Forensic Self-Descriptions Are All You Need for Zero-Shot Detection, Open-Set Source Attribution, and Clustering of AI-generated Images

Supplementary Material

A. Data Composition

In this section, we discuss the composition of the datasets used in our paper.

Tab. 8 summarizes the real image datasets used in our experiments, highlighting their diverse range of resolutions and topics. The datasets include COCO2017 [41], IN-1k [24], IN-22k [61], and MIDB [8, 9], covering resolutions from as low as 32×25 to as high as 5248×6016. This diversity ensures that our method is trained and evaluated on real images that represent a broad variety of scenes, resolutions, and domains, minimizing potential biases and enhancing its generalizability. Notably, our method is trained exclusively on the training samples of real images and does not see the synthetic images during training, supporting its zero-shot detection capability.

Tab. 9 provides an overview of the synthetic image datasets used in our study, which are drawn from OSSIA [28], DMID [19], SB [6], and our own generations. These datasets include synthetic images generated by a wide range of models, such as BigGAN, DALLE variants, StyleGAN, and Stable Diffusion versions, covering diverse resolutions from 256×256 to 1792×1792. Notably, DMID and SB datasets are primarily evaluation-only, with no training samples, except for Latent Diffusion and Guided Diffusion from DMID. This comprehensive collection ensures robust evaluation across diverse generative models, demonstrating the adaptability and generalization of our method to various synthetic sources.

B. Competing Methods Categories and Taxonomy

Tab. 7 presents a comprehensive comparison of various methods for synthetic image detection and source attribution, categorizing them based on their capabilities, training data requirements, training paradigms, and underlying approaches. The capabilities considered are zero-shot detection, open-set recognition, and clustering—key features that determine a method's ability to generalize to unseen data and accurately attribute sources.

Most existing methods rely on supervised learning paradigms and require both real and synthetic images for training. For instance, CnnDet [70] and PatchFor [14] train classifiers on known synthetic sources, focusing on low-level artifacts or standard classification techniques. While these methods can sometimes generalize to similar generative models, they lack zero-shot capabilities and struggle with open-set scenarios where new types of synthetic images emerge. They also do not support clustering, limiting their utility in organizing images based on source similarities.

Some methods, like LGrad [66], UFD [54], DE-FAKE [63], Aeroblade [59], ZED [22], and NPR [67], offer zero-shot detection capabilities. LGrad trains classifiers on gradients of a common CNN, while DE-FAKE and UFD leverage embeddings from models like CLIP and BLIP. Aeroblade is unique in being training-free, using reconstruction errors from pretrained diffusion models. ZED employs a self-supervised approach, using a lossless neural compressor trained on real images. However, despite their zero-shot capabilities, these methods generally do not support open-set recognition or clustering. They are limited to distinguishing real from synthetic images and often cannot attribute images to specific unknown sources or organize them based on source characteristics.

Open-set recognition and clustering are addressed by methods like RepMix [13], POSE [71], Fang et al. [28], and Abady et al. [1]. These methods utilize supervised or open-set training paradigms and require both real and synthetic images for training. RepMix introduces representational mixing to handle unseen classes, while POSE progressively enlarges the embedding space using learned augmentations.

Table 7. Categorization of different capabilities, training data requirement, training paradigm, and high-level idea/approach of competing methods and ours. ✖ means no ability or achieving poor performance, ◑ means having moderate ability or performance, and ✔ means having good to strong ability or performance.

Method	Capabilities			Training Data Requirement	Training Paradigm	Idea / Approach
	Zero-Shot	Open-Set	Clustering			
CnnDet [70]	◑	✖	✖	Real + Synthetic	Supervised	Standard Classifier trained on 1 GAN can generalize to some other GANs
PatchFor [14]	✔	✖	✖	Real + Synthetic	Supervised	Ensemble of Patch-based classifiers trained on low-level artifacts
LGrad [66]	✔	✖	✖	Real + Synthetic	Supervised	Classifier trained on 2D gradients of a common CNN as forensic features
UFD [54]	✔	✖	◑	Real + Synthetic	Supervised	Classifier trained based on CLIP's embedding distances to real and fake reference embeddings
DE-FAKE [63]	✔	✖	✖	Real + Synthetic	Supervised	Classifier trained based on CLIP's and BLIP's text and visual embeddings
Aeroblade [59]	✔	✖	✖	No Data Required	Training-Free	The reconstruction errors using pretrained Diffusion models of synthetic images are lower than that of real images
ZED [22]	✔	✖	✖	Real	Self-Supervised	The coding costs using a lossless neural compressor (trained on real images) of real images are lower than that of synthetic images
NPR [67]	✔	✖	✖	Real + Synthetic	Supervised	Classifier trained on neighboring pixel relationships, which is extracted by subtracting the image by its down-up-sampled version
DCTCNN [29]	✖	✖	✖	Real + Synthetic	Supervised	Classifier trained on DCT of real and synthetic images
RepMix [13]	✖	✔	✖	Real + Synthetic	Supervised	Classifier trained with representational mixing
POSE [71]	✖	✔	✔	Real + Synthetic	Open-Set	Progressively enlarge the embedding space of classes using learned augmentations
Fang et al. [28]	✖	✔	✔	Real + Synthetic	Open-Set	Learned transferable embeddings using ProxyNCA applied on a CNN
Abady et al. [1]	✖	✔	✔	Real + Synthetic	Open-Set	Learned embedding space of classes using siamese network with learned distance metric
FSM [48]	✖	◑	✖	Real	Supervised	Learned embedding space of different camera models using siamese network with learned distance metric
ExifNet [75]	✖	◑	✖	Real	Supervised	Learned embedding space of images' Exif data using siamese network with learned distance metric
CLIP [57]	✖	✔	◑	Real	Self-Supervised	Learned transferable visual embeddings grounded by text captions
ResNet-50 [24]	✖	✔	◑	Real	Supervised	Learned transferable visual embeddings by training on large corpus of real images with many classes
Ours	✔	✔	✔	Real	Self-Supervised	The self-descriptions of the forensic microstructures in real images are naturally different than those of synthetic images. Self-descriptions of images created by different generators are also distinct, attributable and cluster-able.

Table 8. Composition of datasets of real images used in this paper. We note that our method only sees the training samples of real images during training.

Real Images Datasets			
Source	**Image Sizes**	**Train Samples**	**Test Samples**
COCO2017 [41]	51-640 x 59-640	100000	1000
IN-1k [24]	32-5980 x 25-4768	100000	1000
IN-22k [61]	56-1857 x 56-2091	100000	1000
MIDB [8, 9]	480-5248 x 640-6016	22329	1000

Fang et al. and Abady et al. focus on learning transferable embeddings through techniques like ProxyNCA and siamese networks with learned distance metrics. Although these methods can perform open-set recognition and clustering, they lack zero-shot detection capabilities, meaning they require prior exposure to synthetic sources to function effectively.

Our proposed method distinguishes itself by offering all three capabilities: zero-shot detection, open-set source attribution, and clustering, while requiring only real images for training. By modeling forensic microstructures through

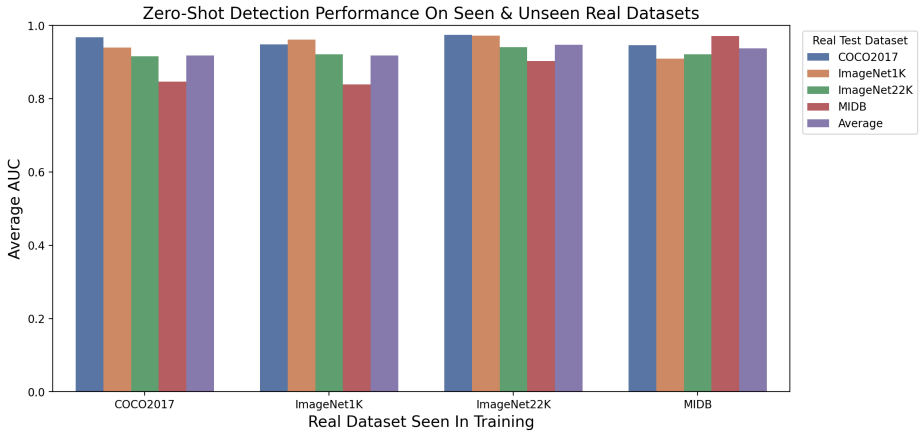

Figure 6. Zero-shot detection performance of our method evaluated on real datasets that are not seen during training. Performance on seen dataset is also provided for comparison.

Table 9. Composition of datasets of synthetic images used in this paper. These datasets are pooled together from OSSIA [28], DMID [19], SB [6], and our own generations. We note that in the zero-shot experiment, our method does not see any synthetic images during training.

Synthetic Image Datasets				
Generator	**Sources**	**Image Sizes**	**Train Samples**	**Test Samples**
BigGAN	DMID	256-512 x 256-512	0	1000
DALLE 2	DMID, SB	1024-1024 x 1024-1024	0	2000
DALLE 3	Ours, SB	1024-1792 x 1024-1792	4000	2000
DALLE M	DMID	256-256 x 256-256	0	1000
EG3D	DMID	512-512 x 512-512	0	1000
FireFly	SB	1536-2304 x 1792-2688	0	1000
GigaGAN	DMID	256-1024 x 256-1024	0	1000
GLIDE	DMID, SB	256-256 x 256-256	0	2000
Guided Dif	DMID	256-256 x 256-256	1000	1000
Latent Dif	DMID	256-256 x 256-256	2000	1000
MJ v5	SB	896-1360 x 896-1360	0	1000
MJ v6	Ours	768-1344 x 896-1536	25000	1000
ProGAN	OSSIA	256-256 x 256-256	25000	1000
Proj.GAN	OSSIA	256-256 x 256-256	25000	1000
SD1.3	SB	512-512 x 512-512	0	1000
SD1.4	OSSIA, SB	512-512 x 512-512	25000	2000
SD1.5	Ours	768-768 x 768-768	10000	1000
SD2.1	SB	576-1408 x 704-1728	0	1000
SD3.0	Ours	1024-1024 x 1024-1024	10000	1000
SDXL	Ours, SB	576-1408 x 704-1728	25000	2000
StyleGAN	OSSIA	256-1024 x 256-1024	25000	1000
StyleGAN2	OSSIA	512-1024 x 512-1024	25000	1000
StyleGAN3	OSSIA	256-1024 x 256-1024	25000	1000
Tam.Xformer	OSSIA	256-256 x 256-256	25000	1000
Total			**252000**	**29000**

diverse predictive filters, we extract residuals that encapsulate intrinsic forensic properties unique to the image creation process. These residuals are used to compute forensic self-descriptions, which naturally differ between real and synthetic images and across different generators. This enables robust zero-shot detection by modeling real-image self-description distributions and detecting deviations. Additionally, the distinctiveness of self-descriptions supports open-set attribution and clustering, providing a generalizable and efficient solution without relying on synthetic training data.

C. Full Zero-Shot Results

In this section, we present zero-shot performances between all real-vs-synthetic dataset pairs. These results are shown in Tab. 10, 11, 12, and 13.

These results, in conjunction with those presented in Tab. 1 and 2 of the main paper, highlight the exceptional generalizability and consistency of our method across a wide range of real sources and synthetic generators. While some other methods achieve high overall average AUC scores, their performance often drops significantly in worst-case scenarios. For instance, NPR demonstrates a strong overvall average AUC of 0.926 but fails on the Firefly generator, with worst-case AUCs as low as 0.239 on the IN-1k dataset. In contrast, our method not only achieves the highest overall average AUC of 0.960 but also maintains consistently high worst-case AUCs, with a minimum of 0.714 on IN-22k, even for challenging generators like GLIDE. This stability reflects our method's ability to generalize effectively to unseen generators.

Compared to other methods that also rely solely on real images for training, such as ZED, our approach demonstrates significant advantages. ZED achieves an average AUC of 0.723 but struggles with specific generators like ProGAN, with worst-case AUCs around 0.375. By leveraging forensic self-descriptions, our method captures intrinsic forensic properties that remain robust across diverse generators, avoiding the pitfalls of methods that depend on synthetic training data or fail to generalize to new generators.

Additionally, our method shows exceptional adaptability in handling challenging cases that cause other methods to fail, such as BigGAN and Firefly. The ability to achieve strong performance even in worst-case scenarios under-

Table 10. Zero-shot detection performance, measured in AUC, between each synthetic generator and COCO2017.

Method	Avg.	ProG	Prj.G	SG	SG2	SG3	BigG	GigaG	Eg3d	Tm.Xf	Glide	G.Dif.	L.Dif.	SD1.3	SD1.4	SD1.5	SD2.1	SDXL	SD3.0	DLEM	DLE2	DLE3	MJv5	MJv6	Firefly
CnnDet	0.756	0.999	0.803	0.994	0.944	0.940	0.923	0.726	0.939	0.654	0.733	0.775	0.752	0.702	0.685	0.521	0.683	0.725	0.702	0.657	0.804	0.477	0.598	0.570	0.834
PatchFor	0.833	0.806	0.953	**0.995**	0.845	0.772	0.939	0.831	0.890	0.918	0.850	0.819	0.952	0.917	0.896	0.885	0.547	0.887	0.751	0.943	0.884	0.564	0.687	0.846	0.620
LGrad	0.819	0.954	0.800	0.972	0.896	0.890	0.862	0.837	0.913	0.729	0.819	0.773	0.871	0.818	0.818	0.827	0.617	0.808	0.859	0.778	0.851	0.734	0.795	0.774	0.657
UFD	0.903	**1.000**	0.976	**0.995**	0.896	**0.990**	**0.997**	0.964	0.988	0.976	0.872	0.894	0.916	0.934	0.928	0.740	**0.946**	0.813	0.732	0.976	0.980	0.680	0.780	0.709	**0.992**
DE-FAKE	0.765	0.728	0.799	0.727	0.894	0.590	0.534	0.646	0.601	0.839	0.905	0.723	0.812	0.795	0.839	0.850	0.694	0.791	0.943	0.795	0.560	0.922	0.775	0.900	0.694
Aeroblade	0.728	0.520	0.718	0.891	0.472	0.664	0.425	0.537	0.714	0.566	0.883	0.720	0.719	0.811	0.872	**0.982**	0.828	0.792	0.741	0.730	0.596	0.745	0.900	0.938	0.706
ZED	0.751	0.462	0.667	0.880	0.811	0.840	0.713	0.727	0.824	0.766	0.663	0.682	0.729	0.812	0.814	0.777	0.702	0.798	0.813	0.830	0.847	0.715	0.803	0.801	0.563
NPR	0.945	0.993	**0.988**	0.994	**0.992**	0.986	0.981	0.959	**0.993**	**0.992**	0.984	0.916	**0.992**	**0.986**	**0.985**	0.971	0.921	**0.975**	0.982	0.970	0.985	0.844	0.935	0.969	0.396
Ours	**0.968**	0.989	0.979	0.905	0.942	0.973	0.990	**0.987**	0.955	0.991	**0.992**	**0.991**	0.989	0.951	0.944	0.892	0.926	0.971	**0.994**	**0.987**	**0.993**	**0.963**	**0.977**	**0.976**	0.987

Table 11. Zero-shot detection performance, measured in AUC, between each synthetic generator and ImageNet-1K.

Method	Avg.	ProG	Prj.G	SG	SG2	SG3	BigG	GigaG	Eg3d	Tm.Xf	Glide	G.Dif.	L.Dif.	SD1.3	SD1.4	SD1.5	SD2.1	SDXL	SD3.0	DLEM	DLE2	DLE3	MJv5	MJv6	Firefly
CnnDet	0.714	0.999	0.751	**0.995**	0.946	0.926	0.903	0.673	0.922	0.599	0.678	0.729	0.702	0.644	0.626	0.458	0.627	0.675	0.646	0.600	0.760	0.424	0.539	0.510	0.792
PatchFor	0.823	0.799	0.948	0.994	0.841	0.763	0.934	0.821	0.876	0.907	0.829	0.804	0.942	0.905	0.882	0.871	0.543	0.874	0.739	0.933	0.868	0.564	0.679	0.834	0.613
LGrad	0.770	0.914	0.738	0.938	0.891	0.820	0.774	0.782	0.812	0.676	0.787	0.728	0.809	0.720	0.731	0.839	0.658	0.777	0.769	0.731	0.803	0.696	0.716	0.742	0.625
UFD	0.862	**1.000**	0.952	0.985	0.850	0.978	**0.993**	**0.939**	0.971	0.953	0.811	0.804	0.874	0.895	0.884	0.661	0.913	0.751	0.643	**0.956**	0.960	0.607	0.705	0.623	0.982
DE-FAKE	0.749	0.641	0.725	0.768	0.872	0.627	0.487	0.554	0.581	0.778	0.814	0.644	0.738	0.823	0.841	0.880	0.710	0.834	0.911	0.735	0.635	0.894	0.810	0.889	0.785
Aeroblade	0.741	0.554	0.734	0.884	0.508	0.690	0.458	0.566	0.733	0.598	0.883	0.735	0.732	0.814	0.869	0.973	0.828	0.802	0.753	0.744	0.618	0.759	0.896	0.931	0.721
ZED	0.676	0.402	0.562	0.790	0.741	0.750	0.632	0.646	0.743	0.692	0.594	0.618	0.672	0.740	0.733	0.690	0.623	0.732	0.756	0.752	0.783	0.651	0.719	0.734	0.473
NPR	0.900	0.979	**0.969**	0.983	0.978	0.964	0.943	0.902	**0.980**	**0.975**	**0.954**	0.882	**0.974**	**0.960**	0.964	0.917	0.816	0.938	0.948	0.908	0.956	0.713	0.847	0.918	0.239
Ours	**0.962**	0.955	0.930	0.984	**0.995**	**0.999**	0.912	0.903	0.975	0.927	0.949	**0.922**	0.925	0.923	**0.979**	**0.977**	**0.978**	**0.993**	**0.978**	0.944	**0.976**	**1.000**	**0.985**	**0.986**	**0.994**

Table 12. Zero-shot detection performance, measured in AUC, between each synthetic generator and ImageNet-22k.

Method	Avg.	ProG	Prj.G	SG	SG2	SG3	BigG	GigaG	Eg3d	Tm.Xf	Glide	G.Dif.	L.Dif.	SD1.3	SD1.4	SD1.5	SD2.1	SDXL	SD3.0	DLEM	DLE2	DLE3	MJv5	MJv6	Firefly
CnnDet	0.733	**0.999**	0.779	0.997	0.956	0.940	0.918	0.694	0.936	0.622	0.704	0.751	0.727	0.670	0.650	0.474	0.651	0.697	0.668	0.622	0.783	0.439	0.560	0.530	0.817
PatchFor	0.845	0.821	**0.958**	**0.998**	0.852	0.789	0.945	0.844	0.897	0.925	0.859	0.832	0.957	0.925	0.904	0.894	0.565	0.895	0.769	**0.949**	0.892	0.594	0.709	0.856	0.643
LGrad	0.866	0.951	0.850	0.965	0.936	0.897	0.871	0.876	0.895	0.812	0.859	0.836	0.893	0.840	0.845	0.910	0.798	0.873	0.867	0.844	0.886	0.816	0.836	0.849	0.776
UFD	0.815	**0.999**	0.921	0.972	0.772	0.959	**0.988**	0.904	0.949	0.919	0.732	0.771	0.807	0.845	0.838	0.568	0.875	0.676	0.553	0.931	0.933	0.527	0.614	0.534	0.970
DE-FAKE	0.617	0.584	0.648	0.558	0.753	0.424	0.383	0.492	0.431	0.706	0.782	0.580	0.672	0.643	0.699	0.706	0.533	0.642	0.825	0.644	0.396	0.795	0.618	0.769	0.527
Aeroblade	0.582	0.405	0.544	0.713	0.378	0.499	0.336	0.420	0.527	0.437	0.752	0.584	0.583	0.617	0.696	0.862	0.637	0.637	0.605	0.579	0.468	0.588	0.742	0.792	0.565
ZED	0.716	0.375	0.603	0.830	0.771	0.789	0.789	0.689	0.775	0.738	0.643	0.665	0.729	0.765	0.766	0.725	0.668	0.782	0.791	0.791	0.809	0.686	0.757	0.752	0.507
NPR	0.900	0.966	**0.958**	0.969	0.966	0.953	0.936	0.903	0.967	**0.962**	**0.947**	**0.891**	**0.962**	0.949	0.948	0.915	0.844	0.968	0.940	0.908	0.929	0.750	0.867	0.917	0.295
Ours	**0.941**	0.930	0.895	0.933	**0.975**	**0.991**	0.912	**0.917**	**0.970**	0.917	0.714	0.852	0.893	**0.971**	**0.969**	**0.977**	**0.966**	**0.988**	**0.983**	0.913	**0.976**	**0.971**	**0.982**	**0.989**	**0.992**

Table 13. Zero-shot detection performance, measured in AUC, between each synthetic generator and MISL Image Database (MIDB).

Method	Avg.	ProG	Prj.G	SG	SG2	SG3	BigG	GigaG	Eg3d	Tm.Xf	Glide	G.Dif.	L.Dif.	SD1.3	SD1.4	SD1.5	SD2.1	SDXL	SD3.0	DLEM	DLE2	DLE3	MJv5	MJv6	Firefly
CnnDet	0.683	**1.000**	0.720	0.999	0.950	0.932	0.900	0.635	0.927	0.551	0.637	0.696	0.664	0.597	0.581	0.407	0.581	0.638	0.604	0.555	0.734	0.373	0.487	0.457	0.769
PatchFor	0.790	0.777	0.919	0.970	0.819	0.741	0.897	0.786	0.836	0.855	0.779	0.765	0.892	0.856	0.832	0.820	0.536	0.832	0.713	0.886	0.818	0.573	0.665	0.790	0.610
UFD	0.612	0.994	0.745	0.856	0.504	0.831	0.947	0.727	0.776	0.723	0.425	0.495	0.547	0.621	0.608	0.272	0.690	0.415	0.255	0.786	0.776	0.270	0.312	0.244	0.883
LGrad	0.824	0.959	0.808	0.978	0.900	0.900	0.872	0.844	0.923	0.730	0.815	0.771	0.881	0.828	0.826	0.839	0.606	0.815	0.864	0.780	0.859	0.732	0.802	0.777	0.655
DE-FAKE	0.791	0.753	0.825	0.759	0.915	0.624	0.563	0.675	0.636	0.862	0.924	0.748	0.836	0.823	0.863	0.875	0.725	0.818	0.960	0.822	0.594	**0.941**	0.804	0.921	0.728
Aeroblade	0.646	0.440	0.606	0.813	0.406	0.547	0.360	0.457	0.578	0.477	0.826	0.645	0.645	0.695	0.783	0.954	0.719	0.708	0.669	0.647	0.517	0.657	0.831	0.885	0.627
ZED	0.747	0.331	0.599	0.872	0.801	0.835	0.729	0.744	0.898	0.763	0.699	0.745	0.760	0.836	0.803	0.774	0.647	0.800	0.812	0.855	0.891	0.713	0.730	0.775	0.513
NPR	0.957	0.994	0.990	0.995	**0.994**	0.991	0.985	0.966	0.994	0.994	0.987	0.963	0.993	**0.990**	**0.986**	**0.980**	**0.947**	**0.990**	**0.987**	0.977	0.988	0.876	0.955	**0.989**	0.449
Ours	**0.971**	**1.000**	**1.000**	**1.000**	0.989	**0.998**	**0.993**	**0.995**	**1.000**	**0.998**	**1.000**	**0.993**	**0.996**	0.959	0.941	0.952	0.903	0.962	0.956	**0.995**	**0.993**	0.931	**0.965**	0.896	**0.896**

scores the effectiveness of our forensic self-description approach. This resilience, combined with the exclusive use of real images during training, positions our method as a reliable and generalizable solution for zero-shot detection of synthetic images.

D. Zero-Shot Performance vs. Thresholds

In this section, we study the detection performance's impact as a result of varying the decision threshold. To do this, we vary a normalized threshold and measure the average accuracy over all real-vs-synthetic dataset pairs with respect to a real dataset. We note that the accuracy is balanced because the number of real and synthetic samples in each pair is identical. the results of this experiment is provided in Fig. 8.

The results in Fig. 8 show that the average accuracy generally increases as the normalized threshold approaches an optimal range, peaking at a certain value before declining. This behavior is consistent across all datasets, though the precise peak accuracy and the threshold at which it occurs vary slightly between datasets. However, all peaks generally occur within the narrow range of thresholds between -0.10 and -0.14. This narrow range highlights the stability of our method's performance across different real datasets, indicating that forensic self-descriptions offer robust generalization to varying real-vs-synthetic scenarios.

This stability has practical implications: a system em-

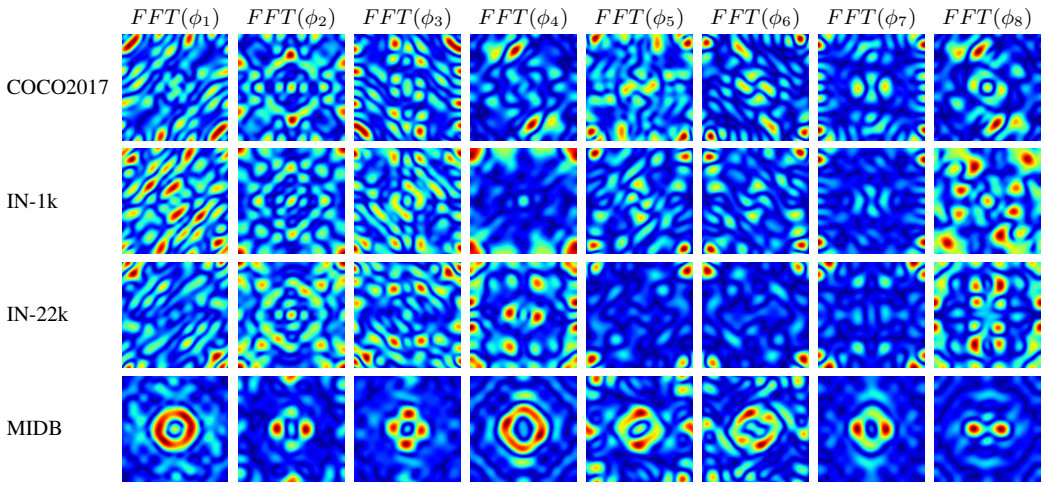

Figure 7. Visualization of the average power spectrum of different filters in the forensic self-descriptions obtained from four real datasets.

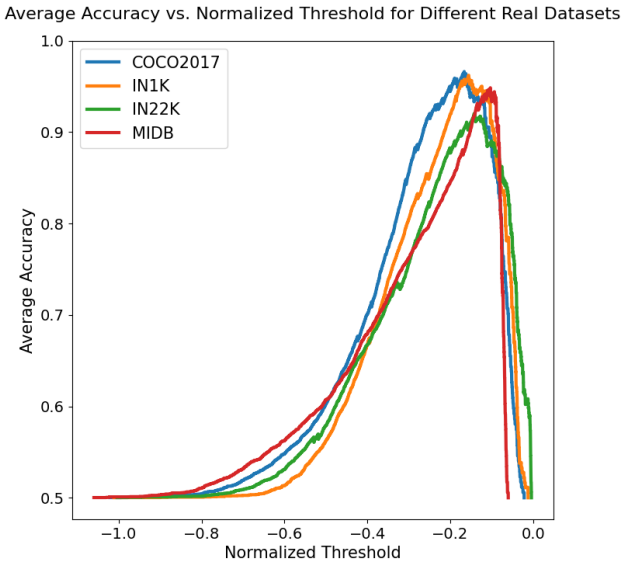

Figure 8. Zero-shot detection performance of our method versus different normalized thresholds.

ploying forensic self-descriptions for zero-shot detection may not require extensive threshold calibration for different datasets. Instead, it can rely on a pre-set threshold determined from a small validation set, simplifying deployment while maintaining consistently high performance across diverse datasets.

E. Impact of Real Training Dataset Choice

In this section, we examine the impact of the choice of the real dataset used for training to the overall zero-shot detection performance. We do this by evaluating the performance of forensic self-descriptions derived from residuals produced by scene content predictive models trained on one real dataset and tested on entirely different real datasets. The results of this experiment are provided in Fig. 6.

The results in Fig. 6 illustrates the robustness and generalization capability of our proposed method when applied to unseen real datasets. Specifically, we achieve consistently high performance across all scenarios, with average AUC values typically remain around 0.94, regardless of the real dataset used for training or testing. This result highlights the fact that our method can maintain its strong performance even when the specific characteristics of real data available during training may differ from those encountered in the wild.

Notably, on MIDB where we observe a slight gap in performance when other datasets are used for training. This effect can be qualitatively explained by examining Fig. 7 in Sec. F, where we observe that the self-descriptions obtained from real images in MIDB are significantly different from those in other datasets. This is because in constrast to other datasets where images are often downloaded from the internet, images in MIDB come directly from a camera without any subsequent post processing or compression. Therefore, for practical applications, this finding shows that better performance may be achievable by training the scene content predictive models on a larger, combined set of real images from diverse sources.

F. Qualitative Study of Forensic Self-Descriptions of Different Real Datasets

In this section, we explore the characteristics of the forensic self-descriptions of real images from different sources. In particular, we examine the power spectrum of different filters in the forensic self-descriptions across real image datasets (COCO2017, IN-1k, IN-22k, and MIDB). We show these visualizations in Fig. 7.

From Fig. 7, we can observe that the power spectra of the filters exhibit consistent patterns across the different

Table 14. Runtime as Images per second (im/s) and Number of Parameters for our method and competing methods in this paper.

Method	Time (im/s)	# Params
Ours	0.11	**2K**
CnnDet	22.72	23M
PatchFor	22.93	191K
LGrad	19.53	46M
UFD	11.13	427M
DE-FAKE	4.90	620M
Aeroblade	5.66	14M
ZED	0.88	809M
NPR	22.92	1.4M
DCTCNN	192.67	170K
RepMix	186.85	24M
Fang et al.	**289.54**	1.2M
POSE	24.53	22M
Abady et al.	17.02	150M
FSM	24.06	437K
ExifNet	19.56	76M
CLIP-ViT-Base	159.31	151M
CLIP-ViT-Large	25.84	427M
ResNet-50	20.74	23M

datasets. For instance, similar spectral structures are observed in $FFT(\phi_2)$ and $FFT(\phi_3)$ of COCO2017, IN-1k, and IN-22k. While the spectral structures of other filters are slightly different across these three datasets, we observe that they are still significantly distinct from those produced by synthetic images (see Fig. 4 in our main paper). This shows that our method of using forensic self-descriptions can accurately distinguish AI-generated images from real images. This is also supported by our experimental results in Sec. 5.3 of our main paper, where our average zero-shot detection performance is 0.960 with a standard deviation of only 0.01. In contrast, other methods have significantly more deviations between different real sources. For instance, NPR suffers big performance drops in IN-1k and IN-22k, ZED in IN-1k, and Aeroblade in IN-22k.

Notably, we see a much bigger difference in the spectral patterns of the self-descriptions of images in the MIDB dataset. This is because real images in this dataset come directly from a camera without subsequent post processing or compression. The fact that our forensic self-descriptions can capture these differences show that our method is highly generalizable and adaptable to many real-world image processing conditions.

G. Space-Time Complexity Analysis

In this section, we examine the runtime and memory cost in terms of the number of parameters of ours and competing methods. We record the average inference runtime per image by performing inference for each method using 1000 images from the ImageNet-1k dataset using a machine with an NVIDIA A6000 GPU.

The runtime and parameter comparison in Table 14 highlights a significant trade-off in our method. Our approach has the lowest number of parameters (2K), making it highly efficient in terms of model size and memory requirements. However, it takes the longest time per image (0.11 image/s), primarily due to the iterative residual modeling process, which requires optimization for each image to accurately capture forensic microstructures. In contrast, other methods such as Fang et al. achieve much faster runtimes (289.54 image/s) by leveraging pre-trained models or architectures optimized for inference speed, albeit at the cost of significantly larger parameter sizes. These results underscore that while our method is highly compact and lightweight, the computational complexity of its residual modeling process remains a bottleneck. In future work, we will address this issue by exploring faster optimization techniques or approximations to further enhance the practicality of our approach without sacrificing its accuracy and generalization capabilities.