally, image forgery operations mainly consist of copy-move, splicing, and inpainting. Usually, specific methods are developed for the detection and localization of specific forgery images, which exploits the specifics of forgery trace for image tampering in a specific type, e.g., (Wu, Abd-Almageed, and
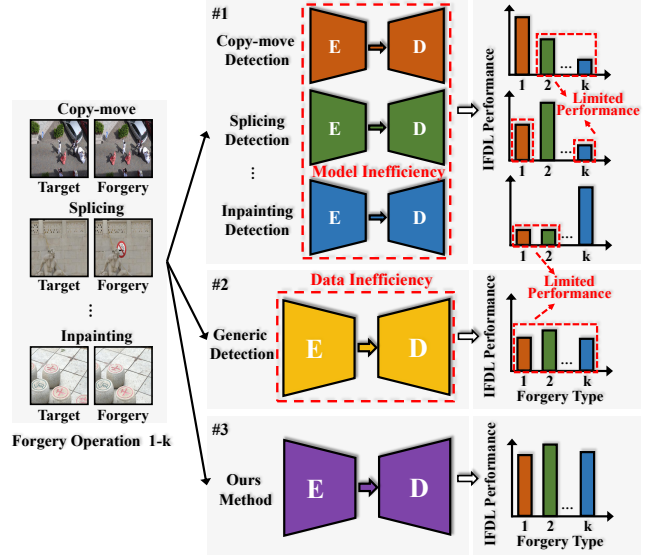
Figure 1: Overview of the existing IFDL methods. Specific IFDL methods are often limited by inefficient models, leading to poor generalization across tampering operations (the first part). Generic IFDL methods are inefficient in exploiting data and difficult to learn both the common and specific tampered features in mixed tamper data (the second part). Our proposed method can achieve promising performance in comprehensive IFDL problems.

Natarajan 2018; Chen et al. 2020) for copy-move, (Bi et al. 2019; Kwon et al. 2022) for splicing and (Li and Huang 2019; Wu and Zhou 2021) for inpainting. Since the training data is limited to the specific forgery, such specific forgery detection methods may overfit and struggle with limited generalization performance. This limitation often results in poor performance, especially when detecting cross-source data, as illustrated in the first part of Fig. 1, where the performance of an inpainting detector significantly degrades when confronted with copy-move or splicing forgeries.

To address the diverse tampering operations, generic forgery detection methods incorporate mixed data including multiple types of tampering. This strategy aims to capture generic tampered features across various forgeries, e.g.,

noise analysis (Bappy et al. 2019; Zhuo et al. 2022; Guillaro et al. 2023), representation learning (Hu et al. 2020; Yu et al. 2024), multi-scale supervision (Dong et al. 2022; Ma et al. 2023) and multi-view feature fusion (Liu et al. 2022; Guo et al. 2023). However, the simultaneous learning of multiple tasks poses its own challenges, as depicted in the second part of Fig. 1. It is challenging to learn both the common and specific natures of diverse tampering traces through simple joint training alone, often leading to performance drops. Additionally, these methods usually entail increased complexity, computational costs, and additional parameters making it hard to meet the practical application scenarios. Therefore, it is crucial to promote a model's ability to discern common traits and specifics among various tampering traces and improve its generalization performance in advancing image forgery detection and localization capabilities.

In this paper, we present a novel Reinforced Multi-teacher Knowledge Distillation (Re-MTKD) framework for the IFDL task. First, we propose the Cue-Net model, built upon a **C**onvNeXt-**U**perNet structure and equipped with a novel **E**dge-Aware Module (EAM). The integration of EAM facilitates forgery artifact detection by fusing low-level and high-level features. To effectively train Cue-Net, we introduce a novel knowledge distillation strategy that utilizes multiple teacher models and activates them during training through a reinforcement learning strategy, named Reinforced Dynamic Teacher Selection (Re-DTS). Specifically, we first train multiple teacher models on different datasets, each focused on a specific type of forgery. By incorporating the Re-DTS strategy, these well-trained teacher models are dynamically selected based on the type of tampering data for knowledge distillation. This approach allows the knowledge from teacher models pretrained on different forgery types to be efficiently transferred to a student model, reducing the risk of overfitting to any specific type of forgery trace. Our contributions to this work are as follows:

- We propose a Reinforced Multi-teacher Knowledge Distillation (Re-MTKD) framework, which consists of ConvNeXt-UPerNet structure with a novel Edge-Aware Module (EAM), which fuses low-level and high-level features, significantly enhancing the ability to detect tampering traces.

- We propose a Reinforced Dynamic Teacher Selection (Re-DTS) strategy to maximize the forgery detection and localization capability of the student model by dynamically selecting teacher models that excel in specific tamper forensic tasks for dedicated knowledge transfer, enabling the student model to learn the commonalities and specifics of multiple tamper traces.

- Comprehensive experiments show that the proposed method achieves superior performances on several recently multiple tampering types of datasets compared with other state-of-the-art methods.

Related Work
Image Forgery Detection and Localization
There are several algorithms for IFDL tasks, including specific forgery detection for copy-move, splicing, inpainting,

and generic forgery detection strategies.

Specific forgery detection. Specific IFDL methods focus on identifying specific tampering traces for specific types of tampered data. For *copy-move detection*, BusterNet (Wu, Abd-Almageed, and Natarajan 2018) and CMSD-STRD (Chen et al. 2020) were designed to localize source/target regions in copy-move tampered images in parallel and in series, respectively. For *splicing detection*, MFCN (Salloum, Ren, and Kuo 2018) focuses on highlighting tampered edges, RRU-Net (Bi et al. 2019) emphasizes residual artifacts, and CAT-Net (Kwon et al. 2022) analyzes compression artifacts in RGB and DCT domains. For *inpainting detection*, HP-FCN (Li and Huang 2019) uses high-pass filtering to enhance inpainting traces, IID-Net (Wu and Zhou 2021) employs neural architecture search for automatic feature extraction, and TLTF-LEFF (Li et al. 2023) introduces a local enhancement transformer architecture. While these methods achieve excellent performance on specific forgery types, they often struggle with other types of tampered data, leading to inefficient model generalization.

Generic forgery detection. With the growth of massive data, recent generic IFDL methods aim to learn the commonalities of various tampering tasks. H-LSTM (Bappy et al. 2019), SATL-Net (Zhuo et al. 2022), CFL-Net (Niloy, Bhaumik, and Woo 2023) and TruFor (Guillaro et al. 2023) integrate spatial and noise domains to detect diverse manipulations. SPAN (Hu et al. 2020) and DiffForensics (Yu et al. 2024) leverage self-supervised learning to identify subtle tampering traces. MVSS-Net (Dong et al. 2022) and IML-Vit (Ma et al. 2023) emphasize multi-scale supervision, while PSCC-Net (Liu et al. 2022) and HiFi-Net (Guo et al. 2023) focus on multi-view feature fusion. Although these methods achieve promising results across various forgery operations, they often suffer performance degradation in joint training due to task incompatibilities and often introduce extra complexity and computational cost.

Knowledge Distillation
Initial knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015) transfers knowledge from a large teacher model to a smaller student model via a teacher-student architecture. This concept was extended by (Zhang et al. 2019) into self-distillation, where the same network serves as both teacher and student, allowing the student to learn from itself. For IFDL tasks, (Yu et al. 2023) proposes a fixed-weight multi-teacher self-distillation strategy to handle five tampering operations in smartphone screenshot images, enabling the student model to learn multiple tampering traces. However, assigning fixed weights to each teacher model may not be optimal, as it prevents dynamic adaptation to different data batches, which can contain varied tampering types, potentially limiting the IFDL performance of the student model.

Reinforcement Learning
Reinforcement learning has shown promising results in areas such as autonomous driving (Sinha et al. 2020), smart games (Silver et al. 2016), and recommendation systems (Feng et al. 2018; Yuan et al. 2021). Its core idea is to maximize long-term rewards by enabling an agent to learn
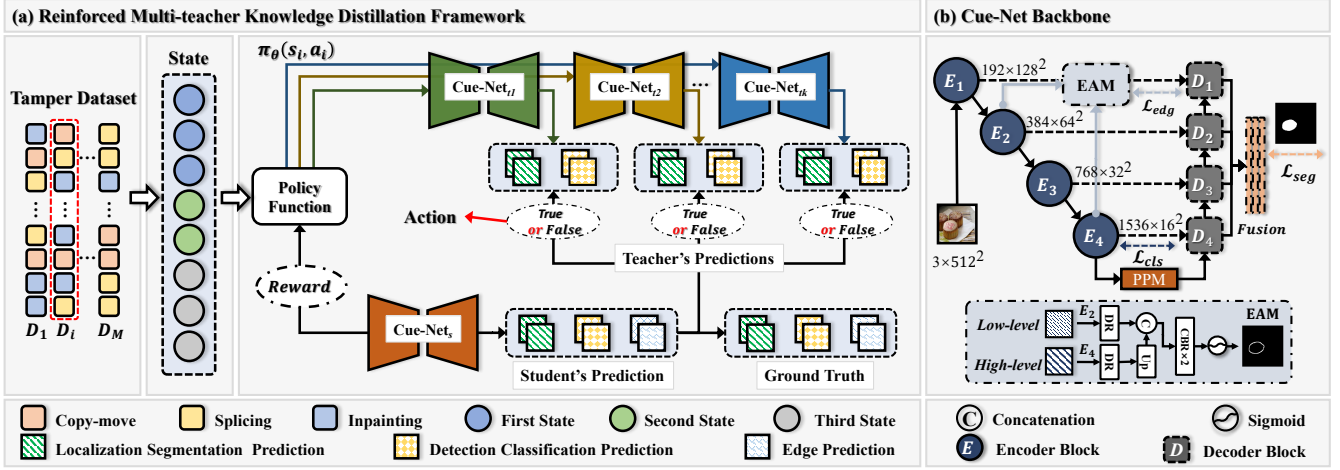
Figure 2: We propose a novel Reinforced Multi-teacher Knowledge Distillation framework, structured by the simple yet effective Cue-Net backbone, for the IFDL task. Within this framework, the proposed Re-DTS strategy dynamically selects teacher models based on different tampering types of data, guiding the student model to effectively learn various tampering traces.

from interactions with the environment and adjust its behavior based on reward signals. In this paper, we introduce a novel Re-MTKD framework that incorporates these principles. Central to this framework is the Re-DTS strategy, dynamically selecting the most appropriate teacher models to transfer specialized knowledge to the student model. This strategy enhances the student model's ability to handle various tampering traces and improves IFDL performance.

Proposed Re-MTKD Framework

Cue-Net Backbone

Pipeline. The architecture of Cue-Net is shown in Fig. 2(b), which is an encoder-decoder structure. For the encoder, assuming the input image is $x \in \mathbb{R}^{h \times w \times c}$, feature extraction is first performed by ConvNeXt v2 (Woo et al. 2023) and feature maps $\{E_j\}_{j=1}^4$ are obtained from low to high levels. Additionally, E_2 and E_4 are used as inputs to the EAM proposed in this paper, which will be presented in the next subsection. The decoder employs UPerNet (Xiao et al. 2018), comprising two components: the Pyramid Pooling Module (PPM) (Zhao et al. 2017) and the Feature Pyramid Network (FPN) (Lin et al. 2017). The FPN outputs feature maps $\{D_j\}_{j=1}^4$, which are resized to their original dimensions using bilinear interpolation and then merged to obtain the localization segmentation result after applying the sigmoid activation function.

Edge-Aware Module. To further enhance fine-grained tampered trace extraction, we design a simple yet effective edge-aware module (EAM) that accurately extracts the edge information of tampered regions for subsequent guidance and fusion. Tampered region edges usually consist of only a few pixels, making it difficult to achieve accurate edge supervision using only high-level feature maps with the lowest resolution. Conversely, using only low-level feature maps with high resolution makes it hard to sense the tampered area. In EAM, we perform edge supervision on multi-level features,

which is derived by fusing low-level feature E_2 and high-level feature E_4 in the encoder, shown as follows,

$$f^e = \sigma(CBR(Cat(DR(E_2), Up(DR(E_4))))), \quad (1)$$

where $\sigma(\cdot)$ is sigmoid normalization and the CBR blocks (Conv + BN + ReLU) implement the fusion of low-level and high-level features, $Cat(\cdot)$ forms a concatenation operation, DR stands for dimension reduction using the CBR blocks and Up for upsampling. Later, the obtained edge prediction map f^e and edge label y^e are used for loss iteration.

Loss function. Following the previous IFDL methods (Liu et al. 2022; Dong et al. 2022; Guo et al. 2023), our method includes three types of supervision: localization segmentation (pixel-level) supervision $\mathcal{L}_{seg}$ and detection classification (image-level) supervision $\mathcal{L}_{cls}$ as standard, and edge detection supervision $\mathcal{L}_{edg}$ as a special design.

For localization segmentation supervision $\mathcal{L}_{seg}$, following (Guillaro et al. 2023), we use a combination of weighted ℓ_{wbce} and ℓ_{dice} (Milletari, Navab, and Ahmadi 2016).

$$\mathcal{L}_{seg} = \lambda_0^s \ell_{wbce}(f^s, y^s) + (1 - \lambda_0^s) \ell_{dice}(f^s, y^s), \quad (2)$$

where λ_0^s is the segmentation balance weight, f^s and y^s are the localization results and localization labels, respectively.

Following ConvNeXt v2 (Woo et al. 2023), we design the detection classification loss $\mathcal{L}_{cls}$ on E_4 using standard ℓ_{bce},

$$\mathcal{L}_{cls} = \ell_{bce}(f^c, y^c), \quad (3)$$

where f^c and y^c are the detection classification results and classification labels, respectively.

For edge supervision $\mathcal{L}_{edg}$ with EAM, we use ℓ_{dice} to better focus on tiny edge regions,

$$\mathcal{L}_{edg} = \ell_{dice}(f^e, y^e). \quad (4)$$

For the final combined loss, which is also the "hard" loss $\mathcal{L}_{\mathbf{hard}}$, weighting from three perspectives yields our nearest supervised loss function,

$$\mathcal{L}_{\mathbf{hard}} = \alpha \cdot \mathcal{L}_{seg} + \beta \cdot (\mathcal{L}_{cls} + \mathcal{L}_{edg}), \quad (5)$$

where $\alpha, \beta \in [0, 1]$. The term "hard" indicates the binary label for each pixel is used for supervision.

Reinforced Dynamic Teacher Selection Strategy

Before introducing the Re-DTS strategy, we will first formulate single-teacher KD. To better improve both the performance of localization segmentation and detection classification of the student model, we choose to transfer specialized knowledge through "soft" loss $\mathcal{L}_{\textbf{soft}}$ shown as follows,

$$\mathcal{L}_{\textbf{soft}} = \mathcal{L}_{seg}(f_{\textbf{s}}^s, f_{\textbf{t}}^s) + \mathcal{L}_{cls}(f_{\textbf{s}}^c, f_{\textbf{t}}^c), \qquad (6)$$

where bolded $\textbf{t}$ and $\textbf{s}$ denote the teacher model and student model, respectively.

Figure 2(a) shows the overview of our Re-DTS strategy. The elements (i.e., *state*, *action*, and *reward*) of reinforcement learning are introduced as follows. To be clear, since all the teacher models share the same training process, we will introduce only one of them and the three teacher models are denoted as $\{\Theta_k^t\}_{k=1}^3$.

State. Our Re-DTS strategy contains a series of environment states, s_1, s_2, ..., which summarise the features of the tampered image to be detected, student model and the candidate teacher model so that reasonable decisions can be made. To formulate the state s_i, we design a state feature vector $\mathbf{F}(s_i)$ containing a concatenation of three features.

The first feature is the vector representation $\mathcal{R}(x_i) \in \mathbb{R}^d$ of i-th batch image x_i, which could be generated by any IFDL representation model Θ^r. In this paper, we obtain the output of the last layer of the decoder D_4 as the feature $\mathcal{R}(x_i)$ by feeding this batch image x_i to Cue-Net, which has not been trained on any IFDL task data.

The second feature $\mathcal{S}(x_i) \in \mathbb{R}^d$ is obtained similarly, except the feature vector is output from the student model Θ^s, trained in real-time on the IFDL task data and designed to effectively represent the content of the tampered image.

The third feature $\mathcal{T}_k(x_i) \in \mathbb{R}^{1+1+1}$, consisting of the localization segmentation prediction result $f_{\textbf{t}k}^s$, the detection classification prediction result $f_{\textbf{t}k}^c$ and the tampering edge prediction result $f_{\textbf{t}k}^e$ of the teacher model Θ_k^t, respectively, are expected to guide the better action execution of the policy network.

$$\mathcal{T}_k(x_i) = Cat(Max(f_{\textbf{t}k}^s), f_{\textbf{t}k}^c, Max(f_{\textbf{t}k}^e)), \qquad (7)$$

where $Max(\cdot)$ is the maxpooling operation.

Therefore, the state feature vector $\mathbf{F}(s_i) \in \mathbb{R}^{2d+3}$ can be formulated as follows:

$$\mathbf{F}(s_i) = Cat(\mathcal{R}(x_i), \mathcal{S}(x_i), \mathcal{T}_k(x_i)), \qquad (8)$$

where such concatenation ensures that the state feature vector $\mathbf{F}(s_i)$ comprehensively reflects the image and model characteristics, enabling more informed and effective decisions by the policy network.

Action. Each teacher model Θ^t is associated with a policy network, and actions $a_i \in \{0, 1\}$ are defined to indicate whether the policy network will be chosen to transfer the specialized knowledge of the k-th teacher model Θ_k^t to the student model Θ^s. We sample the value of a_i using the policy function $\pi_\theta(s_i, a_i)$, where θ is the parameter to be learned. In this paper, we adopt a logistic function as the policy function:

$$
\begin{aligned}
\pi_\theta(s_i, a_i) &= P_\theta(a_i \mid s_i) \\
&= a_i \sigma(\mathbf{W} * \mathbf{F}(s_i) + b) \qquad (9) \\
&\quad + (1 - a_i)(1 - \sigma(\mathbf{W} * \mathbf{F}(s_i) + b)),
\end{aligned}
$$

Algorithm 1: Training Procedure for Re-MTKD

1 Pre-train teacher models Θ_k^t using the corresponding type of tampered data $\mathcal{D}^k$.
2 Pre-train the policy network θ_k by calculating the return under all teacher models Θ_k^t selected.
3 Run Algorithm 2 to jointly train the student model Θ^s and the policy network θ_k until convergence.

Algorithm 2: Re-DTS strategy

Input: Epoch number L. Batch-size number B. Training data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M\}$. k-th teacher model and policy network initialized as $\Theta^t = \Theta_k^t$ and $\theta = \theta_k$.

1 **for** *epoch* $l = 1$ *to* L **do**
2   Shuffle $\mathcal{D}$ to obtain a new training sequence.
3   **for** *each batch* $\mathcal{D}_i \in \mathcal{D}$ **do**
4   Sample teacher selection actions for each batch image $x_i \in \mathcal{D}_i$ with θ by: $a_i \sim \pi_\theta(s_i, a_i)$.
5   Stored (a_i, s_i) to the episode history $\mathcal{H}$.
6   Compute the soft labels of the selected teachers $\mathcal{K}$: $\mathcal{L}_{\textbf{soft}} = \sum_{k \in \mathcal{K}} \mathcal{L}_{\textbf{soft}}^{(k)}$
7   Update the parameter Θ^s of student model by: $\mathcal{L} = \mathcal{L}_{\textbf{hard}} + \omega \mathcal{L}_{\textbf{soft}}$
8   **if** $i \bmod 10B$ **then**
9   **for** *each* $(a_i, s_i) \in \mathcal{H}$ **do**
10   Compute delayed reward following Eq. 10.
11   Update the parameter θ of the policy function following Eq. 11.
12   **end**
13   **end**
14   **end**
15 **end**

---

where $\mathbf{W} \in \mathbb{R}^{2d+3}$ and $b \in \mathbb{R}^1$ are the trainable parameters.

Reward. Reasonable rewards can instruct the teacher model to distill a high-performing student model. For training data $\mathcal{D} = \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_M\}$, with M is the number of batches per epoch, we construct a feature vector s_{ik} for each teacher model Θ_k^t and sample action a_{ik} for the i-th batch of data $\mathcal{D}_i$ according to the policy function $\pi_\theta(s_{ik}, a_{ik})$ to determine whether the action is selected or not. For all sampled teacher models, we integrate the loss $\mathcal{L}_{\textbf{soft}}$ of all teacher models with the loss $\mathcal{L}_{\textbf{hard}}$ to update the student model parameters Θ^s.

We propose three reward calculations. The first is the "hard" loss $\mathcal{L}_{\textbf{hard}}$, comprising localization segmentation loss $\mathcal{L}_{seg}$, detection classification loss $\mathcal{L}_{cls}$ and tampering edge detection loss $\mathcal{L}_{edg}$. The second reward function combines the "hard" loss $\mathcal{L}_{\textbf{hard}}$ as well as the "soft" loss $\mathcal{L}_{\textbf{soft}}$ of the teacher model. Finally, to enhance the performance of the student model on the IFDL task, we use the localization segmentation F1 score and detection classification accuracy score of the student model during training as the third re-

	Methods	Copy-move			Splicing			Inpainting			Multi			Average		
		Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC	Acc	F1	AUC
D-Com	DoaGan*	0.695	0.730	0.756	0.463	0.432	0.516	0.383	0.351	0.522	0.487	0.406	0.560	0.507	0.479	0.588
D-Com	Buster-Net*	0.557	0.715	0.500	0.638	0.779	0.500	0.726	0.841	0.500	0.613	0.760	0.500	0.634	0.774	0.500
D-Com	CMSD-STRD	0.472	0.332	0.502	0.475	0.445	0.530	0.517	0.585	0.558	0.472	0.478	0.494	0.484	0.460	0.521
D-Spl	MFCN	0.529	0.692	0.422	0.637	0.779	0.586	0.725	0.841	0.615	0.621	0.764	0.565	0.628	0.769	0.547
D-Spl	RRU-Net	0.828	0.865	0.863	0.818	0.873	0.872	0.824	0.890	0.899	0.612	0.756	0.624	0.770	0.846	0.815
D-Spl	CAT-Net	0.719	0.758	0.777	0.805	0.861	0.917	0.831	0.888	0.887	0.611	0.730	0.632	0.742	0.809	0.803
D-Inp	HP-FCN	0.557	0.715	0.540	0.636	0.777	0.566	0.726	0.841	0.571	0.613	0.760	0.703	0.633	0.774	0.595
D-Inp	IID-Net	0.505	0.298	0.809	0.648	0.642	0.888	0.670	0.748	0.748	0.526	0.473	0.639	0.587	0.540	0.771
D-Inp	TLTF-LEFF	0.446	0.010	0.755	0.386	0.078	0.808	0.474	0.433	0.790	0.392	0.025	0.646	0.425	0.136	0.750
D-Generic	H-LSTM	0.557	0.715	0.501	0.638	0.779	0.516	0.726	0.841	0.517	0.613	0.760	0.507	0.634	0.774	0.510
D-Generic	SPAN	0.557	0.715	0.500	0.638	0.779	0.500	0.726	0.841	0.500	0.613	0.760	0.500	0.634	0.774	0.500
D-Generic	MVSS-Net	0.598	0.494	0.630	0.843	0.862	0.872	0.835	0.882	0.833	0.641	0.638	0.681	0.729	0.719	0.754
D-Generic	SATL-Net	0.713	0.717	0.815	0.854	0.881	0.934	0.867	0.909	0.920	0.634	0.694	0.666	0.767	0.800	0.834
D-Generic	PSCC-Net	0.473	0.614	0.467	0.657	0.785	0.721	0.706	0.827	0.537	0.615	0.761	0.610	0.613	0.747	0.584
D-Generic	HiFi-Net	0.557	0.715	0.500	0.638	0.779	0.500	0.726	0.841	0.500	0.613	0.760	0.500	0.634	0.774	0.500
D-Generic	IML-Vit	0.675	0.657	0.817	0.869	0.899	0.933	0.774	0.837	0.831	0.639	0.707	0.681	0.739	0.775	0.256
	Re-MTKD (Ours)	**0.888**	**0.900**	**0.952**	**0.893**	**0.918**	**0.970**	**0.973**	**0.981**	**0.994**	**0.703**	**0.778**	**0.790**	**0.864**	**0.894**	**0.927**

Table 1: Image-level Acc, F1 under 0.5 threshold, and AUC scores for image forgery detection. The **best** and 2nd-best results are highlighted. Methods with * use the original paper's pre-trained model, others keep the same training data as our method.

ward. In summary, we have

$$reward_1 = -\mathcal{L}_{\textbf{hard}}$$
$$reward_2 = -(\mathcal{L}_{\textbf{hard}} + \mathcal{L}_{\textbf{soft}})$$
$$reward_3 = -\gamma(\mathcal{L}_{\textbf{hard}} + \mathcal{L}_{\textbf{soft}}) \quad (10)$$
$$+ (1 - \gamma)((F1_{seg} + Acc_{cls})on\mathcal{D}_i),$$

where γ is a hyperparameter to balance the different rewards. We ultimately take the third reward as the default option. Notably, following REINFORCE (Williams 1992), where updates are not rewarded immediately after taking each step, we delayed updates until after $10B$ batches of training, where B notes the batch size.

Optimization. According to the policy gradient theorem (Sutton et al. 1999) and the REINFORCE algorithm (Williams 1992), we compute the gradient to update the current policy as follows:

$$\theta \leftarrow \theta + \xi \sum_i r \sum_k \nabla_\theta \pi_\theta(s_{ik}, a_{ik}), \quad (11)$$

where r is the reward function defined in Eq. 10, and ξ is the learning rate.

Model Training

Algorithm 1 outlines the training procedure for the Re-MTKD framework. We pre-train the model before starting joint training. Teacher models Θ_k^t are pre-trained with specialized knowledge on the corresponding type of tampered data $\mathcal{D}^k$. Then, the policy network θ_k is initialized by selecting feedback from all teacher models during the KD process.

After initialization, the Re-DTS strategy is executed to optimize the student model according to $\mathcal{L}$, as shown in Algorithm 2. In this process, we first fix the policy network θ_k to train the student models Θ^s. After a pre-determined number of iterations, we fix Θ^s, compute the return content, and optimize the teacher selection policy network θ_k. This process is repeated until all training iterations are completed.

Experiments

Experimental Setup

Dataset. Considering the availability and generality, we select ten challenging benchmark datasets to evaluate our method, covering tampering types: copy-move (Com), splicing (Spl), inpainting (Inp), and multi-tampering (Multi). Details of these datasets are provided in the Appendix.

1) For Com, we use CASIA v2 (Dong, Wang, and Tan 2013) and Tampered Coco (Liu et al. 2022) for training and CASIA v1+ (Dong, Wang, and Tan 2013), NIST16 (Nimble 2016) and Coverage (Wen et al. 2016) for testing.

2) For Spl, we use CASIA v2 (Dong, Wang, and Tan 2013) and Fantastic-Reality (Kniaz, Knyaz, and Remondino 2019) for training, and CASIA v1+ (Dong, Wang, and Tan 2013), NIST16 (Nimble 2016), Columbia (Hsu and Chang 2006) and DSO-1 (De Carvalho et al. 2013) for testing.

3) For Inp, we use GC Dresden&Places (Wu and Zhou 2021) for training, and NIST16 (Nimble 2016), AutoSplicing (Jia et al. 2023) and DiverseInp (Wu and Zhou 2021) for testing.

4) For Multi, we use IFC (Challenge 2013), Korus (De Carvalho et al. 2013), and IMD2020 (Novozamsky, Mahdian, and Saic 2020) for testing, as this data involves composite operations of the single tampering types described above.

For the Cue-Net student model Θ^s and the other SOTA methods, training is conducted on all mixed single-tampered data. In contrast, during reinforced multi-teacher knowledge distillation, each teacher model Θ^t is trained on the corresponding type of tampered data.

Implementation details. We use a single Nvidia Tesla A100 GPU (80 GB memory) to conduct experiments on the PyTorch deep learning framework. The parameter configurations for reinforced multi-teacher KD are as follows:

1) For single-teacher model pre-training, we resize the input image to 512×512 and apply the AdamW optimizer.

	Methods	Copy-move			Splicing			Inpainting			Multi			Average		
		F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC	F1	IoU	AUC
D-Com	DoaGan*	0.249	0.169	<u>0.785</u>	0.027	0.021	0.550	0.006	0.004	0.561	0.020	0.013	0.569	0.076	0.052	0.616
	Buster-Net*	<u>0.274</u>	0.185	**0.806**	0.124	0.083	0.715	0.048	0.028	0.666	0.077	0.048	0.705	0.131	0.086	0.723
	CMSD-STRD	0.018	0.011	0.507	0.037	0.023	0.505	0.051	0.032	0.514	0.018	0.011	0.504	0.031	0.019	0.508
D-Spl	MFCN	0.141	0.082	0.634	0.256	0.161	0.707	0.440	0.327	0.735	0.154	0.091	0.669	0.248	0.165	0.686
	RRU-Net	0.165	0.113	0.618	0.454	0.378	0.806	<u>0.604</u>	<u>0.521</u>	<u>0.885</u>	0.232	0.178	0.687	0.364	0.298	0.749
	CAT-Net	0.270	**0.224**	0.746	<u>0.622</u>	<u>0.560</u>	0.877	0.429	0.345	0.817	0.240	0.190	0.717	<u>0.390</u>	<u>0.330</u>	0.789
D-Inp	HP-FCN	0.012	0.007	0.685	0.013	0.007	0.699	0.008	0.004	0.713	0.058	0.033	0.728	0.023	0.013	0.706
	IID-Net	0.022	0.016	0.648	0.157	0.124	0.795	0.273	0.214	0.858	0.043	0.031	0.703	0.124	0.096	0.751
	TLTF-LEFF	0.000	0.000	0.615	0.010	0.007	0.713	0.157	0.132	0.723	0.037	0.003	0.711	0.043	0.035	0.691
D-Generic	H-LSTM	0.143	0.081	0.602	0.199	0.120	0.617	0.344	0.229	0.656	0.134	0.078	0.604	0.205	0.127	0.620
	SPAN	0.052	0.032	0.506	0.098	0.063	0.681	0.455	0.350	0.736	0.137	0.092	0.677	0.186	0.134	0.650
	MVSS-Net	0.165	0.137	0.584	0.521	0.464	0.759	0.571	0.488	0.769	0.241	<u>0.198</u>	0.630	0.374	0.322	0.686
	SATL-Net	0.073	0.050	0.606	0.284	0.221	0.738	0.328	0.241	0.804	0.078	0.054	0.616	0.191	0.142	0.691
	PSCC-Net	0.105	0.066	0.612	0.330	0.241	0.745	0.368	0.274	0.579	<u>0.278</u>	0.194	0.766	0.270	0.194	0.676
	HiFi-Net	0.161	0.093	0.642	0.233	0.145	0.599	0.210	0.138	0.587	0.139	0.084	0.577	0.186	0.115	0.601
	IML-Vit	0.160	0.136	0.773	0.498	0.446	<u>0.903</u>	0.215	0.166	0.795	0.130	0.100	<u>0.772</u>	0.251	0.212	<u>0.810</u>
	Re-MTKD (Ours)	**0.277**	<u>0.221</u>	0.764	**0.676**	**0.606**	**0.934**	**0.789**	**0.714**	**0.965**	**0.444**	**0.372**	**0.858**	**0.531**	**0.468**	**0.861**

Table 2: Pixel-level F1, IoU under 0.5 threshold, and AUC scores for image forgery localization. Same highlighting conventions and conditions as in Table 1 apply.

We set the training hyper-parameters by the learning rate as 1×10^{-4}, the batch size as 24, and the epoch as 50. To balance the performance of forgery detection and localization, we set the weight of forgery localization $\mathcal{L}_{seg}$ to $\alpha = 1$ and λ_0^s is 0.1. The weight β of the detection classification supervision $\mathcal{L}_{cls}$ and edge supervision $\mathcal{L}_{edg}$ is set to 0.2.

2) For reinforced multi-teacher KD, we use the same parameter settings for input image size, optimizer, learning rate, and batch size as for the single-teacher model pre-training but with more training data and only 25 epochs at this training stage. For γ in $reward_3$, we set it to 0.2. For $\mathcal{L}$, we set the constraint factor $\omega = 0.05$ for the overall teacher loss $\mathcal{L}_{\mathbf{soft}}$, and this constraint factor is multiplied by the number of selected teachers to balance the process of teacher model knowledge transfer and student model self-learning.

3) For the policy network, the learning rate is set to 3×10^{-4} and adjusted using CosineAnnealingLR with the Adam optimizer.

Comparison with the State-of-the-Art Methods

For a fair comparison, we focus on methods with available codes or pre-trained models trained on datasets different from the test datasets. We compare methods targeting specific forgery types and generic forgery detection as follows: DoaGan (Islam et al. 2020), Buster-Net (Wu, Abd-Almageed, and Natarajan 2018) and CMSD-STRD (Chen et al. 2020) are designed for copy-move detection. MFCN (Salloum, Ren, and Kuo 2018), RRU-Net (Bi et al. 2019) and CAT-Net (Kwon et al. 2022) are designed for splicing detection. HP-FCN (Li and Huang 2019), IID-Net (Wu and Zhou 2021) and TLTF-LEFF (Li et al. 2023) are designed for inpainting detection. H-LSTM (Bappy et al. 2019), SPAN (Hu et al. 2020), MVSS-Net (Dong et al. 2022), SATL-Net (Zhuo et al. 2022), PSCC-Net (Liu et al. 2022), HiFi-Net (Guo et al. 2023) and IML-Vit (Ma et al. 2023) are designed for generic forgery detection.

Detection evaluation. Table 1 shows the forgery detection performance. We observe that many methods perform very poorly, with the AUC of their performance approaching 0.5, i.e., close to random guessing, e.g., Buster-Net, H-LSTM, SPAN, and HiFi-Net. Thanks to the proposed Re-MTKD framework, which includes Cue-Net and the Re-DTS strategy, our method achieves SOTA performance across all tampering types of datasets. Especially on the multi-tampering dataset, which has more complex types of tampered data, all compared methods perform poorly, but our method is 8.7% ahead of the second place in the AUC score.

Localization evaluation. Table 2 shows the forgery localization performance. Notably, some of the forgery-specific detection methods show competitiveness on the corresponding data, e.g., Buster-Net achieves the optimal AUC scores on copy-move data, CAT-Net achieves the second-best performance on splicing data, and IID-Net achieves competitive AUC scores on inpainting data. Nevertheless, these methods often struggle when applied to other types of forgeries or multi-tampering data, resulting in noticeable performance drops. Interestingly, few generic forgery detection methods consistently demonstrate strong efficacy across both specific and multi-tampering types, underscoring the challenge of learning multiple tampering traces simultaneously. In contrast, our method delivers superior results across all datasets and significantly outperforms the others on average, highlighting its ability to effectively capture both the commonalities and specifics of multiple tamper races.

Ablation Study

This subsection primarily analyzes the effectiveness of key components of the Re-MTKD framework. Table 3 presents the ablation results for the Re-DTS strategy, along with additional strategies for assigning teacher weights. Further details on ablation experiments are provided in the Appendix.
Effectiveness of Teacher Models in KD. Comparing Se-

KD Strategy	Detection				Localization				Average
	Copy-move	Splicing	Inpainting	Multi	Copy-move	Splicing	Inpainting	Multi	
0: -	0.873	0.881	0.794	0.764	0.276	0.611	0.530	0.419	0.643
1: Single-Com	0.884	0.906	0.767	0.721	0.261	**0.678**	0.510	0.436	0.645
2: Single-Spl	0.721	0.883	0.888	0.699	0.264	0.637	0.589	0.390	0.634
3: Single-Inp	0.886	0.901	0.914	0.696	0.259	0.652	0.721	0.372	0.675
4: U-Ensemble	0.880	0.890	0.953	0.741	0.198	0.635	0.600	0.392	0.661
5: Ours ($reward_1 + \mathcal{L}_{\mathbf{soft3}}$)	0.888	0.907	0.943	0.738	0.222	0.632	0.750	0.412	0.686
6: Ours ($reward_2 + \mathcal{L}_{\mathbf{soft3}}$)	0.890	**0.923**	0.949	0.752	0.235	0.645	0.751	0.426	0.696
7: Ours ($reward_3 + \mathcal{L}_{\mathbf{soft1}}$)	0.890	0.891	0.919	0.716	0.257	0.634	0.729	0.432	0.684
8: Ours ($reward_3 + \mathcal{L}_{\mathbf{soft2}}$)	0.865	0.830	0.957	0.694	0.210	0.639	0.683	0.398	0.660
9: Ours ($reward_3 + \mathcal{L}_{\mathbf{soft3}}$)	**0.900**	0.918	**0.981**	**0.778**	**0.277**	0.676	**0.789**	**0.444**	**0.720**

Table 3: F1 performance of various KD strategies on all test sets. (1) **"-"**: Baseline, only the student model Cue-Net is directly trained on task label data without any soft target labels from teacher models. (2) **Single-Com\Spl\Inp:** Single forgery-specific detection teacher model for KD. (3) **U-Ensemble** (Yu et al. 2023): Equal weight assignment to all teacher models, with the student model receiving their average output. (4) **Ours ($reward$):** Our Re-DTS strategy, we compare three different reward functions, as detailed in Eq. 10. (5) **Ours ($\mathcal{L}_{\mathbf{soft}}$):** Our Re-DTS strategy, we compare the different "soft" loss transfer by teacher models in KD, $\mathcal{L}_{\mathbf{soft1}} = \mathcal{L}_{seg}$, $\mathcal{L}_{\mathbf{soft2}} = \mathcal{L}_{cls}$, $\mathcal{L}_{\mathbf{soft3}} = \mathcal{L}_{seg} + \mathcal{L}_{cls}$.

tups #0 - #3, it can be observed that the simple Cue-Net student model struggles to cope with a wide range of tampering attacks. In contrast, a forgery-specific detection teacher can improve the performance of the student model on the corresponding data, e.g., **Singe-Inp** achieves 14% and 19% improvement in forgery detection and forgery localization $F1$ performance on inpainting data compared to Setups #0, respectively. Comparing Setups #0 and #4, **U-Ensemble** provides some performance improvement on the IFDL task, but suffers from performance drops in some manipulation types, e.g., copy-move and multi-tampering. This indicates that averaging weights across multiple teachers fails to adequately capture the distinct commonalities of different tampering operations and may not represent the most effective strategy.

Effectiveness of Re-DTS Strategy. As shown by the last five setups, we compare the effects of different $rewards$ and various teacher knowledge transfer strategies $\mathcal{L}_{\mathbf{soft}}$ in the Re-DTS proposed in this paper. **(i)** $rewards$: Comparing Setups #5, #6, and #9, the performance of the student model can be better improved by adding a reward as the supervision of the knowledge transfer from the teacher model to the student model. For Setup #9, the final reward ($reward_3$) adopted in this paper, the total rewards for the "soft" loss of the teacher model $\mathcal{L}_{\mathbf{soft}}$ and the "hard" loss $\mathcal{L}_{\mathbf{hard}}$, as well as the performance metrics of the student model on forgery localization $F1$ and forgery detection Acc, effectively contribute to the overall performance of the student model. **(ii)** $\mathcal{L}_{\mathbf{soft}}$: Comparing Setups #7, #8 and #9, it can be seen that the combination of specialized knowledge on forgery detection and localization transferred by the teacher model improves the IFDL performance of the student model more efficiently. Our method (Setup #9) ultimately achieves excellent results not only on multiple specific forgery data, but also on the more challenging multi-tampering data, where equally efficient performance is achieved.

Furthermore, we show the embedding space of learned features with t-SNE (Van der Maaten and Hinton 2008) visualization of different KD strategies in Fig. 3. We observe that in our proposed Re-MTKD framework, Cue-Net, when

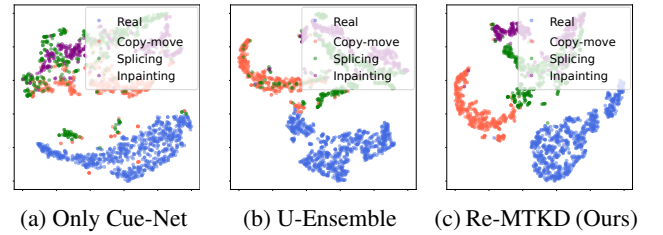(a) Only Cue-Net (b) U-Ensemble (c) Re-MTKD (Ours)

Figure 3: Feature space visualization of different KD strategies on CASIA v1+ and DiverseInp.

trained with the Re-DTS strategy, effectively discriminates between the feature distributions of real and tampered samples, outperforming other KD strategies. This demonstrates the model's ability to learn common features across various tampering types, enabling it to accurately classify all tampered samples as tampered, irrespective of the specific tampering technique. Additionally, the model also captures the specific features of each tampering type, resulting in more clustered and distinct distributions for each category of tampered samples.

Conclusion

In this paper, we propose a novel Reinforced Multi-teacher Knowledge Distillation (Re-MTKD) framework designed for image forgery detection and localization (IFDL). Specifically, we develop a new network called Cue-Net, featuring a ConvNeXt-UPerNet structure equipped with an Edge-Aware Module, which serves as an effective backbone for the IFDL task. We further introduce a Reinforced Dynamic Teacher Selection (Re-DTS) strategy, which dynamically selects specialized teacher models based on different types of tampering data, guiding the student model to effectively learn both the commonalities and specifics of various tampering traces. Extensive experiments across several IFDL tasks demonstrate the superior performance of our proposed method compared to existing state-of-the-art approaches.

References

Bappy, J. H.; Simons, C.; Nataraj, L.; Manjunath, B.; and Roy-Chowdhury, A. K. 2019. Hybrid lstm and encoder–decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 28(7): 3286–3300.

Bi, X.; Wei, Y.; Xiao, B.; and Li, W. 2019. RRU-Net: The ringed residual U-Net for image splicing forgery detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 0–0.

Challenge, T. F. I. I.-T. I. F. 2013. Datasets. https://web.archive.org/web/20171013200331/http://ifc.recod.ic.unicamp.br/fc.website/index.py?sec=5. Accessed: 2024-12-25.

Chen, B.; Tan, W.; Coatrieux, G.; Zheng, Y.; and Shi, Y.-Q. 2020. A serial image copy-move forgery localization scheme with source/target distinguishment. *IEEE Transactions on Multimedia*, 23: 3506–3517.

De Carvalho, T. J.; Riess, C.; Angelopoulou, E.; Pedrini, H.; and de Rezende Rocha, A. 2013. Exposing digital image forgeries by illumination color classification. *IEEE Transactions on Information Forensics and Security*, 8(7): 1182–1194.

Dong, C.; Chen, X.; Hu, R.; Cao, J.; and Li, X. 2022. Mvssnet: Multi-view multi-scale supervised networks for image manipulation detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3): 3539–3553.

Dong, J.; Wang, W.; and Tan, T. 2013. Casia image tampering detection evaluation database. In *2013 IEEE China summit and international conference on signal and information processing*, 422–426. IEEE.

Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; and Zhu, X. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Guillaro, F.; Cozzolino, D.; Sud, A.; Dufour, N.; and Verdoliva, L. 2023. TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20606–20615.

Guo, X.; Liu, X.; Ren, Z.; Grosz, S.; Masi, I.; and Liu, X. 2023. Hierarchical fine-grained image forgery detection and localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3155–3165.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Hsu, Y.-F.; and Chang, S.-F. 2006. Detecting image splicing using geometry invariants and camera characteristics consistency. In *2006 IEEE International Conference on Multimedia and Expo*, 549–552. IEEE.

Hu, X.; Zhang, Z.; Jiang, Z.; Chaudhuri, S.; Yang, Z.; and Nevatia, R. 2020. SPAN: Spatial pyramid attention network for image manipulation localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16*, 312–328. Springer.

Islam, A.; Long, C.; Basharat, A.; and Hoogs, A. 2020. Doagan: Dual-order attentive generative adversarial network for image copy-move forgery detection and localization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4676–4685.

Jia, S.; Huang, M.; Zhou, Z.; Ju, Y.; Cai, J.; and Lyu, S. 2023. AutoSplice: A Text-prompt Manipulated Image Dataset for Media Forensics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 893–903.

Karras, T.; Laine, S.; and Aila, T. 2019. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4401–4410.

Kniaz, V. V.; Knyaz, V.; and Remondino, F. 2019. The point where reality meets fantasy: Mixed adversarial generators for image splice detection. *Advances in neural information processing systems*, 32.

Kwon, M.-J.; Nam, S.-H.; Yu, I.-J.; Lee, H.-K.; and Kim, C. 2022. Learning JPEG Compression Artifacts for Image Manipulation Detection and Localization. *International Journal of Computer Vision*, 130(8): 1875–1895.

Li, H.; and Huang, J. 2019. Localization of deep inpainting using high-pass fully convolutional network. In *proceedings of the IEEE/CVF international conference on computer vision*, 8301–8310.

Li, Y.; Hu, L.; Dong, L.; Wu, H.; Tian, J.; Zhou, J.; and Li, X. 2023. Transformer-Based Image Inpainting Detection via Label Decoupling and Constrained Adversarial Training. *IEEE Transactions on Circuits and Systems for Video Technology*.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2117–2125.

Liu, X.; Liu, Y.; Chen, J.; and Liu, X. 2022. PSCC-Net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(11): 7505–7517.

Ma, X.; Du, B.; Jiang, Z.; Hammadi, A. Y. A.; and Zhou, J. 2023. IML-ViT: Benchmarking Image Manipulation Localization by Vision Transformer. arXiv:2307.14863.

Milletari, F.; Navab, N.; and Ahmadi, S.-A. 2016. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 fourth international conference on 3D vision (3DV)*, 565–571. Ieee.

Niloy, F. F.; Bhaumik, K. K.; and Woo, S. S. 2023. CFL-Net: image forgery localization using contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4642–4651.

1002

Nimble, N. 2016. Datasets. https://www.nist.gov/itl/iad/mig/nimble-challenge-2017-evaluation. Accessed: 2024-12-25.

Novozamsky, A.; Mahdian, B.; and Saic, S. 2020. IMD2020: A large-scale annotated dataset tailored for detecting manipulated images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision Workshops*, 71–80.

Rombach, R.; Blattmann, A.; Lorenz, D.; Esser, P.; and Ommer, B. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 10684–10695.

Salloum, R.; Ren, Y.; and Kuo, C.-C. J. 2018. Image splicing localization using a multi-task fully convolutional network (MFCN). *Journal of Visual Communication and Image Representation*, 51: 201–209.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Sinha, S.; Bharadhwaj, H.; Srinivas, A.; and Garg, A. 2020. D2rl: Deep dense architectures in reinforcement learning. *arXiv preprint arXiv:2010.09163*.

Sutton, R. S.; McAllester, D.; Singh, S.; and Mansour, Y. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(11).

Wen, B.; Zhu, Y.; Subramanian, R.; Ng, T.-T.; Shen, X.; and Winkler, S. 2016. COVERAGE—A novel database for copy-move forgery detection. In *2016 IEEE international conference on image processing (ICIP)*, 161–165. IEEE.

Williams, R. J. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8: 229–256.

Woo, S.; Debnath, S.; Hu, R.; Chen, X.; Liu, Z.; Kweon, I. S.; and Xie, S. 2023. Convnext v2: Co-designing and scaling convnets with masked autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16133–16142.

Wu, H.; and Zhou, J. 2021. IID-Net: Image inpainting detection network via neural architecture search and attention. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(3): 1172–1185.

Wu, Y.; Abd-Almageed, W.; and Natarajan, P. 2018. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European conference on computer vision (ECCV)*, 168–184.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, 418–434.

Yu, Z.; Li, B.; Lin, Y.; Zeng, J.; and Zeng, J. 2023. Learning to Locate the Text Forgery in Smartphone Screenshots. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5. IEEE.

Yu, Z.; Ni, J.; Lin, Y.; Deng, H.; and Li, B. 2024. DiffForensics: Leveraging Diffusion Prior to Image Forgery Detection and Localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 12765–12774.

Yuan, F.; Shou, L.; Pei, J.; Lin, W.; Gong, M.; Fu, Y.; and Jiang, D. 2021. Reinforced multi-teacher selection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 14284–14291.

Zhang, L.; Song, J.; Gao, A.; Chen, J.; Bao, C.; and Ma, K. 2019. Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J. 2017. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2881–2890.

Zhuo, L.; Tan, S.; Li, B.; and Huang, J. 2022. Self-adversarial training incorporating forgery attention for image forgery localization. *IEEE Transactions on Information Forensics and Security*, 17: 819–834.