arXiv:2504.18361v2 [cs.CV] 15 May 2026

COCO-INPAINT: A BENCHMARK FOR DETECTING AND LOCALIZING INPAINTING-BASED IMAGE MANIPULATIONS

Haozhen Yan[1], *Yan Hong*[2], *Jiahui Zhan*[1], *Suning Lang*[1], *Yikun Ji*[1], *Yujie Gao*[1], *Huijia Zhu*[2], *Jun Lan*[2*], *Jianfu Zhang*[1*]

[1] Shanghai Jiao Tong University, [2] Ant Group

orion810@sjtu.edu.cn, lanjun_yelan@163.com, c.sis@sjtu.edu.cn

ABSTRACT

Recent advances in image manipulation have enabled highly photorealistic content generation, but also lowered the barrier to arbitrary editing, raising concerns about multimedia authenticity and security. Existing Image Manipulation Detection and Localization (IMDL) methods mainly target splicing or copy-move forgeries, while benchmarks for inpainting-based manipulations remain limited. To bridge this gap, we present COCO-Inpaint, a comprehensive benchmark specifically designed for inpainting detection and localization, with three key contributions: 1) High-quality inpainting samples generated by six state-of-the-art inpainting models, 2) Diverse generation scenarios enabled by four mask generation strategies with optional text guidance, and 3) Large-scale coverage of 238,302 inpainted images with rich semantic diversity. Our benchmark is constructed to highlight intrinsic inconsistencies between inpainted and authentic regions, rather than superficial semantic artifacts such as object shapes. We further establish a rigorous evaluation protocol with three standard metrics to benchmark existing IMDL methods and reveal current trends and challenges.

***Index Terms*—** AI-Generated Image Forensics, Localization, Diffusion Models

1. INTRODUCTION

With the rapid development of generative methods [1, 2], image generation has achieved remarkable progress in recent years. Image inpainting, as a fundamental task in image generation, aims to fill in the missing regions of an image with plausible content. However, inpainting methods present a double-edged sword: while their operational simplicity and impressive results empower users with intuitive editing capabilities, these very advantages also raise significant concerns regarding digital integrity and information security when misused. This urgently demands effective methods to detect and localize inpainting manipulations.

*Corresponding authors.

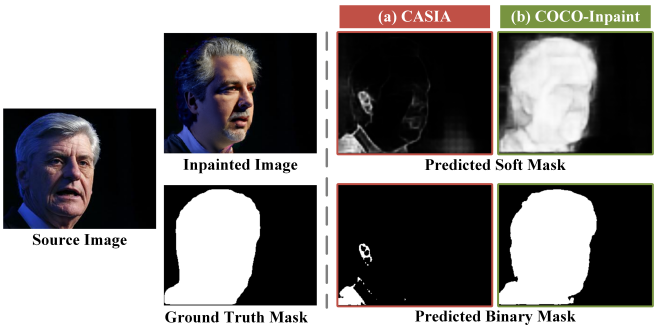


Fig. 1: Comparison of IMDL model performance based on different training datasets. We select a source image from AutoSplice [3] and utilize DALL-E2 [4] to repaint it. (a) presents the prediction of localization using IML-ViT [5] trained on CASIA [6] while (b) is trained on COCO-Inpaint. Both the soft mask and binary mask are provided. Training on COCO-Inpaint substantially enhances the model's detection sensitivity and segmentation accuracy.

Existing Image Manipulation Detection and Localization (IMDL) datasets [6, 7, 8, 9] mainly generate manipulated images by normal *splicing* and *copy-move* operations (*i.e.*, insert objects into other images, or duplicate regions within the same image) from segmentation datasets [10]. These two operations fail to capture the distinct characteristics of different images, limiting the generalizability of existing IMDL methods, as shown in Fig. 1. Other existing studies have also focused on inpainting manipulation. AutoSplice [3] generates 3621 inpainting images from the Visual News dataset using an object segmentation pipeline and a generative model [4]. COCOGlide [11] includes 512 inpainting images from MS-COCO [10] validation set by GLIDE [12]. GIM [13] introduced the IMDL Benchmark, where manipulated images were generated using diffusion-based generative models. However, these datasets remain limited in scale, quality, model diversity, and systematic analysis. To overcome these limitations, we introduce **COCO-Inpaint**, a large-scale, multi-dimensional, high-quality inpainting IMDL benchmark. Built on MS-COCO [10], COCO-Inpaint incorporates multiple representative inpainting models, four

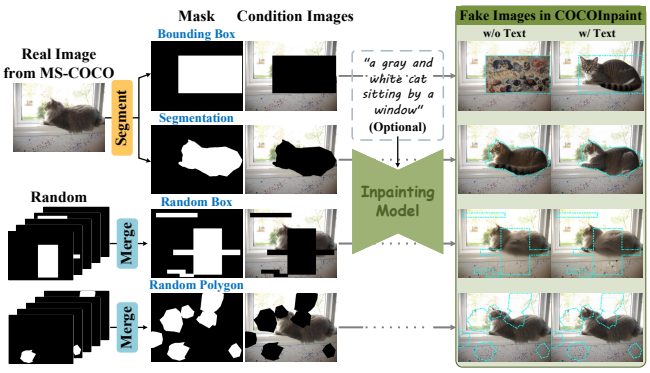


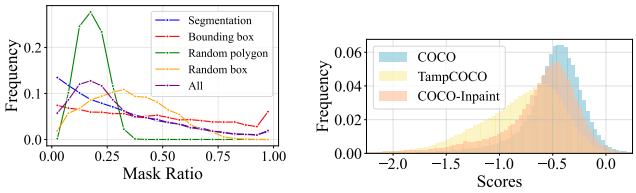
Fig. 2: COCO-Inpaint construction pipeline. Real images are converted into conditional images with 4 mask types. Optional COCO captions for text guidance. Six inpainting models then generate $6 \times 4 \times 2 = 48$ manipulated images.

mask types (segmentation, bounding box, random polygon, random box), and two guidance modes (context-aware and text-guided). The final dataset consists of $\sim 238k$ inpainted images and $\sim 108k$ authentic images, supporting fine-grained evaluation across all model-mask-guidance combinations. As shown in Fig. 1, training IMDL models on COCO-Inpaint substantially improves localization of manipulated regions, whereas training on splicing- or copy-move-based datasets such as CASIA [6] yields limited performance. We further conduct cross-validation experiments across different levels, leading to the following key observations:

- *Model architecture:* Vision Transformer-based models consistently outperform CNN-based models in adaptability to IMDL tasks across diverse experimental conditions.
- *Cross-model generalization:* IMDL models perform well in-distribution but generalize poorly to unseen inpainting models, while multi-model training improves adaptability.
- *Cross-mask evaluation:* The shape of the mask significantly affects generalization. Models trained on randomly generated masks (*e.g.*, random boxes or polygons) exhibit higher robustness than those trained on structured masks.
- *Cross-mask-ratio evaluation:* Models trained with mask ratios between 0.4 and 0.6 achieve better generalization. A balanced proportion of inpainted and authentic regions encourages learning more discriminative features.
- *Cross-prompt evaluation:* IMDL models trained on text-guided inpainted images achieve stronger generalization.

2. COCO-INPAINT

To build a high-quality IMDL benchmark, we consider image quality, mask/context diversity, and generalizability, and introduce COCO-Inpaint, a large-scale benchmark for inpainting-based manipulation analysis (Fig. 2).



(a) (b)

Subset	Text-guided	Context-aware
Segmentation	5.06	17.15
Bounding Box	7.04	31.86
Random Polygon	2.91	3.12
Random Box	3.91	9.20

(c)

Fig. 3: (a): Mask ratio distribution of COCO-Inpaint. (b): ImageReward distribution of COCO, TampCOCO and COCO-Inpaint. (c): FID (↓) under prompt guidance.

Table 1: Summary of IMDL Datasets. HiFi-IFDL mostly contains fully synthesized images; only manipulated ones are counted here. "#Variants" denotes the number of manipulated images per authentic image.

Dataset	#Real	#Fake	Generator	#Model	#Variants
CASIA [6]	7.2K	5.1K	Trad.	-	0-8
IMD2020 [9]	35K	35K	Trad.	-	1
TampCOCO [7]	-	800K	Trad.	-	8
HiFi-IFDL [8]	-	376K	GAN	2	1
AutoSplice [3]	2.3K	3.6K	Diff.	1	1-3
COCOGlide [11]	-	512	Diff.	1	1
GIM [13]	1.1M	1.1M	Diff.	3	1
COCO-Inpaint	108K	238K	Diff.	6	1 or 48

2.1. Inpainting Models

To build a high-quality IMDL benchmark, we aim to cover diverse backgrounds with corresponding bounding boxes and segmentation masks. We adopt MS-COCO [10] as the source dataset due to its rich semantic content and well-annotated object information. Considering both performance and community adoption, six representative inpainting models are selected for manipulating real images, including three inpainting backbones: SD1.5-Inpainting [14], SDXL-Inpainting [15] and Flux.1-Fill-dev [16], and two refined inpainting models: BrushNet [1] and PowerPaint [2]. Stable-Diffusion-3.5-large [14], as a non-inpaint-specific model, is also selected for its powerful inpainting ability.

2.2. Masks and Prompts

Masks are essential in inpainting as they define regions to be filled and encode different guidance signals. Segmentation masks enable object replacement or removal with prompts, while random masks promote background-consistent filling. Prior IMDL datasets [6, 7] rely mainly on segmentation masks, whereas practical inpainting involves more diverse types. To reflect this diversity, we introduce four mask gen-

Table 2: COCO-Inpaint dataset composition. "# w/ 48 variants" and "# w/ 1 variant" indicate images with multiple and single inpainted variants, respectively. "Filtered" denotes invalid inpainted variants removed during quality filtering.

Subset	Authentic	# w/ 48 variants	# w/ 1 variant	Filtered	Total Inpainted
Training	108,353	3,000	105,353	11,051	238,302
Testing	4,551	125	4,426	491	9,935

eration strategies: segmentation masks extracted from MS-COCO objects (merging multiple instances when needed), bounding boxes derived from object masks (merged into unified regions), and randomly generated polygons or boxes with 3–8 shapes each. Existing datasets such as CASIAv2 and Defacto contain small manipulated regions, averaging only 7.6% and 1.7% of image area, which limits evaluation under complex conditions. To address this, we merge multiple Random Polygon or Random Box masks into composite ones, enabling larger and more diverse mask ratios (Fig. 3a).

Recent inpainting models leverage prompts to enhance visual quality and semantic consistency. To assess their impact on IMDL, we consider two settings: context-aware, which relies only on surrounding content, and text-guided, which uses MS-COCO captions. While text guidance may fail with overly specific or misaligned captions, it generally improves fidelity and alignment. Quantitative evaluation with FID [17] (Fig. 3c) shows that text-guided inpainting consistently outperforms the context-aware setting.

2.3. Scale, Diversity, and Quality

To ensure broad coverage and evaluation under diverse conditions, we adopt a hybrid sampling strategy. From MS-COCO, we randomly select 3,000 images for training and 125 for validation, applying all 48 combinations of *models* $\times$ *masks* $\times$ *prompts* ($6 \times 4 \times 2$) to each. The remaining images are processed using a single random combination to increase diversity while controlling dataset size. Original MS-COCO images are retained as authentic references. Dataset composition is shown in Tab. 2. This strategy yields a balanced and scalable benchmark with 238,302 inpainted and 108,353 authentic images (see Tab. 1). Tab. 6 further analyzes the influence of scale on model performance. To reduce preprocessing bias, we apply a maximized CenterCrop to source images before inpainting. This mitigates the resizing artifacts caused by VAE, which require input dimensions divisible by 8 (or 16 for Flux.1). Fig.3b compares ImageReward[18] distributions across datasets. COCO-Inpaint closely aligns with the original COCO, and outperforms TampCOCO, demonstrating higher visual quality and realism. To ensure data quality, we adopt a two-stage post-processing strategy: automatic filtering with a binary classifier, followed by manual curation.

Table 3: Cross-Model Generalization: ACC(%), AUC(%), and F1(%) are reported. In-Distribution: Models are trained and evaluated on the same subset. Cross-Distribution: Models are trained on each training subset and evaluated on all testing subsets. Results are averaged across subsets.

Method	In-Distribution		Cross-Distribution	
	Pixel-Level	Image-Level	Pixel-Level	Image-Level
IID	86.0/87.0/63.1	78.1/85.5/82.5	80.0/66.2/22.4	52.4/61.5/50.4
PSCC-Net	91.8/96.6/82.1	**99.4/100/99.5**	82.2/65.6/**24.6**	52.0/70.8/31.9
ObjectFormer	89.9/92.5/67.0	90.3/98.1/89.7	**85.3**/66.4/19.0	61.2/70.8/37.4
IML-ViT	94.4/97.4/86.6	99.0/99.7/99.3	82.8/**66.9**/23.2	55.0/**75.4**/40.2
SegFormer	**95.9/98.4/89.7**	90.2/100/93.3	83.2/63.2/23.5	**62.1**/66.1/**62.6**

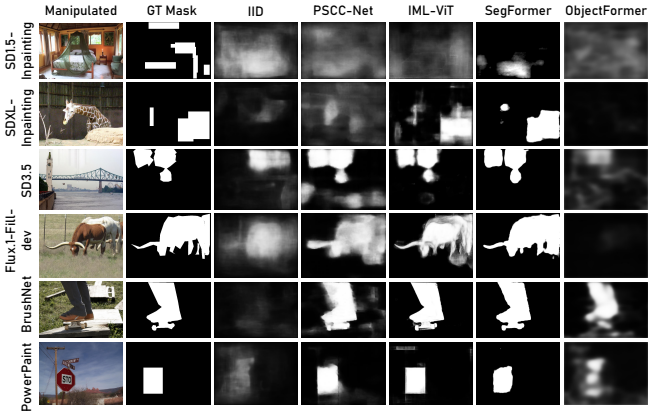


Fig. 4: Qualitative IMDL results on COCO-Inpaint.

3. EXPERIMENTS

3.1. Settings

Baselines. We evaluate five representative IMDL methods: IID [19], PSCC-Net [20], IML-ViT [5], ObjectFormer [21], and SegFormer (MiT-B2) [22]. All methods are tested under their official configurations, with Pixel-F1 early stopping (patience = 5). For SegFormer, global average pooling is applied to the final-stage features, followed by a linear classification head. For IID and IML-ViT, which lack classification heads, we use the maximum value of the prediction map for binary classification.

Metrics. We categorize evaluation into two levels: **Image-Level** and **Pixel-Level**. Image-level metrics evaluate whether an input image has been manipulated (*i.e.*, binary classification). We use three common metrics: Accuracy (ACC), Area Under the Curve (AUC), and F1 score. All image-level metrics are computed directly on the binary classification output. Pixel-level evaluation focuses on the model's ability to localize manipulated regions. Metrics are computed pixel by pixel on ACC, AUC, and F1. We adopt global pixel-level statistics for all pixel-level evaluation, avoiding instability and ensuring consistent, reliable evaluation. Such inconsistencies have led to unfair evaluations in prior studies [20, 5].

Table 4: Performance evaluation results. Models are trained on the full training set and evaluated on different testing subsets. ACC(%), AUC(%), and F1(%) are reported. The best and second-best results are marked in **bold** and underline, respectively.

Testing Dataset		IID [19]		PSCC-Net [20]		ObjectFormer [21]		IML-ViT [5]		SegFormer [22]	
		Pixel-Level	Image-Level	Pixel-Level	Image-Level	Pixel-Level	Image-Level	Pixel-Level	Image-Level	Pixel-Level	Image-Level
Model	SD1.5	83.9/83.8/56.0	72.7/83.0/77.4	84.5/88.7/62.7	96.9/99.8/97.7	80.6/85.1/51.6	86.8/96.5/88.1	**89.5/94.1/73.4**	**98.8/100/99.1**	84.9/92.8/69.5	97.2/100/98.0
	SDXL	86.2/86.7/63.3	75.5/84.8/79.9	91.9/97.3/81.9	96.8/99.9/97.6	84.0/87.5/55.8	85.3/93.7/86.6	**97.2/99.6/93.2**	**99.9/100/99.9**	95.4/99.2/89.5	96.7/100/97.6
	SD3.5	87.4/90.0/67.1	84.4/91.4/88.0	91.1/98.4/81.6	97.6/100/98.2	84.2/86.7/56.1	81.3/90.9/82.1	**98.1/99.8/95.3**	**99.9/100/99.9**	96.4/99.5/91.8	97.1/100/97.8
	Flux.1-Fill-dev	83.9/84.8/55.4	79.5/87.7/83.7	88.7/95.0/75.1	97.1/99.8/97.8	81.0/85.3/50.9	87.4/96.9/88.6	**96.7/99.3/91.9**	**99.8**/99.9/**99.9**	93.6/98.7/85.8	96.9/**100**/97.7
	BrushNet	91.9/94.4/78.3	83.0/91.0/86.6	97.8/99.7/94.7	97.1/99.7/97.9	90.5/94.7/71.5	84.1/92.7/85.1	**99.5/99.7/98.8**	**100/100/100**	98.9/99.9/97.4	96.3/99.7/97.3
	PowerPaint	85.0/85.9/55.9	73.7/84.5/78.1	91.6/96.2/80.0	97.6/100/98.2	82.8/88.8/57.7	88.8/97.3/90.0	**96.3/99.1/90.8**	**100/100/100**	94.3/98.8/87.0	96.9/100/97.7
Mask	Segmentation	87.8/90.8/66.1	77.1/86.0/81.8	91.6/97.1/80.5	97.0/99.8/97.8	86.1/91.3/60.8	86.2/94.6/87.5	**96.6/99.3/91.2**	**99.8/100/99.9**	94.5/98.9/87.1	96.8/99.9/97.7
	Bounding box	88.3/93.9/77.8	82.6/90.5/86.7	93.4/98.4/88.8	97.3/99.9/98.0	87.8/94.0/73.3	87.0/96.2/87.6	**97.4/99.7/95.4**	**99.8/100/99.9**	95.8/99.4/93.0	97.1/100/97.9
	Random polygon	85.0/71.4/24.7	71.7/81.7/76.6	89.2/92.4/60.6	96.9/99.9/97.8	81.5/79.5/32.2	86.5/94.6/87.4	**95.6/98.0/80.6**	**99.5**/99.9/**99.6**	92.7/97.5/73.9	97.3/**100**/98.1
	Random box	83.6/85.4/58.7	81.0/89.2/85.1	89.2/95.6/77.9	97.8/99.9/98.4	83.1/88.3/57.1	87.4/95.6/87.8	**95.3/98.9/89.5**	**99.8/100/99.9**	92.5/98.2/85.0	96.5/100/97.4
Text-guidance	Prompt	84.8/86.0/58.6	76.9/86.0/81.9	90.5/96.3/79.1	97.5/99.9/98.2	83.2/87.0/55.4	85.0/94.4/86.7	**96.1/99.1/90.5**	**99.8/100/99.8**	93.7/98.6/86.2	96.8/100/97.7
	No-prompt	87.1/89.1/66.5	78.5/88.1/83.0	91.0/96.6/80.0	97.2/99.9/98.0	83.7/88.4/58.7	86.6/95.5/88.0	**96.3/99.2/91.0**	**99.7/100/99.8**	93.9/98.8/86.6	97.3/100/98.1
All Testing Dataset		86.7/89.9/62.6	78.9/93.0/82.7	90.7/96.4/79.6	97.4/99.9/98.1	84.1/88.2/55.9	86.1/94.5/86.6	**96.1/99.1/90.7**	**99.7/99.9/99.8**	93.7/98.7/86.4	97.1/99.9/98.0

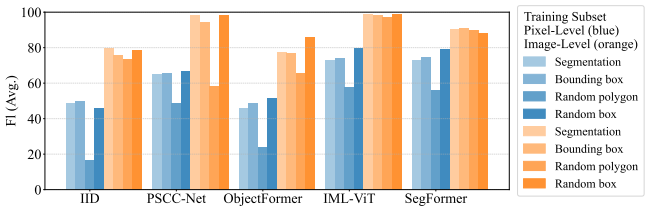


Fig. 5: Cross-mask generalization averaged over mask types.

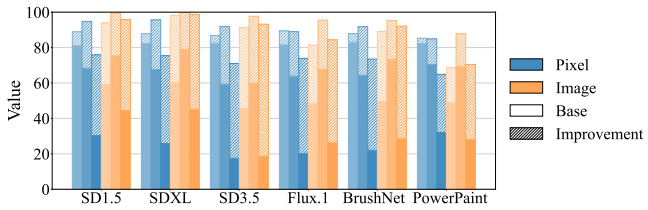


Fig. 6: The results of PSCC-Net trained on the remaining five data subsets and tested on the current subset. The six bars represent Pixel ACC/AUC/F1, Image ACC/AUC/F1, respectively. Improvement indicates the relative gain when trained on a single subset.

3.2. Results and Analysis

Full-Set Performance Evaluation. We first evaluate all baselines trained on the full training set. In addition to the full test set, we assess performance on specific subsets to better understand model behavior under different scenarios. Quantitative results are presented in Tab. 4, with qualitative visualizations in Fig. 4. The results reveal that CNNs exhibit limited performance on inpainting-based manipulations. In contrast, ViT-based architectures (IML-ViT and SegFormer) demonstrate superior robustness by modeling global dependencies through self-attention, enabling the detection of subtle and spatially dispersed manipulations. This advantage is particularly evident when manipulated regions are small (e.g., mask ratio < 0.2), where CNN-based methods such as PSCC often predict near-zero masks, highlighting the importance of global context modeling when local artifacts are sparse. ObjectFormer, due to its explicit learning of mid-level object representations, struggles with complex and diverse mask shapes, resulting in suboptimal performance. For different mask types, object-based manipulations (segmentation and bounding box) are easier to localize than random masks. *E.g.*, the F1 scores of ObjectFormer on random inpainted images (0.322 and 0.571) are notably lower than those on object-based ones (0.608 and 0.733), corroborating this observation. This is likely because random masks disrupt object–semantic alignment, hindering the use of object-level semantics for localization. Finally, text-guided inpainting improves realism but slightly increases localization difficulty across all metrics.

Generalization Evaluation Across Models. Given the rapid evolution of inpainting architectures, generalization to unseen models is crucial for IMDL. To evaluate this, COCO-Inpaint is divided into six subsets by inpainting model, each with separate training and test splits. All IMDL methods are trained and cross-evaluated across these splits. Tab. 3 reports the average performance under in-distribution and cross-model settings. ViT-based models consistently outperform CNNs in both classification and localization. However, localization performance drops sharply across models, with Pixel-F1 reaching only 24.6% due to style discrepancies and distribution shifts. These results highlight poor cross-model generalization, which we attribute to overfitting to generator-specific artifacts. To address this limitation, we advocate training on multiple inpainting models to encourage learning of common manipulation artifacts. As shown in Fig. 6, PSCC-Net trained on five subsets (*i.e.*, SDXL, SD3.5, Flux, BrushNet, and PowerPaint) and tested on the remaining one (*i.e.*, SD1.5) generalizes significantly better to unseen distributions. This cross-model training protocol offers a practical strategy for improving robustness against future inpainting techniques.

Cross-Domain Generalization Evaluation. To assess cross-domain robustness, we evaluate SegFormer pretrained on COCO-Inpaint using the partially manipulated subsets (CNN-syn.) of HiFi-IFDL [8], where CNN-syn. specifically targets

Table 5: SegFormer pretrained on COCO-Inpaint evaluated on the CNN-syn. subset of HiFi-IFDL. OSN-det. and CNN-det., pretrained on their own datasets, are the strongest baselines within HiFi-IFDL.

Model	AUC	F1
OSN-det.	51.4	38.8
SegFormer	84.5	34.6

(a) Pixel-level

Model	AUC	F1
CNN-det.	76.5	60.5
SegFormer	99.3	95.1

(b) Image-level

Table 6: Dataset scale effect results. Models trained on different fractions and tested on the full set. Pixel-F1/Image-F1 are reported.

Scale	IID	PSCC	ObjectFormer	IML-ViT	SegFormer
1	**62.6/82.7**	**79.6**/98.1	55.9/86.6	**90.7**/99.8	**86.4/98.0**
1 / 2	59.6/80.1	76.4/**99.3**	**56.1/86.8**	89.2/**99.9**	85.7/94.9
1 / 4	50.6/70.8	71.8/99.1	49.9/79.1	87.4/99.8	81.2/86.8
1 / 8	46.9/67.9	67.8/95.6	47.1/76.7	83.7/99.3	80.7/92.5
1 / 16	38.6/69.2	61.0/80.9	42.2/76.1	39.7/81.3	77.4/91.4

facial manipulations. In this setting, **SegFormer achieves 84.5% Pixel-AUC for segmentation and 99.3% Image-AUC for classification, demonstrating strong generalization**. SegFormer substantially improves Pixel-AUC and Image-level AUC/F1 over the HiFi-IFDL baselines, although its Pixel-F1 remains slightly lower than OSN-det.

Generalization Evaluation Across Masks. Inpainting models are often trained using different mask strategies [1], often relying on segmentation datasets for mask extraction. LaMa [23] argues that mask type greatly impacts inpainting model performance. To assess robustness, we conduct cross evaluation across mask types (Fig. 5). Models trained with **random box masks generalize the best**, likely because random box masks decouple manipulated regions from object semantics, unlike segmentation or bounding boxes. Irregular polygons are most challenging due to structural ambiguity.

Generalization Evaluation Across Mask Ratio. Mask area has a strong impact on inpainting performance [23]. To study its effect on IMDL, we conduct a cross-evaluation across mask ratio (Fig. 7). Each subset randomly samples 18,000 training images to control scale, while testing on the full-set. **Most IMDL models achieve optimal performance at mask ratios of 0.4–0.6.** This indicates that IMDL performs best when authentic and manipulated regions are balanced.

Generalization Evaluation Across Prompts. Modern inpainting models can operate with or without text prompts [16, 15, 14]. To analyze its impact on IMDL, we conduct cross-validation on prompt and no-prompt subsets (Fig. 8), showing that no-prompt training achieves higher in-distribution scores, while prompt training improves robustness to unseen distributions by supplying richer semantic context.

Analysis of the Benchmark Scale. We analyze scaling effects by training on random subsets of COCO-Inpaint. Performance increases with dataset size, while the marginal gains diminish beyond half-scale (Tab. 6).

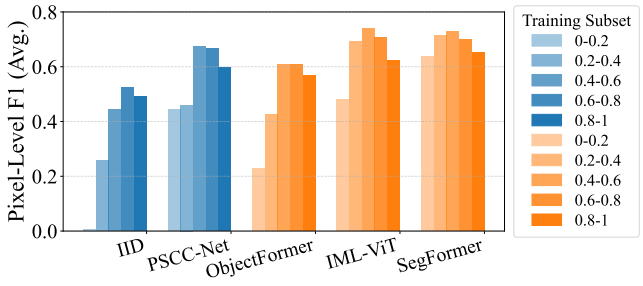


Fig. 7: Cross-Mask-Ratio Generalization: Average pixel-level F1 scores of IMDL models under different mask ratios. Colors distinguish between architectures (CNNs vs. ViTs)

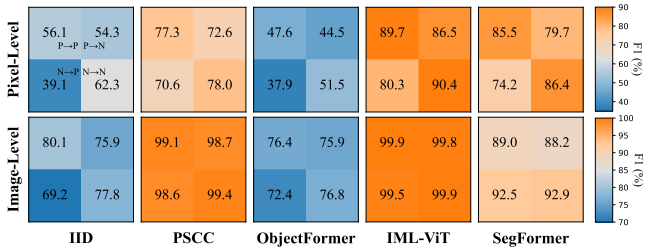


Fig. 8: Cross-Text-Guidance Generalization. F1 scores. $P \rightarrow N$ denotes training with prompts and testing without prompts.

4. CONCLUSION

In this paper, we propose COCO-Inpaint, a benchmark for Inpainting Image Manipulation Detection and Localization (IMDL). COCO-Inpaint utilizes 6 inpainting models (SD1.5-Inpainting, SDXL-Inpainting, SD3.5, Flux.1-Fill-dev, BrushNet, and PowerPaint) to generate manipulated images by 4 mask types (Segmentation, Bounding Box, Random Polygon, Random box) with two prompt guidance types. Based on the hierarchical structure of COCO-Inpaint, multi-level performance and generalization evaluation experiments are conducted, providing a comprehensive and in-depth analysis of the performance of existing IMDL methods in handling inpainting-based manipulated images.

5. ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62302295, 62595733, and 62561160155), the Shanghai Municipal Science and Technology Major Project (Grant No. 2021SHZDZX0102). This work was also supported by Ant Group.

6. REFERENCES

[1] Xuan Ju, Xian Liu, Xintao Wang, Yuxuan Bian, Ying Shan, and Qiang Xu, "Brushnet: A plug-and-play image inpainting model with decomposed dual-branch diffusion," in *ECCV*, 2024.

[2] Junhao Zhuang, Yanhong Zeng, Wenran Liu, Chun Yuan, and Kai Chen, "A task is worth one word: Learning with task prompts for high-quality versatile image inpainting," in *ECCV*, 2024.

[3] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu, "Autosplice: A text-prompt manipulated image dataset for media forensics," in *CVPR*, 2023.

[4] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022.

[5] Xiaochen Ma, Bo Du, Zhuohang Jiang, Ahmed Y Al Hammadi, and Jizhe Zhou, "Iml-vit: Benchmarking image manipulation localization by vision transformer," *arXiv preprint arXiv:2307.14863*, 2023.

[6] Jing Dong, Wei Wang, and Tieniu Tan, "Casia image tampering detection evaluation database," in *ChinaSIP*, 2013.

[7] Myung-Joon Kwon, Seung-Hun Nam, In-Jae Yu, Heung-Kyu Lee, and Changick Kim, "Learning jpeg compression artifacts for image manipulation detection and localization," *IJCV*, 2022.

[8] Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu, "Hierarchical fine-grained image forgery detection and localization," in *CVPR*, 2023.

[9] Adam Novozamsky, Babak Mahdian, and Stanislav Saic, "Imd2020: A large-scale annotated dataset tailored for detecting manipulated images," in *WACV Workshops*, 2020.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, 2014.

[11] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nicholas Dufour, and Luisa Verdoliva, "Trufor: Leveraging all-round clues for trustworthy image forgery detection and localization," in *CVPR*, 2023.

[12] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen, "Glide: Towards photorealistic image generation and editing with text-guided diffusion models," in *ICML*, 2022.

[13] Yirui Chen, Xudong Huang, Quan Zhang, Wei Li, Mingjian Zhu, Qiangyu Yan, Simiao Li, Hanting Chen, Hailin Hu, Jie Yang, Wei Liu, and Jie Hu, "Gim: A million-scale benchmark for generative image manipulation detection and localization," *AAAI*, 2025.

[14] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer, "High-resolution image synthesis with latent diffusion models," in *CVPR*, 2022.

[15] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach, "SDXL: Improving latent diffusion models for high-resolution image synthesis," in *ICLR*, 2024.

[16] Black Forest Labs, "Flux," `https://github.com/black-forest-labs/flux`, 2024.

[17] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *NeurIPS*, 2017.

[18] Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong, "Imagereward: learning and evaluating human preferences for text-to-image generation," in *NeurIPS*, 2023.

[19] Haiwei Wu and Jiantao Zhou, "Iid-net: Image inpainting detection network via neural architecture search and attention," *IEEE TCSVT*, 2021.

[20] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu, "Pscc-net: Progressive spatio-channel correlation network for image manipulation detection and localization," *IEEE TCSVT*, 2022.

[21] Junke Wang, Zuxuan Wu, Jingjing Chen, Xintong Han, Abhinav Shrivastava, Ser-Nam Lim, and Yu-Gang Jiang, "Objectformer for image manipulation detection and localization," in *CVPR*, 2022.

[22] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," in *NeurIPS*, 2021.

[23] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky, "Resolution-robust large mask inpainting with fourier convolutions," in *WACV*, 2022.