Further improving training-free AI-generated image detection

Alec Imhof [*1] **Aurélie Wasem** [*1]

Abstract

AI-generated images produced by modern diffusion models are increasingly realistic, making it difficult to distinguish them from real photographs and raising concerns about misinformation and trust in digital media. Recent work has shown that vision foundation models (VFMs) such as DINOv2 can be used as training-free detectors by exploiting their different sensitivity to image-space perturbations for real versus synthetic images. In this project, we reproduce the RIGID, Gaussian blur, Contrastive Blur, and MINDER detectors of Tsai et al. and extend them in several directions, including the use of a more recent backbone (DINOv3-L), the addition of the SID dataset, and a patch-wise top-k sensitivity formulation. We construct a benchmark of real/fake pairs from ADM, Collaborative Diffusion, and SID, and systematically compare perturbation-based detectors across these domains. Our results largely confirm the original findings on ADM and facial data, while highlighting limitations and robustness issues on more recent generators such as SID.

1. Introduction

AI-generated images have become increasingly realistic and difficult to distinguish from real photographs. This raises concerns about misinformation, political manipulation and the erosion of trust in digital media. Modern diffusion models such as ADM and Stable Diffusion can synthesise highly plausible images across a wide range of domains, from everyday objects to human faces.

Recent work proposes training-free detectors built on pretrained *Vision Foundation Models* (VFMs). Rather than learning a separate classifier, these methods treat the VFM as a feature extractor and exploit the robustness properties of its embeddings: real and synthetic images tend to react differently to perturbations in image space. Tsai et al. (2024) show that, for DINOv2, fake image embeddings are more sensitive to certain perturbations (e.g., Gaussian noise or blur) than those of real images. Their RIGID framework uses Gaussian noise as a perturbation and measures the embedding distance between an image and its noisy versions. They then introduce additional perturbations (Gaussian blur, contrast blur) and the MINDER (MINimum distance detEctoR) algorithm to bridge the performance gap between general and facial datasets. In this project, we reproduce and extend key experiments from Tsai et al. (2024) using our own implementation (Imhof & Wasem, 2025). We construct a dataset of real/fake image pairs drawn from several generative sources (ADM, CollabDiffusion, SID) and evaluate training-free detectors based on DINO-style VFMs. We first focus on DINOv2-L/14 to replicate the main findings of RIGID, Gaussian blur, Contrastive Blur, and MINDER. Then we investigate whether the more recent DINOv3-L backbone can further improve detection performance across heterogeneous generators.

More concretely, our contributions are as follows:

- We construct a dataset of real/fake pairs from several generative sources (ADM, CollabDiffusion, SID), with balanced real and fake images for each subset.
- We implement the baseline detector (RIGID) by extracting DINO-based embeddings and computing robustness-based distances under Gaussian noise perturbations.
- We reproduce and extend the perturbation experiments with Gaussian blur and Contrastive Blur, and implement the MINDER approach that combines noise and blur via a minimum-distance rule.
- As extensions, we:
  - Add images from the more recent SID dataset to test robustness to more modern generative pipelines beyond the original ADM/CollabDiff setup.
  - Evaluate more powerful foundation models such as DINOv3-L as drop-in replacements in the RIGID/MINDER framework.

[*] Equal contribution [1] Master of Science in Computer Science, University of Neuchâtel, Neuchâtel, Switzerland. Correspondence to: Alec Imhof <Alec.Imhof@unine.ch>, Aurélie Wasem <Aurelie.Wasem@unine.ch>.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025. Copyright 2025 by the author(s).

  - Introduce a patch-wise top-k sensitivity extension that computes RIGID/CB/MINDER scores from the most unstable ViT patches, in order to probe spatially local artifacts beyond the global CLS embedding.

2. Dataset and Data Construction

In order to study training-free detection across different types of synthetic images, we construct a benchmark of real/fake pairs from three sources: ADM (general objects), Collaborative Diffusion (faces), and SID (additional synthetic images). For each source, we construct a pool of real images and a pool of fake images, then sample balanced real/fake pairs $(x_{\text{real}}, x_{\text{fake}})$ within each dataset.

2.1. Source Datasets

Following the spirit of Tsai et al. (2024), we consider both *general* and *facial* AI-generated images, and we add a third dataset to test robustness beyond the original setting:

- **ADM (general objects).** We start from the ADM ImageNet-style validation data hosted on Google Drive, which covers a wide range of everyday scenes and objects.
- **Collaborative Diffusion (CollabDiff, faces).** Real facial images and synthetic portraits generated from the official Collaborative Diffusion code and checkpoints (Huang et al., 2023b;a).
- **SID (Synthetic Image Dataset).** A more recent dataset of authentic, fully synthetic, and tampered images, used as a complementary benchmark to test robustness beyond the original ADM/CollabDiff setting.

For all three sources, images are resized to a common format before pairing. Each subset is balanced in terms of the number of real and fake images, and we report results per dataset (ADM, CollabDiff, SID) and the global aggregate.

2.2. Reproducibility

All data-processing steps are scripted and version-controlled in our GitHub repository (Imhof & Wasem, 2025).
This includes:

- download of the original datasets or archives,
- construction of the real/fake pools and the CSV files of image pairs.

By running the shell and Python scripts provided with the documented arguments and paths, it is possible to reproduce the dataset preparation used in our experiments fully.

3. Methods

3.1. Vision Foundation Models

We use DINOv2 and DINOv3 models as our vision encoders:

- **DINOv2-L/14**: the default backbone used by RIGID and by Tsai et al. (2024), providing robust self-supervised features.
- **DINOv3-L**: a more recent and powerful model that we evaluate as a drop-in replacement in the RIGID/MINDER framework.

In our implementation, the model can be selected at the command line via the arguments described in `available_cli_arguments.md` and in the project README. We rely on the official DINOv2 and DINOv3 implementations and documentation (Meta AI Research, 2023; 2024; Hugging Face, 2024a;b).

For each image, we extract the [CLS] token embedding from the final transformer layer and ℓ_2-normalise it before computing cosine similarities.

3.2. Perturbations

We implement the following perturbation operators in image space (pixel values in $[0, 1]$):

- **Gaussian noise** $\delta_{\text{noise}} \sim \mathcal{N}(0, \sigma^2_{\text{noise}})$ added to each pixel, followed by clipping to $[0, 1]$ and renormalization with ImageNet statistics. This yields noisy versions $x + \delta_{\text{noise}}$.
- **Gaussian blur operator** implemented by convolving a 3×3 Gaussian kernel in pixel space, parameterised by a standard deviation σ_{blur} in pixels. We denote the blurred image by $x_{\text{blur}} = \text{Blur}_{\sigma_{\text{blur}}}(x)$. This operator is used as a *building block* for the Contrastive Blur (CB) perturbation described below.

For experiments with **DINOv2**, we do *not* re-tune these hyperparameters: we directly adopt the values of σ_{noise} and σ_{blur} reported as most effective in the original paper (*Understanding and Improving Training-Free AI-Generated Image Detections with Vision Foundation Models*), in order to stay as close as possible to their setting and make our DINOv2 results directly comparable to theirs.

For **DINOv3**, since the original paper does not use this backbone, we perform a systematic grid search over noise and blur strengths. We sweep

$$\sigma_{\text{noise}} \in [0.004, 0.040] \text{ with step } 0.002$$

(19 values) and

$$\sigma_{\text{blur}} \in [0.30, 1.00] \text{ with step } 0.05$$

(15 values). For each noise level we compute the AUROC of the noise-only detector, and for each blur level we compute the AUROC of the Contrastive Blur detector. We then build **MINDER** scores for all 19×15 possible combinations $(\sigma_{\text{noise}}, \sigma_{\text{blur}})$ and evaluate each combination on five settings: global (all datasets combined), global without SID, ADM only, CollabDiff only, SID only. For each setting we rank all MINDER configurations by AUROC and retain the top–10 combinations (see Table 3).

3.3. Distance Computation and Detection Rules

Let f be the VFM encoder and $z = f(x)$ the ℓ_2-normalised embedding of an image x. All distances are cosine distances between normalised embeddings.

RIGID-style distance (Gaussian noise). We apply K i.i.d. samples of Gaussian noise in pixel space and compute

$$D_{\text{noise}}(x) = \frac{1}{K}\sum_{k=1}^{K} D\big(f(x), f(x + \delta_{\text{noise}}^{(k)})\big),$$

where D denotes the cosine distance.

Contrastive Blur (CB). Given the Gaussian blur operator $x_{\text{blur}} = \text{Blur}_{\sigma_{\text{blur}}}(x)$, we define the associated “blur increment” $\delta_{\text{blur}} = x_{\text{blur}} - x$. This allows us to construct both a blurred and a sharpened version of the image:

$$x_{\text{blur}} = x+\delta_{\text{blur}}, \qquad x_{\text{sharp}} = x-\delta_{\text{blur}} \approx \text{clamp}(2x-x_{\text{blur}}, 0, 1).$$

We then measure how different the representations of these two perturbed versions are:

$$D_{\text{CB}}(x) = D\big(f(x_{\text{blur}}), f(x_{\text{sharp}})\big).$$

This matches our implementation, where x_{sharp} is obtained by a contrastive sharpening step $x_{\text{sharp}} = \text{clamp}(2x - x_{\text{blur}}, 0, 1)$ before re-normalisation and embedding.

MINDER (Noise + Contrastive Blur). To combine noise and blur, we follow a MINDER-style formulation based on the *minimum* of the two perturbation distances:

$$D_{\min}(x) = \min\big(D_{\text{noise}}(x), D_{\text{CB}}(x)\big),$$

and use $D_{\min}(x)$ as the detection score. The intuition is that real images tend to be robust to at least one of the two perturbations (noise or blur/sharpen), so at least one of the distances stays small, whereas fake images are sensitive to both, resulting in larger values of $D_{\min}$.

Thresholding and AUROC. Rather than fixing a single threshold, we treat each distance (D_{noise}, D_{CB}, or $D_{\min}$) as a *score* $s(x)$ and evaluate performance using the area under the ROC curve (AUROC). This is consistent with the evaluation protocol of Tsai et al. (2024) and allows us to compare detectors without explicit calibration.

3.4. Patch-wise Top-k Sensitivity

All detectors described so far operate on *global*, image-level representations obtained from the [CLS] token of the VFM backbone. While this global descriptor is semantically rich and effective, it compresses all spatial information into a single vector, which may limit the detector’s ability to react to localised artifacts or region-specific instability in synthetic images.

To investigate whether spatial information can improve training-free detection, we implemented an alternative, *patch-wise* scoring mechanism on top of the same perturbations (Gaussian noise, Contrastive Blur, and their MINDER combination). Let

$$E(x) = (e_0(x), e_1(x), \ldots, e_N(x)) \in \mathbb{R}^{(N+1)\times d}$$

denote the final-layer ViT representation of an image x, where $e_0(x)$ is the CLS embedding and $e_1(x), \ldots, e_N(x)$ are the patch embeddings corresponding to the N spatial tokens. For a perturbation operator T in image space (noise or blur/sharpen), we obtain two sets of patch embeddings,

$$E(x), \qquad E(T(x)),$$

and compute a cosine distance for each patch $i = 1, \ldots, N$:

$$d_i(x, T) = 1 - \big\langle e_i(x),\, e_i(T(x))\big\rangle.$$

This yields a vector of patch-wise instability scores $d(x, T) = (d_1(x, T), \ldots, d_N(x, T))$ for each image x and perturbation T.

Rather than averaging over all patches (which would reintroduce a global smoothing effect similar to the CLS embedding), we sort the patch distances in descending order and retain only the k most unstable patches:

$$d_{(1)}(x, T) \geq d_{(2)}(x, T) \geq \cdots \geq d_{(N)}(x, T),$$

$$D_{\text{top-}k}(x, T) = \frac{1}{k}\sum_{i=1}^{k} d_{(i)}(x, T).$$

The scalar $D_{\text{top-}k}(x, T)$ is then used as the detection score for the corresponding perturbation. In practice we apply this scheme to both the noise perturbation (RIGID) and the Contrastive Blur perturbation, yielding patch-wise scores $D_{\text{noise}}^{\text{top-}k}(x)$ and $D_{\text{CB}}^{\text{top-}k}(x)$, respectively. The MINDER score is then defined as

$$D_{\min}^{\text{top-}k}(x) = \min\big(D_{\text{noise}}^{\text{top-}k}(x),\, D_{\text{CB}}^{\text{top-}k}(x)\big),$$

mirroring the CLS-based definition in the previous subsection.

In our implementation, each perturbation (noise, Contrastive Blur, and MINDER) can be evaluated in two modes: a *CLS* mode that reproduces the original formulation using only the global [CLS] embedding, and a *top-k patch* mode that uses the patch-wise distance $D_{\text{top-}k}(x, T)$ defined above. This allows us to directly compare global versus spatially-aware instability measures within the same codebase and experimental protocol. We hypothesised that focusing on the most unstable patches might yield a stronger or more interpretable signal than the global CLS embedding.

3.5. Implementation Details

Our main implementation lives in `training_free_detect.py` and is exposed via a CLI with arguments documented in `available_cli_arguments.md`. At a high level, the script:

1. Loads a configuration (paths to pairs CSV, model name, perturbation type, batch size, etc.).
2. Instantiates the selected VFM backbone (DINOv2 or DINOv3).
3. Iterates over the real/fake pairs, applies the specified perturbation(s), and computes embedding distances.
4. Logs results (scores and labels) to CSV files under `results_dinov2_large/` or `results_dinov3_large/`.
5. Optionally aggregates results across runs (global summary CSV) for plotting and analysis.

4. Experimental Setup

4.1. Settings

We evaluate:

- **Backbones:** DINOv2-L/14 and DINOv3-L.
- **Perturbations:** Gaussian noise (RIGID), Contrastive Blur (CB), and MINDER (min of noise and CB). We also implement the simple Gaussian blur distance D_{blur}, but all "Blur" figures and tables in this report use the Contrastive Blur variant D_{CB}.
- **Datasets:** subsets from ADM, CollabDiffusion, and SID, with real/fake pairs balanced.

For each setting, we run the detector on all images and compute AUROC per dataset and averaged across datasets.

4.2. Metrics

The main evaluation metric in all our experiments is the area under the ROC curve (AUROC), as in the original RIGID/MINDER paper. For each configuration and each perturbation type (noise, Contrastive Blur, MINDER), we compute:

- a **global AUROC** over all images, and
- **per-dataset AUROCs** for ADM, CollabDiff, and SID.

These values are obtained directly from the per-image scores produced by `training_free_detect.py` and reported in the tables of Section 5.

5. Results

5.1. DINOv2

For DINOv2-L with the hyperparameters suggested in the original paper ($\sigma_{\text{noise}} = 0.009$, $\sigma_{\text{blur}} = 0.55$, $K = 3$ noise samples), our results closely follow the trends reported by Tsai et al. (2024). On the ADM (general-domain) subset, the RIGID detector based on Gaussian noise achieves an AUROC of 0.90, while Contrastive Blur (CB) reaches 0.78 and MINDER 0.87. This confirms that Gaussian noise is the most effective perturbation on general images, whereas blur slightly degrades performance compared to noise, exactly as observed in the original analysis.

On the CollabDiff (facial) subset, the situation is reversed. Contrastive Blur (CB) clearly dominates with an AUROC of 0.91, while RIGID (noise) only reaches 0.70, and MINDER matches CB with an AUROC of 0.91. This is consistent with the finding of Tsai et al. (2024) that deepfake facial datasets often exhibit frequency artifacts that are strongly affected by low-pass filtering, making blur a particularly effective perturbation on faces. Aggregating ADM and CollabDiff, we obtain a global AUROC without SID of 0.80 for noise, 0.85 for blur, and 0.89 for MINDER. This shows that the minimum-distance combination successfully mitigates the noise–vs–blur bias across these two domains and yields the best average performance.

The SID subset behaves differently. Here, Gaussian noise remains reasonably effective with an AUROC of 0.71, but Contrastive Blur (CB) performs close to random guessing (AUROC 0.31), and MINDER is also degraded (AUROC 0.44) because it combines the two perturbations. Since SID is based on a more recent synthetic–authentic image dataset, this suggests that the perturbation-based detectors and hyperparameters tuned on ADM and CollabDiff do not directly transfer to newer data distributions.

Overall, however, our DINOv2-L results reproduce both the qualitative trends and the order of magnitude of the AUROCs reported in the original paper on general and facial domains.

Table 1. AUROC of DINOv2-L for three perturbation-based detectors: RIGID (Gaussian noise), Contrastive Blur (CB), and their MINDER combination, on each dataset and on the global aggregate with and without the SID subset.

DATASET	RIGID	CB	MINDER
GLOBAL	0.736	0.588	**0.679**
GLOBAL W/O SID	0.801	0.846	**0.892**
ADM	**0.901**	0.779	0.871
COLLABDIFF	0.701	**0.913**	0.912
SID	**0.710**	0.310	0.439

5.2. Patch-wise top-k sensitivity (DINOv2).

To better understand the behaviour of the patch-wise extension from Section 3.4, we sweep the number of retained patches k for noise on ADM and for Contrastive Blur on CollabDiff. The resulting AUROCs are shown in Figures 1 and 2. In both cases, AUROC increases almost monotonically with k: when only a few patches are used, the detector is clearly weaker, and performance gradually improves as more patches are averaged. This indicates that focusing exclusively on the most extreme patch responses is too noisy, and that aggregating a larger set of patches stabilises the score. However, the curves saturate around $k \approx 200$ and remain below the CLS-based baselines reported above (e.g. ≈ 0.81 vs. 0.90 on ADM for noise, and ≈ 0.83 vs. 0.91 on CollabDiff for blur). Overall, these experiments suggest that the global CLS embedding already captures a strong, implicitly aggregated instability signal, and our simple top-k patch aggregation does not provide a straightforward accuracy gain, even though it could offer a more spatially interpretable view of where perturbations have the largest effect.

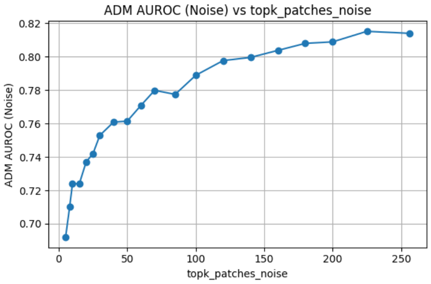

Figure 1. ADM AUROC (noise) as a function of the number of retained patches k in the patch-wise top-k scoring scheme.

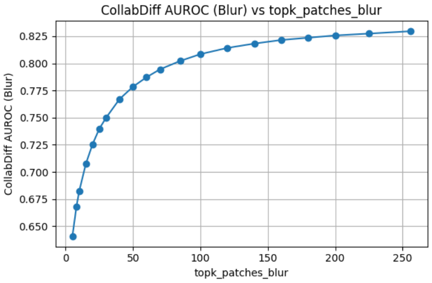

Figure 2. CollabDiff AUROC (Contrastive Blur) as a function of the number of retained patches k in the patch-wise top-k scoring scheme.

5.3. Extending to DINOv3

As described in Section 3.2, we perform a grid search over noise and blur strengths for DINOv3-L, then evaluate MINDER on all 19×15 combinations $(\sigma_{\text{noise}}, \sigma_{\text{blur}})$ and rank them by AUROC on different evaluation sets. From the top configurations we select three representative operating points, each optimising a different objective: (i) pure global AUROC, (ii) global AUROC without SID, and (iii) balanced performance across all three datasets. These three configurations are summarised in Table 2.

Objective 1: Maximising global AUROC. If we focus purely on the global AUROC (all datasets combined), the best configuration is

$$(\sigma_{\text{noise}} = 0.004,\ \sigma_{\text{blur}} = 1.0),$$

which achieves a global AUROC of 0.763, with ADM at 0.877, SID at 0.826, and CollabDiff at 0.456. This operating point is clearly *SID-biased*, it performs very well on SID and reasonably on ADM, but suffers from a low CollabDiff AUROC. It is attractive if SID-like generators are expected to dominate, but risky in scenarios with many facial deepfakes.

Objective 2: Maximising AUROC without SID. If we instead optimise the global AUROC excluding SID (ADM + CollabDiff only), the best configuration becomes

$$(\sigma_{\text{noise}} = 0.004,\ \sigma_{\text{blur}} = 0.40),$$

with a global AUROC without SID of 0.832. This setup yields strong performance on ADM (0.768) and CollabDiff (0.901), but very poor AUROC on SID (0.364). In other words, it essentially overfits to the ADM and CollabDiff distributions and fails to generalise to SID. This choice is reasonable only if we anticipate little or no SID-like data at test time and wish to avoid any influence from SID in the hyperparameter selection.

Objective 3: Balanced performance across datasets. Finally, we consider a more robust criterion: maximising the *worst* per-dataset AUROC,

$$\max_{(\sigma_{\text{noise}},\sigma_{\text{blur}})} \min\{\text{AUROC}_{\text{ADM}}, \text{AUROC}_{\text{CollabDiff}}, \text{AUROC}_{\text{SID}}\}.$$

Under this objective, the best configuration is

$$(\sigma_{\text{noise}} = 0.008,\ \sigma_{\text{blur}} = 0.85),$$

for which the minimum AUROC across ADM, CollabDiff, and SID is 0.629. Concretely, this setting achieves AUROC 0.887 on ADM, 0.707 on CollabDiff, and 0.629 on SID, with a global AUROC of 0.718. While this is not the absolute best on any individual dataset, it avoids catastrophic failures and offers the most "all-terrain" behaviour.

Comparison to DINOv2-L. When we focus only on ADM and CollabDiff, as shown in Figure 3, DINOv3-L does not outperform DINOv2-L. The best DINOv2-L MINDER configuration achieves a higher global AUROC without SID (≈ 0.89) than any of the DINOv3-L configurations we tested. This suggests that, on the two datasets closest to those studied by Tsai et al. (2024), DINOv2-L remains a very strong backbone for training-free detection.

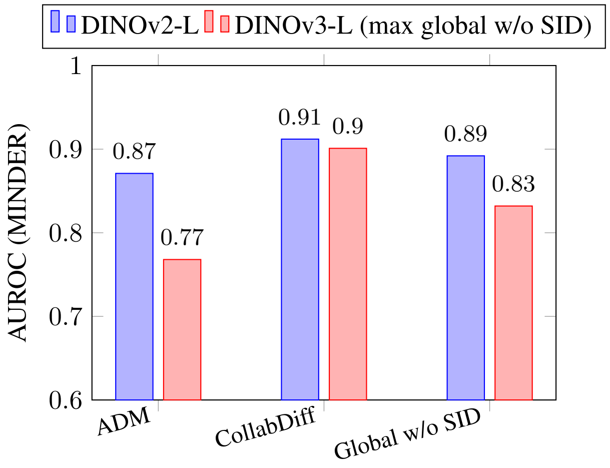

Figure 3. Comparison of MINDER AUROC for DINOv2-L and DINOv3-L under the "max global w/o SID" objective.

However, once SID is taken into account, the picture changes. As shown in Figure 4, the DINOv2-L MINDER detector is strongly penalised by SID (AUROC ≈ 0.44), which pulls its global AUROC down to ≈ 0.68. In contrast, the balanced DINOv3-L configuration, summarised in Table 2 and also illustrated in Figure 4, reaches a higher global AUROC (≈ 0.72) and maintains more homogeneous performance across ADM, CollabDiff, and SID. In this sense, DINOv3-L does not strictly dominate DINOv2-L on the original ADM/CollabDiff domains, but it can surpass DINOv2-L in scenarios where new generators such as SID are present. For this reason, we consider the balanced configuration ($\sigma = 0.008, \sigma_{\text{blur}} = 0.85$) to be the most reasonable hyperparameter choice for DINOv3-L in our setting.

It is plausible that the balanced DINOv3-L configuration works better on SID because DINOv3 is a more recent and higher-capacity vision foundation model than DINOv2-L. Compared to DINOv2, DINOv3 scales both the model and the pre-training data by roughly an order of magnitude (up to 7B parameters trained on ~1.7B images, versus ~1B parameters and 142M images for DINOv2). It is reported to produce higher-quality dense features that outperform previous self- and weakly-supervised foundation models across a wide range of downstream tasks without fine-tuning (Meta AI Research, 2023; Hugging Face, 2024a; Meta AI Research, 2024; Hugging Face, 2024b). Since SID is built from more recent generative pipelines, a stronger and more broadly trained backbone may transfer better to this "modern" synthetic domain. Our results are consistent with this picture: DINOv3-L does not dominate DINOv2-L on the original ADM/CollabDiff domains, but it provides more robust performance once SID is included.

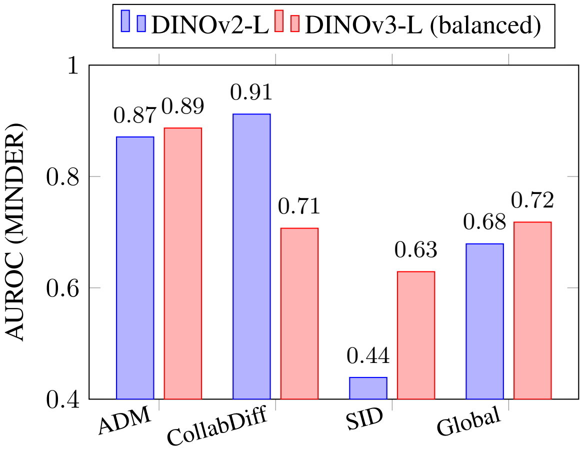

Figure 4. Comparison of MINDER AUROC for DINOv2-L and DINOv3-L under the balanced objective.

6. Conclusion

In this work, we revisited training-free detection of AI-generated images through the lens of robustness in vision foundation models. Building on the RIGID, Contrastive Blur, and MINDER framework of Tsai et al. (2024), we implemented a modular detection pipeline based on DINO-style backbones and evaluated it on a custom benchmark of real/fake pairs from ADM, Collaborative Diffusion, and the more recent SID dataset. Our DINOv2-L experiments largely reproduce the qualitative behaviour reported in the

Table 2. Selected DINOv3-L MINDER configurations illustrating different objectives: maximising global AUROC (Global), maximising AUROC without SID (Global w/o SID), and maximising the minimum per-dataset AUROC (balanced configuration).

$(\sigma_{\text{NOISE}}, \sigma_{\text{BLUR}})$	OBJECTIVE	GLOBAL	GLOBAL W/O SID	ADM	COLLABDIFF	SID
(0.004, 1.00)	MAX GLOBAL	**0.763**	0.696	0.877	0.456	0.826
(0.004, 0.40)	MAX GLOBAL W/O SID	0.615	**0.832**	0.768	0.901	0.364
(0.008, 0.85)	MAX MIN-DATASET (BALANCED)	**0.718**	0.798	0.887	0.707	0.629

original paper: Gaussian noise is the most effective perturbation on general images (ADM), Contrastive Blur dominates on facial data (CollabDiff), and MINDER successfully combines the two to achieve strong average performance across these domains. Taken together, our experiments provide a reproducible DINOv2-L baseline, a comparison between CLS and patch-wise top-k scoring, an extension to DINOv3-L that targets balanced performance across heterogeneous generators, and the addition of a dataset (SID) to test robustness to more modern generative pipelines.

At the same time, our results highlight important limitations of perturbation-based, training-free detectors when moving beyond the original datasets. On SID, Contrastive Blur performs close to random guessing and the MINDER combination is dragged down by the failure of blur, despite still working well on ADM and CollabDiff. This shows that hyperparameters and perturbation choices tuned on a given set of generators do not automatically transfer to newer, distributionally different synthetic data. The behaviour of our patch-wise top-k extension points in a similar direction. While it offers a more spatially interpretable view of instability, simple top-k aggregation of patch distances does not outperform the global CLS embedding in terms of AUROC. This suggests that the original global representation already encodes a strong robustness signal.

Our extension to DINOv3-L further illustrates the trade-offs involved in training-free detection. When we restrict attention to ADM and CollabDiff, DINOv3-L does not surpass DINOv2-L: the best DINOv2-L MINDER configuration still achieves a higher global AUROC without SID. However, once SID is included, a balanced DINOv3-L configuration that maximises the minimum per-dataset AUROC yields a higher global AUROC and a more homogeneous performance profile across ADM, CollabDiff, and SID. In other words, DINOv3-L does not strictly dominate DINOv2-L on the original domains, but it offers a more robust "all-terrain" operating point when faced with heterogeneous and evolving generators.

Several directions remain open. A natural next step is to explore additional backbones (e.g., I-JEPA or vision-language models) and more powerful spatially-aware scoring schemes beyond simple top-k patch aggregation. It would also be important to systematically study robustness under common post-processing operations (compression, resizing, mild editing) and to develop procedures for adapting perturbation hyperparameters to new generators without requiring labelled data.

Overall, our study supports the view that training-free detectors based on VFMs remain a strong baseline for AI-generated image detection, while also illustrating how careful choices of perturbations and backbones are needed to keep up with rapidly evolving generators.

References

Huang, Z., Chan, K. C. K., Jiang, Y., and Liu, Z. Collaborative diffusion. https://github.com/ziqihuangg/Collaborative-Diffusion, 2023a. GitHub repository.

Huang, Z., Chan, K. C. K., Jiang, Y., and Liu, Z. Collaborative diffusion for multi-modal face generation and editing. 2023b. URL https://arxiv.org/abs/2304.10530.

Hugging Face. Dinov2 — transformers documentation. https://huggingface.co/docs/transformers/model_doc/dinov2, 2024a.

Hugging Face. Dinov3 — transformers documentation. https://huggingface.co/docs/transformers/main/model_doc/dinov3, 2024b.

Imhof, A. and Wasem, A. Projectgenai: Training-free detection of AI-generated images with performance modeling and acceleration. https://github.com/AlecImhofUni/ProjectGenAI, 2025. GitHub repository.

Meta AI Research. Dinov2: PyTorch code and models for the dinov2 self-supervised learning method. https://github.com/facebookresearch/dinov2, 2023. GitHub repository.

Meta AI Research. Dinov3: Reference PyTorch implementation and models. https://github.com/facebookresearch/dinov3, 2024. GitHub repository.

Tsai, C.-T., Ko, C.-Y., Chung, I.-H., Wang, Y.-C. F., and Chen, P.-Y. Understanding and improving training-free AI-generated image detections with vision foundation models. *arXiv preprint arXiv:2411.19117*, 2024.

A. DINOv3 MINDER Grid Search

Table 3. Top MINDER configurations for DINOv3 obtained from the grid search over noise σ and blur σ_{blur}.

σ	σ_{blur}	Top-10 for	AUROC (global)	AUROC (global w/o SID)	AUROC (ADM)	AUROC (CollabDiff)	AUROC (SID)
0.004	**0.40**	**global_wo_sid**	**0.614618**	**0.831552**	**0.767692**	**0.901358**	**0.363885**
0.004	0.60	SID	0.727112	0.696341	0.877552	0.456246	0.758591
0.004	0.65	SID	0.740267	0.695043	0.878212	0.450308	0.786109
0.004	0.70	SID; global	0.749056	0.695713	0.878324	0.451804	0.802331
0.004	0.75	SID; global	0.756307	0.699901	0.878196	0.461816	0.811535
0.004	0.80	SID; global	0.760060	0.700706	0.878044	0.464208	0.817505
0.004	0.85	SID; global	0.761243	0.698886	0.877684	0.461144	0.821289
0.004	0.90	SID; global	0.761793	0.697295	0.877288	0.458416	0.823701
0.004	0.95	SID; global	0.762322	0.696594	0.877056	0.457376	0.825252
0.004	**1.00**	**SID; global**	**0.762776**	**0.696013**	**0.876952**	**0.456188**	**0.826463**
0.006	0.45	global_wo_sid	0.598573	0.829959	0.784706	0.888396	0.338150
0.006	0.75	ADM	0.729553	0.751371	0.886028	0.577352	0.704736
0.006	0.90	global	0.743365	0.741982	0.880616	0.564312	0.746232
0.006	0.95	global	0.744168	0.737412	0.879596	0.553956	0.753066
0.006	1.00	SID; global	0.745227	0.734414	0.878852	0.546756	0.758355
0.008	0.45	global_wo_sid	0.572707	0.820283	0.724952	0.906056	0.307794
0.008	0.70	ADM	0.667035	0.758895	0.886212	0.623408	0.557801
0.008	0.75	ADM	0.699955	0.789088	0.892004	0.670404	0.598394
0.008	0.80	ADM	0.714826	0.800931	0.890148	0.704220	0.618209
0.008	**0.85**	**ADM**	**0.718235**	**0.797926**	**0.887016**	**0.707128**	**0.629347**
0.009	0.45	global_wo_sid	0.566224	0.815238	0.705780	0.907660	0.302323
0.009	0.75	ADM	0.684290	0.794049	0.890352	0.686350	0.560075
0.009	0.80	ADM	0.699856	0.809030	0.889536	0.725052	0.577039
0.009	0.85	ADM	0.703520	0.808275	0.886232	0.735480	0.585357
0.010	0.75	ADM	0.670799	0.796826	0.890428	0.691788	0.531046
0.010	0.80	ADM; global_wo_sid	0.685463	0.812236	0.888144	0.731852	0.544745
0.010	0.85	global_wo_sid	0.689248	0.813268	0.884756	0.746692	0.550412
0.012	0.85	global_wo_sid	0.665974	0.814268	0.878824	0.751928	0.504388
0.012	0.90	global_wo_sid	0.666660	0.813554	0.875884	0.760460	0.504534
0.012	0.95	global_wo_sid	0.667958	0.813410	0.874932	0.767032	0.506012
0.012	1.00	global_wo_sid	0.669426	0.813825	0.874836	0.770568	0.508004
0.016	0.40	CollabDiff	0.539401	0.747369	0.527620	0.919508	0.306833
0.018	0.40	CollabDiff	0.538986	0.746064	0.525828	0.919508	0.306794
0.020	0.40	CollabDiff	0.538698	0.745133	0.524872	0.919508	0.306774
0.022	0.40	CollabDiff	0.538559	0.744713	0.524232	0.919508	0.306769
0.024	0.40	CollabDiff	0.538542	0.744612	0.524188	0.919508	0.306769
0.026	0.40	CollabDiff	0.538501	0.744487	0.524052	0.919508	0.306769
0.028	0.40	CollabDiff	0.538477	0.744406	0.523948	0.919508	0.306769
0.030	0.40	CollabDiff	0.538451	0.744303	0.523856	0.919508	0.306769
0.032	0.40	CollabDiff	0.538436	0.744246	0.523788	0.919508	0.306769
0.034	0.40	CollabDiff	0.538438	0.744252	0.523804	0.919508	0.306769