HSIC BOTTLENECK FOR CROSS-GENERATOR AND DOMAIN-INCREMENTAL SYNTHETIC IMAGE DETECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Synthetic image generators evolve rapidly, challenging detectors to generalize across current methods and adapt to new ones. We study domain-incremental synthetic image detection with a two-phase evaluation. Phase I trains on either diffusion- or GAN-based data and tests on the combined group to quantify bidirectional cross-generator transfer. Phase II sequentially introduces renders from 3D Gaussian Splatting (3DGS) head avatar pipelines, requiring adaptation while preserving earlier performance. We observe that CLIP-based detectors inherit text-image alignment semantics that are irrelevant to authenticity and hinder generalization. We introduce a Hilbert-Schmidt Independence Criterion (HSIC) bottleneck loss on intermediate CLIP ViT features, encouraging representations predictive of real versus synthetic while independent of generator identity and caption alignment. For domain-incremental learning, we propose HSIC-Guided Replay (HGR), which selects per-class exemplars via a hybrid score combining HSIC relevance with k-center coverage, yielding compact memories that mitigate forgetting. Empirically, the HSIC bottleneck improves transfer between diffusion and GAN families, and HGR sustains prior accuracy while adapting to 3DGS renders. These results underscore the value of information-theoretic feature shaping and principled replay for resilient detection under shifting generative regimes.

1 INTRODUCTION

The rapid progress of generative models has led to increasingly realistic synthetic images, raising urgent concerns about the spread of misleading digital content. The detection problem is inherently open-world: new diffusion architectures, GAN variants, and 3D Gaussian Splatting (3DGS) rendering pipelines will continue to emerge, remaining unseen during training. Among these, 3DGS has enabled photorealistic, real-time head avatars, expanding the scope of rendered imagery beyond traditional 2D synthesis. As illustrated in Figure 1, synthetic images produced by GANs, Deepfake, and 3DGS exhibit distinct artifacts and statistical patterns, motivating detectors that generalize beyond any single generative family.

A practical detector therefore requires two key capabilities: (i) robust generalization across diverse generation paradigms and (ii) continual adaptability to incorporate new synthetic sources without catastrophic forgetting. Existing systems that rely on the vision backbone of CLIP show promising cross-generator transfer between diffusion and GAN data. However, CLIP embeddings are primarily optimized for text-image alignment, embedding caption semantics that are irrelevant to authenticity and potentially detrimental when the task is purely image-based. As shown in Figure 2, the raw CLIP features cluster together according to object class rather than the real-synthetic boundary.

We tackle these challenges with two complementary components integrated into a single pipeline. First, we introduce a Hilbert-Schmidt Independence Criterion (HSIC) bottleneck on features extracted from the intermediate layers of the CLIP ViT. Specifically, we aggregate multi-layer representations, project them to a compact latent space, and optimize an HSIC objective that minimizes dependence on the input while maximizing dependence on the label. This regularization suppresses nuisance factors (including text-alignment signals) and produces generator-invariant yet discriminative representations, strengthening mutual generalization between diffusion and GAN models.

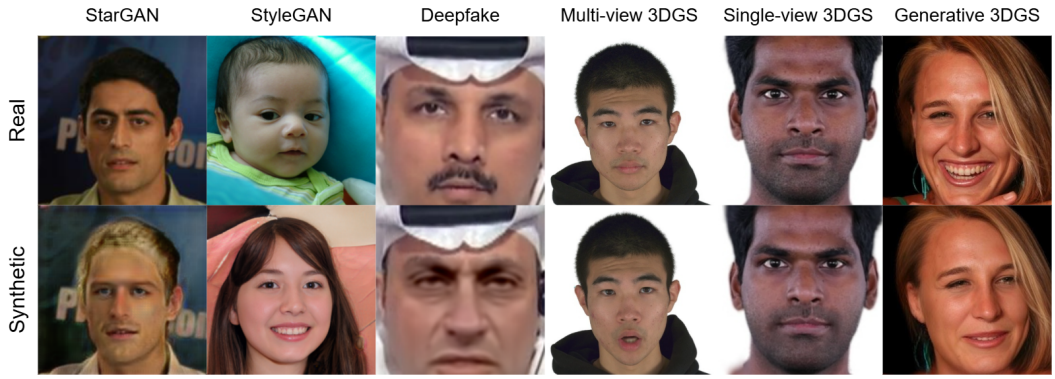

Figure 1: **Diversity of synthetic images across paradigms.** Columns show representative sources of synthetic faces: StarGAN, StyleGAN, Deepfake (face swap), multi-view 3D Gaussian Splatting (3DGS) head avatar, single-view 3DGS head avatar, and generative 3DGS head avatar. Top row: real images. Bottom row: corresponding synthetic examples. The breadth of GAN-based synthesis and rendered 3DGS avatars creates substantial distribution shifts, underscoring the need for detectors that both generalize to unseen sources and continually adapt as new synthetic sources emerge.

Second, for continual learning on rendered domains, we propose HSIC-Guided Replay (HGR). During adaptation, HGR constructs a compact exemplar memory per class by ranking candidates with a weighted score that combines HSIC relevance (information centrality) and k-center coverage (spatial spread). These exemplars are replayed alongside new data to mitigate forgetting and stabilize performance across evolving domains. To support this setting, we curate three 3DGS head avatar datasets covering multi-view reconstruction, single-view reconstruction, and a generative pipeline. Each dataset provides paired real and synthetic frames with identity-disjoint splits and standardized preprocessing. Our evaluation first trains detectors exclusively on diffusion or GAN images to measure cross-generator transfer, and then sequentially adapts them to the curated 3DGS domains under all adaptation orderings, while continuously monitoring prior-domain accuracy.

In summary, our contributions are threefold.

- **HSIC Bottleneck on CLIP Intermediates.** We impose an HSIC bottleneck on intermediate CLIP features to suppress text-alignment nuisances and amplify image-label dependence, substantially improving cross-generator generalization.
- **HSIC-Guided Replay (HGR) for Continual Adaptation.** We introduce an HSIC-driven exemplar selection and weighting scheme that delivers compact yet effective replay, enabling adaptation to 3DGS content while preserving prior accuracy.
- **3DGS Synthetic Image Benchmark.** We curate a 3DGS rendered image suite spanning multi-view reconstruction, single-view reconstruction, and a generative 3DGS pipeline, offering a benchmark to advance research on synthetic image detection.

2 Related Work

2.1 Generalized Synthetic Image Detection and 3DGS Head Avatars

Wang et al. (2020) first explicitly posed the problem of generalization to unseen generators in synthetic image detection, showing that with careful pre/post-processing and augmentation, a classifier trained on one GAN (e.g., ProGAN) can transfer to other generators, indicating shared synthesis artifacts. LGrad (Tan et al., 2023) moved toward generator-agnostic cues by operating in gradient space, yielding strong cross-generator generalization for GAN fakes.

A complementary trajectory leverages large vision-language backbones, and we group these CLIP-based methods together. UniFD (Ojha et al., 2023) demonstrated that operating directly in frozen CLIP feature space—using a nearest-neighbor or a linear probe—provides markedly improved transfer to unseen families, including mutual transfer between diffusion and GAN models. Building on

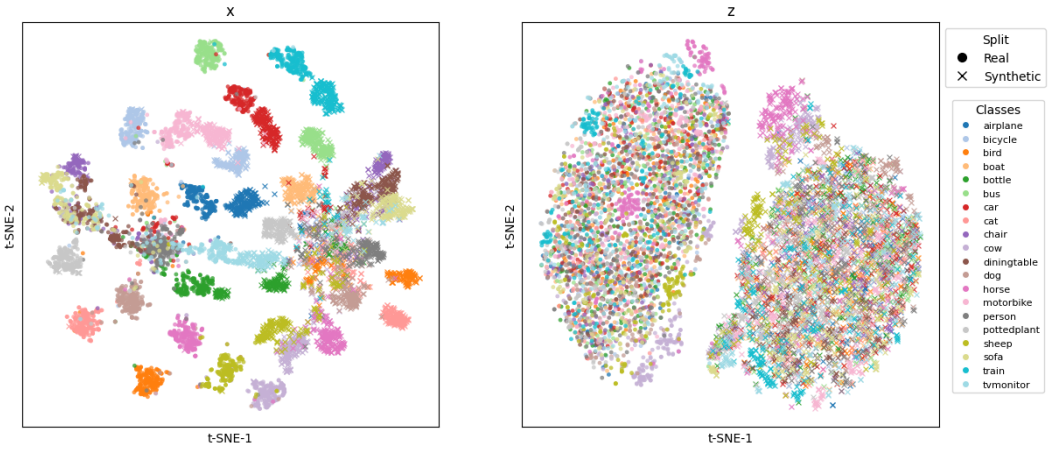

Figure 2: **HSIC bottleneck transforms CLIP semantic clusters into real/synthetic separability.** t-SNE visualization of features before (x, left) and after (z, right) applying the HSIC bottleneck. Points are colored by semantic class, with markers denoting Real ($\bullet$) and Synthetic ($\times$). Pretrained CLIP features x mainly cluster by object category, intermixing real and synthetic samples within each class. By contrast, the HSIC bottleneck suppresses nuisance semantics and reshapes the representation to align with the labels, causing real images to cluster together and synthetic images to cluster together across categories, thereby producing a clearer decision boundary for detection.

this idea, RINE (Koutlis & Papadopoulos, 2024) extracts intermediate encoder-block representations from CLIP rather than only the final layer, capturing richer structural and semantic cues that enhance out-of-distribution generalization. Most recently, VIB-Net (Zhang et al., 2025) couples a pre-trained CLIP backbone with a variational information bottleneck to suppress task-irrelevant factors while retaining discriminative evidence, pushing universal detection performance across generator families. Orthogonal to CLIP-based approaches, NPR (Tan et al., 2024) rethinks source-invariant artifacts by analyzing up-sampling operations common to GANs and diffusion pipelines, proposing neighboring pixel relationships as a simple, local pixel-dependency descriptor that generalizes across a wide range of generators.

Beyond diffusion and GAN imagery, 3D Gaussian Splatting (3DGS) enables photorealistic, real-time rendering with explicit point-based primitives and has rapidly become a foundation for head avatar synthesis, introducing rendered-fake sources that differ from conventional image synthesis. We highlight three representative families that define our rendered-fake domains: *(i) Multi-view 3DGS head avatar*—Gaussian Head Avatar (Xu et al., 2024) proposes dynamic, controllable 3D Gaussians for ultra high-fidelity head modeling, pairing a learned deformation field with an explicit 3DGS representation and multi-view supervision; *(ii) Single-view 3DGS head avatar*—SplattingAvatar (Shao et al., 2024) embeds Gaussians on a deformable mesh to disentangle motion and appearance, supporting real-time rendering from monocular training signals; and *(iii) Generative 3DGS head avatar*—GAGAvatar (Chu & Harada, 2024) predicts 3DGS parameters from a single image for one-shot, animatable, and generalizable avatars. These three categories define the rendered-fake domains in our continual learning protocol, complementing diffusion and GAN synthesis, and support systematic cross-generator evaluation. Despite the progress of generalized detection on diffusion and GAN, detectors that transfer well across these paradigms frequently falter on 3DGS-rendered fakes, motivating methods that couple stronger generalization with principled continual adaptation.

2.2 Continual Learning

We study continual learning in the domain-incremental regime: the label space remains fixed, while the input distribution shifts across domains—first among diffusion or GAN generators, and later to 3DGS categories. Continual methods are commonly grouped into three families: *regularization-based*, *architecture-based*, and *rehearsal-based*. Regularization-based approaches (Kirkpatrick et al., 2017; Zenke et al., 2017; Li & Hoiem, 2018; Aljundi et al., 2018) constrain parameter updates

to protect knowledge from earlier domains. Architecture-based methods (Rusu et al., 2016; Mallya & Lazebnik, 2018; Yan et al., 2021) expand capacity or allocate disjoint subnetworks to reduce interference. Rehearsal-based methods maintain a compact memory and interleave past exemplars with current data. *Class-mean herding* (iCaRL) (Rebuffi et al., 2017) selects exemplars near class centroids to ensure representativeness. *Class-balanced reservoir sampling* (CBRS) (Chrysakis & Pourkamali-Anaraki, 2020) adapts classical reservoir sampling to preserve label balance in nonstationary streams. Coverage-oriented selection via the greedy k-*center* (farthest-first) heuristic (Gonzalez, 1985; Sener & Savarese, 2018) spreads exemplars across the feature space to improve diversity and reduce redundancy. We build on this rehearsal line for the domain-incremental case and introduce HSIC-Guided Replay, which scores and selects exemplars to preserve prior domain coverage while adapting to new domains.

2.3 Hilbert-Schmidt Independence Criterion (HSIC)

HSIC (Gretton et al., 2005) measures statistical dependence between random variables via reproducing kernel Hilbert spaces (RKHS). Let $\mathbf{a} \in \mathbb{A}$ and $\mathbf{b} \in \mathbb{B}$ be random variables with RKHSs $(\mathbb{F}, k)$ and $(\mathbb{G}, \ell)$ induced by feature maps $\phi : \mathbb{A} \to \mathbb{F}$ and $\psi : \mathbb{B} \to \mathbb{G}$. Denote mean embeddings $\boldsymbol{\mu}_{\mathbf{a}} = \mathbb{E}[\phi(\mathbf{a})]$ and $\boldsymbol{\mu}_{\mathbf{b}} = \mathbb{E}[\psi(\mathbf{b})]$. The cross-covariance operator $\mathcal{C}_{\mathbf{ab}} : \mathbb{G} \to \mathbb{F}$ is

$$\mathcal{C}_{\mathbf{ab}} \;=\; \mathbb{E}\big[(\phi(\mathbf{a}) - \boldsymbol{\mu}_{\mathbf{a}}) \otimes (\psi(\mathbf{b}) - \boldsymbol{\mu}_{\mathbf{b}})\big], \tag{1}$$

and the population HSIC is the squared Hilbert-Schmidt norm of this operator:

$$\mathrm{HSIC}(P_{\mathbf{ab}}, \mathbb{F}, \mathbb{G}) \;=\; \big\|\mathcal{C}_{\mathbf{ab}}\big\|_{\mathrm{HS}}^2. \tag{2}$$

Kernel expectation form (population). Let $(\mathbf{a}', \mathbf{b}')$ be an independent copy of $(\mathbf{a}, \mathbf{b})$. Expanding equation 2 yields the following equivalent expression in terms of kernels k and ℓ:

$$\begin{aligned} \mathrm{HSIC}(\mathbf{a}, \mathbf{b}) \;&= \mathbb{E}_{\mathbf{aa'bb'}}\big[k(\mathbf{a}, \mathbf{a}')\,\ell(\mathbf{b}, \mathbf{b}')\big] \;+\; \mathbb{E}_{\mathbf{aa'}}\big[k(\mathbf{a}, \mathbf{a}')\big]\,\mathbb{E}_{\mathbf{bb'}}\big[\ell(\mathbf{b}, \mathbf{b}')\big] \\ &\quad -\; 2\,\mathbb{E}_{\mathbf{ab}}\Big[\,\mathbb{E}_{\mathbf{a}'} k(\mathbf{a}, \mathbf{a}')\;\mathbb{E}_{\mathbf{b}'}\ell(\mathbf{b}, \mathbf{b}')\,\Big], \end{aligned} \tag{3}$$

which is zero if and only if $\mathbf{a}$ and $\mathbf{b}$ are independent under characteristic kernels.

Empirical estimator. Given n i.i.d. samples $\{(a_i, b_i)\}_{i=1}^n$ from $P_{\mathbf{ab}}$, define Gram matrices $\boldsymbol{K}, \boldsymbol{L} \in \mathbb{R}^{n \times n}$ with $\boldsymbol{K}_{ij} = k(a_i, a_j)$ and $\boldsymbol{L}_{ij} = \ell(b_i, b_j)$, and the centering matrix $\boldsymbol{H} = \boldsymbol{I}_n - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. The commonly used (biased) V-statistic estimator is

$$\widehat{\mathrm{HSIC}}(\mathbf{a}, \mathbf{b}) \;=\; \frac{1}{(n-1)^2}\,\mathrm{tr}\big(\boldsymbol{K}\boldsymbol{H}\boldsymbol{L}\boldsymbol{H}\big) \;=\; \frac{1}{(n-1)^2}\,\mathrm{tr}\big(\bar{\boldsymbol{K}}\bar{\boldsymbol{L}}\big), \tag{4}$$

which provides an efficient empirical estimate without requiring density models. In practice, Gaussian RBF kernels are often adopted for k and ℓ, and the bandwidth can be set by the median heuristic. The centered version $\bar{\boldsymbol{K}} = \boldsymbol{K}\boldsymbol{H}$ and $\bar{\boldsymbol{L}} = \boldsymbol{L}\boldsymbol{H}$ removes mean components in RKHS so that equation 4 matches the population definition equation 2 through equation 3.

3 Method

Our detector builds on CLIP ViT features, which, like most pretrained extractors, are not optimized for synthetic image detection and often entangle authenticity with generator identity and text-image semantics. To address this, we introduce an HSIC bottleneck that refines representations for real-versus-synthetic discrimination while suppressing spurious dependencies. For domain-incremental learning, we propose HSIC-Guided Replay (HGR), which selects compact, representative exemplars to balance adaptation to new generators with retention of prior knowledge.

3.1 HSIC Bottleneck

Let the model be $h_\theta = g_{\theta_g} \circ f_{\theta_f}$, where f_{θ_f} is an encoder and g_{θ_g} a classifier. In DualHSIC (Wang et al., 2023), the encoder is a ResNet with L intermediate layers. Given an input x, its label y, and the feature representation at layer j, Z_j ($j = 1, \ldots, L$), the layer-wise HSIC objective is defined as

$$\mathcal{L}_{\mathrm{HB}}(\theta_f) \;=\; \lambda_x \sum\nolimits_{j=1}^{L} \widehat{\mathrm{HSIC}}\big(x, Z_j\big) \;-\; \lambda_y \sum\nolimits_{j=1}^{L} \widehat{\mathrm{HSIC}}\big(y, Z_j\big), \tag{5}$$

where λ_x controls compression of information from the input x and λ_y encourages dependence on its associated label y. Here $y \in \{0, 1\}$ indicates whether x is synthetic (1) or authentic (0).

Unlike DualHSIC, which applies HSIC at every intermediate layer, our approach leverages CLIP ViT as the feature extractor. We form the input x by concatenating features from its 24 intermediate layers and the final layer. The encoder f_{θ_f} then compresses this representation into a compact feature $z = f_{\theta_f}(x)$. Specializing equation 5 to this setting yields

$$\mathcal{L}_{\text{HSIC-Bottleneck}}(\theta_f, \theta_g) \;=\; \lambda_x \,\widehat{\text{HSIC}}(x, z) \;-\; \lambda_y \,\widehat{\text{HSIC}}(y, z). \tag{6}$$

3.2 Training Objective

The classifier g_{θ_g} outputs a logit $u_i = g_{\theta_g}(z_i)$ for each sample, and a probability $p_i = \sigma(u_i)$ via the sigmoid $\sigma(\cdot)$. For binary labels $y_i \in \{0, 1\}$, we use the binary cross-entropy with logits:

$$\mathcal{L}_{\text{BCE}}(\theta_f, \theta_g) \;=\; -\frac{1}{n}\sum_{i=1}^{n}\Big[\, y_i \log \sigma(u_i) \;+\; (1 - y_i)\log\big(1 - \sigma(u_i)\big)\,\Big]. \tag{7}$$

The final objective combines the bottleneck with the classifier:

$$\mathcal{L}_{\text{total}} \;=\; \mathcal{L}_{\text{HSIC-Bottleneck}} \;+\; \mathcal{L}_{\text{BCE}}. \tag{8}$$

3.3 HSIC-Guided Replay (HGR)

We couple HSIC relevance with a k-center coverage term to select exemplars for the rehearsal buffer. A single nonnegative weight $\lambda_{\text{kc}} \geq 0$ controls the relative strength of the k-center regularizer to the HSIC relevance: λ_{kc}=0 yields pure HSIC; larger values emphasize coverage. Exemplar selection is performed per class $c \in \{0, 1\}$, but for notational simplicity, we omit the index c in the following derivation. Let $\mathcal{X} = \{x_i\}_{i \in \mathcal{I}}$ be the candidate set with index set $\mathcal{I}$. At step t, let $S_{t-1} \subset \mathcal{I}$ denote the indices already selected (with $S_0 = \varnothing$), and define the active set as $A_t = \mathcal{I} \setminus S_{t-1}$.

For $x_i \in \mathcal{X}$, compute features $z_i = f_{\theta_f}(x_i)$. Form the Gaussian RBF Gram matrix $\boldsymbol{K}$ on $\{z_i\}_{i \in \mathcal{I}}$ and let $\bar{\boldsymbol{K}}$ be its centered version as in equation 4. The HSIC relevance for index $i \in \mathcal{I}$ is

$$r_i \;=\; \big\|\bar{\boldsymbol{K}}_{i,:}\big\|_2^2. \tag{9}$$

To promote coverage and reduce redundancy, we add a k-center term in feature space, following the coreset view of Sener & Savarese (2018). Define, for $i \in A_t$,

$$d_i(t) = \begin{cases} \|z_i - \mu\|_2^2, & t{=}1, \\ \min\limits_{j \in S_{t-1}} \|z_i - z_j\|_2^2, & t{\geq}2, \end{cases} \quad \text{with} \quad \mu = \tfrac{1}{|\mathcal{X}|}\sum_{j \in \mathcal{I}} z_j\,.$$

so that larger $d_i(t)$ favors points farther from the already-selected set.

Selection rule. HGR selects exemplars by minimizing the λ_{kc}-regularized score

$$s_i(t) = \big(1 - \mathcal{N}(r_i)\big) \;+\; \lambda_{\text{kc}}\,\big(1 - \mathcal{N}(d_i(t))\big), \quad i \in A_t\,. \tag{10}$$

$\mathcal{N}$ denotes a normalization operation. Choose $i_t^\star = \operatorname{argmin}\limits_{i \in A_t} s_i(t)$ and update $S_t = S_{t-1} \cup \{i_t^\star\}$ until $|S_t| = m_c$, where m_c is the number of exemplars assigned to class c. We repeat this procedure for $c \in \{0, 1\}$ and then take the union across classes and domains to form the replay buffer. Intuitively, HGR prefers items that are both HSIC-central (large r_i) and offer coverage (large $d_i(t)$).

3.4 Implementation details

Models are implemented in PyTorch and trained on a single NVIDIA GPU with SGD (learning rate 10^{-4}). The HSIC bottleneck uses a 64-D projection and Gaussian RBF kernels with bandwidth set by the median heuristic. We set $\lambda_x = 900$ and $\lambda_y = 700$ when training on SDV1.4 and $\lambda_x = 500$ and $\lambda_y = 600$ when training on ProGAN. We pick m_c exemplars per class c with $m_0 + m_1 = \lceil \texttt{keep\_frac} \cdot N \rceil$, where N is the total number of training samples and we set `keep_frac`=0.01. We first enforce class balance and then assign any remainder to the larger class.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Task and protocol. We study cross-generator synthetic-image detection in two phases under the following protocol. (1) *Cross-generator generalization*: train a detector on one paradigm (diffusion or GAN) and evaluate on the union of diffusion and GAN targets to measure generalization ability. (2) *Domain-incremental learning*: starting from the model in phase (1), continue training as rendered-fakes from 3DGS head avatar pipelines are introduced sequentially (GHA, SA, GAGAvatar). During this phase, we always monitor performance on previously seen diffusion and GAN test sets to quantify retention alongside adaptation.

Baselines. We benchmark against classical and recent detectors for cross-generator generalization, including CNNSpot (Wang et al., 2020), LGrad (Tan et al., 2023), UniFD (Ojha et al., 2023), NPR (Tan et al., 2024), RINE (Koutlis & Papadopoulos, 2024), and VIB-Net (Zhang et al., 2025). For domain-incremental learning, we compare HSIC-Guided Replay with iCaRL (Rebuffi et al., 2017) and CBRS (Chrysakis & Pourkamali-Anaraki, 2020). All rehearsal-based methods share the same per-class memory budget, and all other training hyperparameters are identical across methods.

Datasets. For *cross-generator generalization*, we use **GenImage**, which contains diffusion-generated images from Stable Diffusion v1.4/v1.5 (Rombach et al., 2022), ADM (Dhariwal & Nichol, 2021), GLIDE (Nichol et al., 2022), Midjourney (Midjourney, 2022), Wukong (Wukong, 2022), and VQDM (Gu et al., 2022). The GAN sources are taken from the collection of Wang et al. (2020), including ProGAN (Karras et al., 2018), CycleGAN (Zhu et al., 2017), BigGAN (Brock et al., 2019), StyleGAN (Karras et al., 2019), StarGAN (Choi et al., 2018), GauGAN (Park et al., 2019), as well as Deepfake (Rossler et al., 2019) and SAN (Dai et al., 2019).

For *domain-incremental learning*, we curate a three-part 3DGS benchmark: (i) Gaussian Head Avatar (Xu et al., 2024) (GHA) trained on NeRSemble (Kirschstein et al., 2023) with identity-disjoint, balanced splits (train: 45,772 real / 45,772 synthetic; val: 9,480 / 9,480; test: 9,782 / 9,782); (ii) SplattingAvatar (Shao et al., 2024) (SA) trained on subjects from NeRFace (Gafni et al., 2021), NHA (Grassal et al., 2022), and IM Avatar (Zheng et al., 2022) (train: 20,322 real / 20,094 synthetic; val: 4,007 / 4,036; test: 5,631 / 5,622); and (iii) GAGAvatar (Chu & Harada, 2024), a one-shot generative 3DGS avatar using FFHQ (Karras et al., 2019) inputs with pose-driven reenactment (train: 55,963 real / 55,963 synthetic; val: 6,995 / 6,995; test: 6,996 / 6,996).

4.2 CROSS-GENERATOR GENERALIZATION

We study out-of-source transfer by training detectors on a single family (diffusion or GAN) and evaluating across the union of diffusion and GAN targets, plus two additional sets (Deepfake, SAN). This setting measures whether a detector trained on one paradigm can generalize to the other without exposure to its data.

Training on diffusion. Table 1 reports cross-generator generalization with diffusion-only training. Specifically, the detectors are trained on the SDV1.4 dataset. Using intermediate ViT features with an HSIC bottleneck yields the strongest overall mean, outperforming both strong baselines and our non-intermediate variant. Improvements are especially robust on GAN targets and remain consistent across several diffusion datasets, while AP is near-saturated for most rows, indicating stable ranking quality. Although the non-intermediate variant occasionally tops individual entries (e.g., StyleGAN, SAN), the intermediate configuration is overall more consistent and generator-invariant, preserving performance across diverse generators and manipulations.

Training on GAN. Table 2 summarizes cross-generator generalization under GAN-only training. Specifically, the detectors are trained on the ProGAN dataset. Using intermediate ViT features with an HSIC bottleneck achieves the best overall mean, outperforming strong baselines while markedly improving transfer to diffusion models. The intermediate configuration secures the best ACC on most diffusion targets and on a subset of GANs (e.g., ProGAN, StarGAN), while remaining competitive elsewhere. AP is near-saturated across large portions of the table, indicating that suppressing

Table 1: **Cross-generator generalization with diffusion-trained detectors (ACC/AP).** Each cell reports ACC/AP (%). Within each dataset row, the highest, second highest, and third highest ACC are shaded red, orange, and yellow, respectively. The SDV1.4 row label cell is shaded green to indicate the diffusion training source. Methods are trained only on diffusion sources and evaluated on diffusion targets (top block), GAN targets (middle block), and two other datasets (bottom). The last row reports the mean across all targets.

Dataset	Method (ACC/AP %)						
	CNNSpot	LGrad	UniFD	NPR	VIB-Net	Ours	Ours w/ intermediate
SDV1.4	99.48/99.98	99.12/99.94	83.55/96.04	100.00/100.00	99.55/100.00	99.33/99.98	99.92/100.00
SDV1.5	99.35/99.83	99.05/99.92	84.80/96.26	99.90/99.97	99.20/99.97	99.24/99.95	99.81/100.00
ADM	50.10/51.10	53.00/58.52	53.35/66.34	73.00/94.70	73.85/95.49	85.82/97.69	91.10/99.61
GLIDE	50.90/58.80	64.24/84.00	75.30/93.73	89.70/95.80	74.25/97.13	94.97/99.26	97.07/99.85
Midjourney	56.42/67.93	76.34/91.06	71.60/92.08	82.30/95.50	88.05/97.81	83.12/97.35	80.86/99.34
Wukong	97.90/99.80	97.53/99.72	73.55/90.98	100.00/100.00	98.25/99.93	98.62/99.92	99.86/100.00
VQDM	50.04/49.92	50.93/56.34	55.10/74.53	68.30/86.30	89.35/97.00	93.69/99.25	99.42/99.99
ProGAN	50.27/53.15	61.61/83.59	58.65/51.77	60.30/83.30	89.70/96.59	97.54/99.64	99.49/99.99
CycleGAN	49.81/50.23	60.74/90.24	59.30/63.42	67.20/94.90	88.60/98.44	97.92/99.86	99.66/99.97
BigGAN	50.10/49.79	48.82/47.51	61.45/75.81	59.20/72.00	91.20/97.17	90.28/98.01	91.75/99.74
StyleGAN	50.98/55.98	61.43/82.74	56.80/54.12	58.00/82.70	74.10/84.31	94.42/98.93	88.15/97.89
StarGAN	49.77/47.07	50.17/99.19	61.45/54.93	73.20/97.30	80.70/97.60	96.05/99.52	100.00/100.00
GauGAN	50.38/56.08	49.70/49.25	55.30/65.99	52.00/66.00	87.15/96.94	87.38/94.51	90.02/97.60
Deepfake	51.98/54.86	50.17/66.49	58.40/70.24	74.80/85.30	72.00/81.32	66.35/76.47	82.15/93.82
SAN	50.22/54.03	56.49/65.09	72.00/83.34	89.60/95.90	81.50/93.27	90.64/96.13	88.58/95.51
Avg	60.51/63.24	65.29/78.24	65.37/75.31	76.50/89.98	85.83/95.53	91.69/97.10	93.86/98.89

Table 2: **Cross-generator generalization with GAN-trained detectors (ACC/AP).** Each cell reports ACC/AP (%). Within each dataset row, the highest, second highest, and third highest ACC are shaded red, orange, and yellow, respectively. The ProGAN row label cell is shaded green to indicate the GAN training source. All methods are trained only on GAN sources and evaluated on GAN targets (top block), two other datasets (middle), and diffusion targets (bottom). The final row reports the mean across all targets.

Dataset	Method (ACC/AP %)							
	CNNSpot	LGrad	UniFD	NPR	RINE	VIB-Net	Ours	Ours w/ intermediate
ProGAN	99.99/99.99	99.80/99.90	99.90/100.00	99.80/100.00	100.00/100.00	99.99/100.00	99.83/100.00	100.00/100.00
CycleGAN	87.59/96.40	86.94/94.01	98.50/99.21	96.10/98.50	99.32/99.99	99.00/99.80	88.38/99.74	93.07/99.99
BigGAN	71.18/87.50	85.63/90.75	94.50/98.31	84.40/87.80	99.60/99.94	95.75/99.29	90.08/99.53	82.05/99.94
StyleGAN	89.95/96.94	91.08/99.80	84.40/97.98	97.70/99.80	88.86/99.44	91.25/98.79	91.37/98.33	93.41/100.00
StarGAN	94.60/94.24	99.27/99.98	95.85/99.35	99.30/99.90	99.55/100.00	98.95/99.72	97.45/99.87	100.00/100.00
GauGAN	81.44/98.28	72.49/79.29	99.50/99.80	82.50/85.50	99.77/100.00	99.70/99.99	82.32/99.99	68.33/99.98
Deepfake	51.69/64.42	56.42/71.71	67.40/82.04	80.20/82.40	80.57/97.90	83.20/92.64	83.98/92.97	82.48/96.82
SAN	50.00/55.89	44.47/45.09	56.50/82.18	69.20/71.60	68.26/94.93	70.50/91.62	86.99/92.25	93.61/97.94
SDV1.4	50.82/52.86	63.03/70.90	63.10/85.48	76.60/84.00	83.96/98.35	71.55/87.24	85.79/92.25	98.82/99.94
SDV1.5	50.88/53.25	63.67/71.72	63.57/82.30	77.90/84.60	83.35/98.33	70.00/86.98	85.14/92.23	98.67/99.85
ADM	60.20/65.14	67.10/71.83	66.90/84.34	69.70/74.60	74.61/96.23	71.45/87.88	81.46/89.61	93.01/97.70
GLIDE	57.85/68.10	66.10/75.96	61.70/84.04	77.30/85.70	80.72/97.87	69.40/88.53	89.72/96.39	97.02/99.56
Midjourney	50.77/56.60	56.20/71.42	57.85/69.10	77.80/85.40	57.12/87.41	61.25/75.68	60.36/66.83	69.40/82.49
Wukong	51.13/51.15	63.60/66.51	71.06/90.13	76.10/80.50	84.95/98.62	75.90/90.92	88.36/95.38	98.62/99.88
VQDM	56.20/69.49	67.02/70.23	85.00/94.96	78.10/81.20	89.79/99.23	86.65/96.51	88.72/96.08	97.49/99.74
Avg	66.95/74.02	72.19/78.61	77.72/89.95	82.84/86.77	86.03/97.88	82.97/93.04	86.66/94.10	91.07/98.25

nuisance variation at intermediate layers yields a representation that generalizes across generation paradigms without sacrificing ranking quality on the source domain.

4.3 Continual adaptation to 3DGS domains

We evaluate sequential adaptation to three 3DGS domains (GHA, SA, GAGAvatar), averaging over all six permutations of arrival order, and report group means for Diffusion, GANs, and Others (Deepfake, SAN), per-domain 3DGS columns, and an all-targets average over 18 datasets. In our setup, *base* trains only on SDV1.4 or ProGAN without 3DGS and *RINE* is a non-continual oracle jointly trained on the base plus 3DGS domains. *iCaRL*, *CBRS*, and our *HGR* are sampling methods for the replay buffer whose scores are averaged across arrival orders. In the tables, boldface marks the best mACC among sampling methods only (iCaRL, CBRS, HGR). With an SDV1.4 start (Table 3), HGR achieves the highest overall mean among sampling methods and even surpasses the non-continual

Table 3: Training from **SDV1.4**. Cells show **mACC/mAP** (%). We evaluate sequential adaptation to three 3DGS domains (GHA, SA, GAGAvatar), averaging over all six permutations of arrival orders. We report group means for Diffusion, GANs, and Others; per-domain 3DGS columns; and an all-targets average over 18 datasets. In our setup, *base* trains only on SDV1.4 without 3DGS; *RINE* is a non-continual oracle trained jointly on SDV1.4 plus {GHA, SA, GAGAvatar}; and *iCaRL*, *CBRS*, and our *HGR* are sampling methods for the replay buffer. **Bold** highlights the best mACC among the sampling methods only.

Method	Diffusion	GANs	Others	GHA	SA	GAGAvatar	Average
base	95.43/99.83	94.85/99.20	85.37/94.67	66.05/78.10	64.65/80.66	50.39/54.56	88.27/94.26
iCaRL	96.00/99.65	91.57/97.61	78.11/93.34	96.01/99.85	94.99/99.80	94.52/99.12	92.40/98.26
CBRS	95.15/99.68	93.15/98.60	77.58/94.24	95.23/99.78	96.58/99.97	**96.02/99.50**	92.66/98.73
HGR	**97.12/99.81**	**94.00/99.07**	**82.31/94.29**	**97.06/99.77**	**98.07/99.99**	95.18/99.07	**94.38/98.92**
RINE	95.54/99.76	95.62/99.39	78.39/96.08	94.57/98.86	98.67/99.94	94.90/99.08	93.75/99.15

Table 4: Training from **ProGAN**. Cells show **mACC/mAP** (%). We evaluate sequential adaptation to three 3DGS domains (GHA, SA, GAGAvatar), averaging over all six permutations of arrival orders. We report group means for Diffusion, GANs, and Others; per-domain 3DGS columns; and an all-targets average over 18 datasets. In our setup, *base* trains only on ProGAN without 3DGS; *RINE* is a non-continual oracle trained jointly on ProGAN plus {GHA, SA, GAGAvatar}; and *iCaRL*, *CBRS*, and our *HGR* are sampling methods for the replay buffer. **Bold** highlights the best mACC among the sampling methods only.

Method	Diffusion	GANs	Others	GHA	SA	GAGAvatar	Average
base	93.29/97.02	89.48/99.99	88.05/97.38	51.25/74.75	55.78/94.09	61.98/73.83	85.28/95.36
iCaRL	76.79/91.10	88.84/94.74	78.90/92.80	95.23/99.86	97.47/99.98	96.93/99.85	84.33/93.96
CBRS	80.33/92.97	89.99/95.34	77.55/90.63	**97.82/99.85**	98.07/100.00	**98.09/99.89**	86.18/94.65
HGR	**82.87/94.33**	**93.94/98.85**	**80.99/90.72**	94.71/99.47	**99.45/100.00**	95.47/99.26	**88.63/96.31**
RINE	82.15/95.58	96.10/99.72	85.15/97.11	90.90/96.66	98.81/99.96	90.98/97.06	89.04/98.27

oracle; on 3DGS, HGR leads on GHA and SA, while CBRS is slightly higher on GAGAvatar. With a ProGAN start (Table 4), HGR again delivers the best overall mean among sampling methods and clearly improves the GANs group relative to the base; within 3DGS, CBRS tops GHA and GAGAvatar, whereas HGR peaks on SA. The base and RINE serve as reference points rather than direct competitors; across both initializations, HGR is the most effective sampling strategy, with CBRS offering targeted gains on specific 3DGS regimes.

4.4 Ablation on HSIC Components and Intermediate Features

Table 5 examines (left) the roles of HSIC(x, z), HSIC(y, z), and intermediate ViT/CLIP representations, and (right) the choice of HSIC kernel/bandwidth. Enforcing HSIC(y, z) consistently strengthens cross-generator generalization by aligning the latent with task-relevant variation, while HSIC(x, z) acts as a complementary regularizer that discourages input-anchored shortcuts and stabilizes optimization; using both terms together yields a representation that preserves prior competencies yet remains adaptable. Enabling intermediate features improves both accuracy and ranking across all configurations, indicating that earlier layers expose generator cues that the HSIC bottleneck can regularize toward invariance.

For the dependence measure, Cosine and IMQ perform competitively, but an RBF kernel with a median heuristic provides the most stable behavior across both SDV1.4 and ProGAN bases, likely due to its scale adaptivity without manual tuning.

4.5 Domain-Incremental Learning Analysis

We report per-dataset detection performance (mACC/mAP) for the arrival order whose final mACC is closest to the mean over all six permutations of arrival orders; in each subtable, the first row lists the chosen sequence (starting from the base), and the column shows the performance after the

Table 5: **HSIC ablations (components/intermediate on the left; kernel/bandwidth on the right).** We report **mACC/mAP** (%) on **SDV1.4** and **ProGAN**. (a) toggles HSIC(x, z), HSIC(y, z), and the use of intermediate ViT features; the full configuration (both HSIC terms + intermediates) attains the best overall performance. (b) compares HSIC kernels and bandwidths (σ); an RBF with the median heuristic performs best and is adopted as default. **Bold** marks the top mACC per dataset.

(a) Ablations on HSIC components and intermediate features. ✓= enabled, ✗= disabled. We report **mACC/mAP** (%) on **SDV1.4** and **ProGAN**.

HSIC(x, z)	HSIC(y, z)	Intermed.	SDV1.4	ProGAN
✗	✗	✗	89.20/96.77	83.30/90.26
✓	✗	✗	89.75/96.66	83.29/90.22
✗	✓	✗	91.64/97.34	87.05/94.45
✗	✗	✓	92.22/98.31	89.22/95.62
✓	✓	✗	91.69/97.10	86.66/94.10
✗	✓	✓	93.39/98.80	90.67/97.78
✓	✗	✓	90.27/97.48	87.12/93.70
✓	✓	✓	**93.86/98.89**	**91.07/98.25**

(b) Ablations on HSIC kernel and bandwidth. "IMQ" denotes the inverse multiquadratic kernel. We report **mACC/mAP** (%) on **SDV1.4** and **ProGAN**.

Kernel	SDV1.4	ProGAN
Cosine	88.70/98.64	91.49/97.22
IMQ	92.79/98.59	89.23/97.50
RBF ($\sigma = 1$)	92.75/98.72	88.82/97.89
RBF ($\sigma = 2$)	92.71/98.55	90.50/97.52
RBF ($\sigma = 3$)	93.14/98.71	90.83/97.48
RBF ($\sigma = median$)	**93.86/98.89**	**91.07/98.25**

Table 6: **Domain-Incremental Learning Analysis.** Each subtable shows the arrival order whose final mACC is closest to the mean over all six permutations of arrival orders; the first row lists the selected sequence, and the column shows the performance after the dataset has arrived. The numbers highlighted in gray indicate that their corresponding 3DGS datasets have not yet been included in the domain-incremental training.

(a) Starting from SDV1.4, then SA, GAGAvatar, and GHA arrived sequentially. Forgetting happened mostly on GANs and Others.

Dataset	**SDV1.4**	**SA**	**GAGAvatar**	**GHA**
Diffusion	95.43/99.83	97.41/99.84	97.59/99.81	96.90/99.79
GANs	94.85/99.20	93.74/98.60	94.36/99.12	93.24/98.79
Others	85.37/94.67	81.01/96.35	76.79/93.93	84.43/93.06
SA	64.65/80.66	97.01/99.83	99.86/100.00	98.99/100.00
GAGAvatar	50.39/54.56	64.91/69.93	99.37/99.98	94.17/99.51
GHA	66.05/78.10	66.67/86.96	66.41/84.82	98.80/99.94
Average	88.27/94.26	90.83/96.66	92.70/98.11	94.36/98.71

(b) Starting from ProGAN, then SA, GHA, and GAGAvatar arrived sequentially. Forgetting happened mostly on Diffusion and Others.

Dataset	**ProGAN**	**SA**	**GHA**	**GAGAvatar**
GANs	89.48/99.99	91.96/99.77	89.56/99.42	93.18/99.39
Others	88.05/97.38	76.84/94.96	83.28/94.53	82.82/92.31
Diffusion	93.29/97.02	85.33/94.85	87.67/94.75	84.66/93.63
SA	55.78/94.09	94.17/100.00	100.00/100.00	99.55/100.00
GHA	51.25/74.75	66.71/88.29	96.23/99.93	92.57/99.03
GAGAvatar	61.98/73.83	50.23/71.24	55.18/68.98	98.49/99.88
Average	85.28/95.36	84.10/95.11	87.17/95.43	89.33/96.40

dataset has arrived. Forgetting concentrates on out-of-domain targets; for example, when training from SDV1.4, degradation is most pronounced on GANs and Others (and symmetrically, diffusion degrades when starting from ProGAN), though the effect remains modest under our rehearsal budget. At the same time, new domains provide positive transfer by exposing the model to additional artifact patterns and scene statistics, yielding consistent gains on datasets related to the most recently trained domain and smaller but nontrivial improvements on some cross-generator targets.

5 Conclusion

We presented a detector that couples an HSIC bottleneck on intermediate CLIP features with HSIC-Guided Replay regularized by a k-center coverage term. The bottleneck filters text-alignment nuisances while preserving discriminative cues, yielding strong cross-generator generalization when training on either diffusion or GAN sources. The replay mechanism selects compact, informative exemplars that stabilize continual adaptation to 3D Gaussian Splatting (3DGS) domains. Across SDV1.4- and ProGAN-based training, our method achieves consistent gains in both mACC and mAP, maintains performance on earlier targets, and shows low variance under different 3DGS orderings. To support rigorous study of rendered-fakes, we curated three 3DGS head avatar datasets spanning multi-view reconstruction, single-view reconstruction, and a generative method.

This work suggests that independence-driven regularization and coverage-aware replay are complementary for open-world forgery detection. Future directions include extending the approach to video and audio-driven avatars, scaling to larger backbones and training regimes, learning to adaptively allocate memory over time, and exploring task-agnostic online updates without explicit domain boundaries.

ETHICS STATEMENT

Our study uses only publicly available image datasets containing human faces, accessed and used under the research licenses of the datasets. We conducted no interaction or intervention with individuals and accessed no private or non-public data. We process facial imagery solely for research on synthetic-content detection, apply data minimization and security safeguards consistent with the dataset licenses, and report only aggregate results. We do not attempt to identify, profile, or target individuals. We have carefully considered potential impacts and do not anticipate ethical risks beyond those commonly encountered in computer vision and machine learning research.

REPRODUCIBILITY STATEMENT

We document the training pipeline and evaluation protocols in both the main paper and the appendix, with explicit hyperparameters and dataset partitions to support exact replication. We also describe implementation settings and reporting conventions to match our results. To further enable reproducibility, we will release the full training and evaluation code, along with runnable scripts, in the camera-ready version.

THE USE OF LARGE LANGUAGE MODELS (LLMS)

We used LLMs to assist with (i) polishing prose (grammar, flow, and clarity), (ii) reorganizing and tightening section structure, captions, and titles, and (iii) brainstorming keywords and query strings to surface related work.

For literature discovery, the model suggested search terms and produced brief summaries to orient the authors. Every citation in the paper was located through standard search engines or digital libraries and then read and verified by the authors; we did not accept model-generated references without inspection. Numerical results, comparisons, and quotes were cross-checked against the original sources.

We edited all model-suggested text for accuracy and originality and ensured that the writing reflects our intent. No confidential or sensitive data were shared with the model. The final manuscript, including all tables and figures, was reviewed end-to-end by the authors for factual correctness and completeness.

REFERENCES

Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Computer Vision - ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part III*, pp. 144–161, 2018.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8789–8797, 2018.

Charis Chrysakis and Farhad Pourkamali-Anaraki. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pp. 1673–1682. PMLR, 2020.

Xuangeng Chu and Tatsuya Harada. Generalizable and animatable gaussian head avatar. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Tao Dai, Jianrui Cai, Yongbing Zhang, Shu-Tao Xia, and Lei Zhang. Second-order attention network for single image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11057–11066, 2019.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat GANs on image synthesis. In *Advances in Neural Information Processing Systems*, pp. 8780–8794, 2021.

Guy Gafni, Justus Thies, Michael Zollhöfer, and Matthias Nießner. Dynamic neural radiance fields for monocular 4D facial avatar reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8649–8658, 2021.

Teofilo F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38:293–306, 1985. doi: 10.1016/0304-3975(85)90224-5.

Philip-William Grassal, Malte Prinzler, Titus Leistner, Carsten Rother, Matthias Nießner, and Justus Thies. Neural head avatars from monocular RGB videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18653–18664, 2022.

Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, pp. 63–77, 2005.

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*, pp. 10696–10706, 2022.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4396–4405, 2019.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, pp. 3521–3526, 2017.

Tobias Kirschstein, Shenhan Qian, Simon Giebenhain, Tim Walter, and Matthias Nießner. NeRSemble: Multi-view radiance field reconstruction of human heads. *ACM Trans. Graph.*, 2023.

Christos Koutlis and Symeon Papadopoulos. Leveraging representations from intermediate encoder-blocks for synthetic image detection. In *Computer Vision - ECCV 2024: 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXII*, pp. 394–411, 2024.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2935–2947, 2018.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Midjourney. `https://www.midjourney.com/home/`, 2022.

Alexander Quinn 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. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 16784–16804, 2022.

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*, pp. 24480–24489, 2023.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2332–2341, 2019.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. iCaRL: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2017.

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*, pp. 10684–10695, 2022.

Andreas Rossler, Davide Cozzolino, Luisa Verdoliva, Christian Riess, Justus Thies, and Matthias Nießner. FaceForensics++: Learning to detect manipulated facial images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, 2016.

Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018.

Zhijing Shao, Zhaolong Wang, Zhuang Li, Duotun Wang, Xiangru Lin, Yu Zhang, Mingming Fan, and Zeyu Wang. SplattingAvatar: Realistic Real-Time Human Avatars with Mesh-Embedded Gaussian Splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Chuangchuang Tan, Yao Zhao, Shikui Wei, Guanghua Gu, and Yunchao Wei. Learning on gradients: Generalized artifacts representation for GAN-generated images detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12105–12114, 2023.

Chuangchuang Tan, Huan Liu, Yao Zhao, Shikui Wei, Guanghua Gu, Ping Liu, and Yunchao Wei. Rethinking the up-sampling operations in CNN-based generative network for generalizable deepfake detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 28130–28139, 2024.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. CNN-generated images are surprisingly easy to spot...for now. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8695–8704, 2020.

Zifeng Wang, Zheng Zhan, Yifan Gong, Yucai Shao, Stratis Ioannidis, Yanzhi Wang, and Jennifer Dy. DualHSIC: HSIC-bottleneck and alignment for continual learning. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

Wukong. `https://xihe.mindspore.cn/modelzoo/wukong`, 2022.

Yuelang Xu, Benwang Chen, Zhe Li, Hongwen Zhang, Lizhen Wang, Zerong Zheng, and Yebin Liu. Gaussian head avatar: Ultra high-fidelity head avatar via dynamic gaussians. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3014–3023, 2021.

Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, pp. 3987–3995, 2017.

Haifeng Zhang, Qinghui He, Xiuli Bi, Weisheng Li, Bo Liu, and Bin Xiao. Towards universal AI-generated image detection by variational information bottleneck network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23828–23837, 2025.

Yufeng Zheng, Victoria Fernández Abrevaya, Marcel C. Bühler, Xu Chen, Michael J. Black, and Otmar Hilliges. IM Avatar: Implicit morphable head avatars from videos. In *Computer Vision and Pattern Recognition (CVPR)*, 2022.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 2242–2251, 2017.

A Appendix

A.1 Robustness of cross-generator synthetic image detection

Figure 3 plots mean accuracy with across-seed standard deviation shown as error bars. Variation is consistently small: for most targets, the standard deviation is roughly 0 to 1.5 percentage points, with slightly larger yet still bounded spreads on a few harder sets such as Midjourney, GauGAN, and Deepfake. Overall accuracies remain high, typically in the 80 to 90% range, and several targets approach 100%. The tight uncertainty bands indicate robustness to random initialization and optimization noise and support the conclusion that our HSIC-based training yields reliable cross-generator synthetic image detection.

A.2 Pretrained model choice

Figure 4 compares three feature extractors: CLIP ViT-L/14, CLIP ViT-B/16, and DINOv2 ViT-L/14. CLIP ViT-L/14 is our default and consistently delivers the highest accuracy across most datasets. CLIP ViT-B/16 follows closely, typically trailing by about 1 to 3 percentage points, which indicates that the method is not overly sensitive to model scale within the CLIP family. DINOv2 ViT-L/14 lags more clearly, particularly on targets produced by GANs and on Deepfake, where the gap can widen to roughly 10 to 30 percentage points. These trends suggest that image-text pretraining in CLIP exposes generator-related artifacts more effectively, while the improvements we observe are largely attributable to the HSIC-based training objective rather than reliance on a single backbone.

A.3 Block-wise effects of intermediate features

Figure 5 reports performance when evaluation is restricted to a single CLIP ViT block. At inference time, we zero out features from all non-target blocks and retain only the target block.

Across most GAN families (ProGAN, StyleGAN, StarGAN; and to a lesser extent CycleGAN and BigGAN), we observe a broad mid-to-late plateau: many adjacent blocks yield comparable mean accuracy, indicating robustness to modest shifts in the chosen block. In contrast, GauGAN exhibits a narrow late-layer peak with steep degradation outside that region. Because our protocol fixes the intermediate block globally for efficiency and fairness, the choice that best serves the majority becomes suboptimal for GauGAN—accounting for its weaker row in Table 2.

When GauGAN performance is prioritized, two lightweight adjustments help: (i) aggregate a small window of adjacent late blocks to form a multi-block feature; or (ii) apply an HSIC pyramid that regularizes over neighboring blocks, capturing late-layer signal while preserving the plateaued behavior on other models.

A.4 Analysis of learned representations

Figure 6 and Figure 7 visualize feature geometry before and after applying the HSIC bottleneck using t-SNE. For each target dataset, the upper panel shows pretrained CLIP embeddings x and the lower panel shows learned embeddings z after training on SDV1.4 in Figure 6 and on ProGAN in Figure 7; blue denotes real and orange denotes synthetic. Relative to x, z forms tighter within-class clusters and larger margins between real and synthetic across generators, with the effect most visible for transfer from diffusion to GAN and from GAN to diffusion. This matches the objective: the HSIC term with labels increases between-class separation, while the HSIC term with inputs reduces reliance on input-specific artifacts and lowers within-class variance. When trained on SDV1.4, real clusters in z become more generator-agnostic, and synthetic clusters move closer together, indicating stronger alignment across models. When trained on ProGAN, z sharpens separation on GAN targets and increases margins on diffusion targets, though small pockets of overlap remain for certain datasets, such as GauGAN. Overall, these t-SNE views corroborate the quantitative results by showing that the HSIC bottleneck reshapes features toward a generator-invariant yet discriminative structure.

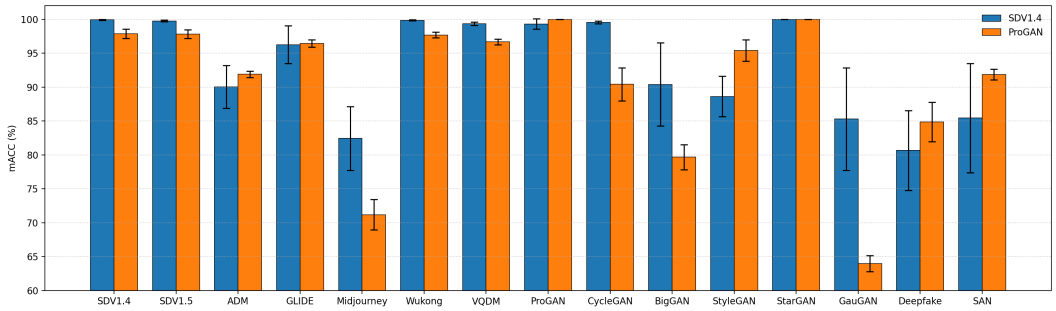

Figure 3: **Per-dataset accuracy with variability.** Bars show the mean accuracy over 5 runs; error bars denote the across-seed standard deviation. Colors compare detectors trained on **SDV1.4** (blue) versus **ProGAN** (orange).

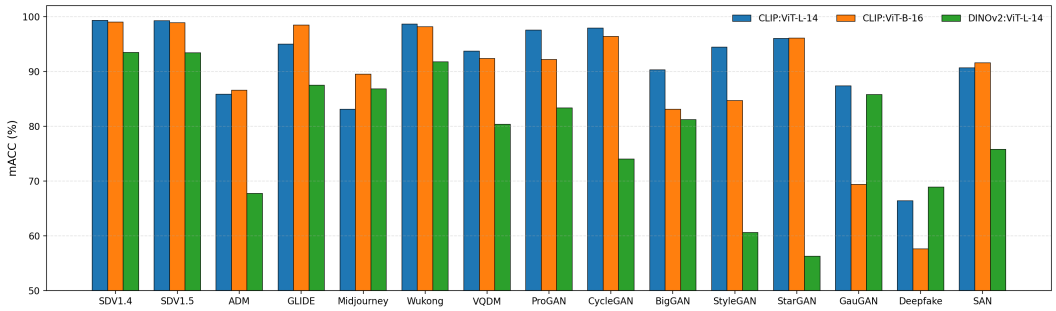

Figure 4: **Pretrained model choice.** We extract frozen features from three pretrained models—CLIP ViT-L/14, CLIP ViT-B/16, and DINOv2 ViT-L/14—and train the detector on SDV1.4. The figure reports accuracy for each target dataset.

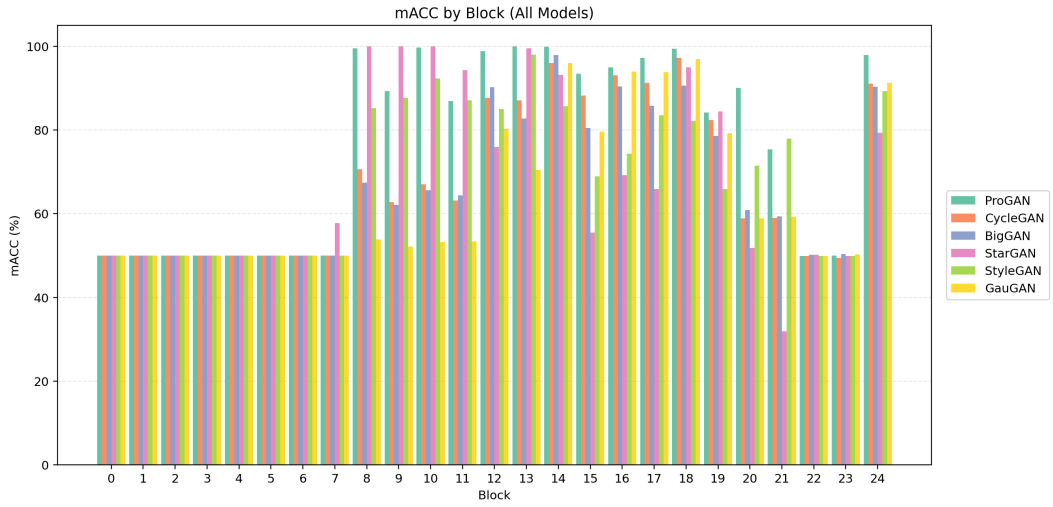

Figure 5: **Where the signal lives: block-wise test accuracy across GAN families.** Each group reports accuracy when the detector uses only one CLIP ViT intermediate block (non-target blocks are zeroed at inference).

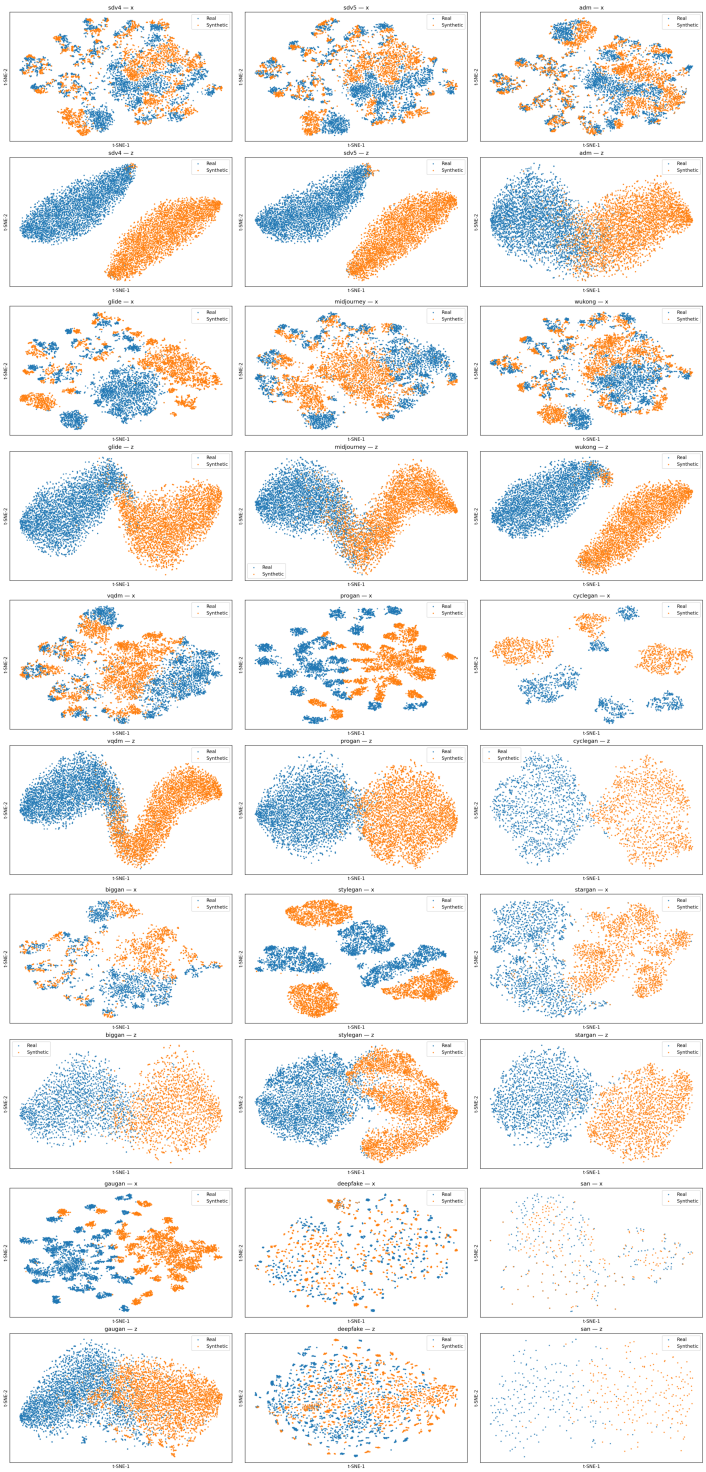

Figure 6: **t-SNE of test features across generators (upper: x, lower: z), trained on SDV1.4.** For each test set (columns grouped by generator; rows per dataset), the upper panel shows embeddings of the pretrained CLIP features x, while the lower panel shows embeddings of the features z learned by training on SDV1.4 with the HSIC bottleneck. Points are colored by ground truth (blue = real, orange = synthetic). Relative to x, the HSIC-trained z generally yields tighter, better-separated clusters of real versus synthetic across both diffusion and GAN generators, indicating improved transferable separability.

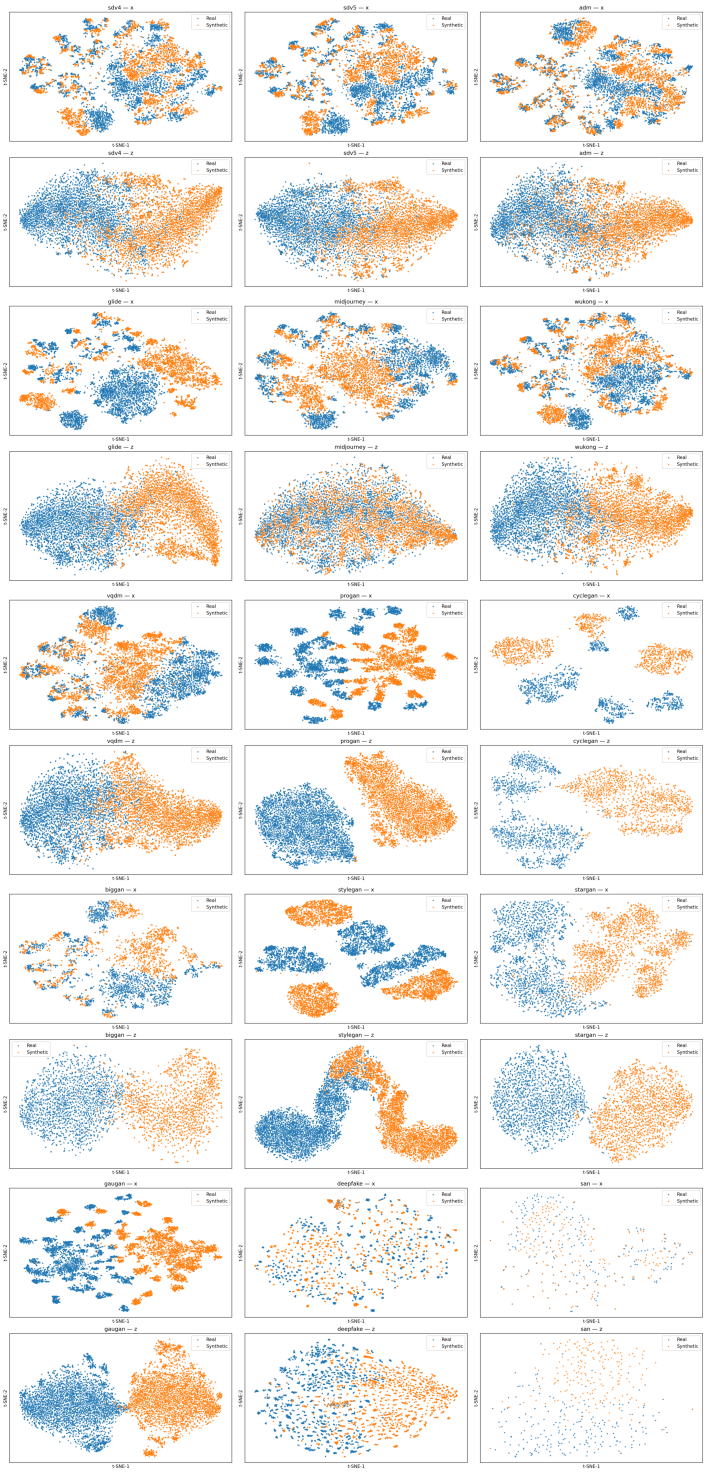

Figure 7: **t-SNE of test features across generators (upper: x, lower: z), trained on ProGAN.** For each test set, the upper panel visualizes embeddings of the pretrained CLIP features x, and the lower panel shows embeddings of the features z learned by training with the HSIC bottleneck on **ProGAN**. Points are colored by ground truth (blue = real, orange = synthetic). Relative to x, the HSIC-trained z typically exhibits tighter clusters and larger margins between real and synthetic across both GAN and diffusion generators, indicating improved transferable separability from a ProGAN-trained detector.