General and Domain-Specific Zero-shot Detection of Generated Images via Conditional Likelihood Supplementary Material

Roy Betser[1,2] Omer Hofman[2] Roman Vainshtein[2] Guy Gilboa[1]
[1]Technion - Israel Institute of Technology [2]Fujitsu Research of Europe
roy.betser@fujitsu.com

Abstract

In this supplementary material, we provide additional implementation details to ensure the full reproducibility of CLIDE. We also present extended explanations of the statistical tests used to assess the normality of whitened embeddings and the correlation between features. Furthermore, we include additional experiments and analyses that provide deeper insights into the behavior and robustness of our approach under different conditions. Specifically, we analyze the impact of semantic content on likelihood values, present additional examples of the flipping classification phenomenon, and conduct a comprehensive ablation study to assess the effect of different hyperparameter choices, likelihood settings and robustness to perturbations. Next, we provide further details on the newly introduced synthetic datasets of damaged car images and invoice images. In addition, we formalize the domain-specific zero-shot generated-image detection *task and its evaluation protocol, provide a head-to-head comparison between* global *and* conditional *likelihood formulations, and include a case study on* medical images *that illustrates applicability, constraints, and failure modes. The sections are structured as follows:*

A. Reproducibility

To ensure full reproducibility, we provide a detailed algorithm (Algorithm 1) along with all necessary implementation details (Results section in the paper). The supplementary material includes our source code, representative set matrices and global whitening matrices, enabling direct validation of our results. In addition, we provide a subset of our synthetic datasets. Due to space limitations, only a portion of the data is included in the supplementary material, while the full dataset, along with our code and pre-computed matrices, will be publicly released upon acceptance.

B. Statistical Tests

Evaluating whether the whitened embeddings follow a normal distribution is performed using the widely used normality tests, Anderson-Darling test [1] and D'Agostino-Pearson test [12]. These tests assess deviations from normality by examining the distributional characteristics of the data. Additionally, we provide details on our correlation measurement metric, the off-diagonal maximal value, along with results for all statistical tests.

B.1. Anderson-Darling Test

The Anderson-Darling (AD) test is a modification of the Kolmogorov-Smirnov test [37] that places more emphasis on the tails of the distribution. It measures the goodness-of-fit of a dataset $X = \{x_1, x_2, \ldots, x_n\}$ to a given cumulative distribution function (CDF), typically the normal distribution. The test statistic is computed as:

$$A^2 = -n - \sum_{i=1}^{n} \frac{2i-1}{n} \left[\ln F(x_i) + \ln(1 - F(x_{n+1-i}))\right], \tag{1}$$

where: - $F(x)$ is the cumulative distribution function (CDF) of the normal distribution, - x_i are the ordered sample values, - n is the sample size.

A higher A^2 value indicates greater deviation from normality. The critical value for rejecting normality is above

0.752. An illustration of the different empirical CDFs of different data distributions is given in Fig. 1.

Algorithm 1 Zero-Shot Generated Image Detection Using Conditional CLIP Likelihood

Require: Inputs:
- I - input image
- X - set of real image embeddings
- k - number of samples used for whitening
- m - number of dimensions used for whitening
- th - threshold value for binary classification.

Outputs:
- $\ell(x \mid X, k, m)$ - conditional likelihood score.
- C - Output class - Real ('0') or Generated ('1').

1: **Step 1:** Embed the input image using CLIP:

$$I \to x.$$

2: **Step 2:** Compute similarity between embedding $x \in \mathcal{R}^d$ and set of embeddings $X^{d \times N}$:

$$Similarity = \frac{Xx^\top}{\|X\|\|x\|}.$$

3: **Step 3:** Select the k closest embeddings from X, based on highest cosine similarity - X_k.

4: **Step 4:** Perform PCA on the covariance matrix of X_k:

$$\mathbf{\Sigma}_{X_k} = V \Lambda V^\top.$$

5: **Step 5:** Select the m highest eigenvalues and their corresponding eigenvectors - Λ_m, V_m

6: **Step 6:** Compute the mean embedding $\mu(X_k)$ and whitening matrix $W(X_k, m)$:

$$W(X_k, m) = \Lambda_m^{-\frac{1}{2}} V_m^\top.$$

7: **Step 7:** Compute the conditional likelihood score $\ell(x \mid X, k, m)$:

$$\ell(x \mid X, k, m) = -\frac{1}{2}\left(m \log(2\pi) + \|W\left(x - \mu\right)\|^2\right),$$

where $\mu = \mu(X_k)$ and $W = W(X_k, m)$.

8: **Step 8:** Compare the likelihood score to a threshold value th:

$$C = \begin{cases} \text{real}, & \text{if } \ell(x \mid X, k, m) > th \\ \text{generated}, & \text{otherwise} \end{cases}$$

9: **Step 9:** Output a binary decision: **real** or **generated**.

B.2. D'Agostino-Pearson Test

The D'Agostino-Pearson test combines measures of skewness and kurtosis to evaluate normality. Given a dataset $X = \{x_1, x_2, \ldots, x_n\}$, it computes: Skewness (g_1), which measures asymmetry in the distribution, and Kurtosis (g_2), which quantifies the distribution's peak relative to the normal distribution. The test is based on the transformation of skewness and kurtosis into z-scores to form the test statistic:

$$K^2 = Z_1^2 + Z_2^2, \tag{2}$$

where: - Z_1 is the standardized skewness statistic, - Z_2 is the standardized kurtosis statistic. Under the null hypothesis of normality, K^2 follows a chi-square distribution with two degrees of freedom. The p-value is then defined as:

$$p = 1 - F_{\chi_2^2}(K^2), \tag{3}$$

where: $F_{\chi_2^2}$ denotes the cumulative distribution function (CDF) of the chi-square distribution with 2 degrees of freedom.

Skewness and kurtosis are computed as:

$$g_1 = \frac{\frac{1}{n}\sum_{i=1}^n (x_i - \mu)^3}{(\frac{1}{n}\sum_{i=1}^n (x_i - \mu)^2)^{3/2}}, \tag{4}$$

$$g_2 = \frac{\frac{1}{n}\sum_{i=1}^n (x_i - \mu)^4}{(\frac{1}{n}\sum_{i=1}^n (x_i - \mu)^2)^2} - 3, \tag{5}$$

where: x_i is a data sample and μ is the data mean, defined as: $\mu = \frac{1}{n}\sum_{i=1}^n x_i$. n is the total number of samples. A skewness value $g_1 > 0$ indicates right-skewed data, while $g_1 < 0$ implies left-skewed data. Similarly, a kurtosis value $g_2 > 0$ suggests heavy tails, whereas $g_2 < 0$ indicates light tails. As both g_1 and g_2 approach zero, the p-value increases, indicating normality. A p-value above 0.05 is typically used as the threshold for normality. An illustration of data with different skewness and kurtosis is available in Fig. 2.

B.3. Correlation Metric

The off-diagonal maximal value (ODM) serves as a quantitative measure of correlation within data samples. It is computed from the covariance matrix $\mathbf{\Sigma}$ of the data X. The ODM is defined as:

$$\text{ODM} = \max_{i \neq j} |\Sigma_{i,j}|, \tag{6}$$

where: - $\mathbf{\Sigma} \in \mathbb{R}^{d \times d}$ is the empirical covariance matrix of X, - $\Sigma_{i,j}$ is the covariance between feature i and feature j, - The maximum is taken over all off-diagonal elements ($i \neq j$).

A low ODM value indicates that off-diagonal elements in $\mathbf{\Sigma}$ are small, suggesting low correlation between different features. Since the covariance matrix's diagonal elements represent feature variances, a nearly diagonal $\mathbf{\Sigma}$ implies that the dataset consists of uncorrelated or weakly correlated features. This metric is particularly useful in whitening transformations, where non-correlated features are desirable.

B.4. Test results

An ablation study is conducted on all three statistical tests, examining the effect of varying k (number of samples used for whitening) and m (number of dimensions used for whitening). The results, presented in Fig. 3, demonstrate that normality is preserved across all tested values of k and m. However, correlation increases significantly for larger dimension values. The tests are performed on 10k real and 10k generated images of general content, randomly sampled from the full evaluation dataset of general images. The representative data used for whitening is MS-COCO [22] validation set (same as in the general image domain in the paper).

C. Semantic Influence on Likelihood Values

To illustrate the influence of semantic content on the likelihood score, we analyze the likelihood values of different images of a zebra and a plate of food (Fig. 4). Since a zebra in the wild and a plate of food have entirely different semantic meanings, we expect these groups of images to exhibit distinct likelihood ranges. Indeed, both real and generated images of zebras consistently show higher likelihood values compared to images of a plate of food, with real zebra images scoring the highest among them. This observation confirms that semantic content significantly impacts likelihood estimation. [2] demonstrate that textual complexity directly influences likelihood values, with more complex text yielding lower likelihood scores. Similarly, we suggest that the greater complexity among different plates of food, compared to the variation among different zebras, contributes to the observed likelihood difference.

The likelihood values in this analysis are computed using a single, pre-computed whitening matrix, based on the MS-COCO validation set, which serves as a representative dataset for general images. Notably, both animals and food are highly prevalent in this dataset, with plates of food being particularly frequent. This example underscores the necessity of minimizing semantic influence on likelihood values to ensure that the distinction between real and generated images is not confounded by semantic priors. By addressing this issue, we refine our detection method to focus mostly on authenticity rather than content. Furthermore, this highlights the need for a conditioning mechanism that adapts likelihood estimation to specific domains, reducing the bias introduced by semantic content and ensuring fair and reliable classification across different image types.

D. Flipping Examples

We present the distribution of real and generated images in the damaged cars domain, per generative model and per detection method, in Fig. 5, for the damaged cars domain. The results indicate that even with a limited number of generative models, all other methods exhibit poor separation and, more importantly, "flipped classification". This phenomenon arises when images generated by different models have opposing relationships with real images. In other words, for a detection method to correctly classify most images from one generative model as generated, it must misclassify a significant portion of images from another model as real. This represents a major drawback in other detection methods, whereas our approach is entirely unaffected. Even when the separation is less pronounced (*e.g.* UnCLIP, bottom row), the classification direction remains consistent.

E. Ablation Study and Perturbations Robustness.

E.1. Ablation Study

We conduct several ablation studies to evaluate our method. First, we examine the impact of varying the number of dimensions used for whitening (m). Results (Fig. 6.a) indicate that the optimal range is 300–400 dimensions, approximately 50% of the original 768 dimensions. Next, we assess the effect of the number of samples used for whitening (k). Results (Fig. 6.b) suggest that reducing the number of samples improves performance. To further investigate, we replace the MS-COCO validation set (5K images) with the test set (40K images) as the representative data. Findings (Fig. 7.a) show that the optimal sample size is around 10% of the representative set. Since PCA requires $k > m$ and 10% of the data is usually insufficient, the optimal choice is approximately $k = m + 100$. To conclude this part we add an experiment with small representative set sizes, from 200 to 1K, in the car domain. We use $k = 500$ when the representative set is 600 or more, and $k = \textbf{rep size} - 100$ otherwise. Results are reported in Tab. 4. In the general image domain, reducing the representative set size lowers AUC only gradually and performance remains strong even with 500 images. However, when the set is cut to about 200 images (limiting the dimensionality to less than 100), the method collapses. In the damaged cars domain, which is much narrower, even 200 images suffice and the performance drop is minimal. This suggests that for domain-specific cases, a small amount of real data is enough.

Additionally, we compare a global whitening matrix (no per-image sampling) to our local scheme (Fig. 7.b), revealing a notable performance drop, though AUC remains above

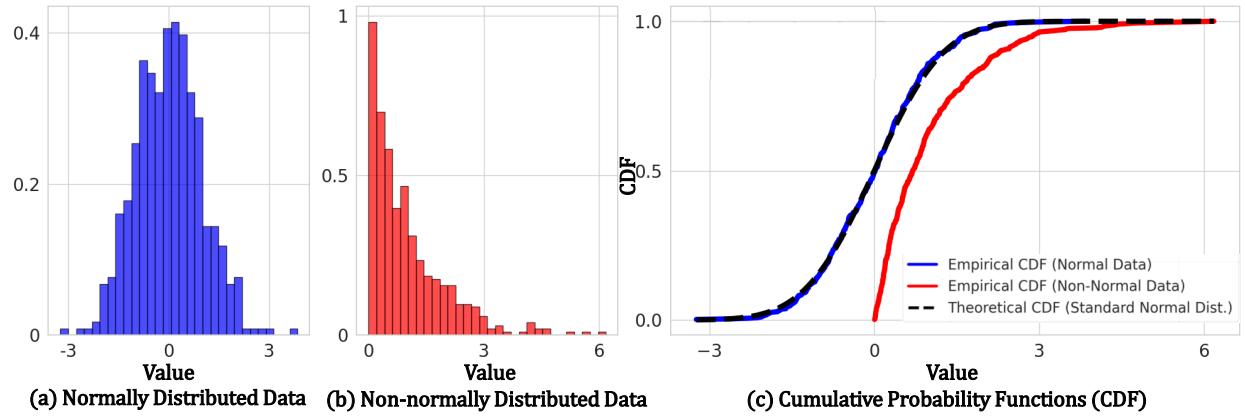


Figure 1. **Anderson-Darling test.** Two histograms of normally distributed (a) and non-normally distributed data (b), and the empirical CDF's compared to the theoretical CDF of normal distribution (c).

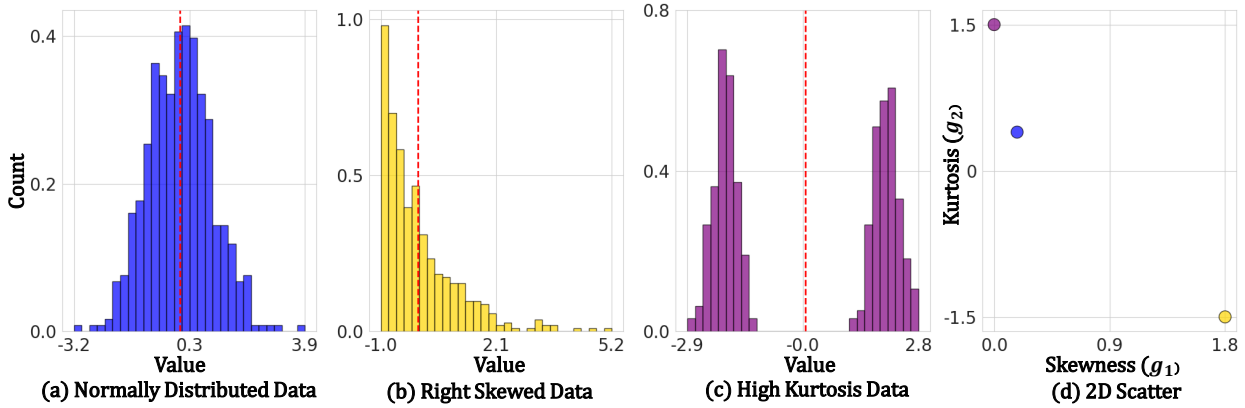


Figure 2. **D'Agostino-Pearson test.** Histograms of three datasets: (a) normally distributed data, (b) right-skewed data, and (c) high-kurtosis data. (d) presents a 2D scatter plot of skewness versus kurtosis, where each distribution is represented by a single point.

0.85 — the highest result among competing methods. In the global setting, we randomly sample a subset of the representative set before evaluation and use a pre-computed whitening matrix for all input images. Each test is repeated five times, and we report the mean result with the standard deviation. All of the experiments above are conducted in the general image domain, where we randomly sample 10K real and 10K generated images.

Finally, we evaluate a combined representative dataset consisting of both the general and damaged cars representative sets, assessing performance across both domains. Results (Fig. 8.a) show that both domains remain stable, achieving results similar to the single-domain representative set case. In the global whitening setting (Fig. 8.b), the general image domain experiences a slight performance drop (as in the single domain case - Fig. 7.b), while the damaged cars domain maintains its performance level. Overall, integrating representative data from multiple domains enhances the robustness of our detection model. For these experiments, we use the same 10K real and 10K generated general images, and 2K real and 2 generated images from the damaged cars domain, as used in Sec. 5.3 of the main paper.

Overall, our method is robust to parameter selection, can generalize across different domains (joint representative set), and can improve efficiency by operating in a global setting.

E.2. Robustness to image perturbations

To evaluate the robustness of our method, we conduct experiments under five common image perturbations: Gaussian blur, JPEG compression, random resizing and cropping, Gaussian noise injection, and color jittering. Each perturbation is applied at five severity levels, ranging from mild (Level 1) to extreme (Level 5). The specific parameter settings for each perturbation type and level are summarized in Tab. 1 and examples of all perturbations in all levels are presented in Fig. 10, Fig. 11 for both real and generated

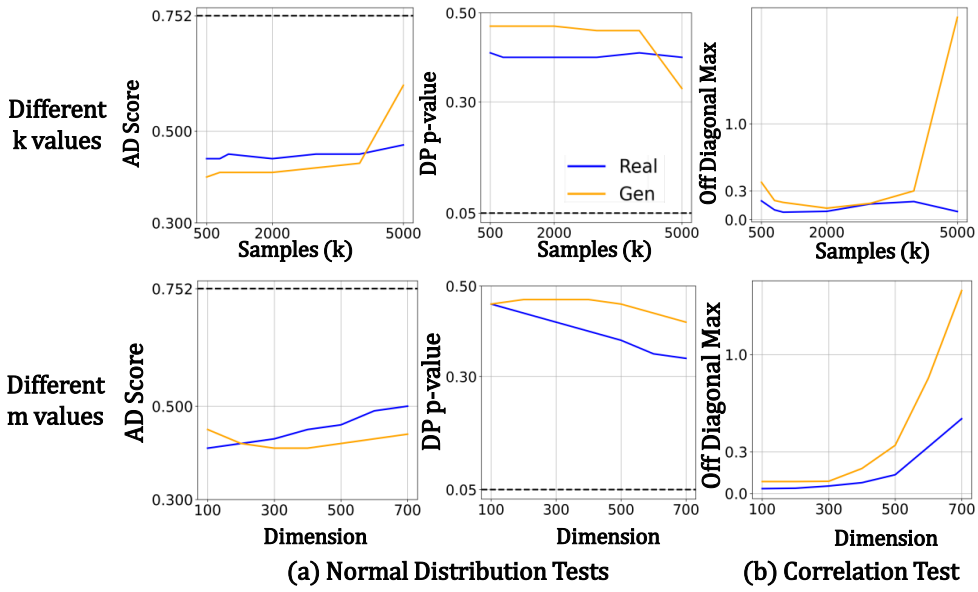


Figure 3. **Statistical test results.** (a) Average values of normality tests computed across all whitened embedding features. (b) ODM values. The top row presents results for varying numbers of samples used for whitening (k), with a fixed number of dimensions ($m = 400$). The bottom row shows results for varying dimensions (m), while keeping the number of whitening samples constant ($k = 1000$). Threshold values for the normality tests are indicated by a dashed black line, where AD values below the threshold indicate normality, and DP values above the threshold suggest normality.

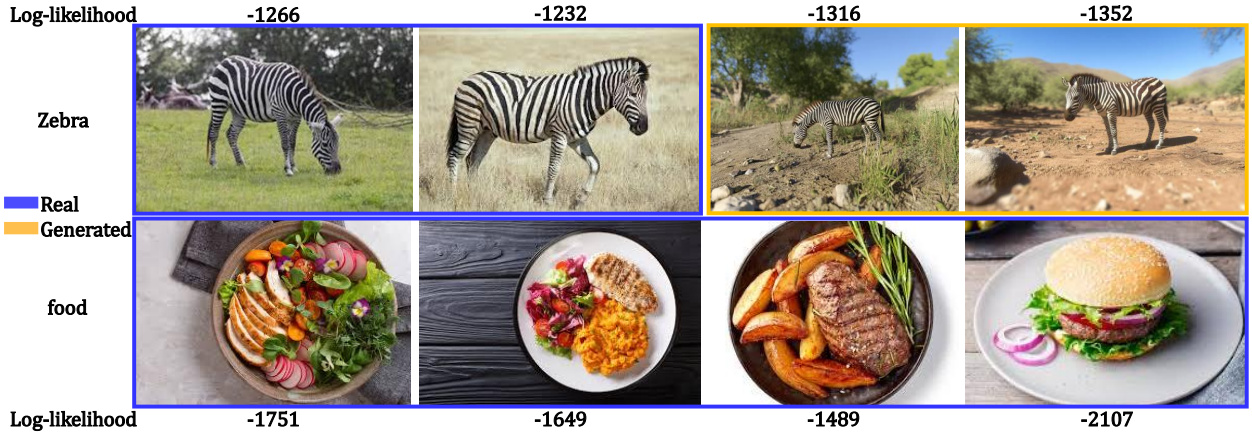


Figure 4. **Zebra and a plate of food likelihood values.** Top row - images of a zebra, two real (left) and two AI generated (right). Bottom row - four real images of a plate of food. All images of a zebra, including generated images, have higher likelihood values compared to images of a plate of food. Real images of a zebra have higher likelihood values compared to generated images.

images. Gaussian blur severity is controlled by the blur radius, with larger radius corresponding to stronger blur effects. JPEG compression levels decrease the image quality percentage, with lower quality introducing more compression artifacts. The resizing and cropping perturbation is parameterized by scale ranges, where lower minimum scales correspond to more aggressive cropping. Gaussian noise severity is adjusted by increasing the standard deviation of the injected noise. Color jittering affects brightness, contrast, saturation, and hue (B/C/S/H), with higher values introducing stronger color distortions. The experiments are performed on the same set of 10K real and 10K generated general images used in all ablation experiments (Sec. 5.3 in the main paper).

Our results, presented in Fig. 9, demonstrate that the method is robust across a wide range of perturbations. Performance remains high (AUC > 0.8) under most conditions, with the exception of the most severe noise (standard devi-

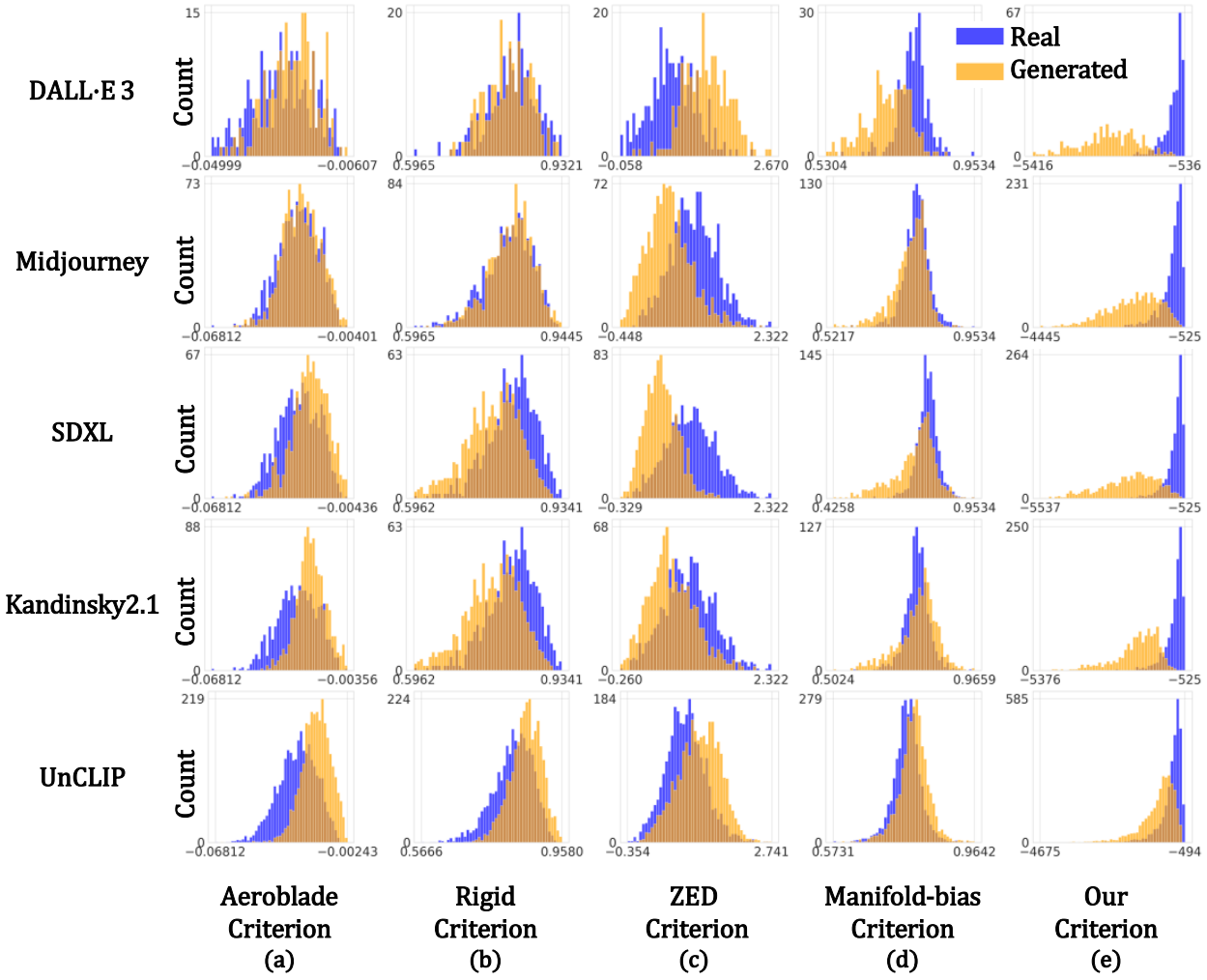


Figure 5. **"Flipped classification" example.** Each row corresponds to a different generative model, while each column represents a different detection method. Our method demonstrates the strongest separation across all generative models, with generated content consistently exhibiting lower criterion values. In contrast, other methods show poor separation and "flipped classification", where images from two generative models are positioned on opposite sides of the real image distribution. Example from the damaged cars domain.

Perturbation Type	**Level 1**	**Level 2**	**Level 3**	**Level 4**	**Level 5**
Gaussian Blur Radius	1	2	3	5	10
JPEG Compression Quality	80%	50%	30%	10%	1%
Resize Crop Scale	(0.85, 0.9)	(0.7, 0.85)	(0.5, 0.8)	(0.3, 0.9)	(0.08, 1.0)
Gaussian Noise Std	0.02	0.05	0.10	0.20	0.50
Color Jitter (B/C/S/H)	0.2/0.4/0.2/0.05	0.4/0.5/0.3/0.07	0.6/0.6/0.4/0.10	0.7/0.7/0.5/0.12	0.8/0.8/0.6/0.15

Table 1. **Perturbation levels.** Summary of the perturbation parameters used at each severity level. The rows correspond to: (1) Gaussian blur radius, where the kernel size is approximately $2 \times \text{radius} + 1$; (2) JPEG compression quality, where lower percentages indicate stronger compression and more visual artifacts; (3) Random resized crop scale ranges, specifying the minimum and maximum scale of the cropped area relative to the original image size; (4) Gaussian noise standard deviation, where higher values correspond to stronger noise levels; (5) Color jitter parameters for brightness, contrast, saturation, and hue adjustments, denoted as B/C/S/H, where larger values introduce more aggressive color perturbations.

ation 0.5) and extreme JPEG compression (quality = 1%). These extreme cases, which introduce heavy pixel-level dis-

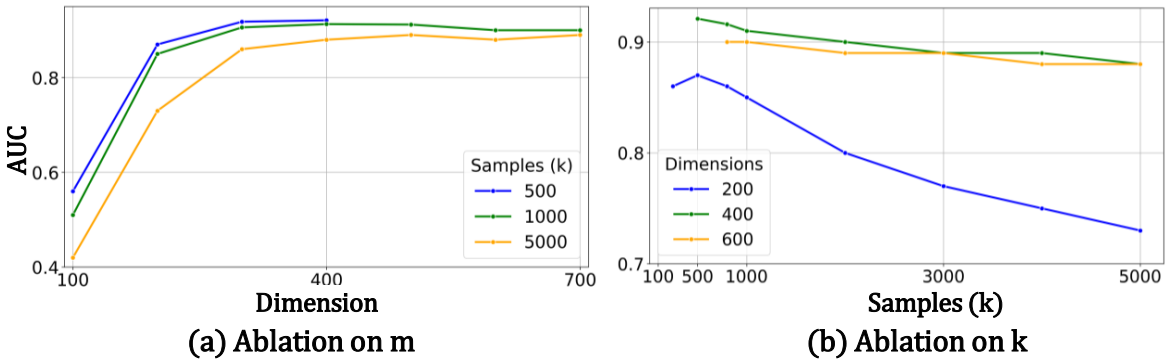


Figure 6. **Ablation on number of samples (k) and number of dimensions (m) used for whitening.** (a) Ablation on m shows that the optimal value is approximately half of the original 768 dimensions. (b) Ablation on k suggests that reducing the number of examples improves performance. Figure 7 further investigates this finding.

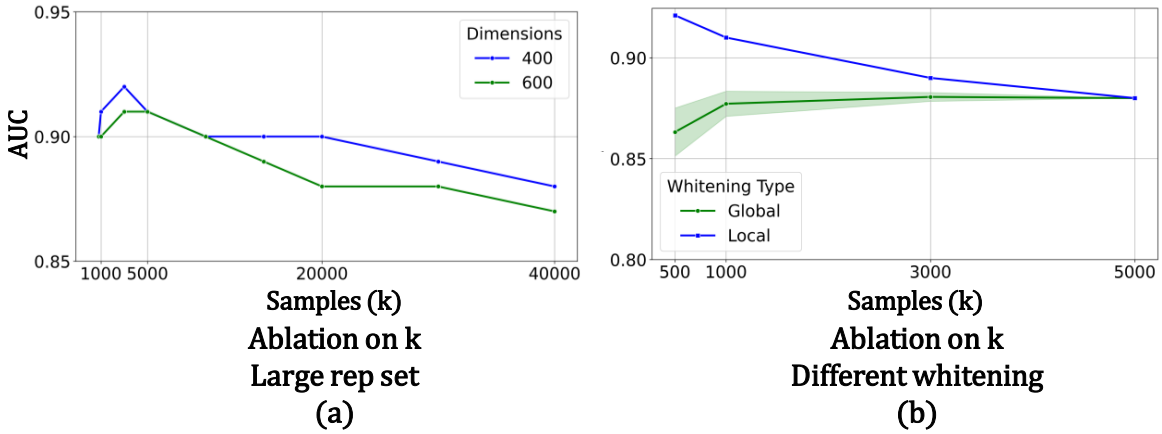


Figure 7. **Additional ablation on the number of samples (k).** (a) Ablation on k using a large representative set (MS-COCO test set, 40K images). With a larger representative set, reducing the number of samples does not monotonically improve results, with the optimal point around 10% of the representative set size. (b) Comparison between our local whitening scheme and a global setting where a single whitening matrix is used for all images. As different subsets can be sampled, we present the mean value with the standard deviation of five different experiments. While the global whitening setting leads to a performance drop, it remains above the highest competitor result (0.85 AUC) in the general image domain. In this experiment $m = 400$ for both settings.

tortions or significant compression artifacts, lead to more noticeable performance degradation. Across other perturbations, including strong color jittering, aggressive cropping, and large blur radius, the method maintains stable detection capabilities, suggesting that it is resilient to common variations encountered in real-world images. These findings further support the robustness and generalization ability of our approach.

F. Synthetic Dataset

F.1. Damaged cars images

CarDD [38], a benchmark dataset of 4K real damaged car images, serves as the real image source in this domain. It is split into training (2,816 images), validation, and test sets (1,184 images combined). To the best of our knowledge, no synthetic dataset exists for damaged car images. Thus, we generate the first publicly available synthetic dataset for this domain.

We synthesize images using two diffusion models (SDXL [30] and Kandinsky2.1 [32]) and two commercial tools (Midjourney [24] and DALL·E 3 [4]). Among these, Midjourney produces the most photorealistic results, while DALL·E 3 performs the poorest. Due to limited access to commercial tools, we generate 1,160 images with Midjourney and 252 with DALL·E 3, while each open-source diffusion model contributes 999 images.

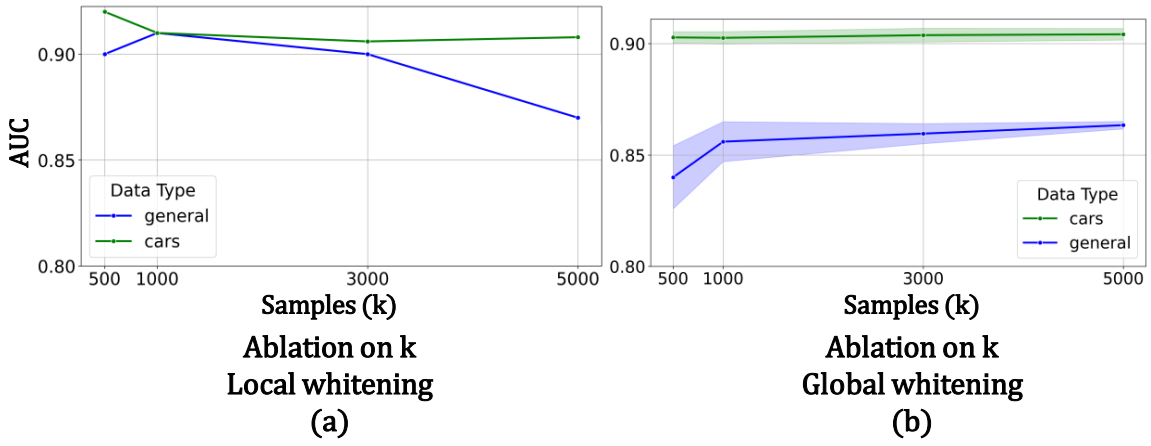


Figure 8. **Ablation of a combined setting.** Evaluation using a representative set combining both general image and damaged cars domains. In the local whitening setting (a), both domains maintain high performance. In the global whitening setting (b), performance remains high for the damaged cars domain, while the general image domain shows a slight drop, similar to Fig. 7.b (single-domain case). Overall, combining representative data from multiple domains enables a robust detection model in both domains. In these experiments $m = 400$ for all settings.

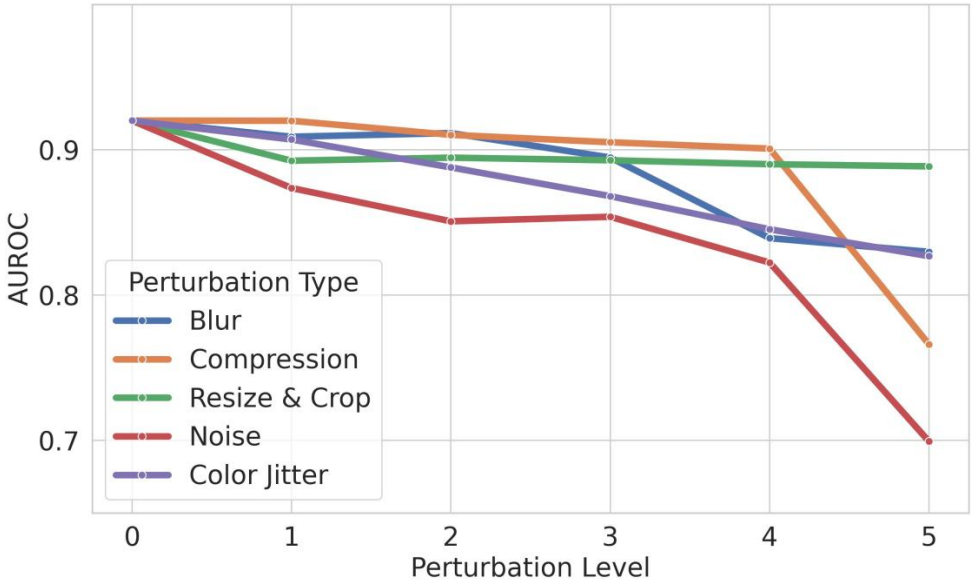


Figure 9. **Robustness to Perturbations.** We evaluate five perturbation types: Gaussian blur, JPEG compression, cropping and resizing, Gaussian noise, and color jittering. For each perturbation, our method is tested across five levels, where Level 1 represents mild perturbations and Level 5 represents severe ones (see Tab. 1 for details). The results show that our method is robust across all perturbations, maintaining high AUC scores (> 0.8) in most cases, with noticeable degradation only at the strongest noise level (std = 0.5) and extreme compression (quality = 1%).

Generating realistic damage-related features (*e.g.* flat tires, broken windshields) remains challenging, as GANs and other diffusion models produced unsatisfactory results. BigGAN [6], for instance, only accepts class labels (*e.g.* "sports car") with no ability to generate damaged vehicles. We also found that earlier diffusion models produce lower-quality images. Specifically, previous versions of Stable Diffusion [34] and open-source implementations of Glide [26] yield very poor results. Thus, we select SDXL and Kandinsky2.1, as they produce the highest-quality results among the open-source diffusion models examined.

The challenge is twofold: first, generating realistic dam-

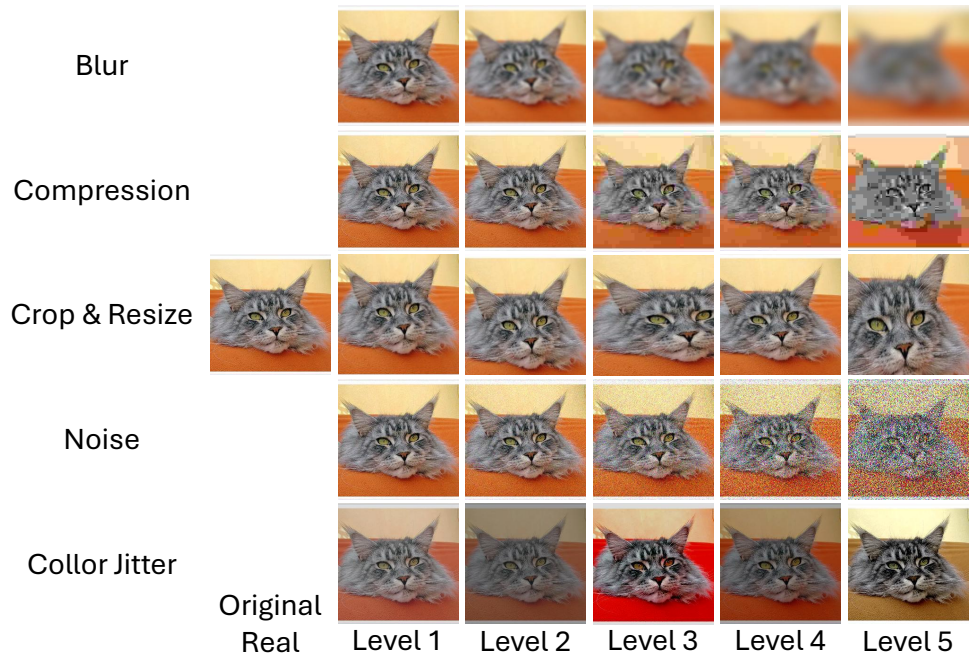


Figure 10. **Perturbation examples for different levels.** The original image is a real image from LAION [35] dataset.

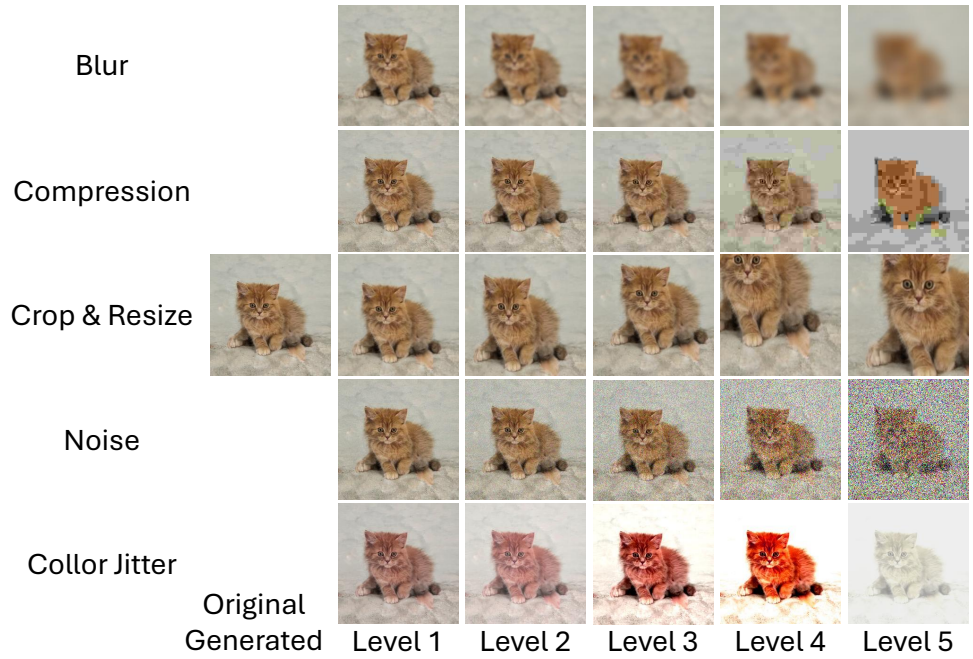


Figure 11. **Perturbation examples for different levels.** The original image is a generated image from [27].

age is non-trivial, often resulting in either unrealistic damage on a realistic car or no damage at all. Second, prompts explicitly describing damage can sometimes lead to unrealistic overall image generation. To improve realism and diversity, we employ the UnCLIP image-to-image framework [31], conditioning the generation on real images. The same 2,816 real images used for evaluation serve as conditioning inputs, making detection more challenging.

The resolution of real images in [38] varies. In contrast, each generative model produces images at a fixed resolu-

Generative Model	ZED D_0				ZED $\lvert D_0 \rvert$				ZED Δ_{01}				ZED $\lvert \Delta_{01} \rvert$				CLIDE (ours)			
	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑
ProGan [17]	0.74	0.75	0.7	0.64	0.74	0.75	0.7	0.64	0.69	0.73	0.65	0.55	0.32	0.42	0.58	0.42	**0.95**	**0.96**	**0.87**	**0.88**
StyleGan [18]	0.75	**0.77**	0.71	0.65	0.75	**0.77**	0.71	0.65	0.48	0.58	0.62	0.46	0.55	0.59	0.6	0.46	**0.76**	0.71	**0.75**	**0.73**
StyleGan2 [19]	0.75	0.8	0.68	0.61	0.75	0.8	0.68	0.61	0.49	0.58	0.61	0.45	0.54	0.6	0.61	0.48	**0.8**	**0.82**	**0.73**	**0.68**
BigGAN [6]	0.45	0.47	0.63	0.5	0.44	0.46	0.62	0.49	0.34	0.44	0.6	0.44	0.69	0.68	0.68	0.61	**0.96**	**0.96**	**0.87**	**0.88**
GauGan [29]	0.33	0.41	0.6	0.44	0.32	0.41	0.59	0.42	0.06	0.32	0.6	0.43	0.74	0.78	0.68	0.62	**0.83**	**0.85**	**0.76**	**0.73**
CycleGan [41]	0.22	0.38	0.6	0.43	0.21	0.37	0.58	0.41	0.28	0.38	0.59	0.43	0.66	0.7	0.65	0.56	**0.98**	**0.98**	**0.9**	**0.91**
CRN [10]	**0.99**	**0.99**	**0.92**	**0.92**	0.96	0.98	0.91	0.91	0.59	0.73	0.6	0.43	0.4	0.58	0.58	0.4	0.98	**0.99**	0.91	**0.92**
SD V1.4 [34]	0.62	0.65	0.65	0.54	0.62	0.64	0.64	0.53	0.35	0.43	0.61	0.46	0.65	0.63	0.66	0.58	**0.9**	**0.91**	**0.82**	**0.82**
SD V1.5 [34]	0.61	0.63	0.65	0.54	0.6	0.63	0.64	0.53	0.37	0.43	0.61	0.46	0.63	0.61	0.65	0.56	**0.9**	**0.91**	**0.83**	**0.82**
Guided DM [14]	0.58	0.55	0.68	0.59	0.56	0.55	0.65	0.56	0.38	0.4	0.63	0.51	0.5	0.47	0.63	0.52	**0.87**	**0.87**	**0.79**	**0.78**
LDM 100 [34]	0.52	0.56	0.63	0.49	0.5	0.55	0.6	0.47	0.45	0.48	0.62	0.49	0.56	0.52	0.64	0.54	**0.92**	**0.93**	**0.84**	**0.85**
LDM 200 [34]	0.53	0.56	0.63	0.5	0.51	0.56	0.61	0.48	0.49	0.5	0.63	0.51	0.52	0.49	0.63	0.52	**0.92**	**0.93**	**0.85**	**0.85**
Glide 50 27 [26]	**0.98**	**0.98**	**0.91**	**0.92**	0.93	0.95	0.87	0.88	0.57	0.54	0.67	0.59	0.43	0.42	0.63	0.53	0.91	0.92	0.84	0.84
Glide 100 27 [26]	**0.98**	**0.98**	**0.91**	**0.92**	0.94	0.95	0.87	0.88	0.64	0.59	0.7	0.63	0.36	0.39	0.62	0.49	0.91	0.92	0.83	0.83
Glide 100 10 [26]	**0.98**	**0.98**	**0.92**	**0.92**	0.93	0.95	0.88	0.88	0.57	0.53	0.67	0.59	0.43	0.42	0.63	0.52	0.91	0.92	0.83	0.83
ADM [14]	0.47	0.46	0.65	0.55	0.46	0.44	0.64	0.53	0.17	0.34	0.6	0.44	0.65	0.63	0.67	0.59	**0.89**	**0.88**	**0.84**	**0.84**
DALL·E 3 [4]	0.45	0.5	0.62	0.47	0.43	0.5	0.59	0.44	0.15	0.34	0.59	0.42	0.77	0.8	0.7	0.66	**0.96**	**0.96**	**0.88**	**0.89**
Midjourney [24]	**0.92**	**0.93**	0.84	0.84	0.9	0.91	0.83	0.83	0.33	0.43	0.6	0.44	0.69	0.71	0.66	0.58	0.9	0.84	**0.85**	**0.85**
VDQM [15]	0.46	0.46	0.63	0.49	0.45	0.46	0.61	0.48	0.2	0.36	0.6	0.43	0.77	0.78	0.72	0.69	**0.92**	**0.93**	**0.84**	**0.84**
Wukong [25]	0.39	0.45	0.6	0.44	0.38	0.45	0.59	0.43	0.34	0.4	0.63	0.5	0.61	0.55	0.66	0.58	**0.95**	**0.96**	**0.88**	**0.88**
All	0.69	0.66	0.69	0.62	0.68	0.66	0.69	0.62	0.53	0.52	0.63	0.5	0.47	0.48	0.61	0.48	**0.91**	**0.91**	**0.84**	**0.84**

Table 2. **Evaluation of generated image detection on general images.** Our method achieves the highest overall performance across all models and surpasses ZED on most generative models. In cases where ZED outperforms our method, the D_0 criterion yields the best results. When evaluating all models combined (bottom row), D_0 achieves the highest score. The best result for each metric in each row is highlighted in bold.

Generative Model	ZED D_0				ZED $\lvert D_0 \rvert$				ZED Δ_{01}				ZED $\lvert \Delta_{01} \rvert$				CLIDE (ours)			
	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑	AUC ↑	AP ↑	F1 ↑	Acc ↑
Artistic Image Domain																				
StyleGan3 [20]	0.38	0.41	0.63	0.5	0.35	0.4	0.65	0.49	0.86	0.85	0.77	0.76	0.14	0.33	0.57	0.4	**0.98**	**0.99**	**0.93**	**0.93**
SD2.1 [34]	0.17	0.34	0.61	0.44	0.16	0.34	0.64	0.47	0.65	0.59	0.67	0.59	0.33	0.39	0.58	0.43	**0.98**	**0.98**	**0.92**	**0.93**
SDXL [30]	0.33	0.43	0.61	0.44	0.34	0.43	0.61	0.44	0.55	0.51	0.64	0.54	0.42	0.44	0.6	0.46	**0.97**	**0.98**	**0.91**	**0.91**
AnimagineXL [23]	0.45	0.59	0.61	0.44	0.49	0.59	0.65	0.49	0.58	0.63	0.61	0.48	0.42	0.49	0.58	0.42	**0.76**	**0.83**	**0.66**	**0.55**
All	0.33	0.41	0.61	0.45	0.34	0.41	0.65	0.48	0.66	0.62	0.67	0.6	0.33	0.39	0.58	0.43	**0.92**	**0.92**	**0.83**	**0.82**
Damaged cars Image Domain																				
Kandinsky2.1 [32]	0.68	0.66	0.68	0.61	0.68	0.66	0.68	0.62	0.9	0.93	0.84	0.83	0.36	0.45	0.61	0.45	**0.97**	**0.98**	**0.93**	**0.93**
SDXL [30]	0.83	0.85	0.76	0.73	0.83	0.85	0.76	0.74	0.64	0.65	0.64	0.5	0.66	0.67	0.66	0.55	**0.97**	**0.97**	**0.91**	**0.91**
DALL·E 3 [4]	0.15	0.33	0.59	0.42	0.14	0.33	0.58	0.41	0.16	0.33	0.62	0.46	0.21	0.35	0.62	0.46	**0.99**	**0.99**	**0.93**	**0.93**
Midjourney [24]	0.78	0.75	0.74	0.71	0.77	0.75	0.74	0.71	0.87	0.87	0.82	0.8	0.37	0.43	0.62	0.47	**0.93**	**0.94**	**0.87**	**0.86**
UnCLIP [31]	0.31	0.38	0.61	0.46	0.31	0.38	0.61	0.46	0.38	0.4	0.64	0.51	0.35	0.4	0.62	0.47	**0.8**	**0.8**	**0.72**	**0.67**
All	0.61	0.54	0.68	0.61	0.61	0.54	0.68	0.61	0.66	0.57	0.71	0.64	0.42	0.44	0.63	0.48	**0.92**	**0.92**	**0.85**	**0.84**
Invoice Image Domain																				
GPT-Image-1 [28]	0.6	0.56	0.65	0.54	0.61	0.73	0.59	0.66	0.56	0.7	0.55	0.64	0.59	0.59	0.38	0.54	**0.91**	**0.92**	**0.82**	**0.82**

Table 3. **Detection performance on domain-specific images.** Our method consistently achieves the highest performance across all generative models and in the combined settings for each domain. Different ZED criteria perform best for different generative models. In the combined evaluations for the art and damaged car domains, the Δ_{01} criterion achieves the highest ZED performance, while in the invoice domain, $|D_0|$ performs best. The best result for each metric in each row is highlighted in bold.

tion, though resolutions differ across models. DALL·E 3 generates images at 1024×1024, Midjourney at 1232×928, SDXL at 1024×1024, Kandinsky at 512×512, and UnCLIP at 256×256. In total, we generate over 6K synthetic images. For per-model comparisons, we balance real and generated images, ensuring consistent real image selection across all detection methods. When using synthetic images from all generative models collectively, we sample 641 images per model and include all 252 images from DALL·E 3. We provide a subset of our dataset in our supplementary material (presented in Fig. 12), organized by generative model for easy access and evaluation. The full dataset, together with our code (also available in the supplementary) will be publicly released upon acceptance.

F.2. Invoice document images

Invoices represent a critical domain with high practical relevance for financial institutions, e-commerce platforms, and document verification systems. While prior work has primarily focused on information extraction and document classification [3, 13, 40], little attention has been given to the detection of synthetically generated invoices. To the best of our knowledge, no publicly available synthetic invoice dataset currently exists.

Rep size	$\|R\| = 200$	$\|R\| = 500$	$\|R\| = 700$	$\|R\| = 1000$	$\|R\| = 1500$	$\|R\| = 2000$	$\|R\| = 3000$	$\|R\| = 5000$
General	0.48	0.84	0.85	0.86	0.87	0.89	0.90	0.91
Rep size	$\|R\| = 200$	$\|R\| = 300$	$\|R\| = 400$	$\|R\| = 500$	$\|R\| = 600$	$\|R\| = 700$	$\|R\| = 800$	$\|R\| = 1000$
Cars	0.904	0.909	0.912	0.914	0.918	0.921	0.924	0.929

Table 4. Detection performance (AUC values) across different representative set sizes - $|R|$. Experiment are performed on the general image domain and on images from the damaged cars domain.

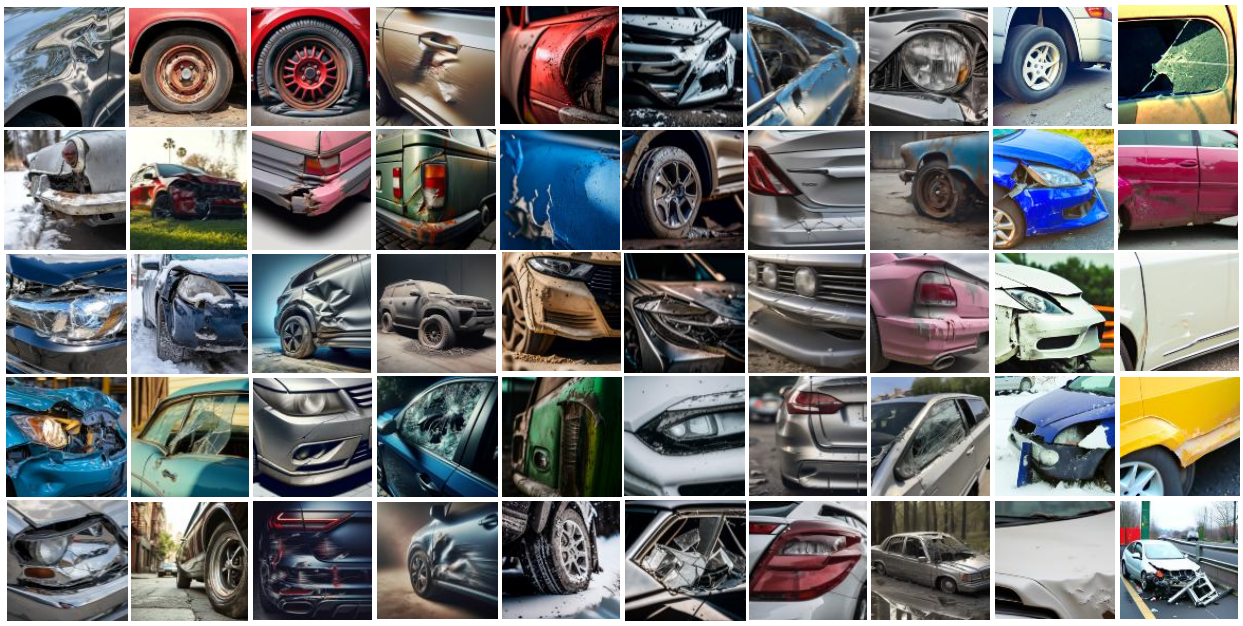


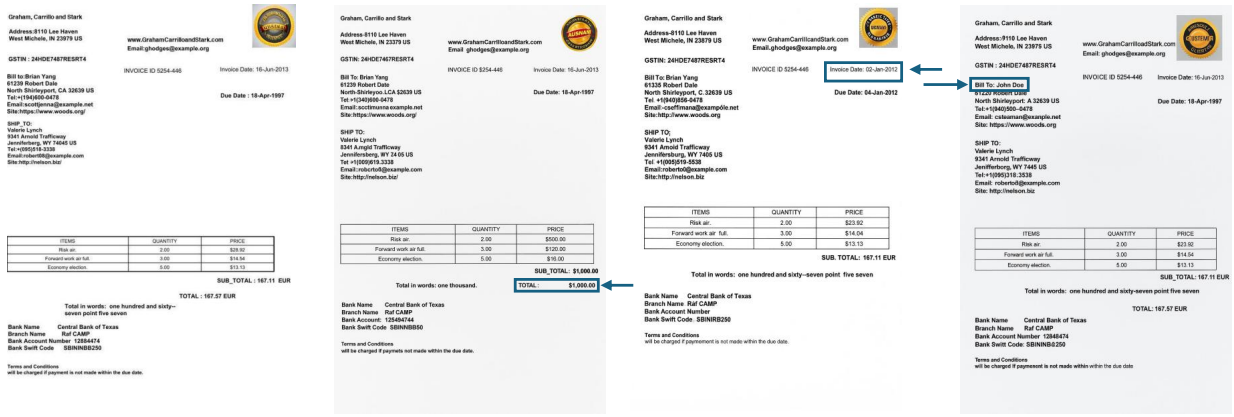
Figure 12. **Synthesized images of damaged cars.** These ten images per generative model are available in the supplementary material as a subset of our generated dataset. The full dataset will be publicly released upon acceptance.

We address this gap by generating the first publicly available synthetic dataset of invoice images. Real invoices are sourced from the InvoiceXpert dataset [39], which contains a diverse collection of real-world invoices. From this dataset, we select 3K images for testing (from the test and validation sets) and an additional 1K images for calibration (from the training set). All synthetic invoices are generated using the GPT-Image-1 model [28], which is, to date, the only model capable of producing high-quality, text-rich, and visually coherent document images.

Each synthetic image is generated by conditioning on a corresponding real invoice from the test set, introducing controlled random variations in key elements such as prices, product names, and dates. This strategy increases the diversity of the generated dataset while maintaining structural and semantic similarity to the original documents, making the detection task substantially more challenging. Examples of a real invoice and generated versions of it are provided in Fig. 13.

Prior generative models, including GANs and earlier diffusion-based models, are not capable of producing text-heavy images with sufficient visual and semantic fidelity. Powerful models such as SDXL [30] and Kandinski 2.1 [32] fail to generate document images with the structural precision and textual accuracy required for this task. Even the UnCLIP framework [31], that receives an image as input, generates poor results with non realistic letters and words. See examples of all three models in Fig. 14. GPT-Image-1 overcomes these limitations, providing high-quality document synthesis for the first time.

All synthetic invoices in our dataset are generated at a fixed resolution of 1024×1536 pixels, consistent with the output format of GPT-Image-1. The real invoices in the InvoiceXpert dataset have varying resolutions, reflecting the diversity of real-world sources. The complete dataset will be publicly released upon acceptance.



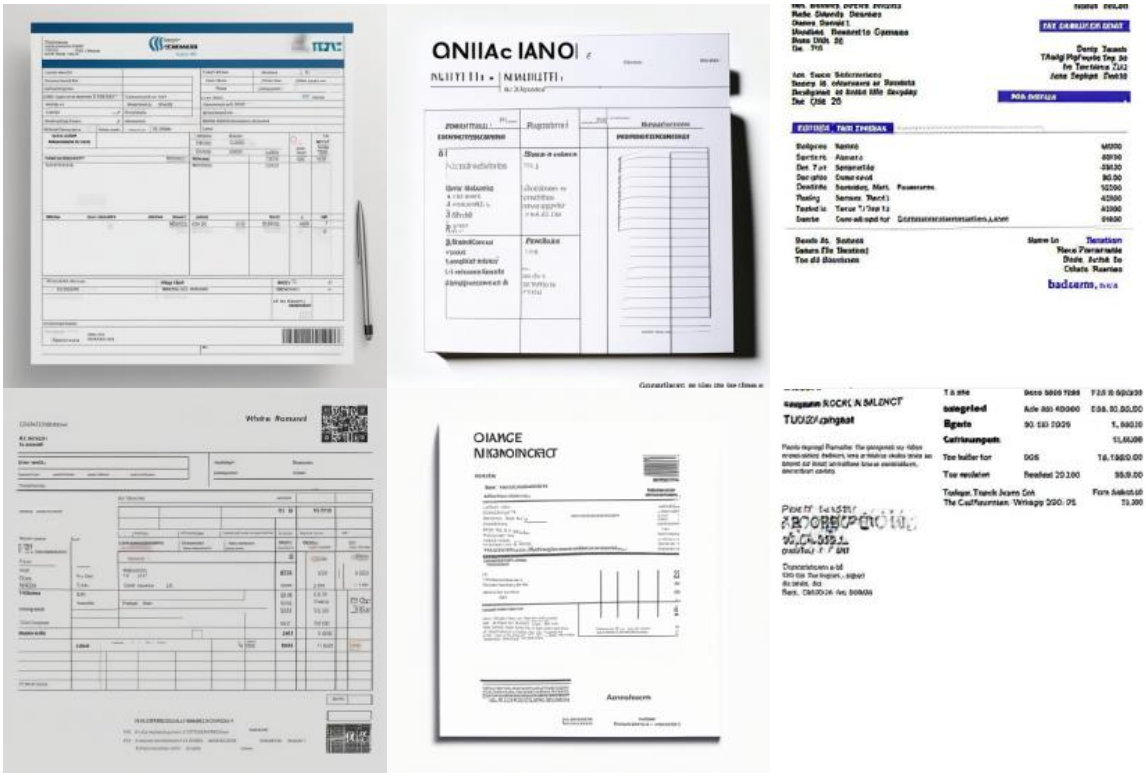
Real Document

Different price
Request – total amount – 1000$

Different date
Request – Jan 2/4, 2012

Different name
Request – name – John Doe

Figure 13. **Synthesized invoice images.** In this example, a real invoice image (most left) is edited three times: second from the left changes the price, second from the right changes the date, and the most right changes the name on the invoice. All other details remain visually and structurally similar.



SDXL
(a)

Kandinsky
(b)

UnCLIP
(c)

Figure 14. **Synthesized invoice images by other models.** Invoice images generated using SDXL, Kandinsky, and UnCLIP. All generated images contain random, non-legible characters and would clearly not be perceived as real by a human observer. The blur in part of the images is originated in the generative output. These low quality generations were omitted from our evaluations.

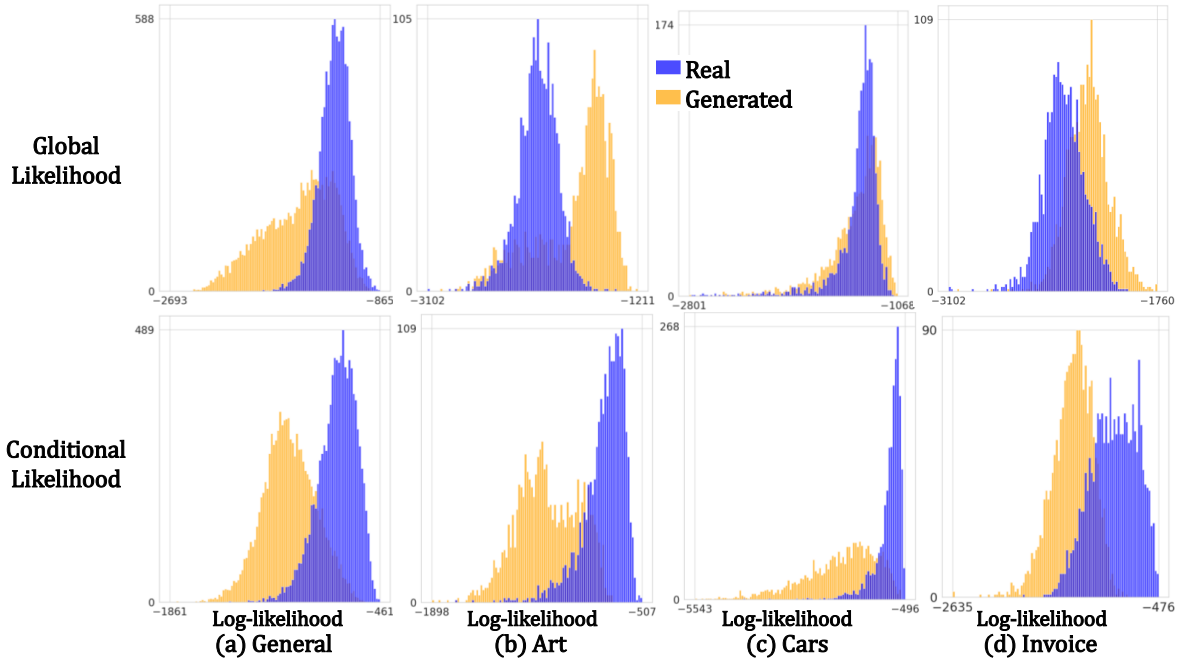


Figure 15. **Global and conditional likelihood comparison.** Per-domain histograms of log-likelihoods for real and generated images. In the general image domain, the global likelihood yields only modest separation and far more overlap than the conditional likelihood; in other domains it shows little separation (*cars*) and even "flipped classification" (*art*, *invoice*).

G. Efficiency Analysis Details

Running time. As presented in Tab. 5, our method operates efficiently, achieving inference speeds comparable to ZED, while other methods, particularly AEROBLADE, exhibit significantly higher latency. We use the official implementations of AEROBLADE and Manifold Bias; however, batch inference is not supported for AEROBLADE, whereas it is for Manifold Bias. On an NVIDIA A100 GPU, the maximal batch size for Manifold Bias is four images. For a fair comparison, we measure inference time of four images across all methods and report the average processing time per image. Notably, our method's runtime can be further reduced by using a global whitening matrix instead of computing one per image. However, this would impact performance (see ablation study in Sec. E.1, Fig. 7). All models are pre-loaded onto the device before time measurement, while image loading is included in the evaluation time.

Generative Model	AEROBLADE [33]	RIGID [16]	ZED [11]	Manifold Bias [7]	CLIDE (ours)
Seconds per image	4.66	0.59	**0.26**	1.66	**0.26**

Table 5. **Running time evaluation.**

Memory consumption. Our approach requires less RAM and GPU memory than most alternatives. Our method primarily relies on the CLIP model, which requires 900MB of disk storage and 2GB of GPU memory during inference. The embeddings of the representative dataset, consuming only 3MB per 1K images, are negligible. In contrast, Manifold Bias exhibits high memory usage, exceeding 30 GB of GPU memory (model memory consumption) during inference. AEROBLADE [33] and RIGID [16] require less GPU memory but still more than our method (using Stable diffusion [34] auto-encoder and DINO model [9] receptively). ZED utilizes the SReC compression model [8], which has low disk storage requirements but higher GPU memory usage for high-resolution images.

Overall, our method is both fast and memory-efficient, making it a lightweight and effective alternative to other zero-shot detection methods.

H. ZED Additional Criteria

In Tab. 2 and Tab. 3 we evaluate all four criteria proposed by ZED [11] - $D_0, |D_0|, \Delta_{01}, |\Delta_{01}|$. In the paper, we report results for D_0, as it achieves the highest AUC in the general image scenario (Tab. 2). Additionally, in the several cases where ZED outperforms our method, D_0 consistently yields the highest results among all ZED criteria. We extend our analysis to all criteria, confirming that none surpass our method in any scenario or for any generative model, except in the cases where D_0 does, and therefore are reported in the paper.

	Global likelihood (W-CLIP)	Conditional likelihood (CLIDE)
Probability modeled	Single, unconditional $p(x)$	Family of conditioned densities $p(x \mid \mathcal{D})$
Conceptual scope	One universal surrogate over "the world"	A *family* of surrogates specialized to target domains
Conditioning variable	None	Domain $\mathcal{D}$ (specified via unlabeled real examples)
Support / geometry	One ambient (globally whitened) space	Domain-specific *subspace* (dominant components)
Directional information	Largely discarded by global norm	Preserved via domain-aligned principal axes
Gaussianity assumption	Approx. i.i.d. Gaussian in the global whitened space	Approx. Gaussian *within* the domain's dominant subspace
Behavior under domain mixture	Can conflate distinct domains at similar radii	Separates domains via subspace alignment
Sufficient statistics	Global mean / covariance	Domain-specific mean / covariance
Limiting case	—	Reduces to global model when $\mathcal{D}$ is broad and $m = d$

Table 6. **Conceptual comparison of global vs. conditional likelihood.** The conditional formulation models $p(x \mid \mathcal{D})$ in a domain-aligned subspace, preserving directional structure that a global norm discards, and thereby improving robustness under domain shift.

I. Domain-Specific Zero-Shot Generated Image Detection

Definition. We introduce the task of *Domain-Specific Zero-Shot Generated Image Detection* (DSZD). In this setting, a detector is provided with access to *real* images from the target domain but receives no generated samples and no task-specific supervision. The goal is to decide, for each test sample in the target domain, whether it is real or generated. Thus, the detector adapts to the distribution of real images in the domain while remaining completely unexposed to generated content.

Setting. Formally, let $\mathcal{D}_{\text{real}}^{\text{target}}$ denote a collection of real images sampled from the target domain distribution $P_{\text{real}}^{\text{target}}$. The detector may access $\mathcal{D}_{\text{real}}^{\text{target}}$ for the purpose of adaptation or calibration, but *no training or fine-tuning of model parameters is allowed*. At test time, the detector receives a sample x drawn either from $P_{\text{real}}^{\text{target}}$ or from some unknown generative distribution $P_{\text{gen}}^{\text{target}}$, with the task of predicting whether x is real or generated. Crucially, $P_{\text{gen}}^{\text{target}}$ is not observed at any stage, reflecting the evolving and potentially adversarial nature of generative models.

Relation to existing settings.

- **Zero-shot detection.** In the standard zero-shot setting [7, 11, 33], the detector has no access to any domain-specific data, neither real nor generated. DSZS extends this by allowing domain adaptation, while maintaining the zero-shot requirement with respect to generated samples.
- **Unsupervised domain adaptation (UDA).** In general vision tasks such as classification, recognition, or out-of-distribution (OOD) detection, UDA assumes access to labeled source-domain data together with unlabeled target-domain data, enabling adaptation to the target distribution without target labels. By contrast, DSZS departs from this paradigm: it prohibits any exposure to generated data and restricts adaptation to the statistics of real images alone. This distinction is important, since UDA has not been formulated in the context of generated image detection, where the diversity and continual evolution of generative models makes the assumption of target-domain generated samples unrealistic.

Justification. This task is motivated by the practical observation that, in many application domains, abundant real images are available, while generated data are diverse, evolving, and difficult to curate exhaustively. Training detectors against all possible generation models is infeasible; yet, real-domain images are easy to collect and provide valuable information about the distribution to be protected. By learning the structure of the real domain, a detector can identify deviations indicative of synthetic content, without relying on prior exposure to such content.

On the naming. The term *Domain-Specific Zero-Shot Generated Image Detection* is chosen to emphasize two key aspects:

(i) **Domain-Specific:** the detector leverages real data from the specific target domain, differentiating it from domain-agnostic zero-shot detection.

(ii) **Zero-Shot:** no generated examples or task-specific supervision are seen, preserving the zero-shot spirit relative to synthetic data.

This name concisely captures the hybrid nature of the task, situating it between existing detection paradigms while clearly distinguishing its assumptions.

J. Global vs. Conditional Likelihood

This appendix complements Sec. 3 by contrasting the *global* likelihood surrogate of W-CLIP with our *domain-conditional* likelihood. In Tab. 6 we enumerate the key differences between the two formulations. Then, in Sec. J.1 we report zero-shot detection results for the global model and for CLIDE, showing a substantial degradation under the global surrogate.

J.1. Global likelihood as a detector across domains

For completeness, we evaluate the *global* likelihood surrogate (W-CLIP style) as a zero-shot detector across all four settings. The global whitening statistics are fit on the MS-COCO validation set, following [5]. As summarized in Tab. 7, the global surrogate shows a *large* decrease even on general images (AUC 0.81 vs. 0.92 for CLIDE; gap to perfect 1.0 is 0.19 vs. 0.08, i.e., $> 2\times$ larger error). Moreover, the global likelihood model degrades sharply on *Artistic*, *Cars*, and *Invoice*, with substantial drops in AUC. This pattern mirrors the declines observed for other zero-shot detectors under domain shift (Tabs. 1–2) and underscores the need to condition on domain-specific statistics. Importantly, the flipped classification (AUC under 0.5), observed in other zero-shot methods, but not in CLIDE, is present in the global likelihood model, making it a poor detector for domain specific cases. The experiment is performed on the same ablation data as Table 4 in the paper. We plot per-domain histograms of log-likelihoods for real vs. generated images in Fig. 15. In the general image domain, the global likelihood provides only moderate separation and shows substantially more overlap than the conditional likelihood. In *cars*, separation is minimal under the global score, and in *art* and *invoice* the ordering reverses - generated often receive higher scores than real (i.e. "flipped classification").

K. Medical image domain

Setup. We use the Chest X-Ray Pneumonia collection (5,865 images) [21], uniformly sample 5,000 real images, and compare against 5,000 images randomly drawn from a large ($\sim 10^5$) PGGAN-generated chest X-ray corpus [36]. See Fig. 16 for examples of images from both datasets. The remaining 865 real images are used as the representative set and the parameters k, m values are identical to all the experiments is Sec. 5 ($k = 500, m = 400$). Our method demonstrates extremely high performance - $AUC = 0.99$.

Method	General	Artistic	Cars	Invoice
Global ([5])	0.81	0.37	0.53	0.18
Conditional (CLIDE)	0.92	0.94	0.93	0.91

Table 7. Detection performance (AUC values) across domains for global (directly based on [5] vs. conditional likelihood (our method - CLIDE).

Real (a) PGGAN (b)

Figure 16. **Real and synthesized X-ray images.**

References

[1] Theodore W Anderson and Donald A Darling. A test of goodness of fit. *Journal of the American statistical association*, 49(268):765–769, 1954. 1

[2] A. Anonymous. Whitened clip as a likelihood surrogate of images and captions, 2025. In review, anonymous during the review process. Supplied as supplemental material wclip.pdf. 3

[3] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 993–1003, 2021. 10

[4] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 7, 10

[5] Roy Betser, Meir Yossef Levi, and Guy Gilboa. Whitened clip as a likelihood surrogate of images and captions. In *International Conference on Machine Learning*. PMLR, 2025. 15

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 8, 10

[7] Jonathan Brokman, Amit Giloni, Omer Hofman, Roman Vainshtein, Hisashi Kojima, and Guy Gilboa. Manifold induced biases for zero-shot and few-shot detection of generated images. In *International Conference on Learning Representations*, 2025. 13, 14

[8] Sheng Cao, Chao-Yuan Wu, , and Philipp Krähenbühl. Lossless image compression through super-resolution. *arXiv preprint arXiv:2004.02872*, 2020. 13

[9] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9650–9660, 2021. 13

[10] Qifeng Chen and Vladlen Koltun. Photographic image synthesis with cascaded refinement networks. In *Proceedings of the IEEE international conference on computer vision*, pages 1511–1520, 2017. 10

[11] Davide Cozzolino, Giovanni Poggi, Matthias Nießner, and Luisa Verdoliva. Zero-shot detection of ai-generated images. In *European Conference on Computer Vision*, pages 54–72. Springer, 2024. 13, 14

[12] Ralph D'agostino and Egon S Pearson. Tests for departure from normality. empirical results for the distributions of b^2 and $\sqrt{b}$. *Biometrika*, 60(3):613–622, 1973. 1

[13] Timo I Denk and Christian Reisswig. Bertgrid: Contextualized embedding for 2d document representation and understanding. *arXiv preprint arXiv:1909.04948*, 2019. 10

[14] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 10

[15] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10696–10706, 2022. 10

[16] Zhiyuan He, Pin-Yu Chen, and Tsung-Yi Ho. Rigid: A training-free and model-agnostic framework for robust ai-generated image detection. *arXiv preprint arXiv:2405.20112*, 2024. 13

[17] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 10

[18] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 10

[19] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8110–8119, 2020. 10

[20] Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021. 10

[21] Daniel S Kermany, Michael Goldbaum, Wenjia Cai, Carolina CS Valentim, Huiying Liang, Sally L Baxter, Alex McKeown, Ge Yang, Xiaokang Wu, Fangbing Yan, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. *cell*, 172(5):1122–1131, 2018. 15

[22] 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 *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 3

[23] Furqanil Taqwa (Linaqruf). Animagine xl 2.0: High-resolution anime image generation. Hugging Face, 2023. 10

[24] Midjourney. Midjourney: An independent research lab exploring new mediums of thought. https://www.midjourney.com/, 2024. Accessed: 2024-05-18. 7, 10

[25] MindSpore. Wukong: A Pre-trained Model for Chinese Text-to-Image Generation. https://xihe.mindspore.cn/modelzoo/wukong, 2024. Accessed: 2024-05-18. 10

[26] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 8, 10

[27] Utkarsh Ojha, Yuheng Li, and Yong Jae Lee. Towards universal fake image detectors that generalize across generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24480–24489, 2023. 9

[28] OpenAI. GPT-Image-1: A multimodal image-generation model. https://platform.openai.com/docs/models/gpt-image-1, 2025. Released Apr 2025. 10, 11

[29] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2337–2346, 2019. 10

[30] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7, 10, 11

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 9, 10, 11

[32] Anton Razzhigaev, Arseniy Shakhmatov, Anastasia Maltseva, Vladimir Arkhipkin, Igor Pavlov, Ilya Ryabov, Angelina Kuts, Alexander Panchenko, Andrey Kuznetsov, and Denis Dimitrov. Kandinsky: an improved text-to-image synthesis with image prior and latent diffusion. *arXiv preprint arXiv:2310.03502*, 2023. 7, 10, 11

[33] Jonas Ricker, Denis Lukovnikov, and Asja Fischer. Aeroblade: Training-free detection of latent diffusion images using autoencoder reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9130–9140, 2024. 13, 14

[34] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 8, 10, 13

[35] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *arXiv preprint arXiv:2111.02114*, 2021. 9

[36] Brad Segal. Synthetic pggan chest x-rays. Kaggle, 2020. 15

[37] Nikolai Smirnov. On the estimation of the discrepancy between empirical curves of distribution for two independent samples. *Bulletin Moscow University*, 2:3–16, 1939. 1

[38] Xinkuang Wang, Wenjing Li, and Zhongcheng Wu. Cardd: A new dataset for vision-based car damage detection. *IEEE Transactions on Intelligent Transportation Systems*, 24(7): 7202–7214, 2023. 7, 9

[39] wiF0n. invoiceXpert: A curated dataset of invoice images. https://huggingface.co/datasets/wiF0n/invoiceXpert, 2025. Accessed June 2025; license: CC-BY-NC-4.0. 11

[40] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 1192–1200, 2020. 10

[41] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 10