This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

E3: Ensemble of Expert Embedders for Adapting Synthetic Image Detectors to New Generators Using Limited Data

Aref Azizpour Tai D. Nguyen Manil Shrestha Kaidi Xu Edward Kim Matthew C. Stamm
Drexel University
Philadelphia, PA, USA
{aa4639, tdn47, ms5267, kx46, ek826, mcs382}@drexel.edu

Code: https://github.com/Aref Az/E3-Ensemble-of-Expert-Embedders-CVPRWMF24

Abstract

As generative AI progresses rapidly, new synthetic image generators continue to emerge at a swift pace. Traditional detection methods face two main challenges in adapting to these generators: the forensic traces of synthetic images from new techniques can vastly differ from those learned during training, and access to data for these new generators is often limited. To address these issues, we introduce the Ensemble of Expert Embedders (E3), a novel continual learning framework for updating synthetic image detectors. E3 enables the accurate detection of images from newly emerged generators using minimal training data. Our approach does this by first employing transfer learning to develop a suite of expert embedders, each specializing in the forensic traces of a specific generator. Then, all embeddings are jointly analyzed by an Expert Knowledge Fusion Network to produce accurate and reliable detection decisions. Our experiments demonstrate that E3 outperforms existing continual learning methods, including those developed specifically for synthetic image detection.

1. Introduction

Over recent years, a number of AI-based techniques have been developed to create visually realistic synthetic images. While these synthetic image generators can be used for creative or artistic purposes, they can also be used to malicious ones. Specifically, they enable the creation of fake images that can be used for misinformation or disinformation.

To address these challenges, considerable research efforts have been directed towards the development of synthetic image detection techniques. Prior work has demonstrated the effectiveness of forensic neural networks in discerning synthetic images by identifying unique traces left behind by different image generators. However, a significant limitation of existing detectors arises when they encounter images generated by previously unseen or emerging techniques, whose traces differ substantially from those in the training data.

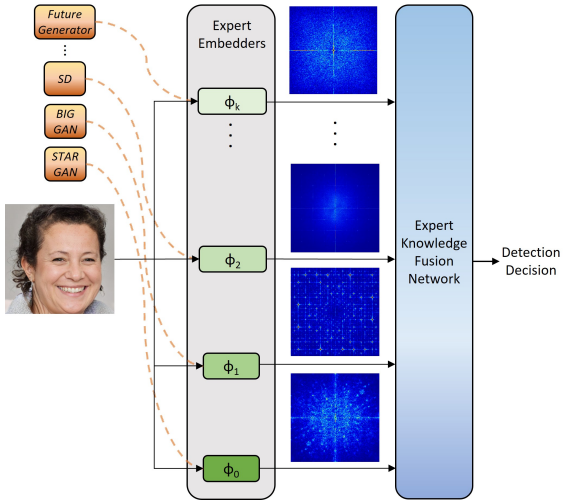

Figure 1. Visualization of the Ensemble of Expert Embedders (E3) framework aimed at enhancing synthetic image detection in response to new generators. Examples of the forensic residual traces for different generator architectures are also depicted.

This poses a critical need for continually updating synthetic image detectors to adapt to new generators. However, this task presents several challenges. Traditional approaches to updating detectors often encounter issues such as catastrophic forgetting and the impracticality of storing and retraining on large datasets [36, 53, 56]. Moreover, limited data availability from newly emerging generators, especially those not yet publicly accessible, further complicates the updating process.

In this paper, we propose a novel approach to address these challenges. Instead of relying on a single network to capture all forensic traces, we introduce an Ensemble of Expert Embedders (E3) framework. Each expert embedder is

created through transfer learning and specializes in capturing traces from a specific image generator. These expert embedders collectively generate a sequence of embeddings, which are then analyzed by an Expert Knowledge Fusion Network (EKFN). The EKFN leverages information from all expert embedders to accurately perform synthetic image detection. Our experimental results demonstrate that the proposed E3 framework significantly outperforms existing continual learning approaches, including those designed specifically for updating synthetic image detectors, and can perform strongly even with very limited data from new generators. The novel contributions of this paper are:

- We propose E3, a new approach that can update synthetic image detectors to accurately detect newly emerging generators, while requiring only minimal amounts of training data to be retained in a memory buffer.
- We develop a novel approach to create a set of expert embedders to accurately capture traces from each new target generator. Each expert can be adapted using a small set of images from its target generator.
- We propose an expert knowledge fusion network able to examine forensic evidence produced by the set of all experts and make accurate detection decision.
- We conduct an extensive set of experiments demonstrating that E3 outperforms competing approaches from continual learning, including approaches specifically designed to update synthetic image detectors. Notably, E3 consistently outperforms competitors across various detector architectures and excels even with limited data from new generators.

2. Background

Synthetic Image Generators. Considerable effort has been devoted to developing systems that can visually understand the world through the process of mimicking; the first of such work is Variational Auto-Encoder by Kingma and Welling [33], which led to the development of Generative Adverserial Networks (GANs) by Goodfellow et al. [18]. This work inspired many other subsequent works [8, 26, 28, 47, 71], which continued to improve visual understanding through enhancing the generation's quality, diversity, and realism. Notably, the introduction of diffusion model for image generation by Ho et al. [21] set the stage for the explosion of research in this area, resulting in many popular generation methods such as: Stable Diffusion [54], DALL·E [50], Midjourney [1], Cascade Diffusion [22], etc. [3, 15, 64, 73]. Due to the rapidly increasing rate of emergence of new generation methods, existing synthetic image detectors face a formidable challenge. These detectors often have limited capabilities, restricting them to detecting only the generators seen during training [13, 59].

Synthetic Image Traces and Detection. Numerous efforts have been undertaken to distinguish synthetic images from real ones. Marra et al. [39] demonstrated that GAN-generated images possess unique "fingerprints" useful for detection and source attribution. This work showed that individual image generators all left behind identifiable artifacts, called forensic traces, that can be used to detect their generated images. While prior work has examined transferable forensic features [43], subsequent research demonstrated the difficulty for a detector to generalize to forensic traces from a new family of generators [13, 41]. To address the rapid release of generators, a broad spectrum of new synthetic image detectors, and more recently, synthetic video detectors, were developed [13, 17, 25, 39, 58, 60–62, 68, 69, 72].

Continual Learning For Synthetic Image Detection. Given the distinct forensic traces left by various generator families, it is crucial for image detectors to continually update their knowledge with new generators. Traditional retraining is often data inefficient and fine-tuning may often lead to catastrophic forgetting [45, 46]. Hence, a number of methods have been developed to allow the base model to adapt without losing prior knowledge [10, 36, 52, 56]. Notably, Marra et al. [40] and Kim et al. [31] have successfully applied such strategies to synthetic image detection, demonstrating the feasibility and effectiveness of continual learning in this domain.

3. Problem Formulation

We begin by assuming that an initial synthetic image detector f_0 has been created to detect images generated by a set of known generators $\mathcal{G}_0$. We will refer to this detector as the baseline detector. We further assume that f_0 was trained using a large baseline training dataset $\mathcal{B}$ consisting of both real and synthetic images made using generators in $\mathcal{G}_0$.

After the baseline detector has been trained, new synthetic image generators $g_k \notin \mathcal{G}_0$ will continue to emerge. As previous research has shown, f_0 will have a difficult time detecting these new generators if the forensic traces or "fingerprints" left by these generators are substantially different than those left by the generators in $\mathcal{G}_0$ [13]. Because of this, the synthetic image detector will need to be continually updated as new generators emerge.

This presents an important set of problems. Training a new detector from scratch can be resource intensive and requires that the baseline dataset $\mathcal{B}$ be stored indefinitely. Instead, it is preferable to retain only a small dataset $\mathcal{M}$, known as the memory buffer, to update the detector. Care must be taken when updating the detector, however, because naively fine-tuning the detector can lead to catastrophic forgetting. When this happens, the detector is able to identify images produced by the new generator, but loses its ability to reliably detect images from previous generators.

Additionally, challenges may arise when access to synthetic images generated by the new generator g_k is limited. For instance, a generator may not be available to the general public, but a small number of examples from the generator may be publicly obtainable. An example of this is OpenAI's Sora generator [9]. Currently, access to Sora is restricted, however OpenAI has shared a small number of sample outputs. Alternatively, a party engaged in a misinformation campaign may have developed a new generator for their purposes. In this case, access to images produced by this new generator is severely limited by the number of examples that have been released into the wild. Training an effective synthetic image detector that can continuously adapt to new generators with limited data is highly non-trivial.

To formalize the problem of updating the detector given these constraints, we define $\mathcal{M}_{k-1}$ as the memory buffer when the k^{th} new generator is introduced. $\mathcal{M}_{k-1}$ consists of two subsets: $\mathcal{R}$ which contains real images and $\mathcal{S}_{k-1}$ which contains images made by each of the previously seen generators. The memory buffer has a fixed size $|\mathcal{M}_\ell| = M$, that remains constant even as more generators emerge. Additionally, we define $\mathcal{D}_k$ as the set of images from the new generator g_k that can be used to update the detector. We assume that $|\mathcal{D}_k| = N$, and that N is significantly smaller than the number of images in $\mathcal{B}$, i.e. we are allowed a small number of images from the new generator.

4. Proposed Approach

While a synthetic image detector f is often thought of as a single network, we can conceptually view it as the composition of an embedder ϕ and a classifier h, such that $f = h \circ \phi$. When adopting this view, the embeddings produced by ϕ capture forensic traces left by synthetic image generators, while the detector maps these embeddings to detection decisions. In practice, lower layers of the network can be thought of as the embedder, while the final layer or layers can be thought of as the classifier.

When a continual learning technique is used to update f, this typically involves using a special process that retrains f to detect images from both a new generator and existing generators using $\mathcal{M}_k$ and $\mathcal{D}_k$. This corresponds to updating ϕ so that it learns an embedding space that is able to jointly capture forensic traces from previous generators as well as the new generator. However, since forensic traces from different generators can be substantially different [13], learning an embedding space that successfully does this with a limited amount of data in $\mathcal{M}_k$ and $\mathcal{D}_k$ can be challenging.

To overcome this challenge, we propose a new framework to update f called *Ensemble of Expert Embedders* (E3). In this framework, we do not attempt to learn a single embedding space to capture the traces left by all generators. Instead, we form a set of expert embedders $\Phi_k = \{\phi_0, \ldots, \phi_k\}$, where each expert embedder ϕ_ℓ is specialized to capture traces from generator g_ℓ. When forming an expert embedder ϕ_k for a new generator g_k, we allow it to experience catastrophic forgetting, since other experts in Φ_k are dedicated to capturing traces from other generators.

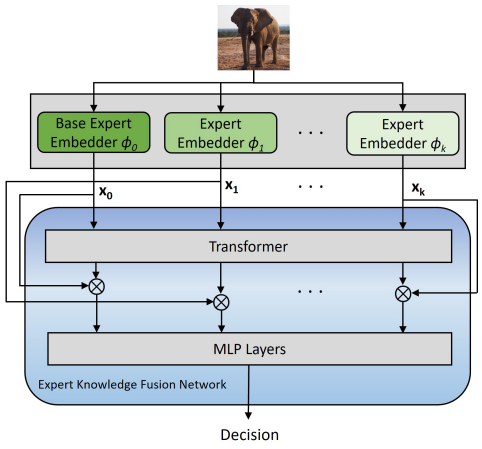

Figure 2. End-to-end architecture workflow: An image is first processed by E3 to generate embeddings, which are passed through a transformer and MLP layers to produce the detection decision.

To analyze an image, it is passed through all embedders in Φ_k to produce a sequence of embeddings $\{\mathbf{x}_0, \ldots, \mathbf{x}_k\}$. Each embedding $\mathbf{x}_\ell$ captures evidence that the image was generated by g_ℓ. This set of embeddings is then analyzed using an Expert Knowledge Fusion Network (EKFN) to produce a single detection decision. As a result, E3 is able to leverage forensic evidence within the union of every expert's embedding space, as opposed to relying on a single embedding space to capture all forensic evidence.

We describe the process to form each expert embedder, update the memory buffer, and train the expert knowledge fusion network in the following subsections.

4.1. Creating A New Expert Embedder

Let $\mathcal{G}_{k-1}$ be the set of currently known synthetic image generators. When a new synthetic image generator $g_k \notin \mathcal{G}_{k-1}$ emerges, we need to update our detector. To do this, we must first create a new expert embedder ϕ_k to capture traces left by g_k. We accomplish this by adapting the baseline detector f_0 using transfer learning. An overview of this process is shown in Fig. 3.

We start by forming a new expert training set $\mathcal{T}_k$ of real and synthetic images that we will use to create ϕ_k. This is done by collecting a set $\mathcal{D}_k$ of N synthetic images created by the new generator. Real images are taken from the current memory buffer $\mathcal{M}_{k-1}$, which contains the subset $\mathcal{R}$ consisting of $M/2$ real images. As a result, the new expert training set is $\mathcal{T}_k = \mathcal{D}_k \cup \mathcal{R}$.

Next, we use $\mathcal{T}_k$ to update f_0 to detect images made by g_k. Specifically, we form a new detector $\hat{f}_k$ by fine tuning

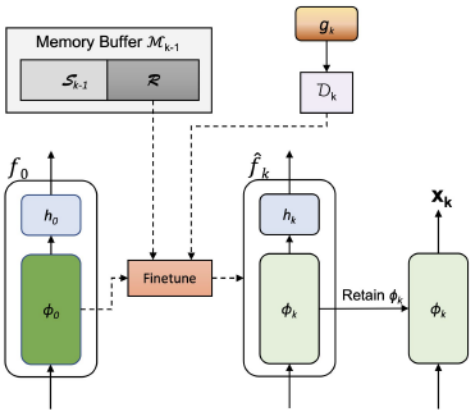

Figure 3. Creation of a new specialized expert, ϕ_k, when a new generator emerges. The baseline detector f_0 is fine-tuned with data $\mathcal{D}_k$ and real images $\mathcal{R}$ from memory buffer. The new classifier $\hat{f}_k$'s embedder ϕ_k is then preserved.

f_0 with $\mathcal{T}_k$ using the loss function

$$\mathcal{L}_\phi = -\sum_{I_\ell \in \mathcal{T}_k} \frac{|\mathcal{D}_k|}{|\mathcal{T}_k|} t_\ell log(\hat{f}_k(I_\ell)) + \frac{|\mathcal{R}|}{|\mathcal{T}_k|}(1-t_\ell)log(1-\hat{f}_k(I_\ell)), \quad (1)$$

where I_ℓ is the ℓ-th image in $\mathcal{T}_k$, and t_ℓ is the class label of the image I_ℓ where 0 means the image is real and 1 means the image is synthetic. By doing this, the resulting detector $\hat{f}_k$ will be adapted to detect images made by g_k. However, it will likely experience catastrophic forgetting and be poorly suited to detect generators in $\mathcal{G}_{k-1}$.

After obtaining the expert detector $\hat{f}_k$, we decompose it into its corresponding embedder ϕ_k and classifier components. The classifier portion is discarded, while ϕ_k is retained. Finally, ϕ_k is added to the set of expert embedders $\Phi_k = \Phi_{k-1} \cup \{\phi_k\}$, where $\Phi_0 = \{\phi_0\}$.

4.2. Expert Knowledge Fusion Network

After adding the new expert embedder ϕ_k to the set of existing expert embedders, we must update the Expert Knowledge Fusion Network (EKFN) ψ so that it can utilize the embeddings produced by ϕ_k. We do this by first updating the memory buffer $\mathcal{M}_k$, then using this data to train ψ to perform detection. This process is described below.

Updating The Memory Buffer. When a new generator g_k emerges, the current memory buffer $\mathcal{M}_{k-1}$ does not include images generated by it. As a result, we need to add images from $\mathcal{D}_k$ to form the updated memory buffer $\mathcal{M}_k$. However, since the memory buffer has a fixed size M, we must remove synthetic images made by previous generators in $\mathcal{G}_{k-1}$ to make space for these new images.

We first note that $\mathcal{S}_{k-1}$, i.e. the subset of $\mathcal{M}_{k-1}$ that corresponds to synthetic images, contains $P_{k-1} = \lfloor M/(2k) \rfloor$ images from each generator in $\mathcal{G}_{k-1}$. We now need to form

Algorithm 1 Training Expert Knowledge Fusion Network

Require: $\mathcal{M}_{k-1}, \mathcal{D}_k, \Phi_k$
// Let $\omega_M^{(k-1)}$ be the procedure to sample images from $\mathcal{M}_{k-1}$ uniformly at random
// Let $\omega_D^{(k)}$ be the procedure to sample images from $\mathcal{D}_k$ uniformly at random
// Let S be the number of training steps
$\mathcal{M}_k \leftarrow \omega_M^{(k-1)}(\mathcal{M}_{k-1}) \bigcup \omega_D^{(k)}(\mathcal{D}_k)$
for step = $1, \cdots, S$ **do**
  $\mathbf{X} \leftarrow \{\}$
  for $i = 1;\ \ i \le |\mathcal{M}_k|;\ \ i \mathrel{+}= 1$ **do**
    $I_i = \mathcal{M}_k[i];\ \mathbf{X}_i \leftarrow \{\}$
    for $j = 1;\ \ j \le |\Phi_k|;\ \ j \mathrel{+}= 1$ **do**
      $\phi_j \leftarrow \Phi_k[j]$
      $\mathbf{X}_i \leftarrow \mathbf{X}_i \cup \{\phi_j(I_i)\}$
    end for
    $\mathbf{X} \leftarrow \mathbf{X} \cup \{\mathbf{X}_i\}$
  end for
  $\mathcal{L} \leftarrow \mathcal{L}_\psi(\mathcal{M}_k, \mathbf{X})$ ▷ Compute loss with Eqn. 2
  $\psi \leftarrow$ BackPropagate$(\mathcal{L}, \psi)$
end for
return ψ

an updated set of synthetically generated images $\mathcal{S}_k$ that contains an equal number of images from all generators in $\mathcal{G}_k$. To do this, we retain $P_k = (k/(k+1))P_{k-1}$ images corresponding to each generator in $\mathcal{G}_{k-1}$ from $\mathcal{S}_{k-1}$ drawn uniformly at random. These images are added to the new updated memory buffer's subset of synthetic images $\mathcal{S}_k$. Finally, we add P_k images drawn uniformly at random from $\mathcal{D}_k$ to $\mathcal{S}_k$ to form a complete set of synthetic images from all generators in $\mathcal{G}_k$ for the memory buffer. The updated memory buffer is now formed as $\mathcal{M}_k = \mathcal{S}_k \cup \mathcal{R}$.

Training The EKFN. Once we have updated the memory buffer, we use $\mathcal{M}_k$ to train the EKFN ψ. To do this, we first use the set of expert embedders Φ_k to extract a sequence of embeddings $\{\mathbf{x}_0^\ell, \ldots, \mathbf{x}_k^\ell\}$ for each image $I_\ell \in \mathcal{M}_k$. Next, we randomly initialize ψ and train it using the loss function

$$\mathcal{L}_\psi = -\sum_{I_\ell \in \mathcal{M}_k} \psi(\mathbf{x}_0^\ell, \ldots, \mathbf{x}_k^\ell) \log t_\ell + (1 - \psi(\mathbf{x}_0^\ell, \ldots, \mathbf{x}_k^\ell)) \log(1 - t_\ell). \quad (2)$$

After training the EKFN, the synthetic image detector has been completely updated and the network is ready to perform detection. The EKFN's training procedure is provided in Alg. 1.

EKFN Architecture. We can think of each embedder ϕ_ℓ as a mechanism that searches for evidence of traces left by generator g_ℓ. The resulting embedding $\mathbf{x}_\ell$ can then be viewed as a measurement of this evidence. When viewed this way, the purpose of the EKFN is to examine all forms

	Abbrv.	Generator Name
Baseline Generators	SG	StyleGAN [28]
	SG2	StyleGAN2 [29]
	ProG	ProGAN [27]
New & Emerging Generators	SD1.4	Stable Diffusion 1.4 [54]
	GLD	Glide [48]
	MJ	Midjourney [1]
	DLM	DALL-E Mini [14]
	TT	Taming Transformers [16]
	SD2.1	Stable Diffusion 2.1 [54]
	CIPS	Cond.Indep. Pix.Synt. [2]
	BG	BigGAN [8]

	Abbrv.	Generator Name
New & Emerging Generators	VQD	Vec. Quant. Diff. [19]
	DIG	Diffusion GAN [63]
	SG3	StyleGAN3 [30]
	GF	GANformer [24]
	DL2	DALL-E 2 [51]
	LD	Latent Diffusion [54]
	EG3	Eff.Geo. 3D GAN [11]
	PG	Projected GAN [55]
	SD1.1	Stable Diffusion 1.1 [54]
	DDG	Denoise Diff. GAN [65]
	DDP	Denoise Diff. Prob. [21]

Table 1. The full names and abbreviations for all synthetic image generators used in this paper.

of evidence in the context of one another in order to make an accurate forensic decision. In light of this, we design the EKFN such that it uses a transformer's self-attention mechanism to examine the sequence of embeddings in the context of one another.

An overview of our EKFN architecture is shown in Fig. 2. The sequence of embeddings is first passed directly to a transformer. Each transformer output is used to weight its corresponding embedding through element-wise multiplication. The resulting weighted embeddings are then passed to an MLP, which makes a final detection decision. In practice, we form the MLP using 2 layers and the transformer using 20 layers in accordance to the results of an ablation study reported on in Sec. 7. We note that simpler EKFN architectures can be utilized to reduce network complexity at the cost of decreased performance.

4.3. E3 Framework Summary

E3 uses the following steps to update a synthetic image detector after the emergence of a new generator:

1. Form a new training set using real images from the memory buffer and a set of images from the new generator.
2. Utilize this training set to create an expert embedder specifically trained to capture forensic traces associated with the new generator, and incorporate it into the ensemble of expert embedders.
3. Update the memory buffer with the new generator's data.
4. Retrain the Expert Knowledge Fusion Network (EKFN) using the data stored in the memory buffer.

5. Experiments

5.1. Experimental Setup

Baseline Detector Dataset. We created a baseline dataset in order to train and evaluate the baseline synthetic image detectors. To form a diverse set of real images, we gathered 72,000 images equally distributed among the CoCo dataset [37], the LSUN dataset [67], and the CelebA dataset [38]. As for the set of synthetic images, we also gathered 72,000 GAN-generated images from the datasets used in prior work [17, 62]. The synthetic images in this set are equally distributed among three GANs: StyleGAN [28], StyleGAN2 [29], and ProGAN [27]. Additionally, to form the set of images used for training, validation, and testing, we split the set of real images and the set of synthetic images each into their own disjoint subsets. Specifically, we used 132,000 images for training with 12,000 images held out for validation, and 12,000 images for testing.

Emerging Generators Dataset. To simulate a constantly evolving environment in which new synthetic image generators rapidly emerge, we created a dataset to train and evaluate our proposed approach in a continual learning setting. To accomplish this, we first selected a diverse set of generators, composed of 19 different generation techniques. These techniques are listed in Tab. 1. All synthetic images in this dataset are assembled from publicly available datasets [17, 49, 58]. Then, from each generator, we gathered 600 images exclusively used for training and 400 images exclusively used for testing.

Baseline Detector Training. We obtained a baseline detector by training MISLnet [6] to distinguish between real and synthetic images in our baseline detector dataset. We chose MISLnet as the architecture for our experiments because it is lightweight and has shown repeatedly to obtain strong performance on detecting synthetic images and other image forensic tasks [4, 5, 12, 42, 44]. We trained MISLnet using a Binary Cross-Entropy Loss function via the ADAM optimizer [32] with a constant learning rate of 5.0×10^{-5}. This model resulted in an accuracy of 0.97 on the testing portion of our baseline detector dataset.

Memory Buffer. Throughout all experiments in this paper, we fixed the size of memory buffer, $\mathcal{M}$, to 1000 images from which 500 are synthetic and 500 are real. The synthetic images are then equally distributed among all previously known generators. We note that the memory buffer is continually updated after each step to include samples of the new generator.

Performance Metrics. To evaluate the performance of our proposed approach and others, we chose the Area Under the ROC Curve (AUC) metric. Additionally, in order to obtain a more complete picture of our performance, we provided the Relative Error Reduction (RER) with respect to the second best performing method, reported as a percentage. This metric is calculated as follows:

$$\text{RER} = 100 \times \frac{\text{AUC}_N - \text{AUC}_R}{1 - \text{AUC}_R}, \tag{3}$$

where AUC_R is the AUC of the referencing method, and AUC_N is the AUC of the method being compared against.

Competing Methods. We compared our method against multiple approaches, including two naive strategies: Baseline, where the baseline detector is never updated, and

Method	Synthetic Image Generators																			Avg. ± Std.
	SD1.4	GLD	MJ	DLM	TT	SD2.1	CIPS	BG	VQD	DIG	SG3	GF	DL2	LD	EG3	PG	SD1.1	DDG	DDP	
Baseline	0.89	0.98	0.85	0.86	0.81	0.89	0.76	0.77	0.85	0.76	0.98	0.73	0.97	0.81	0.98	0.89	0.87	0.67	0.78	0.85 ± .090
Fine-Tune	0.65	0.83	0.65	0.67	0.64	0.69	0.85	0.75	0.88	0.88	0.90	0.92	0.79	0.68	0.89	0.76	0.74	0.87	0.84	0.78 ± .098
LWF [36]	0.91	0.98	0.92	0.82	0.85	0.92	0.85	0.88	0.96	0.94	0.99	0.95	0.94	0.72	0.97	0.92	0.91	0.95	0.94	0.91 ± .066
ER [53]	0.94	0.99	0.97	0.96	0.96	0.94	0.99	0.91	0.98	0.98	0.99	0.99	0.99	0.90	1.00	0.96	0.94	0.99	0.99	0.97 ± .030
DER++ [10]	0.94	0.99	0.96	0.95	0.95	0.93	0.98	0.89	0.97	0.97	0.99	0.98	0.99	0.90	1.00	0.96	0.93	0.99	0.98	0.96 ± .031
UDIL [56]	0.94	1.00	0.97	0.97	0.96	0.94	0.99	0.92	0.98	0.98	0.99	0.99	0.99	0.91	1.00	0.97	0.93	0.99	0.99	0.97 ± .028
ICaRL [52]	0.88	0.96	0.91	0.91	0.89	0.91	0.96	0.87	0.94	0.95	0.91	0.95	0.95	0.86	0.97	0.89	0.88	0.93	0.96	0.92 ± .034
MT-SC [40]	0.91	0.99	0.93	0.93	0.92	0.92	0.98	0.90	0.96	0.96	0.97	0.98	0.97	0.88	0.99	0.94	0.91	0.98	0.98	0.95 ± .034
MT-MC [40]	0.92	1.00	0.95	0.95	0.94	0.93	0.99	0.9.0	0.98	0.98	0.99	0.99	0.99	0.88	0.99	0.96	0.92	0.99	0.99	0.96 ± .036
Ours	**0.99**	**1.00**	**0.99**	**0.99**	**0.98**	**0.98**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**1.00**	**0.96**	**1.00**	**0.99**	**0.99**	**1.00**	**0.99**	**0.99 ± .010**

Table 2. Detection performance of each method, measured in terms of AUC, when adapting to a single new generator. The "baseline" detector has only seen data from the baseline detector dataset and all approaches started from the same baseline detector.

Fine-tuning, where the baseline detector is updated exclusively with data from new generators. Additionally, we benchmarked our approach against seven well-established continual learning strategies: Learning Without Forgetting (LWF) [36], Experience Replay (ER) [53], Dark Experience Replay (DER++) [10], Unified Domain Incremental Learning (UDIL) [56], Incremental Classifier and Representation Learning (ICaRL) [52], Multi-Task Single-Classifier (MT-SC) [40], and Multi-Task Multi-Classifier (MT-MC) [40]. We benchmarked ICaRL, MT-SC, and MT-MC using our own implementations, while to benchmark the rest of the methods, we utilized the code provided by [56].

5.2. Adapting to One New Generator

In this experiment, we evaluated E3's ability to detect synthetic images from one new generator. This is important because in reality we do not know beforehand which generator we need to adapt our detector to. As a result, we repeated this experiment for all 19 generators in our dataset. We then compared our method's performance to other techniques used to update the baseline detector. We note that all approaches started with the same baseline detector network.

We present the results of these experiments in Tab. 2. In this table, each column represents the performance of different algorithms adapting the baseline detector to one specific generator. The final column calculates the average AUC and standard deviation over all generators.

This experiment's results in Tab. 2 show that our method achieves the best detection performance across all possible new generators with an average AUC of 0.99. Compared to the second best method, UDIL, whose an average AUC is 0.97, we obtained a significant relative error reduction of 66%. This result shows that our approach is better at adapting to any one new generator than other approaches.

Additionally, we note that other competing methods' performance has standard deviations of about 3 to 10 times larger than ours. This result can be further examined by looking at certain occasions when the baseline detector experiences a significant performance drop detecting a specific unseen generator. When this happens, other approaches also tend to exhibit a similar drop in performance. For example, all competing methods have a significant drop adapting to Latent Diffusion (LD) and BigGAN (BG). Similarly, LWF experiences a performance drop on DLM, DER++, MT-SC and MT-MC on SD1.1, SD1.4 & SD2.1, and ICaRL on SD1.1 & SD1.4. This result is not surprising, however, because the baseline detector, which was trained on GAN, had to be adapted to detect diffusion models' images. This aligns with findings in prior work, which showed that synthetic image detectors "cannot reliably detect images that present artifacts significantly different from those seen during training." [13]

In contrast, our approach displays strong and stable performance irrespective of which generator was added. This is likely because our approach does not need to rely heavily on learning a single embedding space to capture traces of both existing and the new generators.

5.3. Adapting to Multiple Emerging Generators

In this experiment, we tested E3's ability to adapt to multiple sequentially emerging generators. We conducted this evaluation to mimic real world scenarios in which a detector needs to be able to adapt to detect synthetic images from each and every newly emerging generator. We then compared our method's performance to other dedicated continual learning techniques. After adapting to a new generator, we measured E3's and competing methods' performance in terms of average AUC over the current and the previous generators. We repeated this process until we exhausted all 19 emerging generators in our dataset.

We present the results of this experiment in Tab. 3. These results show that our approach achieved the best performance irrespective of the number of new generators added. Furthermore, we observe that after continually adapting to 19 different generators, our method obtained an average AUC of 0.97, a reduction of only 0.02 when compared to the initial performance of 0.99 average AUC. This result is a significant improvement over the second best method,

Method	Synthetic Image Generators																		
	SD1.4	GLD	MJ	DLM	TT	SD2.1	CIPS	BG	VQD	DIG	SG3	GF	DL2	LD	EG3	PG	SD1	DDG	DDP
Baseline	0.89	0.91	0.86	0.83	0.80	0.79	0.76	0.73	0.73	0.71	0.73	0.71	0.73	0.72	0.74	0.74	0.74	0.72	0.71
Fine-Tune	0.65	0.79	0.73	0.81	0.83	0.83	0.80	0.84	0.82	0.82	0.85	0.85	0.73	0.79	0.81	0.70	0.76	0.83	0.74
LWF [36]	0.90	0.91	0.89	0.89	0.88	0.89	0.86	0.85	0.85	0.85	0.82	0.82	0.72	0.80	0.82	0.81	0.80	0.83	0.79
ER [53]	0.94	0.96	0.96	0.96	0.95	0.95	0.95	0.91	0.93	0.93	0.94	0.94	0.93	0.91	0.93	0.90	0.90	0.93	0.93
DER++ [10]	0.96	0.96	0.92	0.91	0.88	0.89	0.86	0.84	0.83	0.82	0.82	0.82	0.79	0.81	0.82	0.82	0.82	0.82	0.80
UDIL [56]	0.95	0.97	0.96	0.96	0.96	0.95	0.96	0.93	0.94	0.94	0.94	0.94	0.94	0.93	0.94	0.92	0.92	0.93	0.93
ICaRL [52]	0.91	0.93	0.92	0.91	0.90	0.92	0.91	0.89	0.90	0.91	0.90	0.91	0.91	0.87	0.90	0.89	0.90	0.90	0.90
MT-SC [40]	0.87	0.85	0.77	0.76	0.75	0.74	0.76	0.69	0.73	0.74	0.74	0.73	0.74	0.70	0.69	0.70	0.70	0.76	0.74
MT-MC [40]	0.92	0.96	0.94	0.94	0.93	0.91	0.92	0.88	0.89	0.90	0.91	0.92	0.91	0.86	0.90	0.87	0.87	0.91	0.90
Ours	**0.99**	**0.99**	**0.99**	**0.99**	**0.99**	**0.98**	**0.98**	**0.98**	**0.98**	**0.98**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**	**0.97**

Table 3. Detection performance of each method, measured in terms of AUC, when sequentially adapting to a series of new generators. The "baseline" detector has only seen data from the baseline detector dataset and all approaches started from the same baseline detector.

Figure 4. Our method shows slight performance decline when adapting to 19 generators with training image counts reduced from 500 to 50 per generator.

Architecture	Accuracy		AUC		AUC RER
	Ours	UDIL	Ours	UDIL	
MISLnet [6]	**0.922**	0.860	**0.970**	0.932	55.9%
ResNet-50 [20]	**0.817**	0.790	**0.901**	0.890	10.0%
DenseNet [23]	**0.810**	0.800	**0.890**	0.869	16.0%
SR-Net [7]	**0.971**	0.964	**0.996**	0.994	33.3%
Average	**0.880**	0.853	**0.939**	0.921	22.8%

Table 4. The performance of E3 versus UDIL using different basline architectures. AUC-RER is also provided.

UDIL, with 0.93 average AUC. Additionally, it can be seen from Tab. 3 that as the number of new generators introduced increases, the more degraded the performance of some competing approaches becomes (*i.e.* LWF, DER++, MT-SC). In contrast, our method retained strong performance throughout our experiment, with minimal reduction in performance.

Additionally, we note that other competing methods all experience difficulties detecting new synthetic images from certain generators. For example, LWF & ER experienced significant performance drop when adapting to DALL-E 2 (DL2), similarly, UDIL's performance dropped adapting to BigGAN (BG), and ICaRL, MT-SC, MT-MC have substantial difficulties adapting to Latent Diffusion (LD). However, our proposed approach does not experience such issues. This suggests that by fusing the embedding spaces of all past and current expert embedders, we can gracefully adapt to any new generator while avoid having significant performance degradation as the number of generator being added increases.

5.4. Effects of New Generator's Training Data Size

In this experiment, we examined the impact of different training data size from each new generator on our proposed approach. This is important because in real world scenarios, the amount of available data from a new generator is often limited, especially if such generator is a proprietary product. Therefore, to conduct this experiment, we repeated the experiment in Sec 5.3, with different numbers of available data ranging from 50 to 500 data points. We then reported the average AUC after sequentially adding all 19 generators to our dataset.

The results of this experiment are depicted in Fig. 4. These results show that our method's performance experienced minimal reduction in detection performance as the number of available training images decreased. Specifically, we received a very slight drop in performance (0.97 to 0.95 average AUC) while having 5 times less training data, and a small drop in performance (0.97 to 0.94 average AUC) with 10 times less data. Notably, our performance of 0.94 average AUC is still an improvement over UDIL (0.93 AUC), whose training data size is 10 times larger than ours (see Sec. 5.3). This suggests that our approach not only achieves the highest performance in adapting to detect synthetic images from new generators but also requires significantly less data than other methods to attain strong performance.

5.5. Effects of Different Detector Architectures

In this experiment, we examined the effects of different network architectures for the baseline detector. This is to

demonstrate the generality of our approach over the diverse set of detection algorithms in the wild. To do this, we repeated the experiment in Sec. 5.3 with three additional network architectures: ResNet-50 [20], SR-Net [7], and DenseNet [23]. These networks are widely used in prior work for synthetic image detection and other forensic tasks [31, 34, 35, 57, 66, 70]. We then measured accuracy and AUC after adapting to all 19 generators in our dataset. We note that due to space limitation, we will only compare to UDIL, the best performing competitor in both experiments in Secs. 5.2 and 5.3.

We present this experiment's results in Tab. 4. These results show that our proposed approach achieves the best performance irrespective of the underlying detector architecture. In particular, we notably outperform UDIL in terms of both accuracy and AUC when employing MISLnet, ResNet-50, or DenseNet as the network architectures for the baseline detector. Across all architectures, our performance is a 22.8% relative error reduction when compared to UDIL. These results suggest that our proposed method is invariant to different architecture designs and can be applied in a general manner to most, if not all, synthetic image detectors.

6. Discussion

Our method's strong performance can be attributed to its utilization of dedicated embedders for each new generator. These embeddings are specifically learned to capture forensic traces left by their corresponding generators. As a result, our algorithm does not rely solely on a single embedding space, but rather on the union of all embedding spaces from every expert embedder in the ensemble.

However, our approach does come at a cost of increased network size. Despite this, in many cases, the cost of misdetecting AI-generated images, or false-alarming real images as synthetic, is worth the increase in network parameters. Additionally, it is important to note that after our method adapted the baseline detector (based on MISLnet) to 19 different generators, the total number of parameters in our network is 27.8M, a comparable number to a single ResNet-50 model with 23.5M parameters. Furthermore, 25.1M of these parameters are frozen because they come from the ensemble of embedders. In reality, only about 4M parameters in our network are trainable. For perspective, after adding 100 new generators, the trainable portion of our network is only 13.6M parameters.

7. Ablation Results

We conducted an ablation study to assess the impact of different design choices of our Expert Knowledge Fusion Network (EKFN). We present these results in Tab. 5.

Majority Voting. In this setup, we fused the knowledge from each expert embedder using a majority voting strategy. This approach involves each expert embedder producing their own decisions about the input image and the final detection score is decided using a majority vote. As shown in Tab. 5, this approach resulted in a significant drop in both accuracy and AUC. We note that the reason why this version has an AUC of 0.90 with an accuracy of 0.50 is because it produces an uncalibrated decision score, a similar phenomenon observed in Corvi et al. [13].

Setup	Accuracy	AUC
Proposed	**0.93**	**0.97**
Majority Voting	0.50 (-0.43)	0.90 (-0.07)
Knowledge Fusion w/ MLP only	0.91 (-0.02)	0.96 (-0.01)
No Transformer Weighting	0.88 (-0.05)	0.96 (-0.01)
5 Transformer Layers	0.91 (-0.02)	0.97 (-0.00)
10 Transformer Layers	0.91 (-0.02)	0.96 (-0.01)

Table 5. Ablation study of the different design choices of the Expert Knowledge Fusion Network.

Knowledge Fusion with MLP Only. In this experiment, we removed the transformer from the EKFN and evaluated the performance of our approach. Tab. 5 shows that this version experience a performance reduction compared to our proposed method. Hence, the transformer is important to obtain strong performance.

No Transformer Weighting. In this experiment setup, we discarded the weighting between the transformer output and its input. Results in Tab. 5 show that this version achieves worse performance than our proposed method. Hence, this integration approach is important for our method.

Number of Transformer Layers. We tested our approach with the transformer made using only 5 or 10 layers. Tab. 5's results show that this is sub-optimal to our approach in terms of AUC and accuracy.

8. Conclusion

This paper introduces E3, a novel approach for effectively updating synthetic image detectors to accurately detect newly emerging generators with minimal training data. By developing expert embedders tailored to capture traces from each new target generator, our method demonstrates strong adaptability. The proposed expert knowledge fusion network analyzes forensic evidence from all experts, facilitating precise detection decisions. Through extensive experimentation, E3 consistently outperforms competing continual learning approaches, even across various detector architectures and with limited data from new generators.

Acknowledgments. This material is based on research sponsored by DARPA and the Air Force Research Laboratory (AFRL) under agreement number HR0011-20-C-0126 and by the National Science Foundation under Award No. 2320600.

References

[1] Midjourney. https://www.midjourney.com/. Accessed: March 19, 2024. 2, 5

[2] Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14278–14287, 2021. 5

[3] Georgios Batzolis, Jan Stanczuk, Carola-Bibiane Schönlieb, and Christian Etmann. Conditional image generation with score-based diffusion models, 2021. 2

[4] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM workshop on information hiding and multimedia security*, pages 5–10, 2016. 5

[5] Belhassen Bayar and Matthew C. Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *ICASSP*, pages 2152–2156, 2017. 5

[6] Belhassen Bayar and Matthew C. Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018. 5, 7

[7] Mehdi Boroumand, Mo Chen, and Jessica Fridrich. Deep residual network for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 14(5): 1181–1193, 2019. 7, 8

[8] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018. 2, 5

[9] Tim Brooks, Bill Peebles, Connor Holmes, Will DePue, Yufei Guo, Li Jing, David Schnurr, Joe Taylor, Troy Luhman, Eric Luhman, Clarence Ng, Ricky Wang, and Aditya Ramesh. Video generation models as world simulators. 2024. 3

[10] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in neural information processing systems*, 33:15920–15930, 2020. 2, 6, 7

[11] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16123–16133, 2022. 5

[12] Chen Chen, Xinwei Zhao, and Matthew C. Stamm. Mislgan: An anti-forensic camera model falsification framework using a generative adversarial network. In *ICIP*, pages 535–539, 2018. 5

[13] Riccardo Corvi, Davide Cozzolino, Giada Zingarini, Giovanni Poggi, Koki Nagano, and Luisa Verdoliva. On the detection of synthetic images generated by diffusion models. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 2, 3, 6, 8

[14] Boris Dayma, Suraj Patil, Pedro Cuenca, Khalid Saifullah, Tanishq Abraham, Phúc Lê Khac, Luke Melas, and Ritobrata Ghosh. Dall·e mini, 2021. 5

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, pages 8780–8794. Curran Associates, Inc., 2021. 2

[16] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12873–12883, 2021. 5

[17] Shengbang Fang, Tai D Nguyen, and Matthew c Stamm. Open set synthetic image source attribution. In *34th British Machine Vision Conference 2023, BMVC 2023, Aberdeen, UK, November 20-24, 2023*. BMVA, 2023. 2, 5

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[19] 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. 5

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 7, 8

[21] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 2, 5

[22] Jonathan Ho, Chitwan Saharia, William Chan, David J. Fleet, Mohammad Norouzi, and Tim Salimans. Cascaded diffusion models for high fidelity image generation. *Journal of Machine Learning Research*, 23(47):1–33, 2022. 2

[23] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 7, 8

[24] Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *Proceedings of the 38th International Conference on Machine Learning*, pages 4487–4499. PMLR, 2021. 5

[25] Nils Hulzebosch, Sarah Ibrahimi, and Marcel Worring. Detecting cnn-generated facial images in real-world scenarios. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 2

[26] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[27] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stabil-

ity, and variation. In *International Conference on Learning Representations*, 2018. 5

[28] 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. 2, 5

[29] 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. 5

[30] 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. 5

[31] Minha Kim, Shahroz Tariq, and Simon S Woo. Cored: Generalizing fake media detection with continual representation using distillation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 337–346, 2021. 2, 8

[32] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA, 2015. 5

[33] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[34] Sangyup Lee, Shahroz Tariq, Youjin Shin, and Simon S Woo. Detecting handcrafted facial image manipulations and gan-generated facial images using shallow-fakefacenet. *Applied soft computing*, 105:107256, 2021. 8

[35] Weichuang Li, Peisong He, Haoliang Li, Hongxia Wang, and Ruimei Zhang. Detection of gan-generated images by estimating artifact similarity. *IEEE Signal Processing Letters*, 29:862–866, 2022. 8

[36] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2, 6, 7

[37] 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. 5

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018. 5

[39] Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *2019 IEEE conference on multimedia information processing and retrieval (MIPR)*, pages 506–511. IEEE, 2019. 2

[40] Francesco Marra, Cristiano Saltori, Giulia Boato, and Luisa Verdoliva. Incremental learning for the detection and classification of gan-generated images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019. 2, 6, 7

[41] Brandon B May, Kirill Trapeznikov, Shengbang Fang, and Matthew Stamm. Comprehensive dataset of synthetic and manipulated overhead imagery for development and evaluation of forensic tools. In *Proceedings of the 2023 ACM Workshop on Information Hiding and Multimedia Security*, pages 145–150, 2023. 2

[42] Owen Mayer and Matthew C. Stamm. Exposing fake images with forensic similarity graphs. *IEEE Journal of Selected Topics in Signal Processing*, 14(5):1049–1064, 2020. 5

[43] Owen Mayer, Belhassen Bayar, and Matthew C Stamm. Learning unified deep-features for multiple forensic tasks. In *Proceedings of the 6th ACM workshop on information hiding and multimedia security*, pages 79–84, 2018. 2

[44] Owen Mayer, Brian Hosler, and Matthew C Stamm. Open set video camera model verification. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2962–2966. IEEE, 2020. 5

[45] James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995. 2

[46] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, pages 109–165. Elsevier, 1989. 2

[47] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 2

[48] 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*, pages 16784–16804. PMLR, 2022. 5

[49] Md Awsafur Rahman, Bishmoy Paul, Najibul Haque Sarker, Zaber Ibn Abdul Hakim, and Shaikh Anowarul Fattah. Artifact: A large-scale dataset with artificial and factual images for generalizable and robust synthetic image detection. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 2200–2204. IEEE, 2023. 5

[50] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 2

[51] 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. 5

[52] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 2, 6, 7

[53] Matthew Riemer, Ignacio Cases, Robert Ajemian, Miao Liu, Irina Rish, Yuhai Tu, and Gerald Tesauro. Learning to learn without forgetting by maximizing transfer and minimizing interference. *arXiv preprint arXiv:1810.11910*, 2018. 1, 6, 7

[54] 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. 2, 5

[55] Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. In *Advances in Neural Information Processing Systems*, pages 17480–17492. Curran Associates, Inc., 2021. 5

[56] Haizhou Shi and Hao Wang. A unified approach to domain incremental learning with memory: Theory and algorithm. In *Advances in Neural Information Processing Systems*, pages 15027–15059. Curran Associates, Inc., 2023. 1, 2, 6, 7

[57] Brijesh Singh, Arijit Sur, and Pinaki Mitra. Steganalysis of digital images using deep fractal network. *IEEE Transactions on Computational Social Systems*, 8(3):599–606, 2021. 8

[58] Sergey Sinitsa and Ohad Fried. Deep image fingerprint: Towards low budget synthetic image detection and model lineage analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 4067–4076, 2024. 2, 5

[59] Ruben Tolosana, Ruben Vera-Rodriguez, Julian Fierrez, Aythami Morales, and Javier Ortega-Garcia. Deepfakes and beyond: A survey of face manipulation and fake detection. *Information Fusion*, 64:131–148, 2020. 2

[60] Danial Samadi Vahdati and Matthew C Stamm. Detecting gan-generated synthetic images using semantic inconsistencies. *Electronic Imaging*, 35:1–6, 2023. 2

[61] Danial Samadi Vahdati, Tai Duc Nguyen, Aref Azizpour, and Matthew C. Stamm. Beyond deepfake images: Detecting ai-generated videos. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024.

[62] 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*, pages 8695–8704, 2020. 2, 5

[63] Zhendong Wang, Huangjie Zheng, Pengcheng He, Weizhu Chen, and Mingyuan Zhou. Diffusion-gan: Training gans with diffusion. *arXiv preprint arXiv:2206.02262*, 2022. 5

[64] Jianfeng Xiang, Jiaolong Yang, Binbin Huang, and Xin Tong. 3d-aware image generation using 2d diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2383–2393, 2023. 2

[65] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion GANs. In *International Conference on Learning Representations (ICLR)*, 2022. 5

[66] Yassine Yousfi, Jan Butora, Eugene Khvedchenya, and Jessica Fridrich. Imagenet pre-trained cnns for jpeg steganalysis. In *2020 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2020. 8

[67] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[68] Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7556–7566, 2019. 2

[69] Xu Zhang, Svebor Karaman, and Shih-Fu Chang. Detecting and simulating artifacts in gan fake images. In *2019 IEEE international workshop on information forensics and security (WIFS)*, pages 1–6. IEEE, 2019. 2

[70] Junjie Zhao, Junfeng Wu, James Msughter Adeke, Sen Qiao, and Jinwei Wang. Detecting high-resolution adversarial images with few-shot deep learning. *Remote Sensing*, 15(9): 2379, 2023. 8

[71] 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. 2

[72] Mingjian Zhu, Hanting Chen, Qiangyu YAN, Xudong Huang, Guanyu Lin, Wei Li, Zhijun Tu, Hailin Hu, Jie Hu, and Yunhe Wang. Genimage: A million-scale benchmark for detecting ai-generated image. In *Advances in Neural Information Processing Systems*, pages 77771–77782. Curran Associates, Inc., 2023. 2

[73] Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional text image generation with diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14235–14245, 2023. 2