

DeCLIP: Decoding CLIP representations for deepfake localization

Stefan Smeu
Bitdefender

ssmeu@bitdefender.com

Elisabeta Oneata
Bitdefender

eoneata@bitdefender.com

Dan Oneata

POLITEHNICA Bucharest

dan.oneata@gmail.com

Abstract

Generative models can create entirely new images, but they can also partially modify real images in ways that are undetectable to the human eye. In this paper, we address the challenge of automatically detecting such local manipulations. One of the most pressing problems in deepfake detection remains the ability of models to generalize to different classes of generators. In the case of fully manipulated images, representations extracted from large self-supervised models (such as CLIP) provide a promising direction towards more robust detectors. Here, we introduce DeCLIP—a first attempt to leverage such large pretrained features for detecting local manipulations. We show that, when combined with a reasonably large convolutional decoder, pretrained self-supervised representations are able to perform localization and improve generalization capabilities over existing methods. Unlike previous work, our approach is able to perform localization on the challenging case of latent diffusion models, where the entire image is affected by the fingerprint of the generator. Moreover, we observe that this type of data, which combines local semantic information with a global fingerprint, provides more stable generalization than other categories of generative methods.

1. Introduction

This paper addresses the task of localizing manipulations in partially altered images. For example, given a video of a political figure whose mouth has been manipulated to make it look like they are uttering a certain sentence, we want to automatically identify this region as fake. This type of manipulation, where most of the context is real and only a small part is manipulated, is both highly deceptive because of the real context, and easy to achieve because of the wide availability of inpainting techniques. Precise localization of partial manipulations prevents this common type of attack and provides a richer and more interpretable output than detection methods, which output a binary label (fake or real).

The main challenge of deepfake localization (and deepfake detection in general) remains the ability to general-

ize. When training and test data are generated by similar methods, detection is possible [63], but when test data is generated by unseen methods, performance drops sharply [22, 32, 35]. Deepfake detectors, which are typically high-capacity networks, rely on *fingerprints* [40, 69]—imperceptible patterns left by the generator. But these fingerprints are sensitive to the generator (type [7, 52], training data [40, 69], seed [69]) hindering out-of-domain performance. Recently, it has been shown that is possible to replace the very flexible detectors with representations produced by self-supervised models. Specifically, Ojha *et al.* [44] extract features from the pretrained CLIP model [49] and use a linear classifier on top to distinguish fake from real images. This simple approach shows strong generalization across a wide range of generators. However, this method was only applied to fully manipulated images and used to predict image-level labels.

Our idea is to exploit the intrinsic generalization capability of CLIP features for the localization task. To this end, we fill the gap in the literature by first evaluating these self-supervised representations for locally manipulated images and then integrating them for localization. Locally manipulated images are more challenging to detect than fully manipulated images because the features may not capture the fine details as well. Our results show that indeed the use of CLIP features to expose locally manipulated images as fakes fails to a large extent. But we are able to mitigate this problem by equipping the model with a more powerful decoder that can make better use of the local content.

To validate our method we use the Dolos dataset [59]. This dataset consists of face images whose face attributes (such as mouth, hair, eyes) have been inpainted using four methods: two diffusion and two GAN methods. Due to the small local changes, the narrow domain and modern generation methods, many of the images have good perceptual quality. This is different and more challenging from the images on which CLIP features have generally been applied. Many of these datasets (such as the one generated with ProGAN [25], which is used for training) exhibit clear visible semantic artifacts, which may aid generalization.

An intriguing case in Dolos is its subset inpainted with a

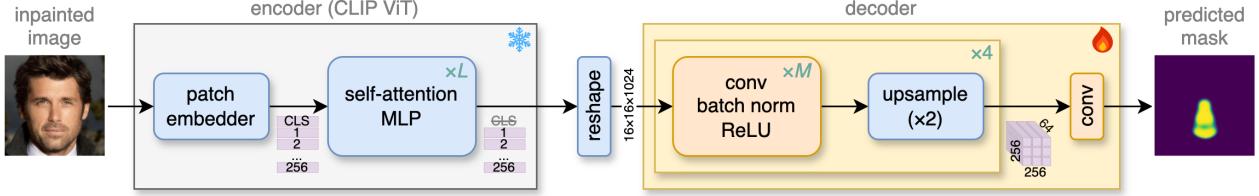


Figure 1. Method overview. We perform manipulation localization by decoding the information from the frozen CLIP embeddings using a learnt convolutional decoder. The embeddings are extracted at an arbitrary layer L and upsampled progressively by the decoder.

latent diffusion model (LDM). The original paper achieved poor results even in the in-domain setting (training and testing on LDM). The authors speculated that this happens because LDM carries the inpainting in the latent space and the final upscaling step leaves artifacts throughout the image. We validate this claim by conducting studies on images with clean background. More importantly, we show that our CLIP-based approach is able to perform localization on the original LDM-inpainted images. Even more, we observe that training on LDM generalizes well to other generators, a behaviour that cannot be achieved by training on a different generator or using conventional data augmentation.

Our work makes the following contributions: (i) We demonstrate that large pretrained representations can effectively be used for deepfake detection and improve generalization over existing methods. (ii) We present a comprehensive study of the factors that contribute to our model: backbone type, layer, decoder type, and decoder size. (iii) We achieve high-accuracy manipulation localization in the challenging case of LDM-inpainted images. Furthermore, we show that training on this type of data improves generalization over other types of data. Our code is available at: <https://github.com/bit-ml/DeCLIP>.

2. Related Work

In response to advances in generative modelling, a growing body of research is devoted to exposing fake content; see [32, 38, 41, 43, 60, 61] for reviews. We survey two directions related to our approach, namely the emerging trend of relying on self-supervised representations for deepfake detection and techniques for the task of deepfake localization.

Self-supervised representations in deepfake detection. Learning transferable representations from unlabelled data has seen impressive progress in recent years [9, 23, 46, 49]. Many of these representations have also been successfully applied to the task of deepfake image detection: in particular, the CLIP representations [49] have been the most widely used [8, 27, 28, 34, 44, 51, 56, 73], but features from other vision-language models (such as BLIP2 [31] or InstructBLIP [9]) or vision-only models (such as DINO v2 [46] or MoCo v3 [5]) have also been employed for deep-

fake detection [4, 26, 42, 51, 67]. These representations are either kept frozen [51] and probed linearly [44], or adapted to the task by full [27, 56] or partial [42] fine-tuning, prompt tuning [4, 27], adapter techniques [26, 27, 34]. The adaptation process can be done most simply by optimising the binary cross-entropy loss [44, 56], but more recent methods have experimented with the contrastive loss [28], a teacher-student paradigm [73], or ways of incorporating the text encoder in the learning process [27, 34, 56]. A similar trend of relying on self-supervised representations can be noticed for deepfake detection on other modalities: video [13, 17, 45] and audio [47, 48, 64].

Manipulation localization. Local manipulations are the result of low-level image editing techniques (splicing [11] copy-move [65], object removal [55]) or deep learning approaches (face swapping [54], in-painting [37, 68]). Many of the detection approaches rely on a combination of frequency information [16, 36, 62] noise information [15, 29, 30, 39, 72] and consistency checks [1, 21]. In terms of architecture, convolutional [30, 36, 39, 66] and self-attention [10, 16, 18, 57, 62] layers are typically employed. The most common loss is the pixel-level binary cross entropy [20, 29, 39] or variations such as focal [30] or Dice loss [15]; this is sometimes coupled with an image-level loss [62, 70] or used in a multitask setting [16]. While supervised learning is the typical setup, localisation maps can also be extracted in a weakly-supervised way [59], providing explanations that can help either humans [14] or algorithms [2] improve. Generalisation is explicitly considered in a few works [15, 29], but these focus on the more traditional manipulations of copy-move and splicing.

3. Overview and preliminaries

Our goal is to perform localization of manipulated areas in images. Our approach is based on CLIP features (Sect. 3.1), since these were shown to yield strong generalization performance for the related task of deepfake detection (classifying if an entire image is fake or real). However, CLIP features were never evaluated in the context of *locally*-manipulated images. Here, we consider the Dolos dataset (Sect. 3.2), a challenging and carefully-constructed

Method	Train data	Test data: Dolos	
		P2 full	P2 local
1 CLIP + linear	ProGAN	93.4	72.8
2 CLIP + linear	Dolos: P2 full	98.9	79.2
3 Patch Forensics	Dolos: P2 full	100.0	95.3
4 CLIP + linear	Dolos: P2 local	97.2	71.4

Table 1. The impact of full versus local manipulations for detection. We report the average precision for image-level deepfake detection on the P2 subset from the Dolos dataset. While CLIP + linear obtains good performance on the fully-generated images from Dolos, it fails to work on locally-generated images.

dataset which disentangles multiple axes of image generation. We perform for the first time (image-level) deepfake detection with CLIP on Dolos (Sect. 3.3) and show that CLIP in its original instantiation struggles detecting local images; we address this problem in the next section.

3.1. CLIP features

CLIP (contrastive language–image pretraining) [49] is a foundation vision-language model trained on over 400M image–text pairs scrapped automatically from the web. Its architecture is composed of two encoders—an image and a text encoder—which are trained to minimize the contrastive InfoNCE loss. Radford *et al.* have shown that this model learns visual features that are highly transferable across various tasks. Recently, Ojha *et al.* [44] have extended this observation by showing that the features extracted from the frozen CLIP image encoder can discriminate between fake and real images. The simple approach of applying a linear classifier on CLIP features works well not only in-domain, but, more importantly, it generalizes better than prior work on a number of different datasets, such as diffusion-generated image, video data, low-level image manipulations. Among the image encoder architectures provided by CLIP, Ojha *et al.* have shown that the visual transformer [12] performs better than the residual network [19].

3.2. Dolos dataset

Dolos [59] is a recently introduced dataset of locally manipulated faces. The dataset has been used to analyse the capabilities of weakly-supervised deepfake methods, and as such it provides a controlled setup over three components of image generation: inpainting type (local, full), model family (P2 [6], LDM [53], LaMa [58], Pluralistic [71]), and training data (CelebA-HQ, FFHQ). We use the inpainting type information to study the effect of local manipulations (Sect. 3.3) and the model family information to study generalization across generators (Sect. 4.2). Regarding the generator training data, we restrict ourselves to the CelebA-HQ variants. The generated images (especially those produced

by diffusion models—P2 and LDM) are highly realistic, making Dolos a challenging out-of-domain dataset.

3.3. Detection with CLIP on Dolos

The majority of datasets considered by Ojha *et al.* [44] are fully-generated images. But how does the CLIP-based model of [44] perform on partially-manipulated images? To answer this question, we consider images from Dolos inpainted with the P2 diffusion model, for which we have both fully and locally generated images. We report average precision for image-level detection. Table 1 shows the results for multiple combinations of methods and training data.

First, we observe that applying the original method of Ojha *et al.* (CLIP + linear trained on ProGAN) on fully-generated images from Dolos yields an average precision of 93.4% (row: 1, col: P2 full). This performance is similar to the average performance of 93.3% reported in Table 2 from [44], which indicates good generalization, as we move to a different domain (from general images to faces) and to a different generator (from GAN to diffusion).

However, the pretrained CLIP + linear model does not work as well on local manipulations, as the performance drops from 93.4% to 72.8% (row 1). This conclusion is also supported by the results on face swap manipulations (another type of local manipulations): Ojha *et al.* report 82.5% AP (Table 9, col: 9, “Deepfakes”), which is the second lowest performance out of the 19 datasets used therein.

Importantly, the performance on local manipulations is not improved even if train on in-domain data: training on either P2 full or P2 local still yields only 79.2% (row 2) or 71.4% (row 4), respectively. On the other hand, Patch Forensics [3], which was used as a baseline in [59], is not affected local manipulations: it achieves an average precision of 95.3% (row 3). This result serves as our motivation for developing a patch-based approach on CLIP features. The full model, which we describe in the next section, is able to match the performance of Patch Forensics on local images, while maintaining good generalization performance.

4. Deepfake localization with CLIP

Given an image that has been manipulated locally, our aim is to produce a map showcasing where the manipulation has occurred: values close to 1 indicate that the corresponding pixel has been altered; values close to 0 indicate that the pixel is authentic. We assume a fully supervised setting in which we have access to images and groundtruths maps; this setup is reminiscent to the one encountered in the object segmentation task.

Our main idea is to leverage high quality pretrained image representations and couple them with an appropriate decoder trained for deepfake manipulation localization. This is achieved by two components: an image encoder, which encodes the image as a low-resolution grid of features, and a

decoder, which upscales the encoded representations to the higher-resolution of the input image. The resulting method, which we name DeCLIP, is shown in Figure 1.

Encoder. We extract representations from both pre-trained CLIP image architectures (visual transformer and residual network) at various layers. For the visual transformer, we choose the ViT-L/14 variant, which operates on 16×16 patches of size 14×14 and has 24 self-attention layers; each layer outputs 256 1024-dimensional embeddings of the input patches and one additional global CLS token, which we discard. For the residual network, we use the ResNet-50 variant. This variant has four blocks: after the first block the output is a 56×56 256-dimensional embedding; with each subsequent block the embedding dimension doubles, while the spatial resolution halves.

Decoder. We decode the information from the CLIP representations using a convolutional-based architecture. This architecture consists of four blocks, each sequencing M sub-blocks and a $\times 2$ bilinear upsampling layer. A sub-block is composed of a 5×5 convolutional layer followed by batch normalization and ReLU activations. To project the output to the grayscale mask space, we use a final 5×5 convolutional layer. The decoders mentioned throughout the paper are conv- $\{4, 12, 20\}$, where the number indicates the total number of sub-blocks ($4M$) and controls the decoder size.

The choices regarding the encoder backbone, layer at which features are extracted, decoder size are analysed in Sects. 4.2 and 4.3.

4.1. Experimental setup

Dataset and metrics. We report results on the four locally-manipulated subsets from the Dolos dataset (Sect. 3.2): LaMa, Pluralistic, LDM, P2. We consider all 16 train-test combinations (4 train \times 4 test) and report the intersection over union (IoU) between the predicted binary mask and the groundtruth mask. We obtain binary predictions by using a fixed threshold of 0.5 over the continuous predictions. To ease comparison between methods, we report aggregated metrics. We consider the averaged IoU based on whether the train and test dataset match. **ID IoU** (in-domain intersection-over-union) is computed as the average IoU over the 4 combinations when the training and test sets match. It serves as a topline, measuring the difficulty of the chosen datasets (how well the detector can learn patterns inflicted by a fixed deepfake generator). **OOD IoU** (out-of-domain intersection-over-union) is computed as the average IoU over the 12 combinations when the training and test sets differ. It measures the generalization to unseen data and the model’s capability to handle diverse variations in the data.

Implementation details. We adapt the training setup for deepfake detection from [44] for localization. We optimize the binary cross-entropy loss between the predicted

	Backbone	Decoder	IoU \uparrow	
			ID	OOD
<i>Patch Forensics variants</i> [3, 59]				
1	Xception (L2)	linear	69.3	20.4
2	Xception (L2)	conv-20	70.4	12.6
<i>CLIP variants</i> [44]				
3	ViT-L/14 (L24)	linear	36.2	18.5
<i>Other methods</i>				
4	PSCC-Net [36]		81.0	21.6
5	CAT-Net [29]		18.5	17.9
<i>DeCLIP variants</i>				
6	ViT-L/14 (L21)	conv-20	67.9	32.6
7	RN50 (L3)	conv-20	71.0	32.0
8	ViT-L/14 (L21) + RN50 (L3)	conv-20	73.8	34.7

Table 2. Comparison of deepfake localization methods on Dolos. We report in-domain (ID) IoU (averaged over the four datasets in Dolos) and out-of-domain (OOD) IoU (averaged over the twelve train-test combinations, where train and test sets differ). DeCLIP performs better in the OOD scenario, while also showing good ID performance, being outperformed only by PSCC-Net.

and groundtruth masks. The hyper-parameters are kept the same. Specifically, we use the Adam optimizer with an initial learning rate of 10^{-3} , which is reduced by a factor of 10 using a patience of five epochs. The training is stopped when the learning rate decreases under 10^{-6} .

4.2. Main results

Baselines. We compare DeCLIP against the following baselines: (i) Patch Forensics was proposed in [3] and used in [59] for weakly-supervised localization and cross-generator localization on the Dolos dataset. The original method extracts features from block 2 of the Xception network and projects them to binary predictions using a 1×1 convolutional layer. Since this last step is equivalent to a linear decoder, we also experiment with a stronger 20-layer convolutional decoder. (ii) CLIP:ViT-L/14-linear is the method proposed in [44] for image-level detection, which we adapt minimally for localization: instead of using the CLS token we use the feature map extracted at the last (L24) layer of ViT-L/14 encoder and learn a linear patch classifier on top. (iii) PSCC-Net [36] learns to extract local and global features from images and estimates manipulation masks at multiple scales. (iv) CAT-Net [29] uses discrete cosine transform coefficients to learn compression artifacts to localize image manipulations.

Quantitative results. Our main results are shown in Table 2. For all methods we use all 16 train-test combinations of generators in Dolos and average IoU results for ID and OOD setups. Patch Forensics (row 1), the method that was

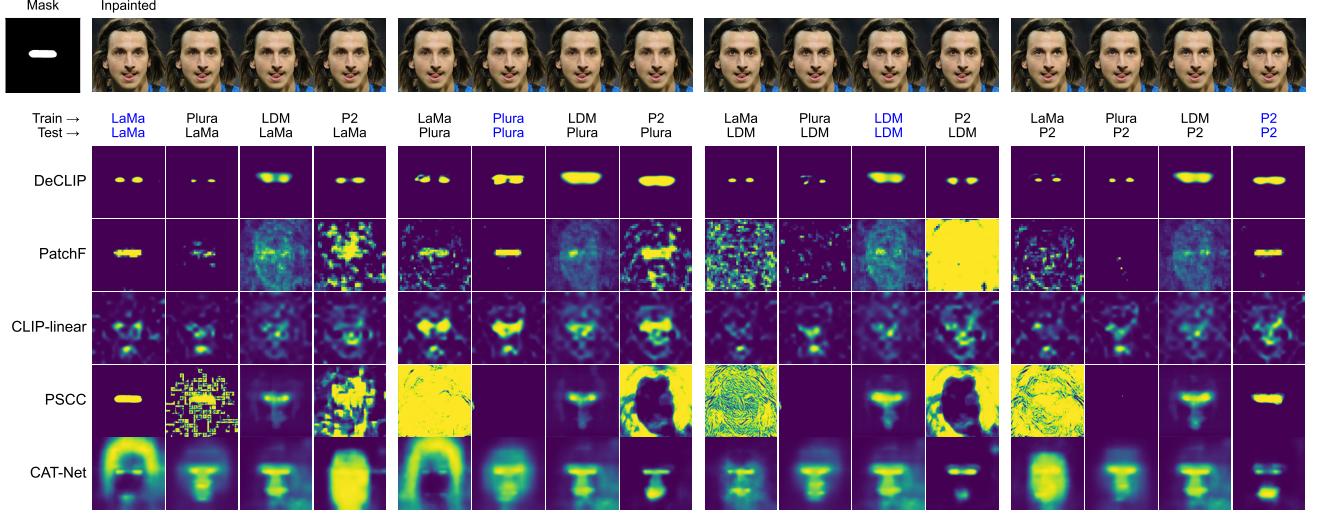


Figure 2. Sample predictions for DeCLIP (second row) and four other methods (Patch Forensics, CLIP-linear, PSCC, CAT-Net) on all 16 train–test combinations from the Dolos dataset. The in-domain combinations are highlighted in blue; the others are out-of-domain combinations. The black-and-white image in the top left corner shows the inpainting mask (white is the inpainted region) and the rest of the images in the first row are the inpainted images with one of the four test datasets (LaMa, Pluralistic, LDM, P2).

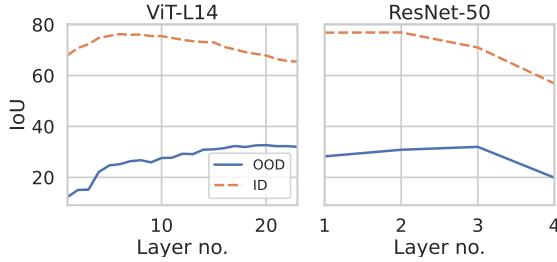


Figure 3. The impact of the layer at which the features are extracted for the ViT-L/14 (left) and ResNet-50 (right) backbone. We report IoU performance on the Dolos dataset both in-domain (ID, orange dashed line) and out-of-domain (OOD, blue solid line).

originally applied for this task, shows good ID performance but performs poorly in OOD. Our starting point, the original CLIP method (row 3) has much worse performance in ID than Patch Forensics and comparable performance in OOD. Rows 6–8 show variants of our method, DeCLIP, which bring a significant boost to the original CLIP, improving both in ID and OOD setups. Compared to Patch Forensics, it has similar good performance in ID, but a 50% relative improvement in OOD. We also experiment with adding a larger decoder to Patch Forensics (row 2), but it did not help improve the OOD performance. Rows 4 and 5 show comparisons with other methods, PSCC-Net and CAT-Net that have been re-trained and tested in the exact same scenarios as DeCLIP. Both have significantly lower performance in ODD compared to DeCLIP. Their behaviour is different in ID: PSCC has good performance, while CAT-Net very poor.

Qualitative results. We show examples of output localization masks for all train-test setups produced by DeCLIP (ViT-L/14) and other methods in Figure 2. Notice that DeCLIP produced more consistent and clean masks even in the harder, OOD scenarios. PSCC-Net and Patch Forensics generally perform well in ID (with the exception of LDM-LDM case), but struggle in the OOD scenarios. CAT-Net and CLIP:ViT-L/14-linear seem to learn general face features, not related to the actual inpainted region.

4.3. Ablations

Backbones and representations depth. For both ViT-L/14 and ResNet-50 backbones we vary the depth of the layer at which we extract the pretrained representations. Our results are shown in Figure 3. When using ResNet-50, representations extracted at lower convolutional blocks (L1, L2) are best for manipulation localization in ID, while representations extracted at L3 block is best for OOD. The last block, L4 has lower performance both in ID and OOD. In the case of ViT-L/14, a similar trend can be seen with ID localization being higher when using features extracted at lower layers (L7) while OOD localization accuracy increasing when using higher level features (L21). Unlike ResNet-50, for ViT-L/14 there is no significant drop in performance at the last layer in the OOD scenario.

Decoder architecture. We experiment with three types of decoders: linear, convolutional and self-attention. For the convolutional one we vary the depth and choose 4, 12 and 20 sub-blocks. The self-attention decoder has 2 attention blocks. Each attention block has 16 heads, a hidden size

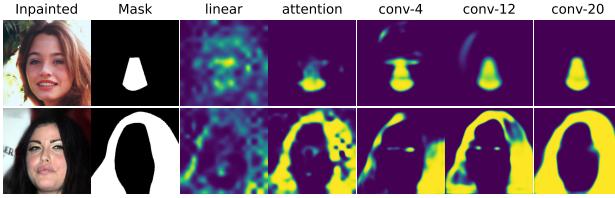


Figure 4. Predicted masks obtained with different decoders. All results use DeCLIP ViT-L/14 variant. First row shows the LDM–P2 scenario, while the second P2–LaMa. The larger convolutional decoder produces more smooth and precise results.

of 1024, associated with a MLP of size 4096. For all decoders, we use bilinear upsampling. Results are shown in Table 3 for DeCLIP with ViT-L/14 backbone. The convolutional decoder outperforms both the linear and the self-attention one. Moreover, the larger the decoder, the better the performance both in ID and OOD scenarios. This indicates that localization manipulation needs larger decoders to properly make use of pretrained representations. Visual examples are shown in Figure 4 for two train-test scenarios: LDM–Pluralistic and P2–LaMa. The identified manipulation mask becomes more precise (less erosion, fewer holes) as we move from the linear decoder to the attention-based one and convolutional decoders.

Method		IoU \uparrow		
Backbone	Decoder	Params.	ID	OOD
ViT-L/14	linear	1.0×10^3	39.9	19.2
ViT-L/14	attention	25.1×10^6	61.8	27.9
ViT-L/14	conv-4	17.4×10^6	65.3	28.6
ViT-L/14	conv-12	34.8×10^6	66.8	31.0
ViT-L/14	conv-20	52.2×10^6	67.9	32.6

Table 3. Influence of decoder type and size on manipulation localization with DeCLIP ViT-L/14. The convolutional decoder outperforms linear and self-attention ones. The larger decoder (with 20 convolutional sub-blocks) performs best in both ID and OOD.

4.4. Detailed results

In the previous sections, in order to summarize the localization performance of different models we used aggregate measures over in-domain and out-of-domain train–test combinations. Here we provide a more detailed view by showing the results of each train–test combination. Figure 5 shows these results for the Patch Forensics method [59] and for DeCLIP with either ViT-L/14 or ResNet-50 backbone. The diagonal of these cross-generator matrices shows the in-domain performance for each dataset. We see that Patch Forensics is slightly more accurate for three of the generators (P2, LaMa, Pluralistic), but it fails completely on the LDM generator. DeCLIP, on the other hand, is more stable on all four datasets and even gives good results on

	Patch Forensics	Patch Forensics				DeCLIP ViT-L/14				DeCLIP RN-50			
		LDM	P2	LaMa	Plura	LDM	P2	LaMa	Plura	LDM	P2	LaMa	Plura
test on	Patch Forensics	18.1	19.5	11.4	0.5	49.1	29.7	13.6	9.5	44.1	24.0	22.0	10.6
		11.5	84.1	10.3	0.2	43.7	66.4	20.1	16.4	37.8	73.8	33.8	10.5
		7.5	38.0	88.7	44.9	42.6	30.7	75.6	20.3	23.5	50.0	84.4	60.9
		11.5	41.4	48.5	86.4	55.5	59.1	50.3	80.4	35.4	34.6	40.5	81.6
test on	DeCLIP ViT-L/14	LDM	P2	LaMa	Plura	train on	LDM	P2	LaMa	Plura	train on	LDM	P2
		49.1	29.7	13.6	9.5		43.7	66.4	20.1	16.4		44.1	24.0
		42.6	30.7	75.6	20.3		42.6	30.7	75.6	20.3		37.8	73.8
		55.5	59.1	50.3	80.4		55.5	59.1	50.3	80.4		23.5	50.0
test on	DeCLIP RN-50	LDM	P2	LaMa	Plura	train on	LDM	P2	LaMa	Plura	train on	LDM	P2
		44.1	24.0	22.0	10.6		44.1	24.0	22.0	10.6		44.1	24.0
		37.8	73.8	33.8	10.5		37.8	73.8	33.8	10.5		37.8	73.8
		23.5	50.0	84.4	60.9		23.5	50.0	84.4	60.9		23.5	50.0

Figure 5. Detailed cross-generator performance on the Dolos dataset for three methods: Patch Forensics [59], DeCLIP with ViT-L/14 backbone at layer 21, DeCLIP with ResNet-50 backbone at layer 3. Both DeCLIP variants use the conv-20 decoder.

LDM data (44.1% and 49.1%, respectively). Looking at the columns, we can see how well one dataset transfers to the others. Interestingly, we see that when training on LDM, DeCLIP also generalizes well to other test datasets. This is not the case when training on Pluralistic, whose performance on P2 and LDM is low for all the methods shown; this suggests that Pluralistic fingerprints have very little in common with those produced by diffusion-based generators (P2 or LDM). The case of LDM is worth further investigation, which we do in the next section (Sect. 5). Finally, we observe the complementarity of the two DeCLIP variants. Even if on average their in-domain and out-of-domain IoUs are similar (see Table 2, rows 6–7), there are train–test combinations where the two backbones show contrasting behaviours: for example, from Pluralistic to LaMa, the ResNet backbone performs better (60.9 versus 20.3); from P2 to Pluralistic, ViT performs better (59.1 versus 34.6). For this reason, concatenating the representations from the two backbones improves both ID and OOD performance compared to their individual results (see Table 2, row 8).

5. The case of LDM-inpainted images

We have seen in Sect. 4.4 that localizing manipulations in images inpainted with LDM is more challenging than performing this task on images inpainted with other techniques (P2, LaMa, Pluralistic). Furthermore, we observed that training DeCLIP on LDM data gives a strong out-of-domain performance. What is the reason for this?

First, we recall that LDM provides an atypical case of image inpainting. Unlike the other three inpainting methods considered, LDM inpainting takes place in the *latent* space. As such, the generated latent image must be projected back to the pixel space. This upscaling step is performed by a variational autoencoder (VAE) network, which leaves artifacts throughout the entire generated image, not just in the inpainted regions as for the other three methods. These artifacts, although imperceptible, are detectable by the networks and is what makes localization challenging. In what follows we conduct multiple analyses to understand: (i) the

Train	Out region		In region		Test data: LDM variants			Test data: OOD datasets		
	Content	Fingerprint	Content	Fingerprint	LDM/real	LDM/clean	LDM	P2	LaMa	Plura
1 LDM/real	real		real	✓	62.1	62.2	24.5	56.9	23.2	43.2
2 LDM/clean	real		fake	✓	39.7	67.3	38.3	59.6	27.4	56.8
3 LDM	real	✓	fake	✓	17.1	53.1	49.1	43.7	42.6	55.5

Table 4. The impact of the LDM fingerprint on the inpainted (in) and background (out) regions. We report IoU using DeCLIP with ViT-L/14 backbone, layer 21 and conv-20 decoder.

Method	Params. ×10 ⁶	Test data			
		LDM	P2	LaMa	Plura
PatchF. (linear) [3]	0.2	18.1	19.5	11.4	0.5
PatchF. (conv-20)	42.6	39.2	25.6	18.5	24.1
PSCC [36]	3.6	41.5	23.6	26.2	27.6
DeCLIP	52.2	49.1	43.7	42.6	55.5

Table 5. Comparison in terms of IoU of localization methods trained on LDM. DeCLIP uses features from ViT-L/14 layer 21 and conv-20 decoder.

impact of the model capacity; (ii) the impact of the fingerprint left by LDM on the background; (iii) the relationship between the LDM fingerprint and data augmentation; (iv) the performance on general content images.

Larger models improve performance on LDM, but capacity alone is not sufficient for generalization. Patch Forensics fails at localizing even ID manipulations on LDM images, while DeCLIP works better. An important difference between the two is the larger decoder employed by the latter. We verify whether the reason for the difference in performance is solely based on network capacity. We consider two larger variants: Patch Forensics but with the conv-20 decoder (42.6M parameters) and PSCC [36] (3.6M parameters); both of these networks are trained from scratch on the LDM subset. The results in Table 5 show that indeed the network capacity is responsible to a degree for the good performance, as the larger variant of Patch Forensics improves substantially over the smaller baseline. However, PSCC is better than both Patch Forensics variants, while DeCLIP achieves the best performance, at a number of parameters comparable to the larger Patch Forensics variant.

LDM background fingerprint provides stable out-of-domain performance. In-domain performance on LDM is lower than that on the other three generation methods. Conversely, the generalization performance of LDM is much stronger than that of the other generation methods. We investigate the role played by the LDM fingerprint in this behaviour. To disentangle this aspect, we create two variants of LDM datasets:

- LDM/clean, which uses a fingerprint-free background. This variant is created by replacing the background

(the complement of the mask) of the LDM-generated images with information from the original real image.

- LDM/real, which consists of real images with fingerprint on the masked region. This variant is created by passing the real images through LDM using an empty mask and then cleaning the background.

The results are shown in Table 4. Cleaning the background fingerprint improves in-domain results: from 49.1 to 62.1 and 67.3 (see the diagonal along the “LDM variants” columns); this is to be expected since there are no distractors on the background. Relying solely on the fingerprint information (row 1) gives good results on two of the out-of-domain datasets (56.9 on P2 and 43.2 on Plura), suggesting that LDM shares a similar fingerprint to these methods. Conversely, the poor result on LaMa (23.2) indicates that its fingerprint is different. By further manipulating the content of the target region (row 2), we notice stronger results on all three datasets. A possible reason is that this setup is similar to that of the other datasets: the background is clean, while the manipulated region is affected by both low-level and semantic changes. However, the original LDM (row 3) ensures the most consistent out-of-domain performance, improving the performance on LaMa—the most challenging dataset—from 27.4 to 42.6. This might happen because LDM forces the model to disregard the fingerprint and focus on semantic information, which is more transferable.

Low-level data augmentations induce a similar, but weaker effect than the LDM fingerprint. A possible explanation for the improved generalization showcased when training on LDM-inpainted data is that the VAE decoding acts as data augmentation: it introduces low-level artifacts on the entire images, forcing the detection model to be robust to low-level changes and more aware to semantic inconsistencies. This observation raises the question: Would a different low-level data augmentation help with generalization? We experiment with three types of augmentations (Gaussian blur, color jitter and JPEG compression), which we apply on all images from the LDM/clean dataset. The results in Table 7 show the augmentations have a similar effect to that produced by the LDM fingerprint: they level up the results across datasets, by improving performance on the LDM and LaMa datasets, while sometimes hurting per-

Method	COCO-SD	AutoSplice			
		75	90	100	
<i>Pretrained models</i>					
1	PSCC [36]	8.2	1.8	3.6	48.2
2	CAT-Net [29]	0.1	33.1	69.7	76.1
3	TruFor [15]	6.7	20.7	34.1	55.6
4	MantraNet [66]	2.0	3.2	3.2	13.1
<i>Models trained on COCO-SD</i>					
5	Patch Forensics	16.4	28.1	28.4	28.2
6	PSCC	33.4	48.5	50.6	49.4
7	CAT-Net	15.5	0.6	0.8	4.7
8	DeCLIP	51.1	49.6	49.9	50.1

Table 6. Comparison in terms of IoU to other methods on the more general domain provided by MS COCO. DeCLIP shows a more stable performance across all datasets. Values in bold indicate best results in each of the two sections.

Train data	Augmentation			Test data			
	Blur	Color	JPEG	LDM	P2	LaMa	Plura
LDM/clean				38.3	59.6	27.4	56.8
LDM/clean	✓			43.9	52.5	33.9	55.4
LDM/clean		✓		39.7	59.6	23.0	54.4
LDM/clean			✓	47.0	53.9	35.4	43.4
LDM/clean	✓	✓	✓	43.9	48.7	33.4	49.3
LDM				49.1	43.7	42.6	55.5

Table 7. Data augmentation on LDM/clean. We report IoU on Dolos for the DeCLIP model trained on LDM/clean augmented with either blur, color jitter or JPEG compression.

formance on P2 and Pluralistic. However, none of the augmentations nor their combination helps on average as much as the fingerprint artifacts present in LDM.

Conclusions translate to more general domains. While Dolos dataset enabled a careful analysis of deepfake localization in a challenging and realistic setting, it covers only a narrow domain: faces. We verify whether the main conclusions apply to more general images. We inpaint almost 11k images (9k train, 829 validation, 985 test) from MS COCO [33] with Stable Diffusion [53], which is also an LDM model. We select the mask of a random object whose area is larger than 5% and prompt the inpainting model with the original image caption. This dataset, which we name COCO-SD, is used for training the localization models. To evaluate generalization, we use the AutoSplice dataset [24]. This dataset consists of images manipulated by DALLE-2 [50], where specific objects are replaced with other objects. It has three variants differing in terms of the JPEG compression applied: the original uncompressed one (quality factor: 100) and two compressed ones (quality factors of 75 and 90). Table 6 presents the results for DeCLIP, Patch Foresnics, PSCC and CAT-Net. We also show results of pretrained models, for the two

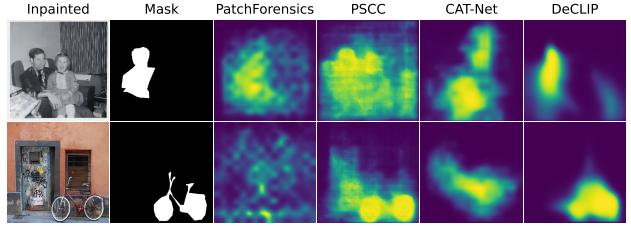


Figure 6. Manipulation localization results on COCO-SD, which has a more challenging set of masks and diverse content. DeCLIP offers a more precise localization of the manipulated area.

last ones, MantraNet [66] and TruFor [15]. We observe that the pretrained models, especially CAT-Net and PSCC, are unstable: they have good performance on some of the datasets, but equally poor on others. MantraNet performs poorly on all datasets, while TruFor has good performance on AutoSplice-100 but lower on its JPEG compressed variants and COCO-SD. Instead, we see that training on the LDM-based COCO-SD dataset offers more stable performance. This is especially true for Patch Forensics, PSCC and DeCLIP, which perform similarly across all three AutoSplice variants. In terms of model comparison, DeCLIP works better in-domain, while slightly outperforming PSCC on the out-of-domain AutoSplice dataset. The visual examples in Figure 6 show that DeCLIP produces more precise localization of the forged object.

6. Conclusion

Our paper presents DeCLIP—a first attempt at decoding large self-supervised representations for manipulation localization. Through extensive experiments we showed that, not only is manipulation localization feasible using these features, but they also significantly improve generalization capabilities in OOD scenarios, with train–test generator mismatch. We conducted a comprehensive analysis of the factors that contribute to the successful decodings of those features: backbone type, layer depth, decoder type and size. Our findings reveal that larger, convolutional decoders improve the quality of the predicted masks compared to linear or self-attention ones. Moreover, VIT-L/14 and ResNet-50 backbones show contrasting behavior when looking at detailed train-test scenarios that can be exploited by combining representations from both backbones. Finally, we showed that, contrary to prior assumptions, manipulation localization can be effectively performed even in the challenging case of LDM data. Interestingly, learning on this type of data offers robustness and improves generalization to other types of local manipulations.

Acknowledgements. This work was funded by EU Horizon projects AI4TRUST (No. 101070190) and ELIAS (No. 101120237).

References

- [1] Susmit Agrawal, Prabhat Kumar, Siddharth Seth, Toufiq Parag, Maneesh Singh, and R. Venkatesh Babu. SISL: Self-supervised image signature learning for splicing detection & localization. In *Proc. CVPR Workshops*, pages 22–32, 2022. 2
- [2] Aidan Boyd, Patrick Tinsley, Kevin W. Bowyer, and Adam Czajka. CYBORG: Blending human saliency into the loss improves deep learning-based synthetic face detection. In *Proc. WACV*, pages 6108–6117, 2023. 2
- [3] Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? Understanding properties that generalize. In *Proc. ECCV*, pages 103–120, 2020. 3, 4, 7
- [4] You-Ming Chang, Chen Yeh, Wei-Chen Chiu, and Ning Yu. AntifakePrompt: Prompt-tuned vision-language models are fake image detectors. *CoRR*, abs/2310.17419, 2023. 2
- [5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proc. ICCV*, pages 9620–9629, 2021. 2
- [6] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon. Perception prioritized training of diffusion models. In *Proc. CVPR*, pages 11462–11471, 2022. 3
- [7] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *Proc. ICASSP*, pages 1–5, 2023. 1
- [8] Davide Cozzolino, Giovanni Poggi, Riccardo Corvi, Matthias Nießner, and Luisa Verdoliva. Raising the bar of AI-generated image detection with CLIP. In *Proc. CVPR Workshops*, pages 4356–4366, 2024. 2
- [9] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven C. H. Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Proc. NeurIPS*, 2023. 2
- [10] Sowmen Das, Md. Saiful Islam, and Md. Ruhul Amin. GCA-Net: Utilizing gated context attention for improving image forgery localization and detection. In *Proc. CVPR Workshops*, pages 81–90, 2022. 2
- [11] Jing Dong, Wei Wang, and Tieniu Tan. CASIA image tampering detection evaluation database. In *China Summit and International Conference on Signal and Information Processing*, pages 422–426, 2013. 2
- [12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. ICLR*, 2021. 3
- [13] Chao Feng, Ziyang Chen, and Andrew Owens. Self-supervised video forensics by audio-visual anomaly detection. In *Proc. CVPR*, pages 10491–10503, 2023. 2
- [14] Camilo Fosco, Emilie Josephs, Alex Andonian, Allen Lee, Xi Wang, and Aude Oliva. Deepfake caricatures: Amplifying attention to artifacts increases deepfake detection by humans and machines. *CoRR*, abs/2206.00535, 2022. 2
- [15] Fabrizio Guillaro, Davide Cozzolino, Avneesh Sud, Nick Dufour, and Luisa Verdoliva. TruFor: Leveraging all-round clues for trustworthy image forgery detection and localization. In *Proc. CVPR*, pages 20606–20615, 2022. 2, 8
- [16] Xiao Guo, Xiaohong Liu, Zhiyuan Ren, Steven Grosz, Iacopo Masi, and Xiaoming Liu. Hierarchical fine-grained image forgery detection and localization. In *Proc. CVPR*, pages 3155–3165, 2023. 2
- [17] Alexandros Haliassos, Rodrigo Mira, Stavros Petridis, and Maja Pantic. Leveraging real talking faces via self-supervision for robust forgery detection. In *Proc. CVPR*, pages 14930–14942, 2022. 2
- [18] Jing Hao, Zhixin Zhang, Shicai Yang, Di Xie, and Shiliang Pu. Transforensics: Image forgery localization with dense self-attention. In *Proc. ICCV*, pages 15055–15064, 2021. 2
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, pages 770–778, 2016. 3
- [20] Xuefeng Hu, Zhihan Zhang, Zhenye Jiang, Syomantak Chaudhuri, Zhenheng Yang, and Ram Nevatia. SPAN: Spatial pyramid attention network for image manipulation localization. In *Proc. CVPR*, pages 312–328, 2020. 2
- [21] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A Efros. Fighting fake news: Image splice detection via learned self-consistency. In *Proc. ECCV*, pages 101–117, 2018. 2
- [22] Marija Ivanovska and Vitomir Struc. On the vulnerability of deepfake detectors to attacks generated by denoising diffusion models. In *Proc. WACV Workshops*, pages 1051–1060, 2024. 1
- [23] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc V. Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *Proc. ICML*, volume 139, pages 4904–4916, 2021. 2
- [24] Shan Jia, Mingzhen Huang, Zhou Zhou, Yan Ju, Jialing Cai, and Siwei Lyu. AutoSplice: A text-prompt manipulated image dataset for media forensics. In *Proc. CVPR Workshops*, pages 893–903, 2023. 8
- [25] Tero Karras, Timo Aila, Samuli Laine, and Jaakkko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 1
- [26] Mamadou Keita, Wassim Hamidouche, Hessen Bouguesfa Eutamene, Abdenour Hadid, and Abdelmalik Taleb-Ahmed. Bi-LORA: A vision-language approach for synthetic image detection. *CoRR*, abs/2404.01959, 2024. 2
- [27] Sohail Ahmed Khan and Duc-Tien Dang-Nguyen. CLIPping the deception: Adapting vision-language models for universal deepfake detection. In *Proc. ICMR*, pages 1006–1015, 2024. 2
- [28] Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. *arXiv preprint arXiv:2402.19091*, 2024. 2
- [29] 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. *Int. J. Comput. Vis.*, 130(8):1875–1895, 2022. 2, 4, 8

- [30] Ang Li, QiuHong Ke, Xingjun Ma, Haiqin Weng, Zhiyuan Zong, Feng Xue, and Rui Zhang. Noise doesn't lie: Towards universal detection of deep inpainting. In *Proc. IJAI*, pages 786–792, 2021. 2
- [31] Junnan Li, Dongxu Li, Silvio Savarese, and Steven C. H. Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. ICML*, volume 202, pages 19730–19742, 2023. 2
- [32] Li Lin, Neeraj Gupta, Yue Zhang, Hainan Ren, Chun-Hao Liu, Feng Ding, Xin Wang, Xin Li, Luisa Verdoliva, and Shu Hu. Detecting multimedia generated by large AI models: A survey. *arXiv preprint arXiv:2402.00045*, 2024. 1, 2
- [33] 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 *Proc. ECCV*, pages 740–755, 2014. 8
- [34] Huan Liu, Zichang Tan, Chuangchuang Tan, Yunchao Wei, Yao Zhao, and Jingdong Wang. Forgery-aware adaptive transformer for generalizable synthetic image detection. *CoRR*, abs/2312.16649, 2023. 2
- [35] Ping Liu, Qiqi Tao, and Joey Tianyi Zhou. Evolving from single-modal to multi-modal facial deepfake detection: A survey. *arXiv preprint arXiv:2406.06965*, 2024. 1
- [36] Xiaohong Liu, Yaojie Liu, Jun Chen, and Xiaoming Liu. PSCC-net: Progressive spatio-channel correlation network for image manipulation detection and localization. *IEEE Trans. Circuits Syst. Video Technol.*, 32(11):7505–7517, 2022. 2, 4, 7, 8
- [37] Andreas Lugmayr, Martin Danelljan, Andrés Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. RePaint: Inpainting using denoising diffusion probabilistic models. In *Proc. CVPR*, pages 11451–11461, 2022. 2
- [38] Asad Malik, Minoru Kuribayashi, Sani M. Abdullahi, and Ahmad Neyaz Khan. Deepfake detection for human face images and videos: A survey. *IEEE Access*, 10:18757–18775, 2022. 2
- [39] Hannes Mareen, Dante Vanden Bussche, Fabrizio Guillaro, Davide Cozzolino, Glenn Van Wallendael, Peter Lambert, and Luisa Verdoliva. Comprint: Image forgery detection and localization using compression fingerprints. In *Proc. ICPR*, volume 13644, pages 281–299, 2022. 2
- [40] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do GANs leave artificial fingerprints? In *IEEE Multimedia Information Processing and Retrieval*, pages 506–511, 2019. 1
- [41] Yisrael Mirsky and Wenke Lee. The creation and detection of deepfakes: A survey. *ACM Computing Surveys (CSUR)*, 54(1):1–41, 2021. 2
- [42] Huy H. Nguyen, Junichi Yamagishi, and Isao Echizen. Exploring self-supervised vision transformers for deepfake detection: A comparative analysis. *CoRR*, abs/2405.00355, 2024. 2
- [43] Thanh Thi Nguyen, Quoc Viet Hung Nguyen, Dung Tien Nguyen, Duc Thanh Nguyen, Thien Huynh-The, Saeid Nahavandi, Thanh Tam Nguyen, Quoc-Viet Pham, and Cuong M Nguyen. Deep learning for deepfakes creation and detection: A survey. *Computer Vision and Image Understanding*, 223:103525, 2022. 2
- [44] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proc. CVPR*, pages 24480–24489, 2023. 1, 2, 3, 4
- [45] Trevine Oorloff, Surya Koppisetty, Nicolò Bonettini, Divyraj Solanki, Ben Colman, Yaser Yacoob, Ali Shahriari, and Gaurav Bharaj. AVFF: Audio-visual feature fusion for video deepfake detection. *CoRR*, abs/2406.02951, 2024. 2
- [46] Maxime Oquab, Timothée Darct, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Mahmoud Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael G. Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Hervé Jégou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piott Bojanowski. DINOv2: Learning robust visual features without supervision. *CoRR*, abs/2304.07193, 2023. 2
- [47] Octavian Pascu, Adriana Stan, Dan Oneata, Elisabeta Oneata, and Horia Cucu. Towards generalisable and calibrated audio deepfake detection with self-supervised representations. In *Interspeech*, 2024. 2
- [48] Alessandro Pianese, Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Training-free deepfake voice recognition by leveraging large-scale pre-trained models. *arXiv preprint arXiv:2405.02179*, 2024. 2
- [49] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *Proc. ICML*, 2021. 1, 2, 3
- [50] 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. 8
- [51] Tal Reiss, Bar Cavia, and Yedid Hoshen. Detecting deepfakes without seeing any. *CoRR*, abs/2311.01458, 2023. 2
- [52] Jonas Ricker, Simon Damm, Thorsten Holz, and Asja Fischer. Towards the detection of diffusion model deepfakes. In *Proc. VISIGRAPP*, pages 446–457, 2024. 1
- [53] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. CVPR*, pages 10684–10695, 2022. 3, 8
- [54] Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *Proc. CVPR*, pages 1–11, 2019. 2
- [55] Min-Cheol Sagong, Yoon-Jae Yeo, Seung-Won Jung, and Sung-Jea Ko. RORD: A real-world object removal dataset. In *Proc. BMVC*, page 542, 2022. 2
- [56] Koushik Srivatsan, Muzammal Naseer, and Karthik Nandakumar. FLIP: Cross-domain face anti-spoofing with language guidance. In *Proc. ICCV*, pages 19628–19639, 2023. 2
- [57] Joel Stehouwer, Hao Dang, Feng Liu, Xiaoming Liu, and Anil K. Jain. On the detection of digital face manipulation. In *Proc. CVPR*, pages 5780–5789, 2019. 2

- [58] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor S. Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proc. WACV*, pages 3172–3182, 2021. 3
- [59] Dragos-Constantin Tantaru, Elisabeta Oneata, and Dan Oneata. Weakly-supervised deepfake localization in diffusion-generated images. In *Proc. WACV*, pages 6246–6256, 2024. 1, 2, 3, 4, 6
- [60] Diangarti Tariang, Riccardo Corvi, Davide Cozzolino, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. Synthetic image verification in the era of generative artificial intelligence: What works and what isn’t there yet. *IEEE Security & Privacy*, 2024. 2
- [61] Luisa Verdoliva. Media forensics and deepfakes: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):910–932, 2020. 2
- [62] 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 *Proc. CVPR*, pages 2354–2363, 2022. 2
- [63] Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A. Efros. CNN-generated images are surprisingly easy to spot... for now. In *Proc. CVPR*, 2020. 1
- [64] Xin Wang and Junichi Yamagishi. Investigating self-supervised front ends for speech spoofing countermeasures. In *Odyssey 2022: The Speaker and Language Recognition Workshop*, pages 100–106, 2022. 2
- [65] Bihai Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. COVERAGE—A novel database for copy-move forgery detection. In *Proc. ICIP*, pages 161–165, 2016. 2
- [66] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. ManTra-Net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *Proc. CVPR*, pages 9543–9552, 2019. 2, 8
- [67] Zebin You, Xinyu Zhang, Hanzhong Guo, Jingdong Wang, and Chongxuan Li. Are image distributions indistinguishable to humans indistinguishable to classifiers? *CoRR*, abs/2405.18029, 2024. 2
- [68] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proc. ICCV*, pages 4471–4480, 2019. 2
- [69] Ning Yu, Larry S. Davis, and Mario Fritz. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proc. ICCV*, October 2019. 1
- [70] Zequin Yu, Jiangqun Ni, Yuzhen Lin, Haoyi Deng, and Bin Li. DiffForensics: Leveraging diffusion prior to image forgery detection and localization. In *Proc. CVPR*, pages 12765–12774, 2024. 2
- [71] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proc. CVPR*, 2019. 3
- [72] Jiaying Zhu, Dong Li, Xueyang Fu, Gang Yang, Jie Huang, Aiping Liu, and Zheng-Jun Zha. Learning discriminative noise guidance for image forgery detection and localization. In *Proc. AAAI*, pages 7739–7747, 2024. 2
- [73] Mingjian Zhu, Hanting Chen, Mouxiao Huang, Wei Li, Hailin Hu, Jie Hu, and Yunhe Wang. GenDet: Towards good generalizations for AI-generated image detection. *CoRR*, abs/2312.08880, 2023. 2

DeCLIP: Decoding CLIP representations for deepfake localization

Supplementary material

1. Additional detailed results

In Figure 1 we provide additional detailed results on all train–test scenarios obtained using the PSCC method and DeCLIP on concatenated representations. Specifically, for DeCLIP we stack together the features from the 21st layer of CLIP ViT-L/14 and the features from 3rd layer of CLIP ResNet-50. The representations extracted from ResNet-50 are bilinearly upsampled from $14 \times 14 \times D$ to $16 \times 16 \times D$ to match the spatial resolution of the features extracted by ViT-L/14; here D denotes the feature dimension. The representations from both networks have the same dimension $D = 1024$. By concatenating the features along the last axis, we obtain a block of size $16 \times 16 \times 2048$, which then fed as input to the conv-20 decoder.

Compared to PSCC, DeCLIP shows better generalization capabilities (results in the out-of-domain setups, off-principal diagonal), especially when trained on LDM and P2. PSCC generally has better in-domain performance (principal diagonal), with the exception of the harder LDM–LDM case, where DeCLIP performs better (51.1% compared to 41.5% IoU).

		PSCC			
		LDM	P2	LaMa	Plura
test on	LDM	51.1	27.4	21.6	4.7
	P2	44.7	74.0	29.2	8.1
	LaMa	43.5	42.4	86.5	51.9
	Plura	56.4	54.2	32.0	83.7
train on		LDM	P2	LaMa	Plura

Figure 1. Detailed cross-generator performance (IoU) on the Dolos dataset (all 16 train–test combinations) for DeCLIP that used both ViT-L/14 and ResNet-50 representations and PSCC.

2. Additional qualitative results on Dolos

In Figures 3, 4, 5, 6 we show detailed visual results on Dolos dataset for all train–test scenarios for DeCLIP as well

as four other methods trained and tested in the same way: Patch Forensics, CLIP-linear, PSCC and CAT-Net. The results show that although some train–test scenarios are considerably harder than the other, DeCLIP offers a plausible manipulation mask in the majority of cases. We showcase different types of masks, from the very small ones that cover only eyes to larger ones that correspond to face and hair. Patch Forensics and PSCC usually work well in domain (with the exception of LDM–LDM scenario), but generally struggle in the out-of-domain cases. CLIP-linear and CAT-Net struggle both in domain and out of domain, producing masks with arbitrary activations that follow the face characteristics.

3. Additional qualitative results on COCO-SD

In Figures 7 and 8 we provide additional results on COCO-SD dataset for DeCLIP, Patch Forensics, PSCC and CAT-Net. Notice that even in a diverse visual domain, with arbitrary-shaped inpainted regions, DeCLIP has a more stable and precise localization of the manipulated area. The dataset is particularly hard as the inpainted objects are often parts of a larger one (e.g. the tie, the drawing of the dog on a cup), represent a single entity among similar of the same type (the doughnut, the bowl). Even in these conditions, DeCLIP provides plausible maps of the inpainting.

4. Illustration of LDM images

In Figure 2 we show how fingerprint and fake content are distributed in different types of LDM images. Green color corresponds to real content, red color corresponds to fake content and red dots symbolize fingerprint.

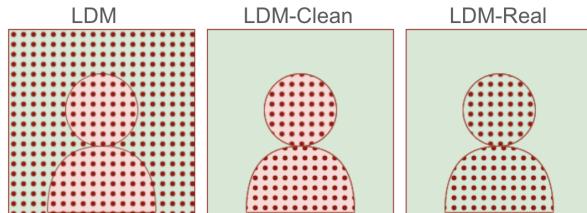


Figure 2. Schematic view of different types of inpaintings with LDM considered in Section 5, Table 4 in the main paper.

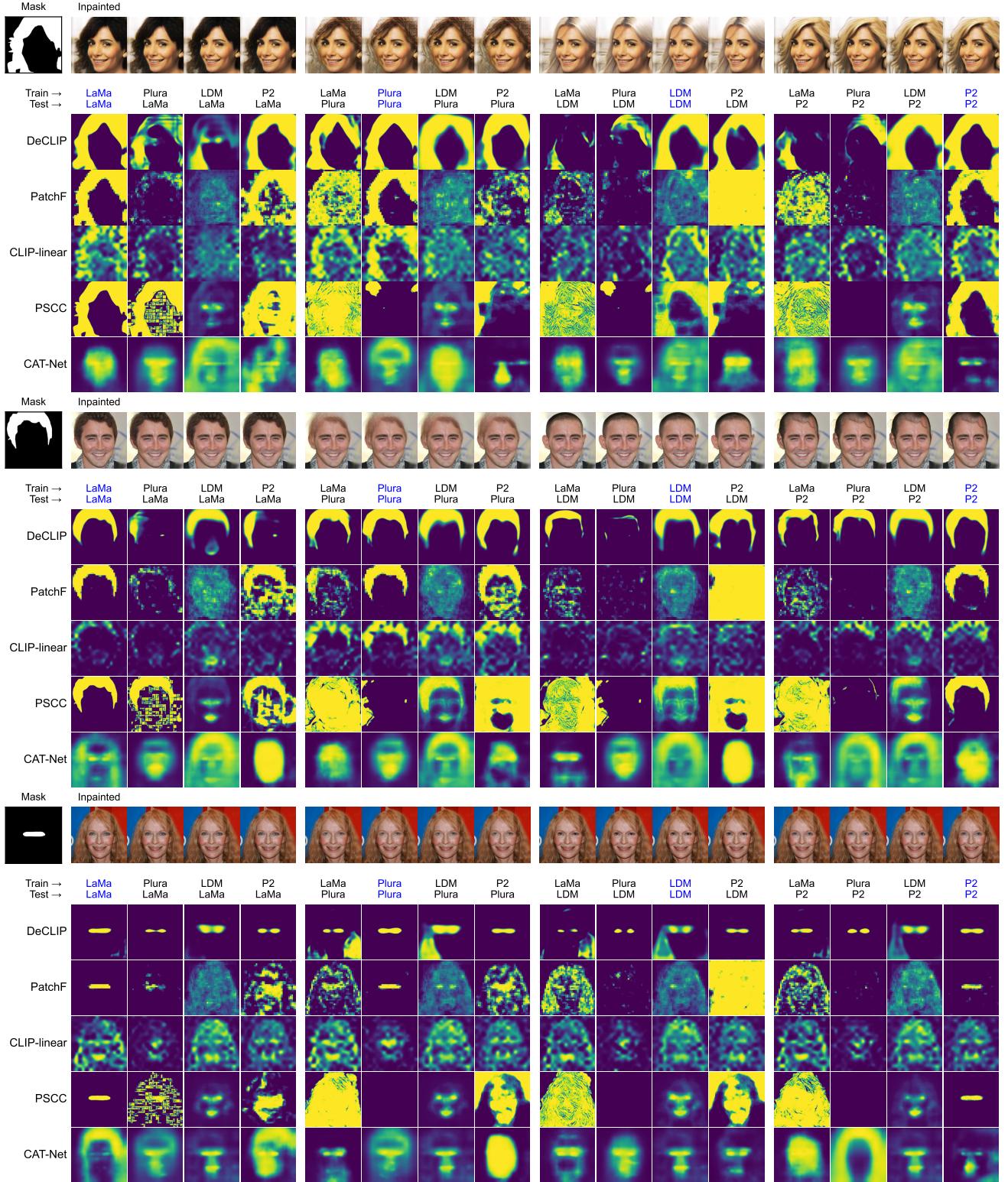


Figure 3. Sample predictions for DeCLIP (second row) and four other methods (Patch Forensics, CLIP-linear, PSCC, CAT-Net) on all 16 train–test combinations from the Dolos dataset. The in-domain combinations are highlighted in blue; the others are out-of-domain combinations. The black-and-white image in the top left corner shows the inpainting mask (white is the inpainted region) and the rest of the images in the first row are the inpainted images with one of the four test datasets (LaMa, Pluralistic, LDM, P2).

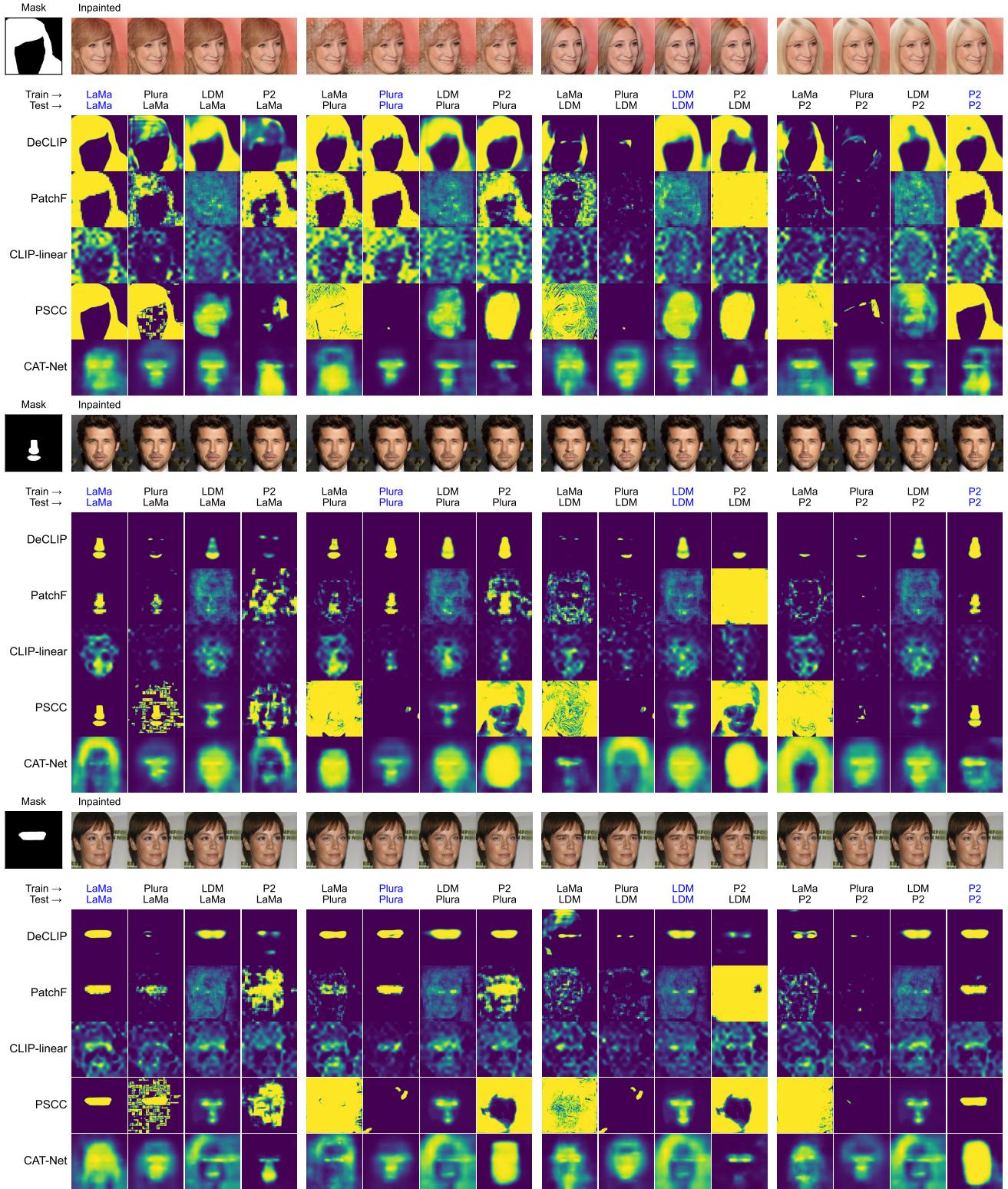


Figure 4. Sample predictions for DeCLIP (second row) and four other methods (Patch Forensics, CLIP-linear, PSCC, CAT-Net) on all 16 train–test combinations from the Dolos dataset. The in-domain combinations are highlighted in blue; the others are out-of-domain combinations. The black-and-white image in the top left corner shows the inpainting mask (white is the inpainted region) and the rest of the images in the first row are the inpainted images with one of the four test datasets (LaMa, Pluralistic, LDM, P2).

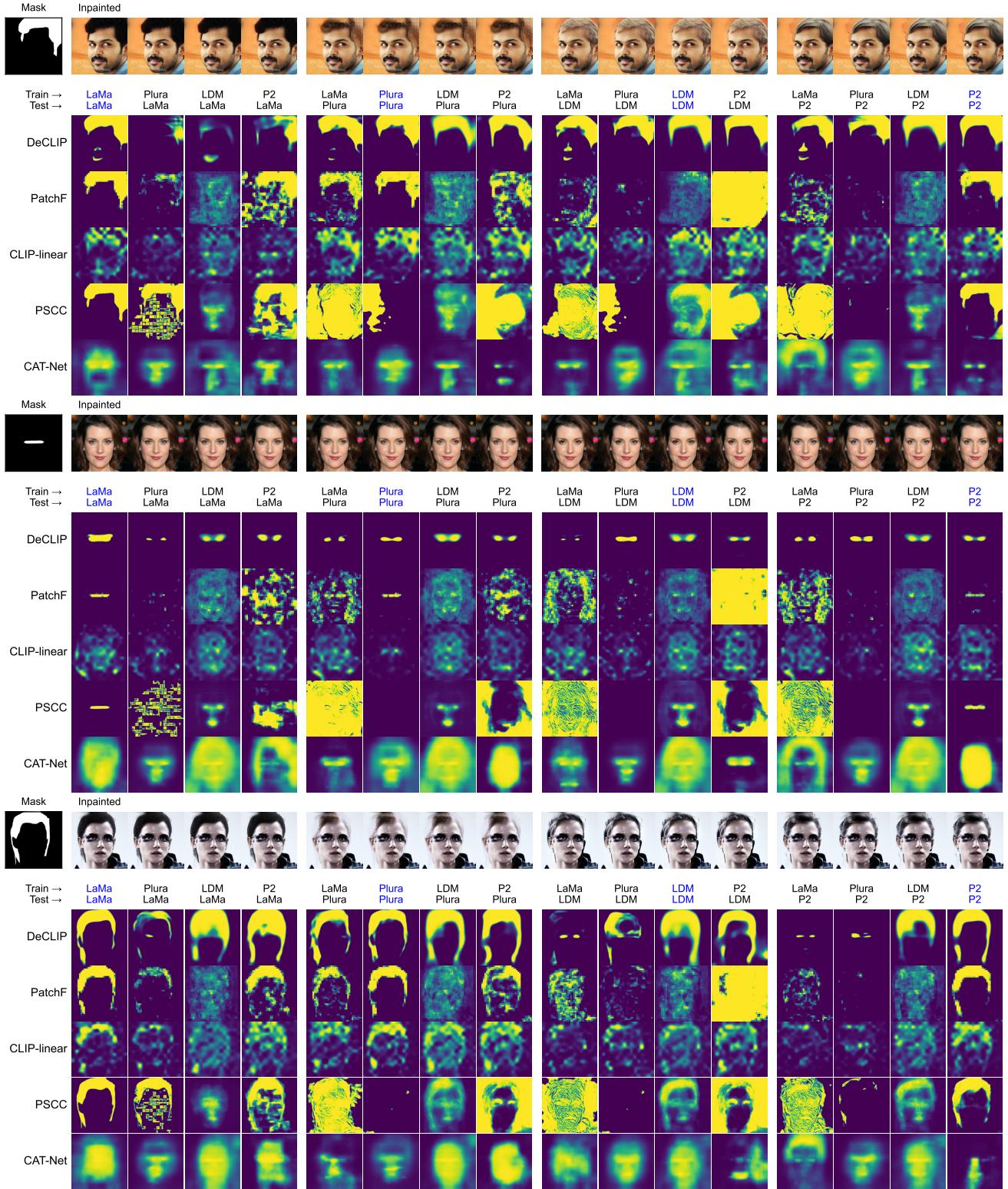


Figure 5. Sample predictions for DeCLIP (second row) and four other methods (Patch Forensics, CLIP-linear, PSCC, CAT-Net) on all 16 train–test combinations from the Dolos dataset. The in-domain combinations are highlighted in blue; the others are out-of-domain combinations. The black-and-white image in the top left corner shows the inpainting mask (white is the inpainted region) and the rest of the images in the first row are the inpainted images with one of the four test datasets (LaMa, Pluralistic, LDM, P2).

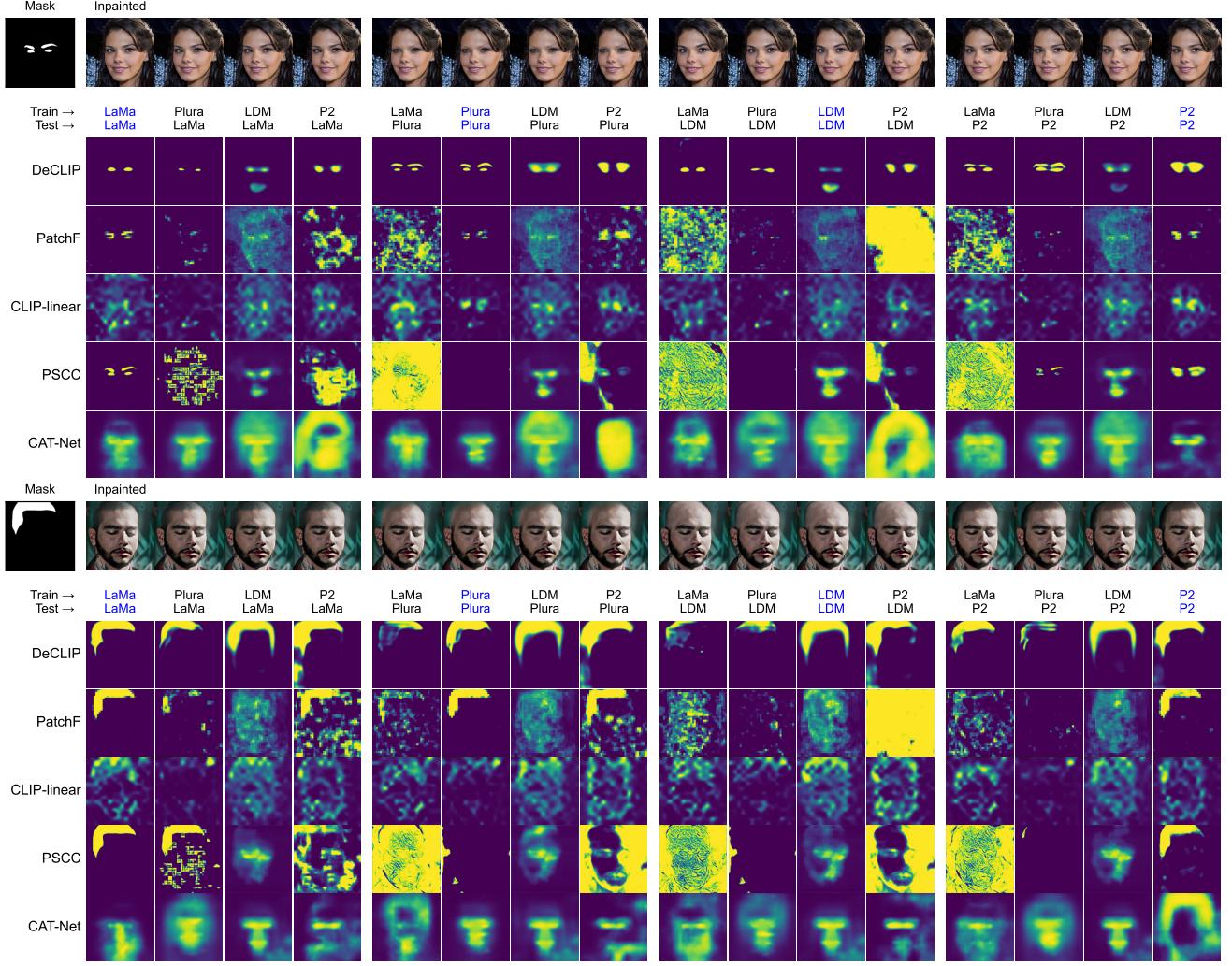


Figure 6. Sample predictions for DeCLIP (second row) and four other methods (Patch Forensics, CLIP-linear, PSCC, CAT-Net) on all 16 train–test combinations from the Dolos dataset. The in-domain combinations are highlighted in blue; the others are out-of-domain combinations. The black-and-white image in the top left corner shows the inpainting mask (white is the inpainted region) and the rest of the images in the first row are the inpainted images with one of the four test datasets (LaMa, Pluralistic, LDM, P2).

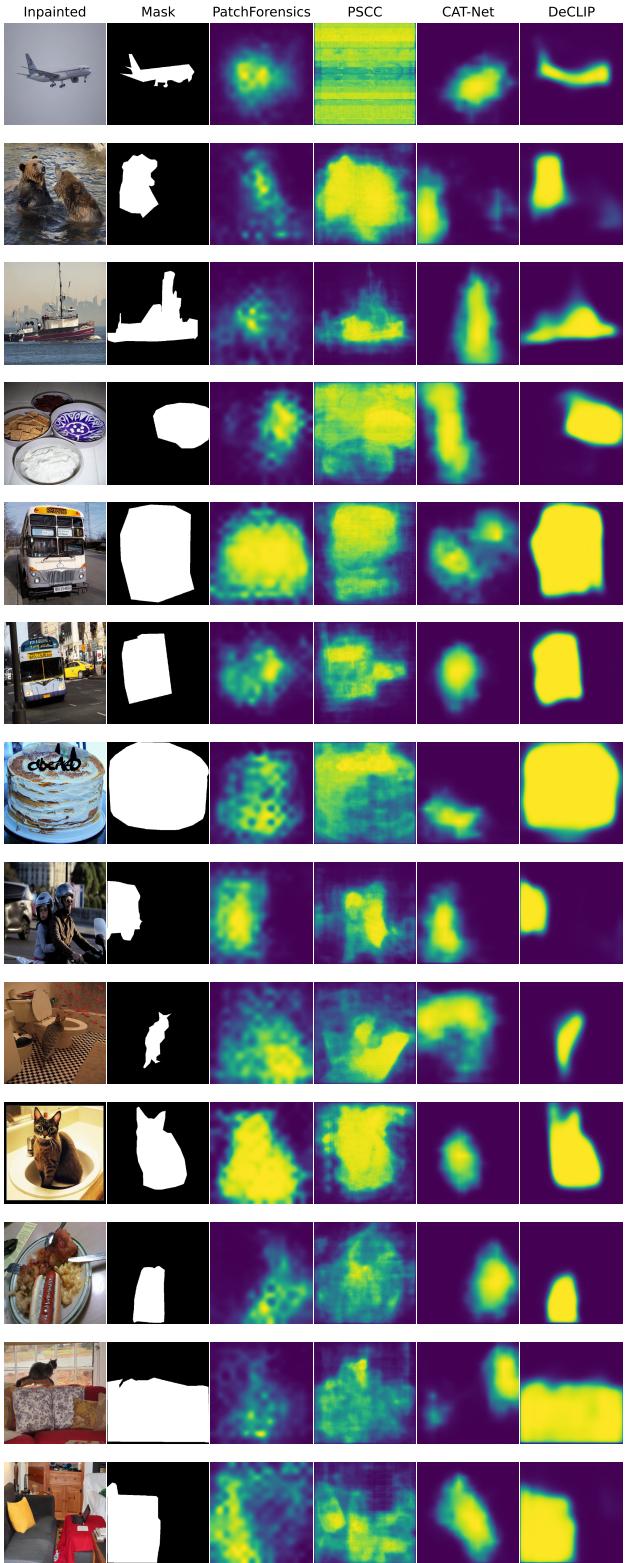


Figure 7. Manipulation localization results on COCO-SD, which has a more challenging set of masks and diverse content. DeCLIP offers a more precise localization of the manipulated area.

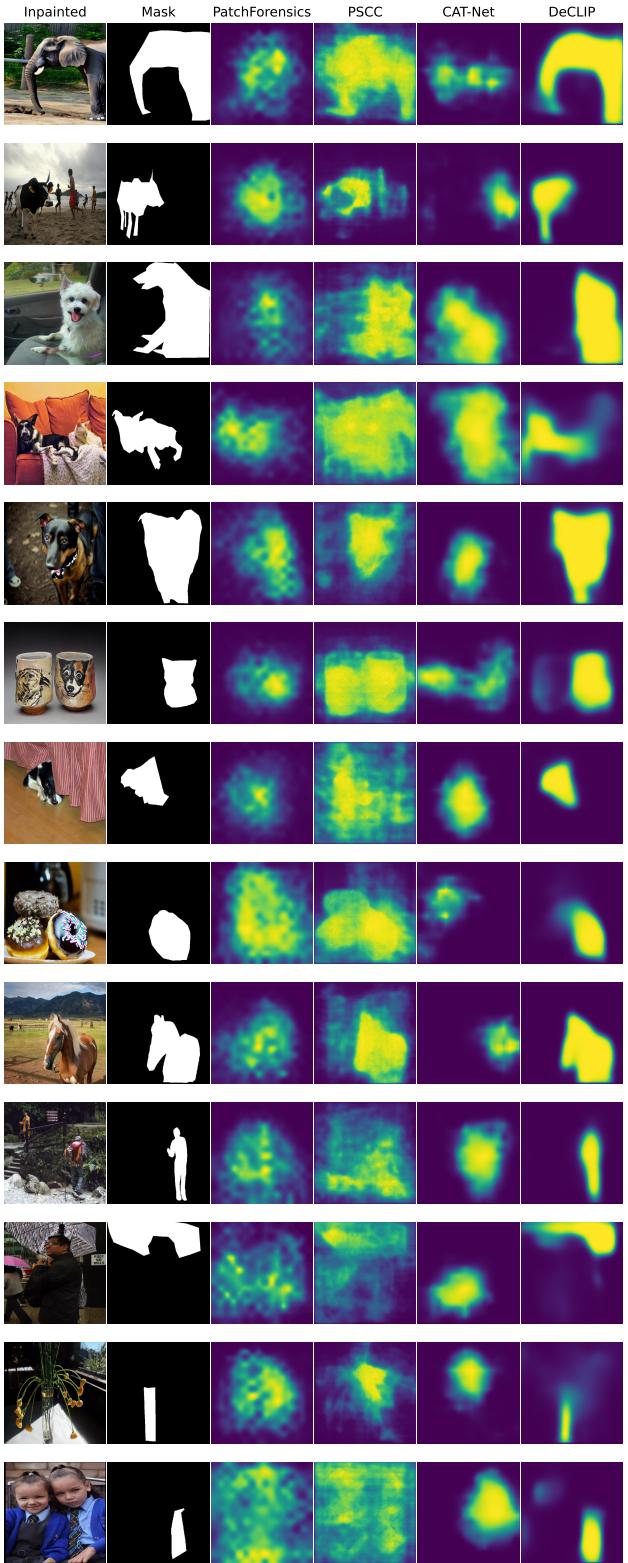


Figure 8. Manipulation localization results on COCO-SD, which has a more challenging set of masks and diverse content. DeCLIP offers a more precise localization of the manipulated area.