广州大学
GuangZhou University

Employing Reinforcement Learning to Construct a Decision-Making Environment for Image Forgery Localization

运用强化学习构建图像篡改定位的决策环境

汇报人：伍俊　指导老师：陈艳利

第一次汇报　日期：2025年9月18日

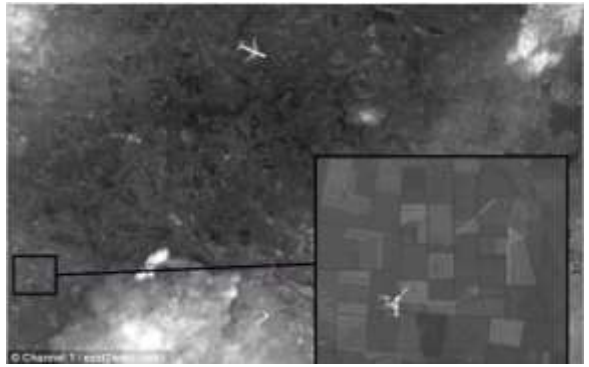
R. Peng, S. Tan, X. Mo, B. Li and J. Huang, "Employing Reinforcement Learning to Construct a Decision-Making Environment for Image Forgery Localization," in IEEE Transactions on Information Forensics and Security, vol. 19, pp. 4820-4834, 2024

目　录

1.1 图像篡改背景

手工制作伪造图像

手工制作的伪造图像是**指人为地使用图像编辑软件对原始图像进行修改**。此手段已经在**虚假新闻、国际冲突、电子商务、版权保护等**多个方面产生严重的影响。



俄罗斯媒体公布的乌克兰战机击落马航MH17的伪造照片，产生了不良的国际影响

2022年9月，安徽法院审结了一起利用修图工具非法制造发票的案件

2004年小布什竞选时选用的宣传图片，在选举时干扰了民众的决策，对选举的结果也造成了不小的影响

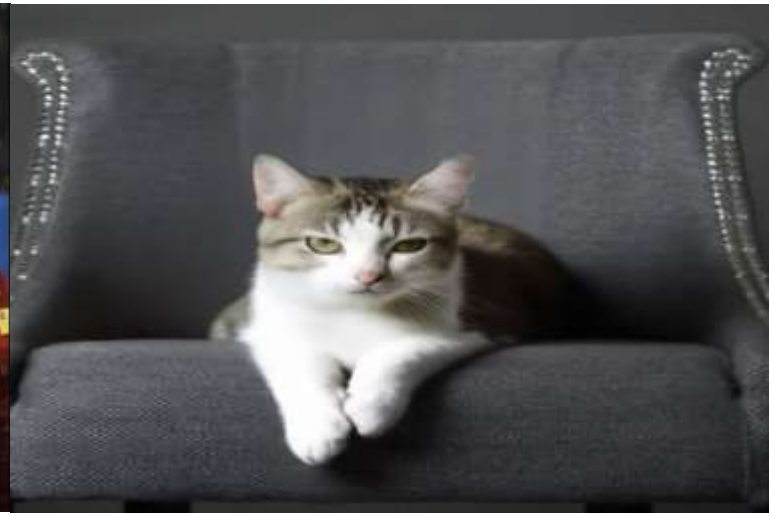
1.2 图像篡改背景

人工智能生成的伪造图像

人工智能的兴起使网络中产生了大量的伪造图像，然而这些图像通常很难通过肉眼分辨真伪，此手段已经在**隐私侵犯与人格尊严践踏，虚假新闻等**多个方面产生严重的影响。

DeepNude**软件可以自动“脱掉”女性身上的衣服，此类软件通过制作色情图像对社会影响极大**

MidJourney**软件生成的中国队赢得世界杯画面，足以以假乱真**



Dall·E2**软件生成的宠物照，难以辨别真伪**

1.3 现有篡改检测方法

机器学习方法

- **主流方法**：基于**人工提取的特征**（如：纹理特征，颜色边缘特征）采用传统机器学习方法（SVM 分类器、随机森林）来进行篡改检测。
- **缺点**：**依赖人工特征，泛化性差**，只能针对特定篡改方式设计，**难以应对新型复杂篡改**

深度学习方法

- **主流方法**：深度学习在图像篡改检测中**主要依赖端到端的卷积神经网络**，通过**学习图像的高层语义和底层纹理特征实现篡改区域定位。**
- **缺点**：普遍存在**泛化能力有限**和**鲁棒性不足**的问题：一旦跨数据集测试，性能显著下降；在面对 JPEG 压缩、噪声、缩放、模糊等常见图像退化时，检测精度也会明显受损。

1.4 什么是强化学习?

两大主体

Agent：智能体

Environment：外部环境

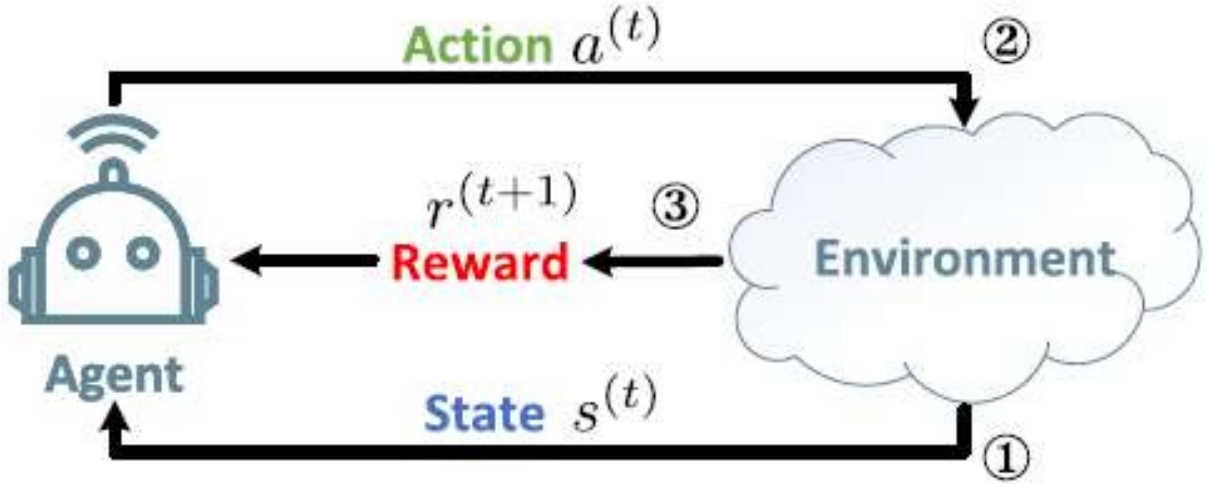
四大要素

Action：动作，时间步i的动作。

State：状态，时间步i的状态。

Percent：当前状态s转到下一状态s’的概率。

Rewards：奖励，采取动作a所获得的奖励。



行为决策过程MDP=(S,A,P,R)

强化学习对问题的理想化建模

1.5 为什么使用强化学习?

现有检测方法的局限性

不足之处

- **过度依赖数据**：性能过度依赖于训练数据的数量、质量和分布
- **决策僵化**：其决策过程是静态且固定的，无法根据图像质量或篡改强度进行动态调整。
- **缺乏探索**：作为一种被动的监督学习，它缺乏对未知环境进行主动探索的机制。

需求

能否有自适应的方法**在动态空间中进行探索**，从而更准确地进行**篡改定位**?

强化学习

强化学习能够在**大型状态-动作空间中进行探索，找到最优策略**，是这项任务的天然候选者!

应用

在复杂未知图像篡改场景中进行检测定位

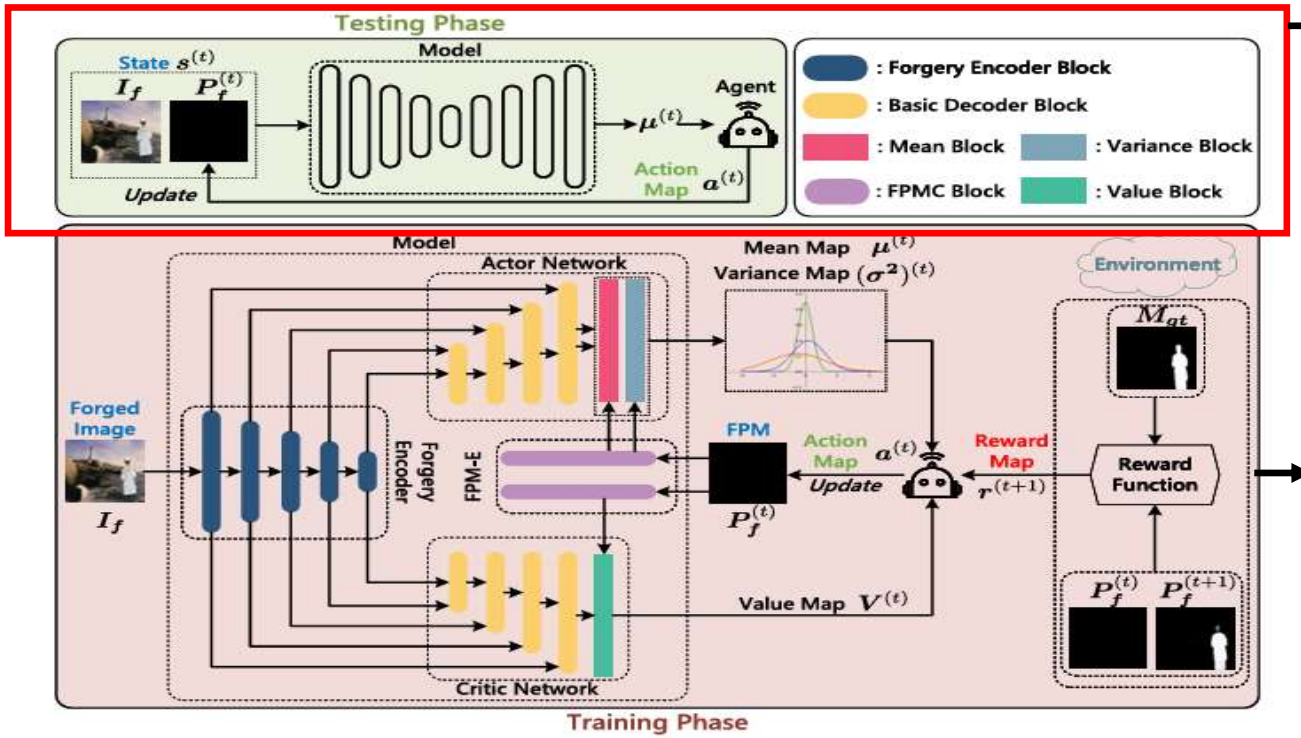
解决

1.6 本文贡献

- **提出新颖的强化学习框架：** 首次提出了一种新颖的基于强化学习的像素级图像伪造定位框架CoDE
- **定义连续动作空间：** 定义了一个基于高斯分布的连续动作空间，**即使没有相应的数据增强，也能显著提高模型对各种图像后处理攻击的鲁棒性 。**
- **设计专门的奖励函数：** 考虑到实际图像篡改中**常见的篡改区域稀疏分布**，专门设计了一个奖励函数，**使代理能够从反馈奖励中有效地学习最优策略 。**
- **优越的实验性能：** 在多个基准数据集上进行的综合实验证明，**CoDE显著优于现有最先进的方法**，并且在**抵抗在线社交网络传输导致的图像降级方面表现出卓越的鲁棒性 。**

目　录

2.1 整体框架



测试阶段

输入篡改图像和初始概率图，Actor 直接输出确定性动作迭代更新概率图，最终经过固定阈值二值化得到篡改区域掩码。

训练阶段

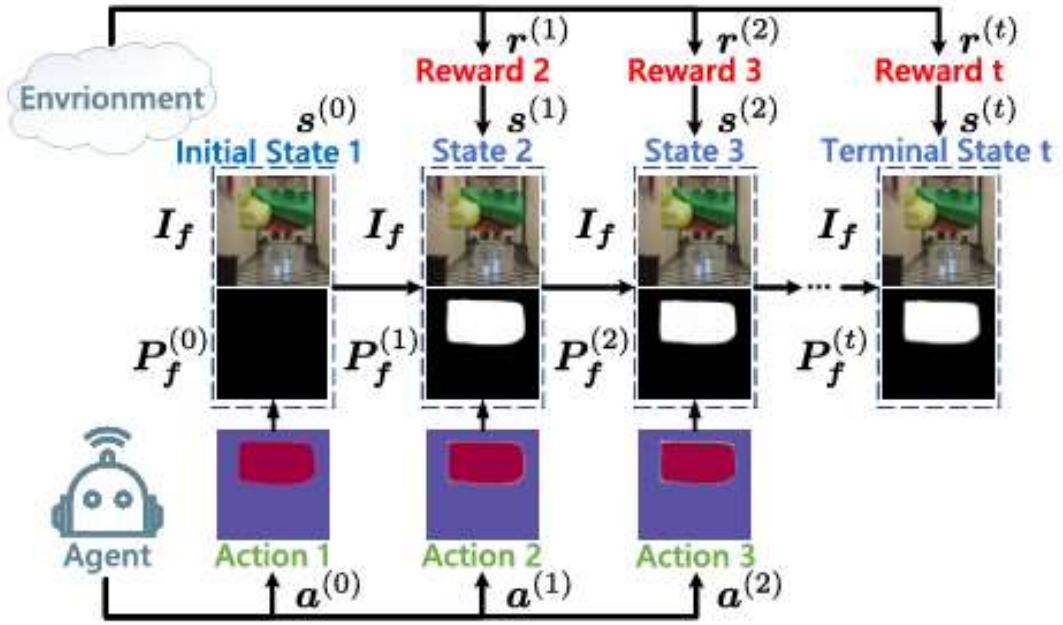
将图像与伪造概率图（状态）输入模型，经 Actor 生成动作迭代更新，Critic 进行状态评估，并结合奖励函数优化策略，最终通过多步交互学习到精准的篡改定位能力。

2.2 问题建模

将图像篡改定位任务转化成MDP决策过程

确定MDP关键要素

- **状态**：篡改图像 + 当前概率图
- **动作**：调整像素伪造概率
- **奖励**：鼓励正确更新篡改区域



最终目标

通过**状态 - 动作 - 奖励的闭环迭代**，让智能体**逐步优化每个像素的篡改概率估计**，最终输出与真实篡改掩码高度匹配的定位结果，实现**像素级精准检测。**

2.3 状态空间表示

MDP建模的内在要求

状态提供了智能体在每个时间步做出决策所需的所有信息

体现当前的决策历史-伪造概率图$P_f^{(t)}$

伪造概率图$P_f^{(t)}$：模型对图像中每个像素是否为篡改区域的**当前判断结果的动态快照**

环境的核心信息-伪造图像I_f

I_f：提供判断真伪的原始特征

在每一步，比对 **“Map”** -$P_f^{(t)}$ 与 **“World”** -I_f 的差异，生成动作来**修正地图**，逐步精炼其定位结果

2.3 状态空间表示

如何进行状态空间编码？

拼接编码模式

- **方法**：将伪造图像I_f与伪造概率图$P_f^{(t)}$在通道维度拼接，作为输入，一次性送入编码器
- **优点**：结构简单，编码器设计直接
- **缺点**：每一步状态更新时都需要重新处理整幅图像和概率图，计算量大、推理速度慢

VS

双流编码模式

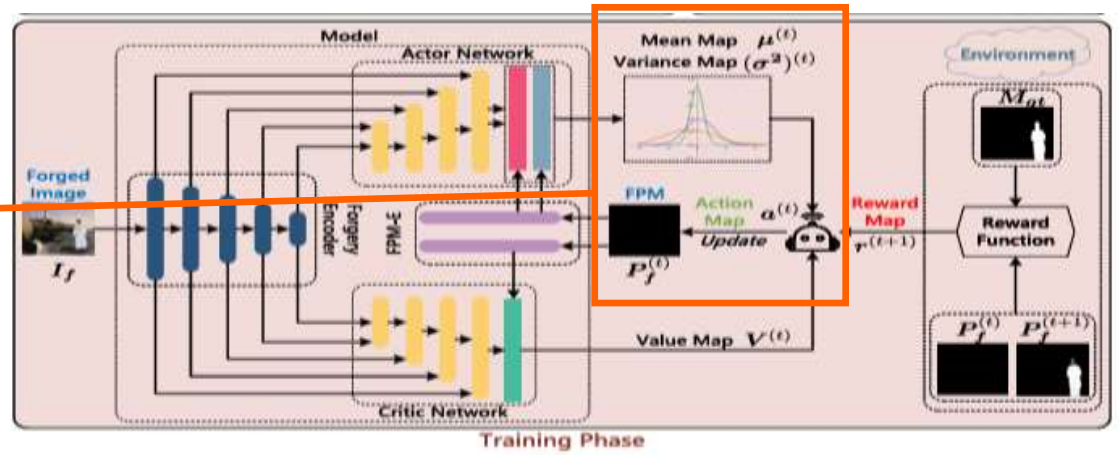
- **方法**：设计双编码器——伪造图像编码器提取一次性图像特征，伪造概率图编码器逐步更新概率图特征
- **优点**：显著减少推理时间
- **缺点**：实现上比拼接模式稍复杂

结论：采用双流编码模式能有效解决多步状态更新的效率瓶颈

2.4 动作空间表示

目标：更新伪造概率图，使其逐步逼近真实篡改区域

每个像素的智能体都会**根据一个独立的高斯分布来采样一个连续的动作值a 。这个动作值代表了该像素伪造概率的微调量（增加或减少）**



Q1：为什么选择连续动作空间而不是离散动作空间?

A1：连续动作能对伪造概率做更细粒度、可微分的调整，相比离散动作更灵活，提升定位精度与复杂场景下的鲁棒性。

Q2：为什么选择的是高斯连续分布而不是其他连续分布?

A2：高斯分布参数少，表达能力强，采样与反向传播方便（可用重参数化技巧），在探索与利用之间平衡性好，训练稳定。

2.5 奖励机制

奖励函数

引导智能体逐步将每个像素的篡改概率更新至接近其真实标签

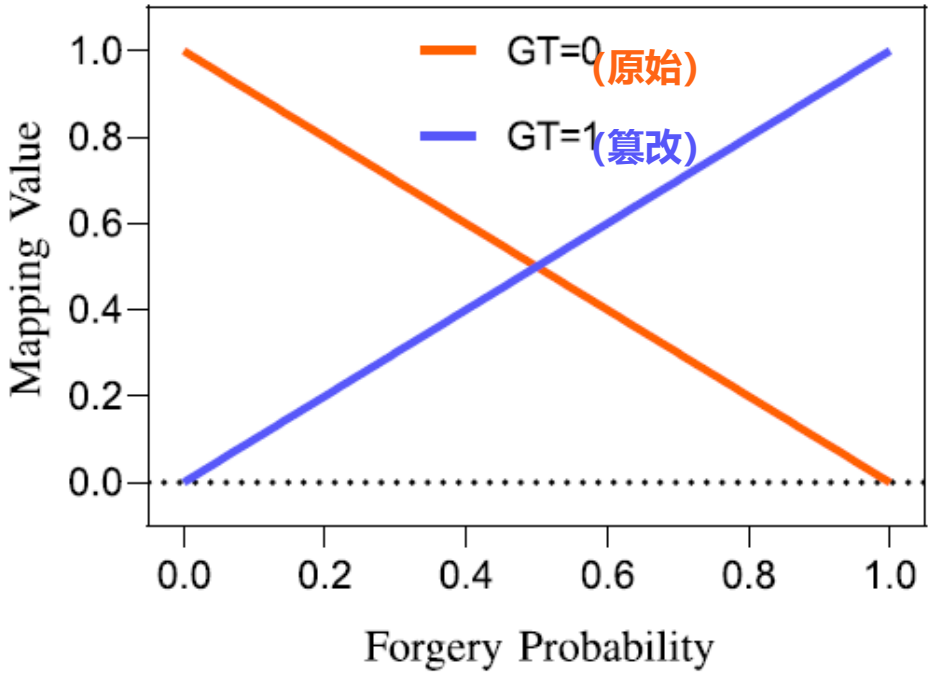
当前伪造概率 x 和真实标签 GT切合度

$$f_{lin}(x, GT) = \begin{cases} 1-x, & \text{if } GT = 0, \\ x, & \text{if } GT = 1. \end{cases} = GT \cdot x + (1-GT) \cdot (1-x)$$

奖励与概率变化呈线性关系

像素点篡改概率

描述像素点是否被篡改
- GT=0：该像素点未被篡改
- GT=1：该像素点被篡改



指引智能体朝着正确更新伪造概率的方向前进

$$r_{bs}^{(t)} = f_{lin}(x^{(t)}, GT) - f_{lin}(x^{(t-1)}, GT)$$

t-1时刻伪造概率与真实标签切合度

t时刻伪造概率与真实标签切合度

2.5 奖励机制

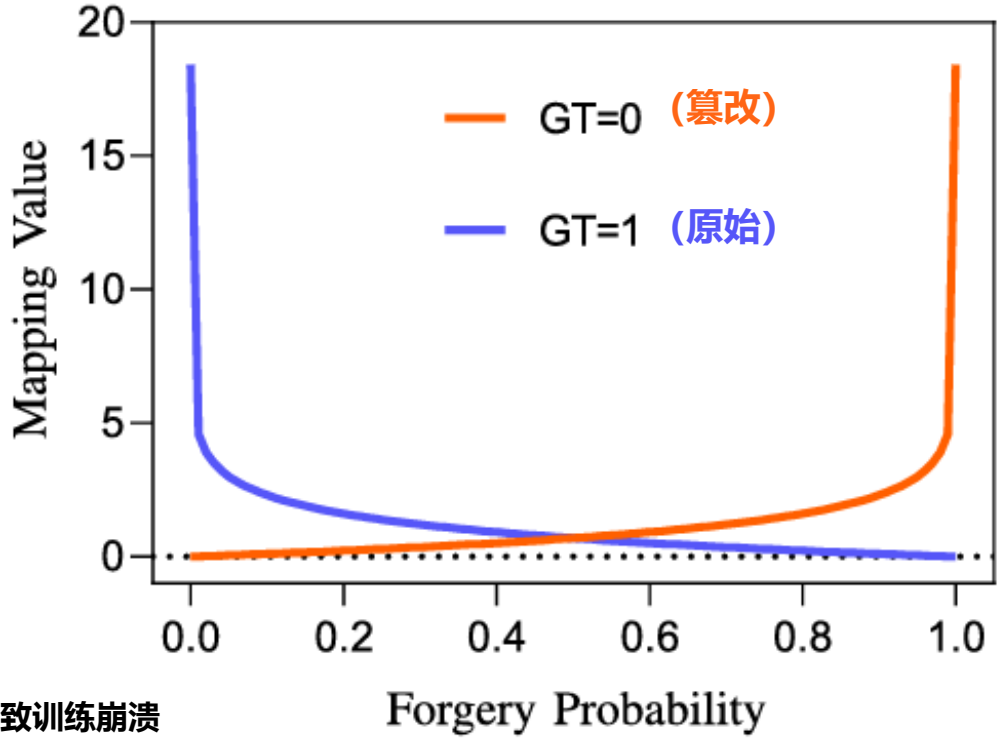
数据分布失衡：原始区域像素数量远超篡改区域

面临挑战

- **策略缺陷引发惩罚**：训练初期智能体策略不成熟，在原始区域大量误判，持续获得较高的负奖励。
- **负奖励导向保守策略**：频繁惩罚迫使智能体倾向于极保守的微小更新，严重抑制有效学习。

引入基于二元交叉熵的映射函数

- 在篡改区域采取正确行动时应获得更大幅度的奖励提升
- 减少在原始区域采取错误行动的惩罚



二元交叉熵的映射函数

$$f_{bce}(x, GT) = \begin{cases} -\log(1 - x + \epsilon), & \text{if } GT = 0, \\ -\log(x + \epsilon), & \text{if } GT = 1. \end{cases}$$

设为1×10^{-8}，避免边界陷入无限值导致训练崩溃

差分奖励机制

$$r_{bce}^{(t)} = -(f_{bce}(x^{(t)}, GT) - f_{bce}(x^{(t-1)}, GT))$$

2.6 A3C算法

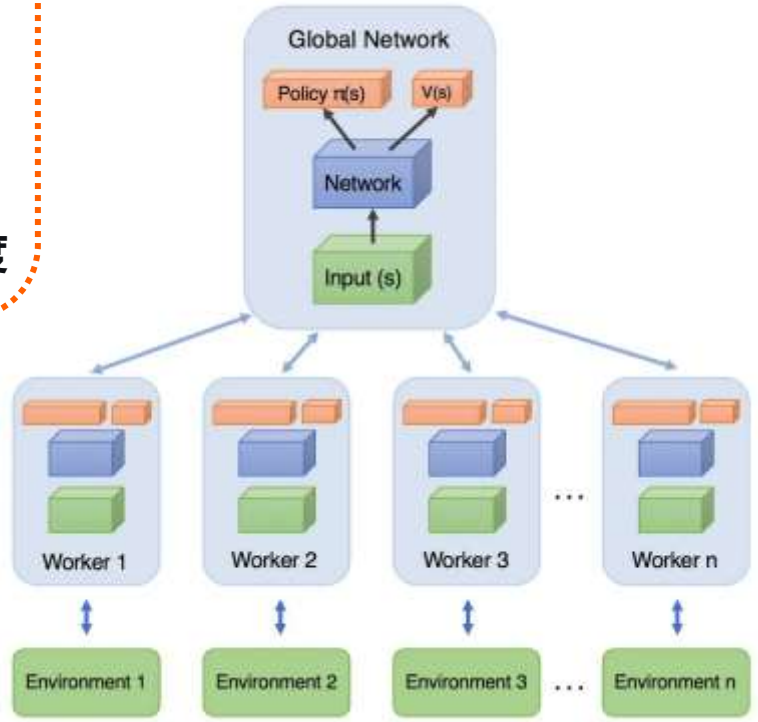
A3C算法简介

通过**多线程异步并行**方式训练全局网络，以提高学习效率和稳定性。

行动者（Actor）：作为策略网络，负责根据状态输出动作；

评论家（Critic）：作为价值网络，评估当前状态的价值以引导策略更新。

允许多个线程同时与环境交互并异步更新全局网络，从而集成多样化经验，加快训练速度



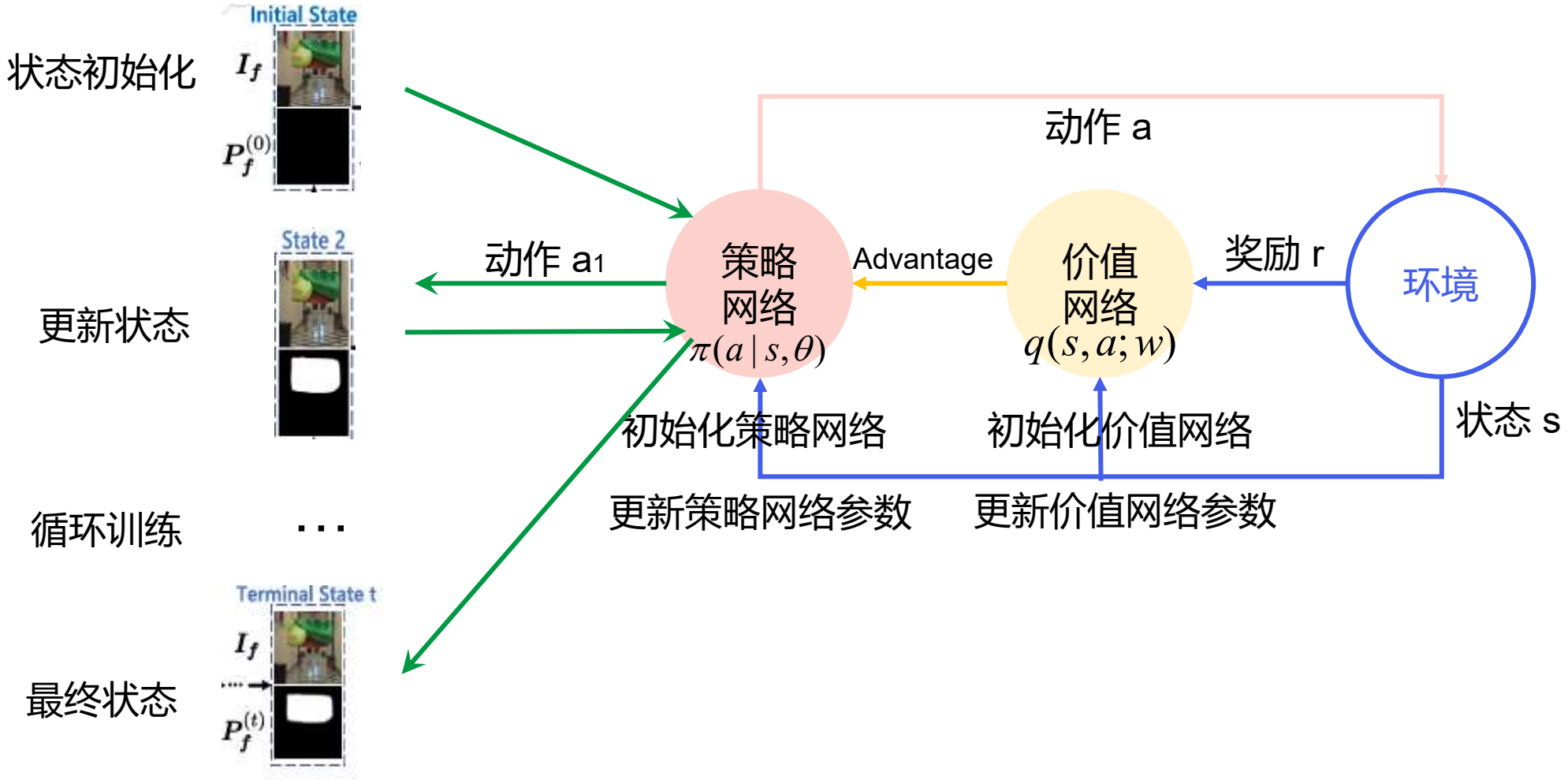
为什么采用A3C算法训练

① **高效率与稳定性**：通过**异步并行**训练，能**更快地收敛并避免训练不稳定。**

② **支持连续动作空间**：完美适配模型**基于高斯分布的连续动作空间**，实现**像素级精细微调。**

③ **出色的泛化能力**：**多智能体探索不同状态**，提升模型在**复杂情况下的泛化能力。**

2.7 RL训练流程



目　录

3.1 实验设计

- **实验目标：**验证CoDE在图像篡改定位中的**有效性**、**泛化性**和**鲁棒性**。

- **训练模式**
  - **基准训练：**模型分别在每个基准数据集上单独训练， 并在对应的测试集上评估。
  - **预训练：**模型仅在外部的数据集上训练，然后直接在目标数据集的测试集上进行评估。
  - **微调：**模型首先用预训练权重初始化，然后在目标数据集的训练集上进一步微调，最后在目标数据集的测试集上评估。
- **实验内容**
  - **消融实验：**验证BCE奖励、双流编码、连续动作空间的有效性。
  - **对比实验：**与多种SOTA方法对比
  - 泛化性能测试：验证泛化性能
  - **鲁棒性测试：**验证后处理攻击和社交网络传输图像

- **数据与评估**
  - **数据：**采用多类型篡改（拼接、复制-移动和移除）数据集（核心数据集NIST16，CASIA）
  - **指标：**使用**F1-Score**和**IoU**作为核心评估指标。

3.2.1 验证BCE奖励函数有效性

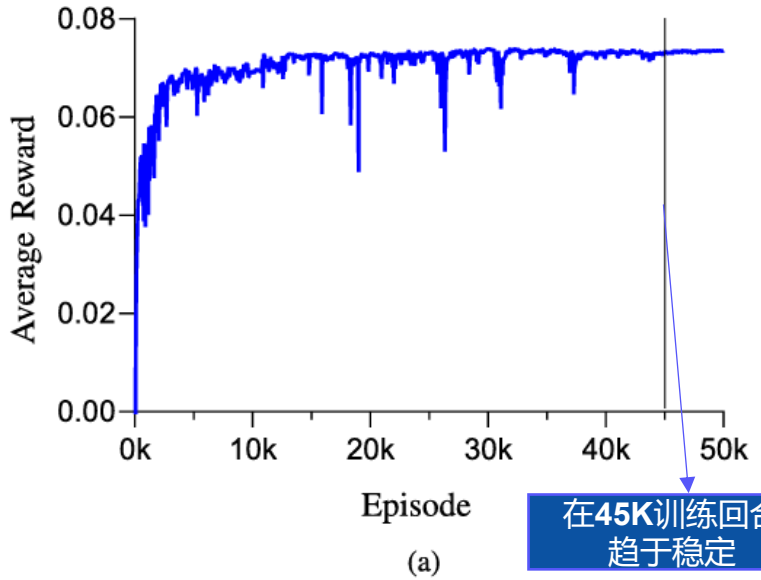
验证使用BCE奖励函数相对基础奖励函数在性能上是否有提升

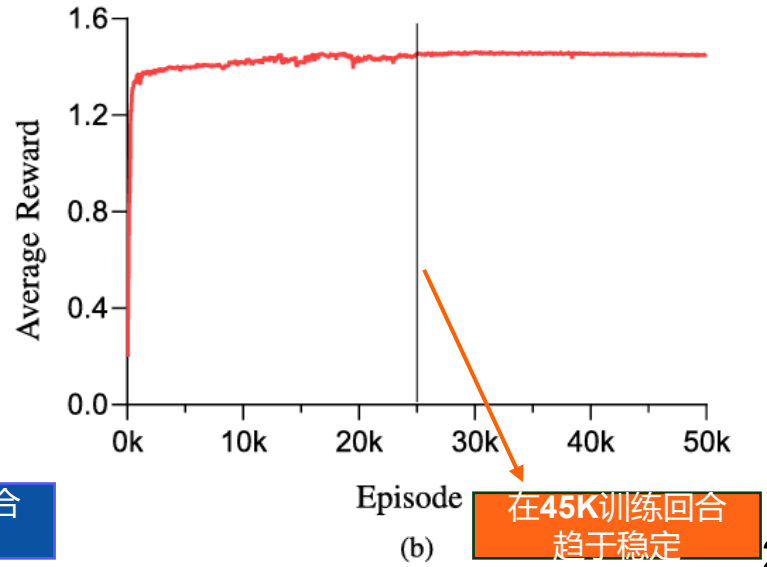
- **BCE奖励函数性能提升显著：**在NIST16和CASIA v1数据集上的像素级F1分数分别提升10%和12.6%，**优于基础奖励函数**。

- **BCE奖励函数收敛速度更快：** BCE奖励函数仅需**25k训练回合**即趋于稳定，而基础奖励函数需**45k回合**，波动更大、收敛缓慢。

ABLATION STUDY FOR REWARD FUNCTION

Reward Function	NIST16 [41]		CASIA v1 [40]	
	F_1	IoU	F_1	IoU
Basic Reward	0.821	0.790	0.512	0.472
BCE-based Reward	**0.921**	**0.886**	**0.638**	**0.557**





3.2.2 验证双流模式有效性

状态由伪造图像I_f和伪造概率图$P_f^{(t)}$组成

思考I_f和$P_f^{(t)}$的组合模式

- 拼接模式：将I_f与$P_f^{(t)}$在通道维度进行拼接
- 双流模式：分别编码I_f和$P_f^{(t)}$

ABLATION STUDY FOR STATE PROCESSING MODE

State Processing Mode	NIST16		CASIA $v1$		推理时间 Inference Time (ms)
	F_1	IoU	F_1	IoU	
Concat	0.870	0.799	0.386	0.324	115
Twin-flow	**0.921**	**0.886**	**0.638**	**0.557**	**65**

解释

在NIST16上学习难度较低，拼接模式已能取得良好性能。

双流模式性能提升幅度小

相比拼接模式，双流模式在NIST16和 CASIA v1数据集上F1分数分别提升0.041(4.7%)和 0.252(65.3%)

结论

双流模式能显著降低模型开销

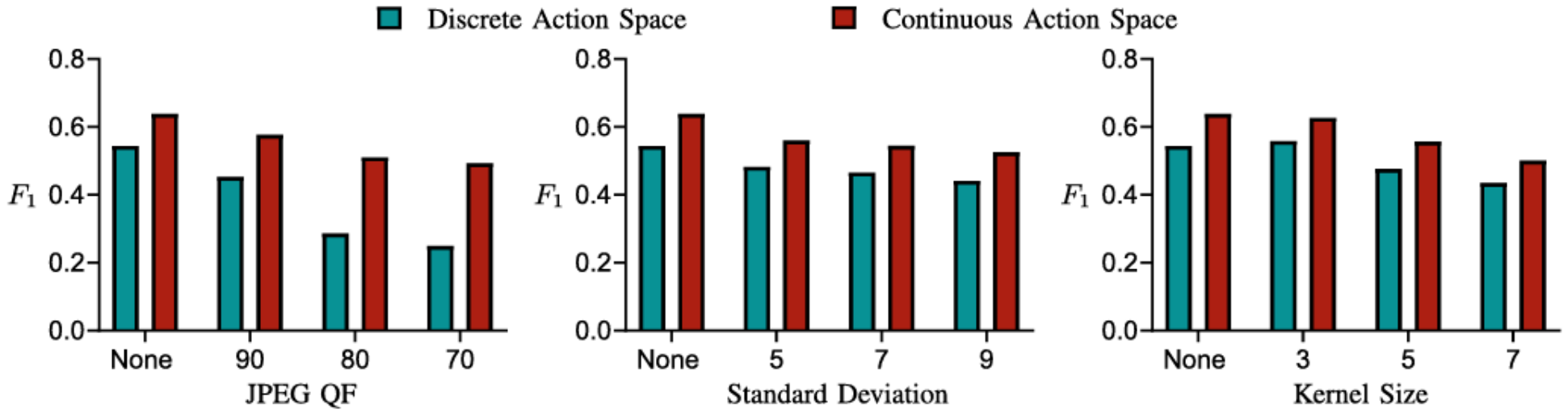
双流模式推理速度更快

双流模式平均推理时间为65毫秒，较拼接模式更快

3.2.3 验证连续动作空间有效性

离散与连续动作空间在JPEG压缩、高斯噪声和高斯模糊攻击下的鲁棒性对比

- 离散动作空间：[±0.5,±0.2,±0.1,+0.0]
- 连续动作空间：基于高斯分布的连续动作空间



- **性能显著优于离散空间**：基于高斯分布的**连续动作空间在定位性能上实现显著提升。**
- **抗干扰能力更强**：在JPEG压缩、高斯噪声与高斯模糊等后处理攻击下，**连续动作空间表现出更优的鲁棒性。**

3.3 对比实验

COMPARISON ON THE F_1 PERFORMANCE (USING FIXED THRESHOLD) UNDER BENCHMARK-TRAINING AND FINE-TUNING TRANING MODES. FIRST RANKING IS SHOWN IN BOLD

Dataset	Training Mode	MVSS-Net [25]		PSCC-Net [26]		CAT-Net v2 [29]		IF-OSN [31]		CoDE	
		F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
Columbia	benchmark -training	0.887	0.821	0.934	0.901	0.886	0.830	0.955	0.922	**0.983**	**0.965**
Coverage		0.410	0.292	0.395	0.260	0.403	0.302	0.551	0.418	**0.721**	**0.634**
CASIA $v1$		0.464	0.372	0.355	0.261	0.293	0.199	0.531	0.467	**0.638**	**0.588**
NIST16		0.895	0.837	0.641	0.528	0.647	0.549	0.832	0.767	**0.921**	**0.884**
FF++		0.952	0.910	0.917	0.849	0.955	0.896	0.939	0.911	**0.973**	**0.948**
OpenForensics		0.716	0.607	0.575	0.449	0.788	0.684	0.784	0.698	**0.796**	**0.714**
CocoGlide		0.865	0.796	0.906	0.844	0.712	0.596	0.880	0.830	**0.930**	**0.886**
Weighted Avg.		0.619	0.537	0.518	0.426	0.485	0.393	0.664	0.601	**0.747**	**0.699**
Columbia	fine -tuning	0.952	0.946	0.951	0.925	0.962	0.942	0.974	0.952	**0.983**	**0.968**
Coverage		0.625	0.528	0.581	0.479	0.455	0.375	0.776	0.697	**0.824**	**0.736**
CASIA $v1$		0.504	0.426	0.564	0.480	**0.703**	0.622	0.588	0.528	0.690	**0.630**
NIST16		0.825	0.730	0.700	0.611	0.446	0.381	0.903	0.858	**0.922**	**0.874**
FF++		0.949	0.904	0.836	0.723	0.953	0.888	0.971	0.943	**0.973**	**0.948**
OpenForensics		0.692	0.581	0.574	0.449	0.605	0.511	**0.792**	**0.704**	0.791	0.700
CocoGlide		0.887	0.828	0.918	0.865	0.662	0.553	0.913	0.866	**0.928**	**0.888**
Weighted Avg.		0.641	0.565	0.645	0.558	0.692	0.612	0.714	0.658	**0.779**	**0.723**

在**基准训练（Benchmark-training）**和**微调（Fine-tuning）**两种模式下，CoDE 在多项数据集上的 F1 与 IoU 均全面优于主流方案。

3.4 泛化性能分析

COMPARISONS ON GENERALIZATION PERFORMANCE IN TERMS OF F_1-SCORE (USING FIXED THRESHOLD). FIRST RANKING IS SHOWN IN BOLD

Testing Set	MVSS-Net [25]		PSCC-Net [26]		CAT-Net v2 [29]		IF-OSN [31]		TruFor [45]		CoDE	
	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU	F_1	IoU
Columbia	0.677	0.588	0.866	0.825	0.792	0.742	0.713	0.614	0.807	0.748	**0.881**	**0.844**
Coverage	0.458	0.381	0.394	0.282	0.290	0.232	0.266	0.180	**0.525**	**0.451**	0.464	0.362
CASIA $v1$	0.432	0.379	0.627	0.538	0.703	0.622	0.509	0.465	0.693	0.629	**0.723**	**0.637**
NIST16	0.305	0.248	0.298	0.227	0.301	0.230	0.331	0.252	0.362	0.291	**0.420**	**0.339**
FF++	0.105	0.076	0.066	0.040	0.202	0.152	0.204	0.126	**0.504**	**0.366**	0.394	0.293
OpenForensics	0.052	0.034	0.104	0.065	0.120	0.097	0.068	0.041	**0.329**	**0.259**	0.324	0.230
CocoGlide	0.333	0.257	0.422	0.333	0.363	0.288	0.265	0.207	0.360	0.292	**0.489**	**0.387**
Weighted Avg.	0.293	0.243	0.367	0.301	0.399	0.337	0.324	0.265	0.499	0.419	**0.518**	**0.428**

- 相较于五种方法，我们的CoDE分别实现了0.225 （76.8%）、0.151（41.1%）、0.119（29.8%）、 0.194（59.9%）和0.019（3.8%）的**显著提升**。
- 我们在评估中加入了基于GAN的数据集（FF++）和基于扩散模型的数据集（CocoGlide）。**即使与TruFor相比，我们提出的 CoDE也展现出至少相当的优异性能。**

3.5 鲁棒性能分析

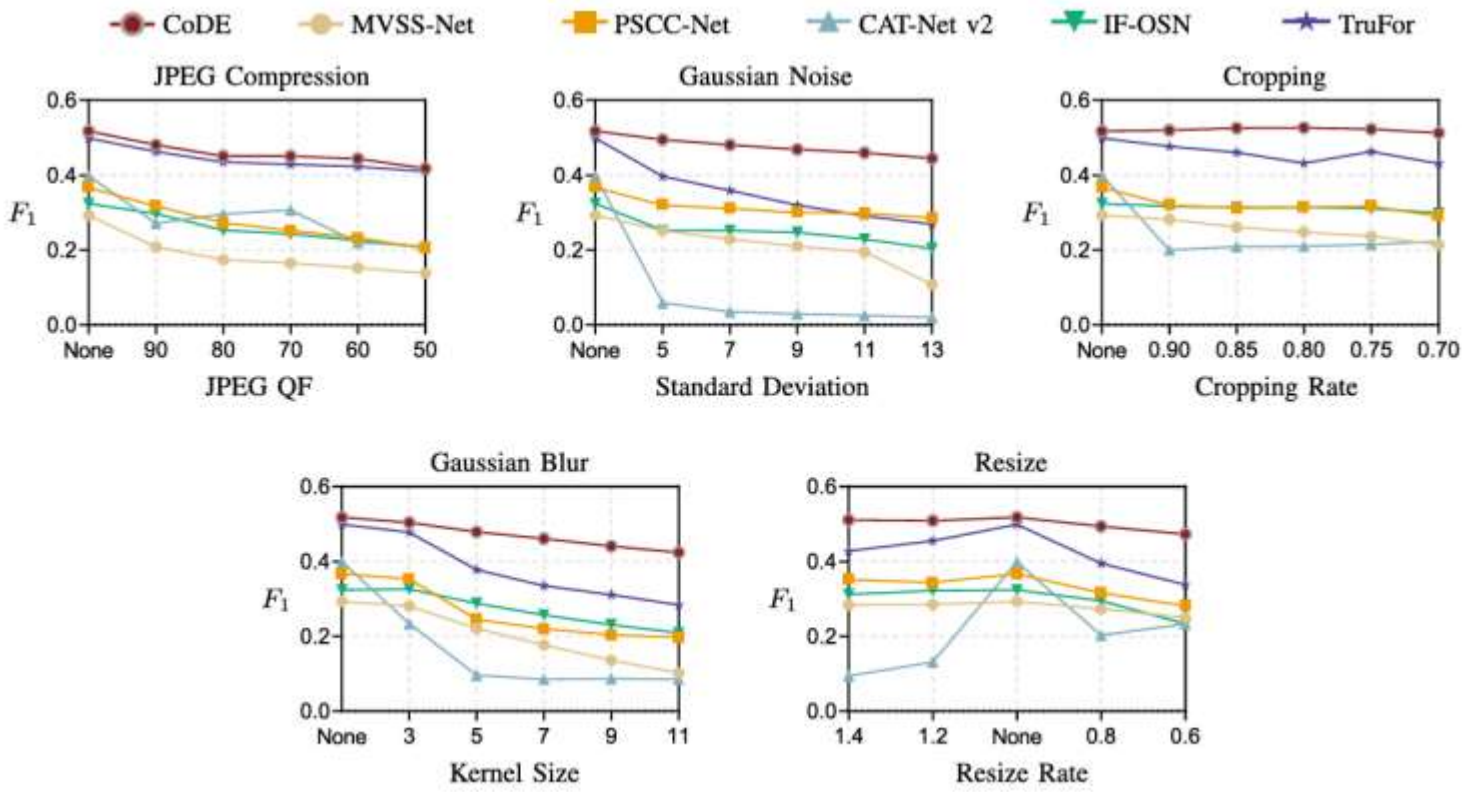


Fig. 10. Comparisons on robustness against JPEG compression, Gaussian noise, cropping, Gaussian blur and resize over all testing datasets.

与所有对比方法相比，CoDE的性能下降幅度明显更小，**表现出更强的鲁棒性 。**

目　录

4.1 总结

核心动机

- **解决瓶颈**：克服现有深度学习篡改检测方法**泛化能力差**和**鲁棒性不足**（尤其对图像后处理）两大缺陷。
- **应对稀疏**：解决篡改区域像素**极度稀疏**带来的类别不平衡与难以定位的挑战。
- **范式探索**：引入**强化学习（RL）** 的探索与决策机制，替代传统的纯监督学习。

创新点

- **首次将像素级篡改定位建模为马尔可夫决策过程（MDP）**，每个像素是一个智能体，通过多步迭代决策协同定位。
- 设计**基于高斯分布的连续动作空间**，实现像素级更新的精细调控，极大提升鲁棒性。
- 提出**针对稀疏篡改的BCE奖励函数**，有效平衡学习过程，加速收敛。
- 采用**双流状态编码器**，分离处理图像与概率图，在保证性能的同时大幅提升效率。

4.2 思考

思考点一

Q：时间开销问题--CoDE 在每个像素上部署一个智能体，并使用多步迭代更新，导致计算复杂度较高，训练时间可能较长。

A：引入**分层强化学习（HRL）** 或 **注意力机制**，**优先在疑似伪造区域进行精细操作**，减少无关区域的计算。

思考点二

Q：对真实世界中复杂后处理的鲁棒性仍有提升空间:虽然论文测试了多种后处理攻击，但未考虑**复合攻击**（如同时进行 JPEG 压缩 + 噪声 + 模糊）的极端情况。

A:

- 在训练中引入**对抗训练（Adversarial Training）**，使用对抗样本增强模型鲁棒性。
- 设计更复杂的**混合后处理攻击测试集**，模拟真实社交网络中的图像退化过程。

感谢聆听

欢迎老师、同学们批评指正