Appendix

A. Details of Baseline Detectors

In this section, we introduce the baseline methods utilized in our experiments.

- **CNNDet** [59]: This pioneering work collects a well-established dataset comprising real images from LSUN [69] and fake images generated by ProGAN [19]. It trains a ResNet-50 [25] on this dataset, incorporating random JPEG compression and Gaussian blurring as data augmentation techniques. The study demonstrates that fake images generated by early generative models, such as GANs, are relatively easy to detect.
- **FreqFD** [18]: This paper analyzes the differences between real and fake images in the frequency domain, highlighting that up-sampling operations typically introduce texture-level artifacts in the spatial domain. Consequently, it trains a classifier on images after applying the DCT, focusing on detection in the frequency domain.
- **Fusing** [29]: This method performs detection on inputs at multiple scales and fuses the features from these different scales to make a final decision. It generally improves detection performance compared to CNNDet.
- **LNP** [37]: This approach observes that the noise patterns of real images exhibit consistent characteristics in the frequency domain, whereas fake images differ significantly. It trains a detector to distinguish image authenticity based on these noise patterns.
- **LGrad** [52]: This method notes that the gradients of fake and real images are distinguishable when approximated using a pre-trained StyleGAN model. It trains a classifier on these gradient features to differentiate between real and fake images.
- **UnivFD** [43]: This approach leverages a pre-trained CLIP model as a feature extractor and performs classification based on the extracted embeddings.
- **DIRE** [60]: This method observes that fake images generated by diffusion models can be easily reconstructed using a pre-trained diffusion model, such as ADM [11]. It trains a classifier on the reconstruction errors to distinguish between real and fake images.
- **FreqNet** [54]: This approach enhances frequency artifacts by designing a sophisticated network that conducts an in-depth extraction of the frequency footprints of fake images.
- **NPR** [55]: This method extracts up-sampling artifacts by up-sampling the inputs and comparing nearby patch differences. This simulation effectively reveals generative footprints.
- **DRCT** [5]: This approach leverages a pre-trained diffusion model, such as SD-v1.4 [48], to reconstruct both real and fake images. It then applies contrastive loss to train the classifier to distinguish reconstructed real images (serving as hard fake samples) from the original real images.

B. Details of Used Models

In this section, we introduce the details of the image generation models used in our experiments.

- **Generative Adversarial Network (GAN) [20].** GAN is a representative class of generative models. In a GAN, two neural networks (i.e., generator and discriminator) compete in a zero-sum game, where the gain of one network comes at the loss of the other. Multiple GAN models are involved in our experiments, including ProGAN [30], StyleGAN [31], StyleGAN2 [32], BigGAN [3], CycleGAN [72], StarGAN [7], GauGAN [46], and whichfaceisreal (WFIR) [28].
- **Ablated Diffusion Model (ADM) [12].** This is a relatively early diffusion model developed by OpenAI. It is capable of achieving conditional generation by leveraging gradients from a classifier. It is open-sourced with MIT license.
- **Glide [42].** This is a 3.5 billion-parameter diffusion model developed by OpenAI, which uses a text encoder to condition on natural language descriptions with classifier-free guidance. This model is with MIT license.
- **VQ-Diffusion (VQDM) [22].** This model is built on a vector quantized variational autoencoder (VQ-VAE), with its latent space modeled using a conditional version of the Denoising Diffusion Probabilistic Model. This model is open-sourced with MIT license.
- **wukong [64].** This is a text-to-image diffusion model trained on Chinese text description and image pairs.
- **Latent Diffusion Model (LDM) [48].** This model is the first to implement the diffusion process within the latent space of pretrained autoencoders. It strikes a near-perfect balance between reducing complexity and preserving details, significantly enhancing visual quality while operating under constrained computational resources. This model is with MIT license.
- **Stable Diffusion (SD) [48].** Stable Diffusion is a series of models based on the LDM architecture with fixed, pretrained CLIP encoders. There are multiple Stable Diffusion models involved in our experiments, i.e., SD-v1.4, SD-v1.5, SD-2, SD-2-1, SDXL, SDXL-turbo, and SD-3-medium. These models are with creativemlopenrail-m license.
- **tiny-sd and small-sd [49].** These models are distilled from a fine-tuned SD-v1.5 model (SG161222/Realistic_Vision_V4.0).

Compared to the original model, the distilled models offer up to 100% faster inference times and reduce VRAM usage by up to 30%. These models are with creativemlopenrail-m license.

- **Segmind Stable Diffusion 1B (SSD-1B) [23].** The SSD-1B is a distilled version of Stable Diffusion XL (SDXL), reduced by 50% in size, delivering a 60% increase in speed while still preserving high-quality text-to-image generation performance. This model is with apache-2.0 license.
- **Segmind Mixture of Diffusion Experts (SegMoE-SD) [67].** This is a mixture of expert model combined by 4 SD-v1.5 models using the SegMoE merging framework. Comparing to the single model, this mixture of expert model has better adherence and better image quality. This model is with apache-2.0 license.
- **Playground (PG) [35].** Playground is a model family trained from scratch by the research team at Playground. These models are based on LDM architecture with two fixed, pre-trained text encoders (OpenCLIP-ViT/G and CLIP-ViT/L). Four models in this model family are used in our experiments: PG-v2-256, PG-v2-512, PG-v2-1024, and PG-v2.5-1025. These models are with playground-v2-community and playground-v2dot5-community licenses.
- **PixArt-XL (PAXL) [6].** These models reduces the training cost by decomposing training into three stages and using an efficient diffusion transformer. Two PAXL models with different generation resolutions (i.e., PAXL-2-512 and PAXL-2-1024) are used in our experiments. These models are with openrail++ license.
- **Latent Consistency Model (LCM) [40].** The LCMs are distilled from pre-trained classifier-free guided diffusion models. These distilled models can directly predict the solution of the corresponding ODE in latent space, significantly reducing the need for multiple iterations. Specifically, two LCM models are involved in this paper: LCM-sdv1-5 and LCM-sdxl, which are distilled from SD-v1.5 and SDXL, respectively. These models are with openrail++ license.
- **FLUX [2].** FLUX is a set of state-of-the-art text-to-image models developed by the Black Forest Lab. These models excel in prompt adherence, visual quality, image detail, and output diversity. In our experiments, we utilize FLUX.1-sch and FLUX.1-dev. The weights for these two models are open-sourced under the Apache-2.0 license and the FLUX-1-dev-non-commercial-license, respectively.
- **DALL-E [44].** DALL-E is a series of closed-source text-to-image AI systems built by OpenAI. Both DALL-E 2 and DALL-E 3 are included in our experiments.
- **Midjourney [41].** Midjourney is a series of closed-source text-to-image models developed by Midjourney, Inc. In our experiments, we used version Midjourney-v6.
- **Other In-the-Wild Sources.** We also incorporate additional in the wild sources to generate the images in CO-SPYBENCH/in-the-wild. In addition to DALL-E 3 and Midjourney-v6, we use Civitai, instavibe.ai, and Lexica. These website platforms generate images based on models like Stable Diffusion [48], FLUX [2], and Lexica Aperture [34], respectively.

C. Illustrations of JPEG Compression's Impact on Texture-level Artifacts

To further investigate how JPEG compression affects texture-level artifacts and why artifact detectors struggle with lossy formats (as outlined in Section 2), we conduct a frequency domain analysis. We compute the average frequency energy for 500 real images and 500 synthetic images. To ensure the analysis captures only the core content of the images, we first apply denoising with a pre-trained model [9] before performing the Fourier transform. Figure 9 presents the frequency representations of real images, synthetic images, and their JPEG-compressed versions in separate rows. In the absence of JPEG compression, synthetic images display abnormal patterns in the high-frequency regions (non-central areas) compared to real images. However, JPEG compression significantly reduces these differences between real and synthetic images, suggesting that compression diminishes the artifacts critical for detection.

D. Limitations of Existing Test Datasets

To investigate the limitation of existing test datasets, we evaluate two latest detectors, UnivFD [43] and DRCT [5], both trained on the DRCT-2M/SD-v1.5 dataset (which includes real images from MSCOCO [36] and fake images generated by SD-v1.5 using MSCOCO captions) across various test scenarios.

Lack of Evaluation on Latest Models. Synthetic images produced by the latest generative models tend to exhibit higher visual quality, making them more challenging to detect. To illustrate this issue, we evaluate the two detectors on synthetic images generated by SD-v2, SDXL, FLUX.1-schnell, and FLUX.1-dev [2]. As shown in Figure 10a, both detectors perform well on SD-v2 and SDXL, which are included in DRCT-2M. However, their performance significantly degrades when applied to FLUX models, which are not covered by the existing dataset.

Lack of Evaluation on Diverse Objects. Synthetic images generated by text-to-image models can vary significantly based on the diversity of input captions, as different captions prompt the generation of various objects. Achieving high performance

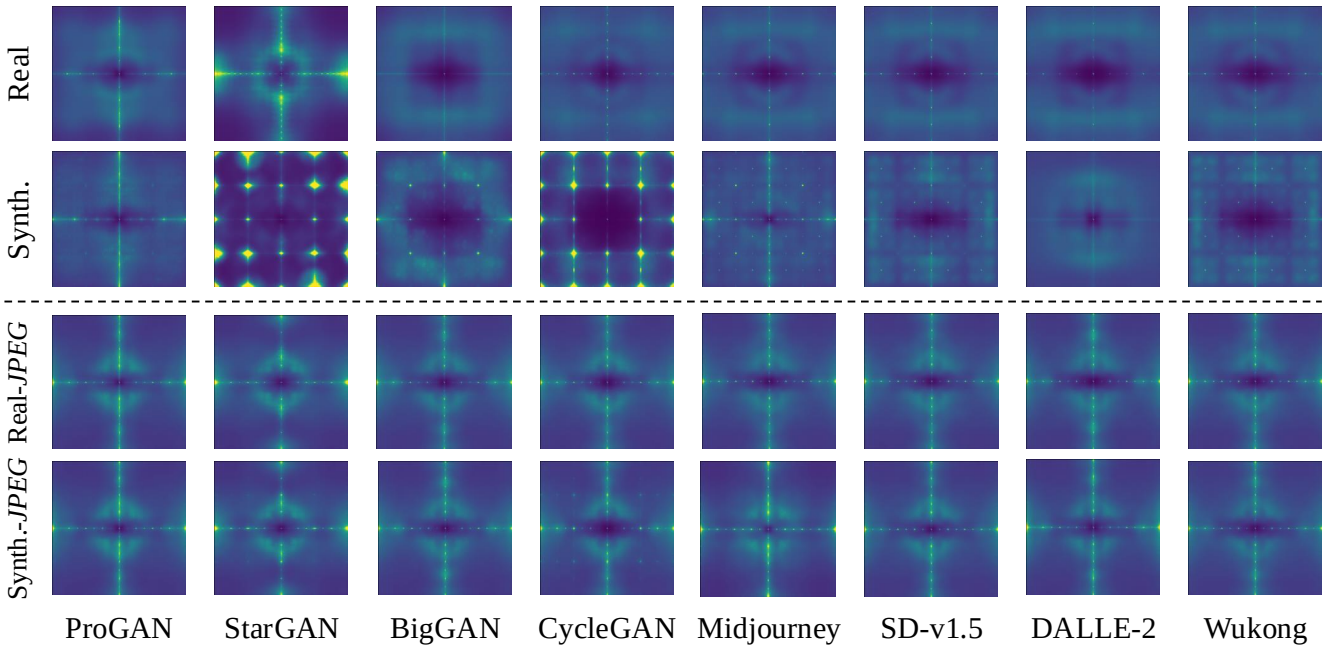
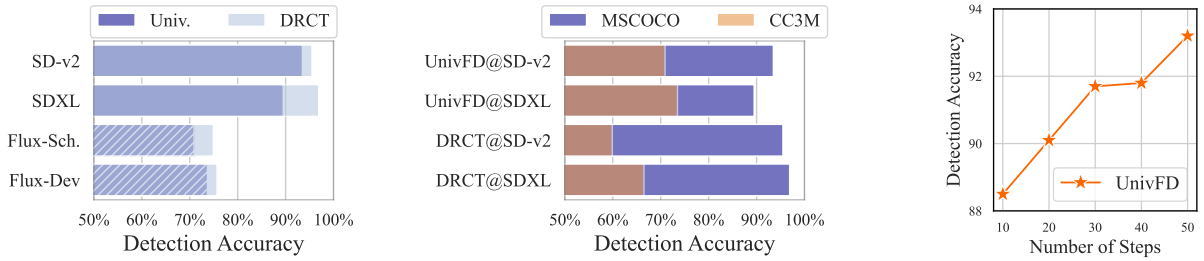


Figure 9. Frequency analysis of the impact of JPEG compression on texture-level artifacts



(a) Advanced model (b) Diverse captions (c) Number of steps

Figure 10. Detection performance varies significantly across different types of fake image generation.

on a limited set of similar captions does not guarantee effective detection across a wider range of objects. To explore this, we evaluate two detectors on synthetic images generated by SD-v2 and SDXL using captions from MSCOCO (included in the training set) and CC3M [4]. The results, shown in Figure 10b, reveal that while the detectors perform well on images generated from MSCOCO captions, their accuracy declines significantly on images generated from CC3M captions. This indicates that existing test sets do not sufficiently represent the diversity of image objects.

Lack of Evaluation on Various Generation Parameters. Additionally, existing synthetic image datasets often fix certain generation parameters, which can artificially inflate detection performance. For instance, DRCT typically uses 50 inference steps for all models. However, our observations indicate that the number of inference steps impacts detection performance. In Figure 10c, we evaluate UnivFD on SD-v2 synthetic images generated with varying numbers of inference steps. The results reveal that images generated with more inference steps are easier to detect, with a 5% accuracy difference between images generated with 10 steps versus 50 steps.

E. Limitation of Simply Combining Two Types of Detectors

We evaluate the effectiveness of directly combining two types of existing detectors to create a new one. In this experiment, we use the state-of-the-art artifact detector, NPR [55], and the semantic detector, UnivFD [43]. To combine the two detectors, we concatenate the artifact and semantic feature vectors before each downstream classifier and then retrain the classifier. The results, presented in Figure 11, leverage the DRCT-2M/SD-v1.4 dataset for training and test on synthetic images generated

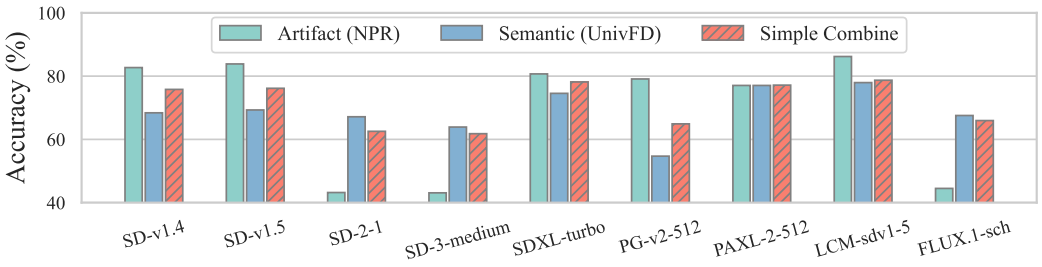


Figure 11. Limitation of simple combination of two types of detectors.

by various models with random JPEG compression. As shown, the simple combination merely averages the performance of the two detectors rather than enhancing it. This outcome arises because the two types of detectors are effective in distinct scenarios (as discussed in Section 1), and direct combination fails to create a synergistic effect, yielding only an aggregate rather than a complementary result.

F. Discussion of Overfitting Problem in Semantic Detector Training

As discussed in Section 2, semantic detectors often struggle with generalization due to the overfitting problem. For example, a ResNet-50 [25] trained on the CNNDet dataset [59] with data augmentation may achieve perfect performance on unseen ProGAN images but fail to detect samples from BigGAN. This happens because the training data is limited to ProGAN-generated images, leading to overfitting on this specific model. UnivFD [43] mitigates this issue by using a pre-trained CLIP model (ViT-L-14-224 [47]) as a feature extractor without modifying its weights. This approach leverages the extensive training of CLIP on billions of images, enabling it to capture the semantic meaning of inputs through its text-image self-supervised learning.

To further explore the overfitting problem, we fine-tune a pre-trained CLIP model on the DRCT-2M/SD-v1.5 dataset (comprising over 300,000 real images from MSCOCO and fake images generated by SD-v1.5) using the OpenCLIP [27] training pipeline. We then train a synthetic image detector on these fine-tuned features and compare its performance with the original CLIP model. The results, shown in Figure 12, reveal that fine-tuning generally degrades performance on unseen models due to overfitting. However, performance on SD-v2 improves, likely due to its similarity to SD-v1.5. This finding suggests that fine-tuning CLIP can be risky. Instead, we propose a better data augmentation during training can help. For example, DRCT [5] uses a pre-trained diffusion model to reconstruct real images as hard synthetic samples. However, this approach introduces significant computational overhead and may unfairly inflate the dataset size. Our solution using feature interpolating, introduced in Section 3.2, improves accuracy by approximately 3% over the original UnivFD (presented in Figure 12).

G. Comparison of Using Different CLIPs as Backbone Models

We explore using more advanced CLIP models to better handle the latest generative models. We evaluate three top-performing CLIP models from OpenCLIP [27]: ViT-H-14-224 [63], ViT-H-14-378 [13], and ViT-SO400M-14-384 [70]. As shown in Figure 13, ViT-SO400M-14-384 outperforms the others, potentially due to its use of Sigmoid loss and higher resolution, which enable more effective and robust semantic understanding. Based on these findings, we propose to use ViT-SO400M-14-384 combined with feature interpolation to enhance the generalization capabilities of semantic detectors, as introduced in Section 3.2.

Table 4. **Comparison with AIDE [65].** CO-SPY outperforms AIDE on Chemeleon dataset and CO-SPYBENCH, and demonstrates greater resilience to lossy formats, e.g., JPEG.

Acc.	AIGCDetect		Chameleon		CO-SPY	
	Raw	JPEG	Raw	JPEG	Raw	JPEG
AIDE	92.77	73.08	61.93	55.24	85.15	74.61
CO-SPY	87.75	79.76	67.63	63.19	91.45	87.06

Table 5. **Comparison of different backbones.** Empirical results show that CLIP achieves the best semantic feature extraction among the evaluated backbone models.

Acc.	CLIP-ViT	ResNet-50	ConvNeXT	EVA02
SD-v1.5	90.91	79.66	75.05	71.95
PG-v2.5-1024	93.05	71.32	72.11	70.65
PAXL-2-1024	92.37	81.95	74.85	74.73

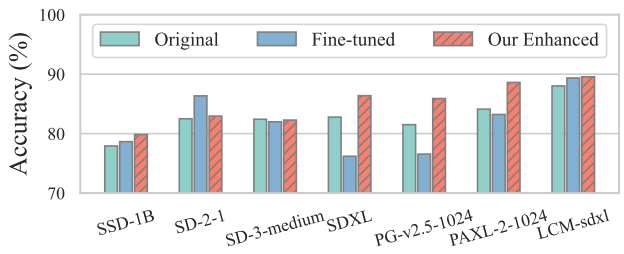


Figure 12. Enhanced training for semantic detectors using CLIP

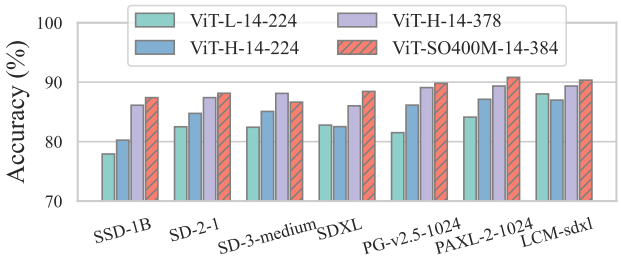
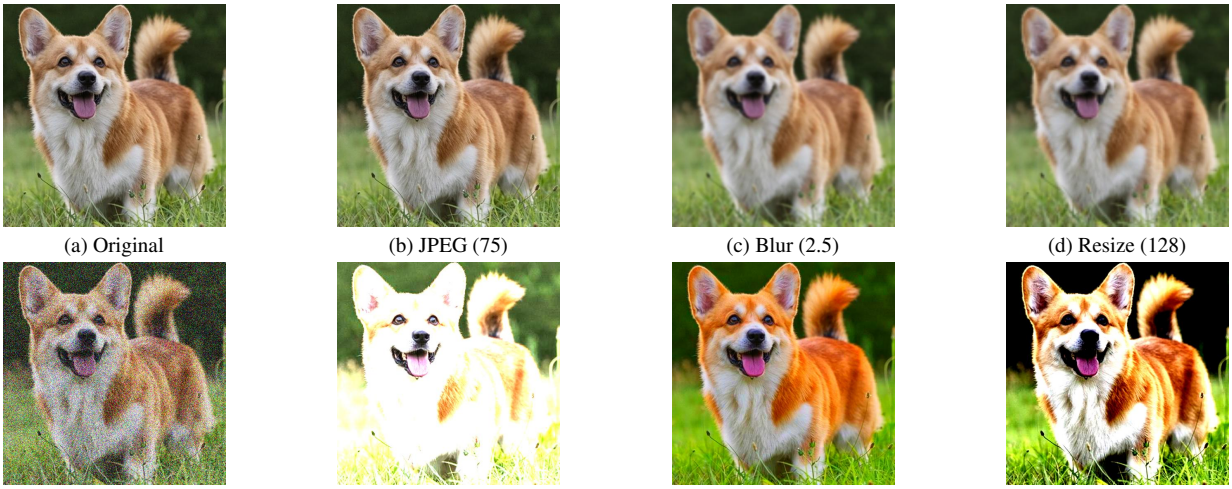


Figure 13. Comparison between latest CLIP models



(a) Original (b) JPEG (75) (c) Blur (2.5) (d) Resize (128)

(e) Noise (0.25) (f) Brightness (2.5) (g) Saturation (2.5) (h) Contrast (2.5)

Figure 14. **Demonstration of various post-processing functions.** Sub-figure (a) shows the original image and the subsequent figures (b)-(h) illustrate the effect of different functions, with the parameter value presented in the parentheses.

H. Comparison with Another SOTA Baseline AIDE

We compare Co-Spy with AIDE [65] by training on the same DRCT/SD-v1.5. We then evaluate on AIGCDetect [71], Chameleon [65], and Co-SpyBench (also testing with JPEG compression). Observe in Table 4 that while AIDE slightly outperforms Co-Spy on AIGCDetect w/o JPEG, it performs worse on the others, especially w/ JPEG. We attribute this to AIDE's reliance on pixel-level artifacts, which is vulnerable to lossy formats. By contrast, Co-Spy fuses enhanced artifact and semantic features, being more robust and generalized.

I. Backbone Selection: Why Choose CLIP?

In the default setting of Co-Spy, we use a pre-trained CLIP [27] as the backbone model to extract semantic features. In the study, we evaluate various pre-trained backbones, i.e., CLIP-ViT (default choice), ResNet-50 [25], ConvNeXT [39], EVA02 [14], by training on DRCT/SD-v1.5 dataset. The result are shown in Table 5, CLIP achieves the best performance, due to its large-scale vision-language pretraining. Hence, we choose CLIP-ViT for semantic feature extraction.

J. Evaluation on Pixel-space Diffusion Models

In the main experiment, we primarily focus on evaluating latent diffusion models [48]. In this study, we evaluate 3 pixel-based diffusion models trained on LSUN-bedroom (1,000 synthetic images with 1,000 real ones). Observe from the Table 6 that artifact-based detection shows a lower accuracy than stable diffusion (over 80% accuracy), as pixel-space diffusion do not rely on a latent decoding stage (see Figure 4) and thus exhibit fewer up-sampling artifacts. Consequently, the VAE-based artifact detector is less effective (around 70%). However, pixel-space diffusion are largely outdated and produce lower-quality outputs, enabling Co-Spy to detect their semantic inconsistencies (about 85%).

Table 6. **Evaluation on Pixel-Space Diffusion Models.** The performance of CO-SPY, particularly its artifact detector, slightly degrades on pixel-space diffusion models since they do not use a VAE-based architecture. However, due to their lower generation quality, CO-SPY can still effectively detect them based on semantic features.

Method	Semantic		Artifact		CO-SPY	
	AP	Acc.	AP	Acc.	AP	Acc.
ADM	83.96	80.20	74.31	69.96	84.48	80.30
iDDPM	84.07	81.30	75.13	70.10	87.34	84.10
PNDM	82.90	80.65	77.94	71.90	86.20	82.70

Table 7. **Comparison with existing baselines, trained on CNNDet [59] and evaluated on AIGCDetectBenchmark [71].** Note that all images undergo random JPEG compression to simulate real-world scenarios. The results are measured in average precision (AP) and accuracy, with a decision threshold of 0.5. The highest AP scores are highlighted in red, and the highest accuracy scores are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Detector	CNNDet		FreqFD		Fusing		LNP		LGrad		UnivFD		DIRE		FreqNet		NPR		CO-SPY	
	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
ProGAN	100.0	100.0	88.50	75.48	100.0	99.98	91.04	80.95	80.19	68.84	99.95	99.01	90.54	85.80	87.19	74.95	79.81	75.11	98.45	98.99
StyleGAN	98.79	67.68	81.06	68.93	98.74	76.84	87.35	77.10	79.98	64.11	95.71	74.79	85.18	71.90	83.04	70.26	77.22	73.85	92.32	81.38
BigGAN	90.01	58.23	62.32	59.48	94.88	73.88	82.18	72.47	67.20	63.25	96.80	86.67	74.52	65.10	76.31	69.55	70.66	67.50	95.24	84.25
CycleGAN	97.75	81.64	73.83	64.38	98.27	88.68	88.73	79.03	79.74	71.65	98.91	93.68	71.50	63.50	85.30	72.86	78.80	71.99	91.05	90.58
StarGAN	96.86	82.07	86.28	74.11	98.49	88.74	91.13	78.94	82.13	71.89	98.79	94.67	94.42	82.00	85.69	68.46	74.74	74.29	94.65	86.39
GauGAN	98.94	79.84	64.72	59.18	98.75	83.83	67.39	62.94	67.12	61.21	99.74	97.50	80.90	72.90	78.01	71.87	66.24	65.18	96.66	83.86
StyleGAN-2	98.33	63.75	82.94	66.06	97.80	70.26	84.40	74.26	74.15	61.09	95.15	66.24	78.73	72.80	84.49	69.15	77.84	74.92	90.67	82.45
WFIR	91.04	55.30	45.09	46.85	94.02	77.80	78.56	68.25	65.87	58.30	93.41	70.75	62.70	60.40	50.33	48.55	54.70	51.45	82.34	73.05
ADM	64.23	50.54	58.53	58.31	60.12	51.17	82.19	72.81	50.25	51.72	88.50	64.74	70.00	64.80	77.29	67.30	72.16	67.28	87.54	77.66
Glide	71.26	51.53	64.96	59.28	62.39	51.82	87.51	77.57	61.58	59.07	87.43	62.04	57.52	57.50	72.84	66.32	76.42	71.59	79.31	70.64
Midjourney	53.76	50.51	61.07	59.70	50.81	50.62	74.20	66.54	63.57	59.31	49.05	49.83	54.62	51.30	74.39	60.85	69.01	64.19	85.29	67.70
SD-v1.4	55.62	50.07	56.19	56.07	53.09	50.08	76.69	67.49	62.13	60.46	66.43	51.23	52.66	50.90	65.30	58.25	76.25	70.95	78.96	74.17
SD-v1.5	55.22	50.06	55.54	55.17	52.46	50.12	75.59	67.00	60.96	59.30	65.95	51.23	53.17	52.00	65.20	57.88	76.41	71.14	78.62	76.96
VQDM	73.28	51.40	62.45	58.84	73.03	53.32	73.42	64.98	50.46	52.04	95.95	80.23	65.87	58.70	73.43	66.12	74.08	69.58	90.59	80.42
wukong	52.65	50.08	59.17	59.47	52.79	50.21	74.13	64.17	65.87	62.46	76.35	54.20	51.86	51.00	61.35	54.73	75.74	69.31	77.38	74.67
DALL-E 2	47.16	49.95	45.87	47.00	37.03	49.65	80.95	74.15	57.12	53.90	64.90	50.60	52.85	50.50	55.45	53.75	76.35	74.10	72.92	73.00
Average	77.81	62.04	65.53	60.52	76.42	66.69	80.97	71.79	66.77	61.16	85.81	71.71	68.56	63.19	73.47	64.43	73.53	69.53	87.00	79.76

K. Detection Performance using CNNDet Training Set

We conduct experiments on the CNNDet training set and evaluate the performance of the converged detectors on the AIGCDetectBenchmark, which comprises synthetic images generated by 16 different generative models. To simulate real-world scenarios, we assess the detectors on images subjected to random JPEG compression. The results are presented in Table 7, where CO-SPY achieves an average accuracy improvement of 8% over the best baseline, UnivFD. Although the performance of CO-SPY on GAN-based images is slightly lower than that of UnivFD, likely due to the impact of JPEG compression on the artifact detector, CO-SPY demonstrates superior performance on the more challenging task of detecting diffusion-generated fake images. This improved generalization is attributed to the adaptive fusion mechanism in CO-SPY, which dynamically integrates both semantic and artifact features, enabling more comprehensive decision-making.

In addition, we do not apply any transformation to the input images during training and assess the detection performance on the raw inputs (same as the evaluation setup in most baselines [37, 55]). The results are shown in Table 8, where each raw shows the test result on different generative models and the last raw presents the averaged result. Observe that CO-SPY achieves the best performance with 96.72% AP and 87.75% accuracy in average, outperforming the best AP (87.13% of UnivFD) for over 9% and the best accuracy (80.69% of NPR) for over 7%. This can be attributed to the enhanced artifact and semantic feature extraction in CO-SPY and its comprehensive decision based on both features. Notably, CO-SPY achieves slightly lower but comparable high performance on GAN-generated synthetic images. This is because the baseline detectors tend to overfit on the training data, whose synthetic samples are also generated by GANs. Therefore, they may focus on low-level features typically spread on GAN-generated images but not generalizing to others. On the other hand, CO-SPY makes comprehensive decisions, and hence it generalizes to diffusion-generated images.

Table 8. **Comparison with existing baselines, trained on CNNDet [59] and evaluated on AIGCDetectBenchmark [71].** Note that no post-processing is applied to the inputs. The results are measured in average precision (AP) and accuracy, with a decision threshold of 0.5. The highest AP scores are highlighted in red, and the highest accuracy scores are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Detector	CNNDet		FreqFD		Fusing		LNP		LGrad		UnivFD		DIRE-G		FreqNet		NPR		CO-SPY	
	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
ProGAN	100.0	100.0	100.0	99.86	100.0	100.0	99.75	97.31	99.32	87.78	100.0	99.81	91.54	91.80	100.0	99.58	99.95	99.84	100.0	99.86
StyleGAN	99.19	72.61	95.81	86.56	99.26	82.92	98.55	92.31	95.53	77.93	97.48	80.40	85.18	71.90	99.78	89.91	99.74	97.52	99.94	96.29
BigGAN	90.39	59.45	70.54	69.77	95.65	78.47	94.51	84.95	80.01	74.85	99.27	95.08	74.52	69.10	96.05	90.45	84.39	83.20	99.52	92.00
CycleGAN	97.92	84.63	88.06	70.82	98.47	91.11	97.09	86.00	96.66	90.12	99.80	98.33	71.50	66.80	99.63	95.84	97.83	94.10	99.33	98.03
StarGAN	97.51	84.74	100.0	96.87	99.05	91.40	99.94	85.12	99.00	94.15	99.37	95.75	94.42	88.50	99.80	85.67	100.0	99.70	100.0	96.05
GauGAN	98.77	82.86	74.42	65.69	98.60	86.27	76.51	71.74	83.45	72.86	99.98	99.47	80.90	72.90	98.63	93.41	81.73	79.97	99.95	90.90
StyleGAN-2	99.03	69.22	95.59	80.17	98.84	78.97	98.98	94.14	90.85	72.25	97.71	70.76	78.73	72.80	99.58	87.89	99.97	99.34	99.94	97.89
WFIR	91.27	56.60	43.54	45.30	95.07	81.95	74.03	61.80	70.26	57.30	94.22	72.70	62.70	60.40	51.06	49.20	61.55	59.75	92.12	71.65
ADM	64.70	51.04	60.30	61.82	60.26	51.68	80.78	71.94	51.92	55.18	89.80	67.46	70.00	64.80	92.13	84.06	73.22	68.95	95.31	73.28
Glide	71.61	52.78	67.69	58.34	60.45	52.85	72.21	62.29	67.69	68.64	88.04	63.09	57.52	57.50	89.78	82.78	81.01	75.51	98.87	88.82
Midjourney	53.45	50.60	48.71	46.93	48.78	50.79	79.54	70.12	62.77	58.83	49.72	49.87	54.62	53.10	80.88	71.02	80.33	74.57	89.78	88.70
SD-v1.4	55.77	50.14	43.55	45.01	52.27	50.13	63.97	59.29	66.47	67.24	68.63	51.70	52.66	52.40	77.10	65.56	80.44	75.58	93.30	86.01
SD-v1.5	55.68	50.07	43.09	44.27	51.99	50.07	64.16	59.26	65.91	66.40	68.07	51.59	53.17	53.00	77.95	65.84	81.23	76.36	93.20	86.35
VQDM	72.62	52.15	69.45	65.34	71.55	53.93	66.82	62.70	54.30	56.92	97.53	86.01	65.87	58.70	90.42	82.29	74.66	72.98	97.73	82.35
wukong	52.71	50.08	46.96	48.48	51.53	50.13	61.99	56.75	69.84	68.41	78.44	55.14	51.86	48.70	69.43	58.59	75.42	72.23	92.34	78.77
DALL-E 2	47.16	49.85	39.58	36.00	37.89	49.50	87.88	76.25	64.21	57.45	66.06	50.80	52.85	51.40	55.40	55.75	70.86	61.40	96.23	77.05
Average	77.99	63.55	67.96	63.83	76.23	68.76	82.29	74.50	76.14	70.39	87.13	74.25	68.63	64.61	86.10	78.61	83.90	80.69	96.72	87.75

Table 9. **Comparison with existing baselines, trained on DRCT [5] and evaluated on GenImage dataset [73].** The results are measured in average precision (AP) and accuracy, with a decision threshold of 0.5. The highest AP scores are highlighted in red, and the highest accuracy scores are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Detector	CNNDet		FreqFD		Fusing		LNP		UnivFD		DIRE		FreqNet		NPR		DRCT		CO-SPY	
	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
ADM	47.93	50.12	48.68	50.14	55.17	50.43	86.69	57.39	58.97	53.77	68.60	53.95	78.21	58.27	75.00	59.30	81.74	76.81	81.63	67.25
Glide	72.70	51.98	72.85	52.13	77.29	51.43	96.33	74.48	81.92	69.64	85.61	63.49	92.47	68.42	99.12	81.12	92.83	86.60	95.90	93.02
Midjourney	73.53	53.12	65.11	50.52	77.94	52.02	74.55	54.33	88.35	78.08	63.94	53.23	67.55	51.22	88.50	58.27	89.38	82.39	92.26	83.45
SD-v1.4	99.91	98.28	94.27	64.83	99.98	99.34	99.88	98.20	96.58	90.13	97.51	86.48	94.18	68.83	99.54	93.30	94.40	88.45	96.92	96.83
SD-v1.5	99.87	98.17	93.85	64.61	99.93	99.28	99.84	97.95	96.43	89.94	97.53	86.56	94.39	68.89	99.55	93.23	94.42	88.44	96.95	96.68
VQDM	53.22	51.08	61.33	51.58	61.18	50.75	88.51	58.46	65.56	56.14	64.55	52.48	75.60	57.25	66.59	53.40	90.89	84.07	90.57	78.83
wukong	99.76	96.14	92.10	61.82	99.92	97.66	99.60	95.51	95.13	87.38	95.66	80.34	91.64	63.62	98.51	82.68	93.99	87.75	96.72	95.93
BigGAN	41.61	49.50	70.80	56.17	45.01	49.78	42.67	45.15	67.07	57.25	44.76	48.62	40.56	44.20	35.18	39.45	74.58	74.10	65.39	65.20
Average	73.57	68.55	74.87	56.48	77.05	68.84	86.01	72.68	81.25	72.79	77.27	65.64	79.32	60.09	82.75	70.09	89.03	83.58	89.54	84.65

L. Detection Performance using DRCT Training Set over GenImage Test Set

The test results on GenImage [73] are presented in Table 9, where CO-SPY achieves slightly better performance compared to the latest detector, DRCT. The reason is that DRCT leverages SD-v1.4 to reconstruct real images, thereby creating challenging synthetic samples and effectively increasing the amount of training data. Despite this advantage, CO-SPY still outperforms DRCT due to its comprehensive decision-making approach.

M. Illustration of Various Post-processing Transformations

We illustrate the effect of various post-processing transformations (evaluated in Section 4.3) in Figure 14.

N. Ablation Study on the Strength of Feature Interpolation as Data Augmentation

In this section, we perform an ablation study to evaluate the impact of feature interpolation strength (as introduced in Section 3.2) on detection performance. We utilize the DRCT [5] training set and CO-SPYBENCH as the test set. We randomly select 0% (no augmentation), 20%, 40%, 60%, and 80% of the data in each batch for feature interpolation. The results are presented in Figure 15. Observe that incorporating feature interpolation as an augmentation technique generally enhances CO-SPY's detection performance by approximately 3%. Specifically, interpolation probabilities of 40% and 60% yield the most significant improvements. Consequently, we adopt a default probability of 50% for performing random feature interpolation.

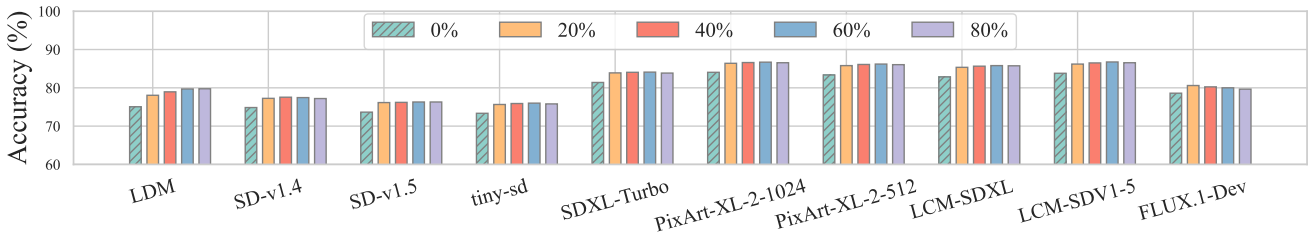


Figure 15. Ablation study on the probability of random feature interpolation

Table 10. **Ablation study on CO-SPY, trained on DRCT [5] and evaluated on CO-SPYBENCH.** The results are measured in average precision (AP) and accuracy, with a decision threshold of 0.5. The highest AP scores are highlighted in red, and the highest accuracy scores are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Method	Only Semantic		Only Artifact		Avg		Max		Min		Simple Concat		CO-SPY	
	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.	AP	Acc.
LDM	79.86	66.07	67.78	58.19	77.50	63.00	74.76	68.16	78.52	56.10	85.25	84.51	98.91	95.04
SD-v1.4	93.46	86.90	95.10	87.62	97.16	92.54	95.26	85.92	96.82	88.60	87.95	83.54	97.80	91.95
SD-v1.5	93.55	86.85	95.21	87.78	97.27	92.78	95.34	85.97	96.94	88.66	96.84	87.64	98.02	91.31
tiny-sd	87.77	76.84	83.76	74.82	90.70	80.94	86.27	79.84	91.41	71.82	85.98	82.35	95.99	84.80
SegMoE-SD	89.70	80.35	91.26	83.85	94.22	88.33	91.68	84.22	93.99	79.98	91.26	68.01	97.39	89.49
SDXL-turbo	95.08	89.15	95.57	88.03	97.86	93.42	96.22	86.01	97.64	91.17	97.90	82.89	99.17	95.39
SDXL	87.79	77.22	75.54	65.25	86.72	74.66	83.57	76.45	86.85	66.02	77.80	76.27	91.68	74.12
PG-v2-512	85.24	72.85	66.59	58.10	80.83	66.36	78.68	70.98	79.17	59.97	79.17	64.12	85.02	64.86
PG-v2-256	89.08	79.20	68.30	59.05	85.21	70.05	82.89	75.15	81.35	63.10	86.83	79.82	90.22	72.92
PAXL-2-1024	97.14	92.80	81.85	72.58	94.61	90.53	95.04	85.67	92.66	79.71	89.93	87.27	97.94	93.94
PAXL-2-512	97.31	92.93	89.98	81.97	97.06	92.83	96.10	86.13	96.26	88.77	94.49	95.27	98.63	94.96
LCM-sdxl	96.95	92.29	92.94	84.52	97.86	93.59	96.43	86.44	97.38	90.37	87.64	79.57	98.72	96.20
LCM-sdv1-5	96.77	92.45	97.88	88.58	98.86	94.37	98.60	86.45	98.57	94.58	91.98	89.87	99.63	97.14
FLUX.1-sch	93.31	84.93	84.18	75.02	92.99	86.24	91.70	83.50	91.89	76.45	83.27	75.49	95.52	85.24
Average	91.64	83.63	84.71	76.10	92.06	84.26	90.18	81.49	91.39	78.24	88.31	81.19	96.04	87.67

O. Ablation Study on Feature Fusion

In this section, we conduct an ablation study of CO-SPY to examine the integration of semantic and artifact features. The experiments are performed using the DRCT training set, and the performance is evaluated on CO-SPYBENCH. The results are presented in Table 10, which compare the default CO-SPY setting with several alternative and straightforward configurations. These alternatives include (1) using only semantic features for detection, (2) only artifact features, (3) averaging the semantic and artifact scores from two detectors, (4) taking the maximum score between the semantic and artifact detectors, (5) outputting the minimum score, and (6) simply concatenating the semantic and artifact vectors without an adaptive regulator. As shown in the table, the default setting of CO-SPY outperforms these straightforward combinations in most cases, demonstrating the effectiveness of our design. This superior performance is attributed to the regulators that dynamically assign adaptive coefficients to the semantic and artifact features, allowing the model to handle different test cases effectively. In contrast, simple concatenation leads to overfitting on one feature, resulting in reduced effectiveness.

P. Detection Performance on More Evaluation Metrics

In addition to AP and accuracy, we consider F1 and ROC-AUC scores in Table 11, and TPRs at low FPRs in Table 12. The experiment is conducted on DRCT training set and CO-SPYBENCH evaluation set. Observe that CO-SPY consistently outperforms the existing baselines regarding the four new metrics, demonstrating its general high effectiveness.

Q. Details of CO-SPYBENCH and CO-SPYBENCH/in-the-wild

The first component of CO-SPYBENCH focuses on generating synthetic images using state-of-the-art open-source models. We emphasize text-to-image generation due to its simplicity and widespread adoption. To ensure diversity within the dataset, we incorporate several key variations:

1. **Different Generative Models**: We include 22 diffusion models, such as the latest FLUX [2], to cover a wide range of generative architectures.

Table 11. **Comparison with existing baselines, trained on DRCT [5] and evaluated on CO-SPYBENCH dataset.** The results are measured in F1 score (%) and ROC-AUC score (%). The highest F1 scores are highlighted in red, and the highest ROC-AUC scores are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Detector	CNNDet		FreqFD		Fusing		LNP		UnivFD		DIRE		FreqNet		NPR		DRCT		CO-SPY	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC	F1	AUC
LDM	72.53	89.65	16.38	72.46	79.64	98.16	83.01	96.08	77.97	87.97	50.57	85.53	67.85	91.89	82.42	92.05	81.81	89.29	95.01	99.02
SD-v1.4	89.32	97.56	41.64	92.67	99.16	99.97	95.92	99.31	80.24	89.38	80.78	96.83	57.39	91.08	90.49	97.70	83.78	92.25	91.68	97.86
SD-v1.5	89.08	97.52	40.76	92.58	99.11	99.97	96.23	99.35	80.25	89.46	81.08	97.01	56.71	91.08	90.94	97.77	83.66	91.86	90.93	97.94
SSD-1B	52.68	88.67	0.16	48.90	15.15	81.79	74.94	94.16	74.54	86.29	26.08	73.16	2.76	46.31	0.95	52.55	78.16	82.88	80.81	95.33
tiny-sd	52.30	89.02	9.50	82.20	70.45	98.02	78.38	95.35	75.20	86.34	44.66	88.48	45.32	89.03	87.57	96.92	82.57	89.19	82.98	95.87
SegMoE-SD	67.42	92.21	7.87	83.12	64.24	97.18	85.27	96.96	82.90	90.70	49.99	89.68	47.24	89.65	93.70	98.15	77.37	81.40	88.77	97.46
small-sd	59.82	91.10	10.86	84.01	79.08	99.11	77.34	95.28	75.85	87.16	55.10	91.90	49.52	90.88	88.43	97.12	83.80	91.42	84.27	96.18
SD-2-1	55.92	86.84	0.99	52.98	31.77	92.62	30.27	81.47	81.31	89.75	48.35	88.55	11.55	62.35	13.16	71.39	78.48	83.09	87.67	96.99
SD-3-medium	39.34	79.70	1.19	60.64	9.93	81.00	20.00	75.69	77.13	87.34	26.42	76.76	4.48	55.06	8.69	72.18	77.18	81.35	80.43	95.25
SDXL-turbo	87.79	95.94	37.63	93.89	32.81	95.94	81.15	95.78	84.35	91.16	63.96	89.74	52.11	88.53	81.40	96.10	82.95	91.83	95.37	99.07
SD-2	50.95	87.13	0.63	50.16	21.59	88.34	22.67	77.24	70.78	83.96	35.70	84.32	8.12	56.52	12.76	73.43	77.40	81.69	81.42	95.32
SDXL	42.04	86.04	0.08	42.62	4.52	74.18	76.73	94.18	60.40	73.01	10.99	64.66	1.46	44.72	0.50	45.32	77.44	81.80	67.12	91.76
PG-v2.5-1024	19.80	63.76	0.08	48.50	2.21	78.65	74.86	94.40	78.68	83.54	12.27	60.18	0.81	55.59	0.34	50.43	73.00	78.14	87.78	96.91
PG-v2-1024	45.99	84.76	0.08	50.77	8.51	86.42	16.16	75.61	79.02	84.02	26.08	77.17	1.27	54.11	2.93	64.55	67.10	71.82	88.38	97.13
PG-v2-512	32.25	79.04	0.75	57.24	6.67	69.89	5.60	57.22	52.88	70.16	16.71	72.12	2.42	38.39	5.46	64.46	80.09	85.27	49.40	83.82
PG-v2-256	45.58	81.43	2.01	59.07	4.86	73.09	23.30	70.63	59.42	73.86	36.35	80.12	3.84	41.80	7.78	56.13	75.60	79.57	64.89	88.92
PAXL-2-1024	27.67	70.12	0.32	56.44	13.91	88.13	19.13	75.46	80.59	85.51	20.36	69.07	3.58	66.05	12.66	73.57	73.56	77.56	93.83	98.47
PAXL-2-512	50.33	82.76	9.54	82.89	54.79	96.50	67.18	92.67	80.83	85.73	41.11	79.03	29.08	84.80	78.20	95.68	77.83	82.04	94.93	98.90
LCM-sdxl	78.47	91.85	10.40	82.84	58.83	98.16	83.78	95.92	78.48	82.70	55.48	88.48	42.31	86.53	11.68	70.57	83.65	92.78	96.22	99.11
LCM-sdv1-5	92.00	97.17	53.74	94.92	76.96	98.93	90.40	97.80	80.17	84.41	74.11	92.71	69.98	93.32	93.61	98.57	82.16	88.75	97.19	99.67
FLUX.1-sch	26.95	71.29	1.34	57.68	4.60	77.11	22.12	74.25	72.70	79.85	24.99	72.22	7.22	63.70	19.07	79.39	69.38	74.26	83.53	96.08
FLUX.1-dev	30.89	69.85	1.42	53.77	11.54	83.43	21.36	72.30	76.22	82.31	25.14	72.24	2.30	52.32	11.24	71.46	72.24	77.68	84.56	96.51
Average	54.96	85.16	11.24	68.20	38.65	88.94	56.63	86.69	75.45	84.30	41.19	81.36	25.79	69.71	40.64	77.98	78.15	83.91	84.87	96.07

Table 12. **Comparison with existing baselines, trained on DRCT [5] and evaluated on CO-SPYBENCH dataset.** The results are measured in TPR at 10% FPR and 1% FPR. The highest T-10 (TPR at 10% FPR) are highlighted in red, and the highest T-1 (TPR at 1% FPR) are highlighted in blue. Note that only CO-SPY's results are highlighted if they match the best performance achieved by the baselines.

Detector	CNNDet		FreqFD		Fusing		LNP		UnivFD		DIRE		FreqNet		NPR		DRCT		CO-SPY	
	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1	T-10	T-1
LDM	75.54	41.38	37.72	13.22	95.70	78.20	89.10	49.66	75.88	21.56	64.20	25.47	74.84	33.66	84.76	38.02	65.28	19.32	98.32	78.26
SD-v1.4	99.66	93.86	77.28	35.30	99.98	99.58	99.22	85.40	74.10	21.54	91.43	57.33	70.66	20.74	97.48	43.58	73.28	24.50	95.42	61.52
SD-v1.5	99.84	93.72	75.84	34.10	99.98	99.40	99.36	86.36	74.22	21.70	92.53	57.43	69.90	20.56	97.68	44.24	70.94	23.36	95.00	63.38
SSD-1B	69.80	20.48	6.92	0.18	61.34	18.76	82.38	36.28	57.38	13.16	38.07	9.73	5.20	0.26	2.64	0.00	43.40	6.92	85.62	38.44
tiny-sd	84.48	32.74	44.38	8.64	95.92	73.82	86.82	38.10	56.78	8.08	64.57	19.77	60.50	12.62	96.54	25.04	61.62	13.80	88.10	40.84
SegMoE-SD	79.26	28.94	45.22	7.16	94.02	66.76	92.94	48.18	66.74	15.44	67.80	25.07	63.20	12.96	99.36	43.44	36.94	4.86	94.08	53.18
small-sd	93.22	49.52	47.98	9.48	98.88	83.98	86.78	36.70	61.84	11.26	73.77	27.57	67.40	13.44	97.52	25.62	68.86	19.04	89.74	40.16
SD-2-1	75.64	28.70	10.08	0.92	79.40	33.52	38.88	6.50	69.80	18.60	64.40	23.20	17.34	1.80	21.78	0.90	42.32	7.24	92.88	51.92
SD-3-medium	54.08	13.96	13.26	1.22	53.02	13.18	26.68	3.08	60.98	16.44	40.77	9.97	8.40	0.82	18.04	0.56	38.08	5.50	87.10	35.62
SDXL-turbo	92.08	65.70	82.08	33.48	90.68	43.06	88.76	38.66	77.52	21.38	72.83	37.70	64.70	12.68	90.90	28.22	74.46	25.90	98.54	80.52
SD-2	69.54	21.38	9.10	0.72	70.12	24.66	29.46	4.16	53.78	10.90	53.97	13.77	12.08	1.14	21.58	0.84	39.56	5.72	86.90	34.50
SDXL	61.38	13.00	2.90	0.08	46.88	7.18	83.18	38.00	46.52	8.76	23.10	3.70	3.50	0.18	1.04	0.04	39.36	5.44	73.12	23.60
PG-v2.5-1024	47.58	8.72	4.32	0.10	34.62	3.72	83.92	32.80	79.20	23.62	21.67	3.83	4.10	0.04	1.66	0.02	34.12	8.04	92.46	45.76
PG-v2-1024	71.64	20.80	4.94	0.10	62.52	12.48	25.62	2.24	80.50	26.56	41.00	9.13	4.02	0.12	7.74	0.04	19.14	1.42	93.54	48.24
PG-v2-512	52.40	10.54	11.22	0.76	43.76	9.54	9.34	0.72	39.92	5.74	30.67	5.83	3.78	0.34	13.28	0.32	49.22	8.56	52.80	11.76
PG-v2-256	60.88	17.80	13.36	1.88	40.56	7.22	27.42	4.60	50.76	7.34	49.23	14.97	6.80	0.66	13.30	0.90	33.90	4.68	68.88	21.86
PAXL-2-1024	56.92	17.40	9.46	0.36	62.08	16.38	26.80	3.44	91.60	43.48	32.50	7.50	12.10	0.18	21.02	0.78	30.92	3.84	97.28	64.74
PAXL-2-512	73.96	35.34	45.52	8.36	89.88	55.02	76.98	24.08	93.94	48.38	52.53	18.53	46.22	5.62	90.54	21.58	39.86	6.42	98.34	75.92
LCM-sdxl	87.82	49.48	48.90	9.18	97.12	64.46	88.18	55.42	91.36	44.90	69.03	28.30	58.02	9.30	18.28	0.90	76.68	27.44	99.32	77.02
LCM-sdv1-5	94.16	74.98	84.30	46.26	98.40	78.82	94.56	68.20	93.22	46.48	82.37	48.57	78.12	31.86	98.10	67.46	61.98	11.88	99.68	92.68
FLUX.1-sch	40.54	7.14	11.48	1.02	43.24	6.96	27.70	4.40	66.06	20.14	37.83	9.10	13.28	1.22	29.80	2.42	28.92	5.62	89.50	39.12
FLUX.1-dev	46.86	12.36	10.30	1.24	54.26	14.22	24.92	5.24	76.98	30.20	39.23	9.57	5.06	0.34	18.12	1.28	35.50	7.12	90.96	48.60
Average	72.15	34.45	31.66	9.72	73.29	41.41	63.14	30.56	69.96	22.08	54.70	21.18	34.06	8.21	47.33	15.74	48.38	11.21	89.44	51.26

2. **Diverse Caption Inputs**: Captions are collected from five well-known image-text datasets, including MSCOCO [36], CC3M [4], Flickr [68], TextCaps [50], and SBU [45]. Captions are randomly selected to generate a variety of image descriptions.
3. **Varied Generation Configurations**: We randomize generation parameters by setting the number of inference steps between 10 and 50 and adjusting the guidance scales from 3.0 to 7.0. These settings are chosen to reflect typical and reasonable values used in image generation.

Table 13. Configuration of CO-SPYBENCH.

Abbreviation	Model Name (on Huggingface)	Release Date	Image Count
LDM	CompVis/ldm-text2im-large-256	Jul. 2022	25,000
SD-v1.4	CompVis/stable-diffusion-v1-4	Aug. 2022	25,000
SD-v1.5	runwayml/stable-diffusion-v1-5	Oct. 2022	25,000
SSD-1B	segmind/SSD-1B	Jul. 2023	25,000
tiny-sd	segmind/tiny-sd	Jun. 2023	25,000
SegMoE-SD	segmind/SegMoE-SD-4x2-v0	Jan. 2024	25,000
small-sd	segmind/small-sd	Jul. 2023	25,000
SD-2-1	stabilityai/stable-diffusion-2-1	Dec. 2022	25,000
SD-3-medium	stabilityai/stable-diffusion-3-medium-diffusers	Jun. 2024	25,000
SDXL-turbo	stabilityai/sdxl-turbo	Nov. 2023	25,000
SD-2	stabilityai/stable-diffusion-2	Nov. 2022	25,000
SDXL	stabilityai/stable-diffusion-xl-base-1.0	Jul. 2023	25,000
PG-v2.5-1024	playgroundai/playground-v2.5-1024px-aesthetic	Feb. 2024	25,000
PG-v2-1024	playgroundai/playground-v2-1024px-aesthetic	Dec. 2023	25,000
PG-v2-512	playgroundai/playground-v2-512px-base	Dec. 2023	25,000
PG-v2-256	playgroundai/playground-v2-256px-base	Dec. 2023	25,000
PAXL-2-1024	PixArt-alpha/PixArt-XL-2-1024-MS	Nov. 2023	25,000
PAXL-2-512	PixArt-alpha/PixArt-XL-2-512x512	Nov. 2023	25,000
LCM-sdxl	latent-consistency/lcm-lora-sdxl	Nov. 2023	25,000
LCM-sdv1-5	latent-consistency/lcm-lora-sdv1-5	Nov. 2023	25,000
FLUX.1-sch	black-forest-labs/FLUX.1-schnell	Aug. 2024	25,000
FLUX.1-dev	black-forest-labs/FLUX.1-dev	Aug. 2024	25,000

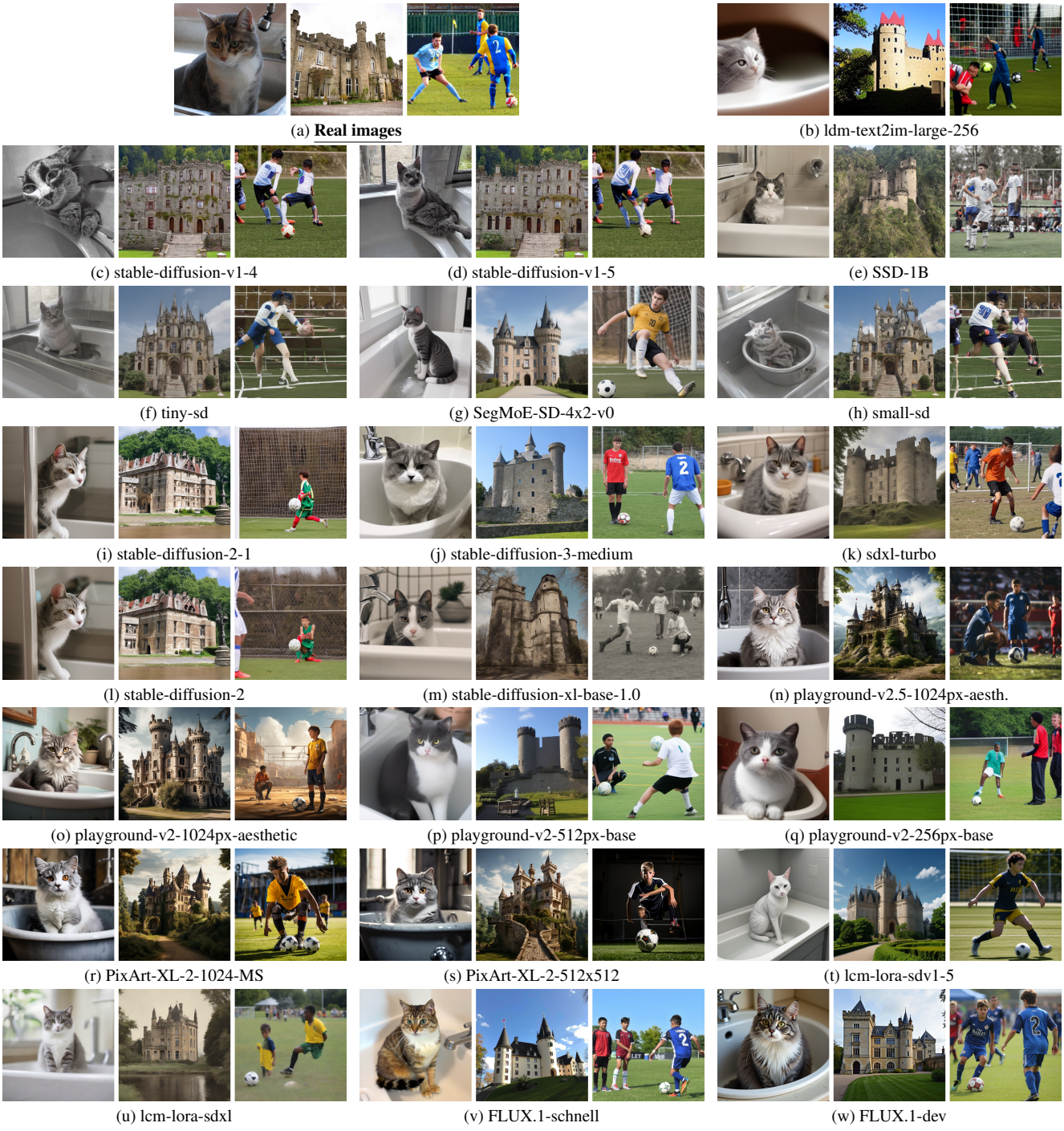
Table 14. Configuration of CO-SPYBENCH/in-the-wild.

Abbreviation	Source Website	Generative Model	Image Count
Civitai	https://civitai.com/	Stable Diffusion [48]	10,000
DALL-E 3	huggingface	DALL-E 3 [44]	Around 1M
instavibe.ai	https://www.instavibe.ai/discover	FLUX [2]	30,000
Lexica	https://lexica.art/	Lexica Aperture [34]	9,000
Midjourney-v6	huggingface	Midjourney-v6 [41]	520,000

4. **Broad Range of Real Images**: To maintain a balanced and comprehensive benchmark, we include an equal number of real images from the aforementioned caption datasets. This ensures that the benchmark provides a robust comparison between real and synthetic images.

Details of the used models are provided in Table 13. Additionally, Figure 16 presents illustrations of various synthetic images.

CO-SPYBENCH/in-the-wild. The second component of CO-SPYBENCH comprises synthetic images sourced from the Internet, providing a realistic evaluation environment for detection methods. Specifically, we collect images from five popular platforms, such as Civitai [8] and Lexica [34]. These images are generated and post-processed by developers or users, without our control over their creation processes. Consequently, this dataset closely mirrors practical, real-world scenarios, offering a robust evaluation of detection methods. We provide descriptions of each source in Table 14 and illustrations in Figure 17.



(a) **Real images** (b) ldm-text2im-large-256

(c) stable-diffusion-v1-4 (d) stable-diffusion-v1-5 (e) SSD-1B

(f) tiny-sd (g) SegMoE-SD-4x2-v0 (h) small-sd

(i) stable-diffusion-2-1 (j) stable-diffusion-3-medium (k) sdxl-turbo

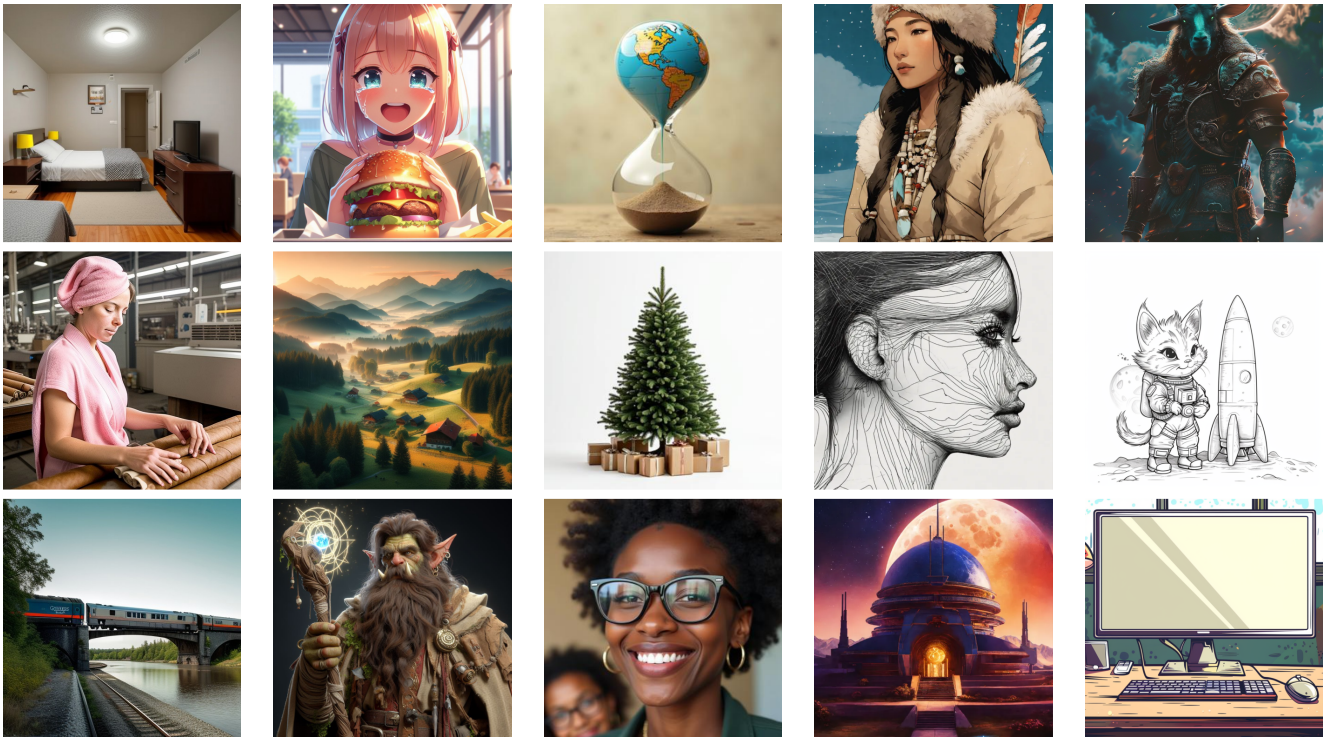
(l) stable-diffusion-2 (m) stable-diffusion-xl-base-1.0 (n) playground-v2.5-1024px-aesth.

(o) playground-v2-1024px-aesthetic (p) playground-v2-512px-base (q) playground-v2-256px-base

(r) PixArt-XL-2-1024-MS (s) PixArt-XL-2-512x512 (t) lcm-lora-sdv1-5

(u) lcm-lora-sdxl (v) FLUX.1-schnell (w) FLUX.1-dev

Figure 16. **Demonstrations of CO-SPYBENCH dataset.** Sub-figure (a) shows three real images from MSCOCO-2017, CC3M and TextCaps, respectively. The subsequent figures are generated using a series of state-of-the-art text-to-image generative models. The text captions used for generation are derived from the descriptions of the three real images: (Left) "A grey and white cat sitting in a sink."; (Middle) This castle dates back to the 19th century.; and (Right) The young man on the soccer team sponsored by Bailey Scaffolding waits carefully as the number 2 player on the opposing team prepares to kick the ball.. The url of each real image is provided using hyper-reference.



(a) Civitai (b) DALLE-3 (c) instavibe.ai (d) Lexica (e) Midjourney-v6

Figure 17. **Demonstration of CO-SPYBENCH/in-the-wild dataset.**