RepMix: Representation Mixing for Robust Attribution of Synthesized Images

Tu Bui[1 [0000−0001−6622−9703]], Ning Yu[2], and John Collomosse[1,3]

[1] University of Surrey t.v.bui@surrey.ac.uk
[2] Salesforce Research ning.yu@salesforce.com
[3] Adobe Research collomos@adobe.com

Abstract. Rapid advances in Generative Adversarial Networks (GANs) raise new challenges for *image attribution*; detecting whether an image is synthetic and, if so, determining which GAN architecture created it. Uniquely, we present a solution to this task capable of 1) matching images invariant to their semantic content; 2) robust to benign transformations (changes in quality, resolution, shape, etc.) commonly encountered as images are re-shared online. In order to formalize our research, a challenging benchmark, Attribution88, is collected for robust and practical image attribution. We then propose RepMix, our GAN fingerprinting technique based on representation mixing and a novel loss. We validate its capability of tracing the provenance of GAN-generated images invariant to the semantic content of the image and also robust to perturbations. We show our approach improves significantly from existing GAN fingerprinting works on both semantic generalization and robustness. Data and code are available at https://github.com/TuBui/image_attribution.

Keywords: GAN Fingerprinting, Image Attribution, Fake Image Detection, Dataset Benchmarking

1 Introduction

Generative imagery is transforming creative practice through intuitive tools that enable controllable and high quality image synthesis. The photo-realism achievable by recent Generative Adversarial Networks (GANs) is often indistinguishable from real imagery [41]; it is difficult for a lay user to tell if an image is synthetic, or to tell images generated by one GAN from those generated by another. Yet, understanding the provenance of visual media has never been more important – to help ensure creative rights, and to mitigate the spread of misinformation due to abuses of GAN technology. In the near future, parameterizable generative imagery may even begin to challenge or replace traditional stock photography. Tools to trace an image to the GAN that created it are urgently needed to ensure the authenticity and proper attribution of images shared online.

Recent work has already shown initial success at detecting synthetic imagery [51,22,50] and attribution of generative imagery [56,2,17] ('GAN fingerprinting') to a GAN source. Particularly, Wang *et al.* [51] suggest that today GANs share some common technical flaws that could be easily distinguished from real images. However, image attribution is generally more challenging than synthesis

detection due to the diversity in GAN classes; also it is inconclusive what sort of fingerprint a GAN model leaves in its output imagery. Existing image attribution methods, despite reporting near-saturated performance, have two setbacks. First, they mostly focus on attributing images to specific GAN models, which is impractical because a single change in training data, training metaparameters (*e.g.* learning rate, optimizer, training iterations ...) or even random seed results in a different GAN model [56]. It would be more practical to attribute synthetic imagery to the underlining GAN architecture rather than specific GAN models. Second, the effects of perturbations on synthetic images are largely underestimated. Current works often experiment with few image transformations such as blurring, JPEG compression, random crop [56,17,51] which does not reflect the real-life perturbations that online imagery is subjected through redistribution. Such perturbations could deteriorate GAN fingerprint which is reported to lay between the medium and high frequency bands in an image [63].

The foremost contribution of this paper is a solid benchmark for image attribution, where a GAN class is represented by several GAN models trained on different semantic datasets, and images are subjected to various sources of perturbations. We then propose a novel method to robustly determine the fakeness of an image, and if so, which GAN architecture was used. Both our benchmark and proposed method address two key limitations of existing approaches:

1. Semantic generalization. Existing GAN fingerprinting methods trained on images of a particular class of object (e.g. faces) typically fail on images of other object classes. This is because prior works focus on attribution to one of several GAN models seen at training time. Uniquely, we address the new problem of attributing images of unseen semantic class to the GAN *architecture* that created them. In doing so, we formalize a new problem (attribution to GAN architecture rather than model), and propose a novel representation mix-up training strategy so as to equip GAN fingerprinting with semantic generalization over unseen models producing images containing unseen object classes.

2. Robustness to benign transformation. Images often undergo non-editorial (benign) transformations, such as quality, resolution, or format change as they are redistributed online [11,40,8]. Existing GAN fingerprinting techniques exploit artifacts in the GAN generated images in the pixel domain [56] or frequency domain [17] that are removed or corrupted via redistribution process, causing attribution to fail. In some cases, GANs are actively trained to introduce such artifacts. We employ a contrastive training strategy to enable our GAN attribution model to discriminate GAN architectures passively, based upon artifacts that are seldom removed via benign transformation upon images.

2 Related work

Generative Adversarial Networks (GANs) [19] have shown outstanding performance in many downstream image synthesis tasks: photo-real blending and in-painting [54], super-resolution [31], facial portrait generation [13], manipulation [43], and texture synthesis [42,55]. GANs have been also applied to

bridge multiple modalities such as geometry [1], audio [48], or sketch [36]. Our work focuses upon unconditional GANs [27,29,28,26,5,6,39] to avoid introducing additional constraints when producing synthetic images.

Content provenance explores the attribution of media to a trusted source (*e.g.* a database or blockchain [10,9]). Image provenance systems typically rely upon embedded metadata [11,3], watermarking [21,14,44,4] or perceptual hashing [64,34,32,12] to perform visual search robust to the kinds of non-editorial transformation encountered online. Some methods are trained to fail in the presence of digital manipulation [40], whilst others are explicitly trained to match such content and highlight any manipulation [8,7] between the query and matched original. Regardless of applications, robustness and generalization are crucial for content provenance. This is usually addressed via data manipulation (augmentation, data mixing, adversarial attack), implicit representation learning (kernel methods, disentanglement) or explicit learning strategy (ensemble, meta-learning) [49]. In this aspect, RepMix can be considered as a blend of data manipulation (new data is created by mixing existing data points) and representation learning (mixing is performed at feature level).

Digital forensics methods detect and localize image manipulations in the 'blind' *i.e.* without a comparator. The recent 'deep fake detection challenge (DFDC) [16] identified several approaches to detect GAN generated images or image regions, either upon its statistical properties [62,52] or current limitations of GAN methods (*e.g.* human blinking [33]). Our approach contributes most directly to this area, seeking to determine both the presence, and the source of, synthetic imagery. As such we are aligned with recent GAN fingerprinting work. Prior work has explored this problem mainly for facial images, seeking to identify the model [56,17,15] or the architecture and metaparameters [2]. All these works are passive; the practicality of GAN identification is limited by reliance upon fragile signals within an image that are easily destroyed by benign transformation. In order to mitigate this, Yu *et al.* instead propose to modify the GAN training to inject a robust fingerprint into the synthetic image [57,58]. However such approaches require active participation of the GAN creator, and all remain limited to images of a single semantic class. Our fingerprinting approach is passive and robust to both unseen semantic classes and benign transformation, presenting a further step toward practical GAN attribution in the wild.

Most of the above approaches attribute images towards specific GAN models. Although Ding *et al.* [15] attempts to learn an architecture-specific attributor, their work only covers GAN models of different training seeds. Reverse Engineering [2] shows that GAN architecture parameters could be traced even for unseen GAN models, however such fine-grain attributions mean each GAN class is represented by 1 GAN model; and the robustness of the model is still inconclusive. Recently, Girish *et al.* [18] proposes to automatically discover a new GAN cluster for unseen synthesized images, at the cost of iterative evolution of the attributor. While we share a similar goal with [18] in term of architecture-specific attribution, our work scope limits at a closed world problem (*i.e.* attribution on

Fig. 1. Illustrating the construction of Attribution88; a new dataset and benchmark that we contribute for synthetic image detection and attribution.

a fixed set of GAN classes), instead we focus on the generalization on unseen semantic and transformations.

3 The Attribution88 benchmark

The most popular attribution dataset in literature is introduced by Yu *et al.* [56], containing 5 classes (Real + 4 GANs) of a single semantic object. Each GAN class is represented by one GAN model, thus the learned fingerprint could be entangled with semantic features. This is also not an absolute benchmark since only the GAN models are released (rather than the synthesized images) and there is not a fixed train/test split. Existing approaches [56,17,2] report different results on this dataset, even for the common baselines. Additionally, the reported performance is near saturated. It is important to have a fixed and more challenging benchmark for image attribution. The new benchmark should have GAN classes tied to the GAN design/architecture rather than specific GAN models, meaning images from the same GAN class could come from different model training instances. While we could simply vary the training random seeds (*e.g.* [15]) or other metaparameters to create different model instances of a same GAN, we leave the configuration of these parameters of each GAN model fixed to recommended settings for optimal generative quality. Instead, for each GAN class, we train multiple models on different sets of image objects (semantics). The new benchmark is more challenging as attribution must be agnostic to semantics.

We introduce **Attribution88** - a new dataset made of 8 generator classes and 11 semantics (Fig. 1). We start with 5 generator classes (Real, Progan [26], Cramergan [5], Mmdgan [6], Sngan [39]) as proposed in [56], then add 3 most recent classes of the StyleGAN family (Stylegan [28], Stylegan2 [29] and Stylegan3 [27]). For semantics, we choose 10 objects and scenes from the LSUN dataset [53] plus the popular CelebA face dataset [35]. We note that CelebA is structurally aligned and well curated as compared with other semantic sets, but it is widely used for image attribution/synthesis and adds diversity to our benchmark. For each semantic set, we randomly select 100k images for training the

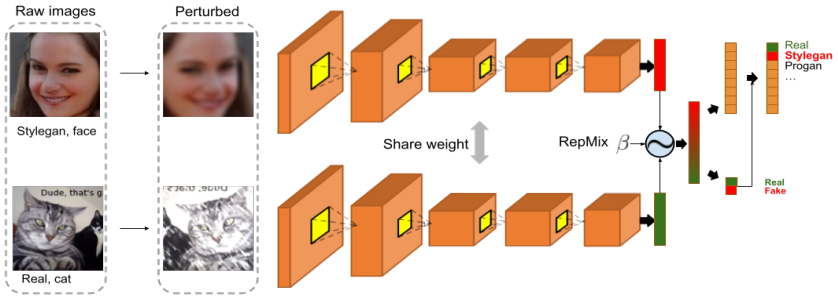


Fig. 2. CNN architecture of our image attribution model. A pair of images is passed though the earlier layers of the CNN model, gets mixed in the RepMix layer before passing to later layers. Training is regulated by a compound loss (see Sec. 4.2).

7 GAN models above, and a disjoint 12k images to serve as *real* images for the attribution task. We use pretrained GAN models when available, otherwise they are trained from scratch using public code, outputting 128×128 images (more details in Sup.Mat).

Next, we generate 100K images per GAN model, resulting in 7.7M synthesized images. Since some images have visible artifacts, we clean them to improve challenge and quality by first extracting perceptual features (of synthesized and real images) using InceptionV3 [61]. We then use K-Means (k=100) to cluster the synthesized images, determine the closest real image for each, and sort the synthesized images according the distance to its closest real image. We then pick top-k (k=120) images in each group, assuming the images closest to a real one have the highest quality. This process helps retain a balance between diversity and realism of images. Overall, we obtain 12K images for each of 8 generator sources (Real plus 7 GANs) and 11 semantics, totally ∼1M images. We further partition each set to 10K training, 1K validation and 1K test images. In our experiments, we expose only 6 semantics (*CelebA Face, Bedroom, Airplane, Classroom, Cow, Church Outdoor*) in training and evaluate on all test images (including 5 unseen semantic classes: *Bridge, Bus, Sheep, Kitchen, Cat*).

Perturbations. Images circulated online are subjected to benign perturbations, from mild transformations such as image resizing to strong ones like noises and enhancement effects. It is important to be robust against these. To this end, we employ ImageNet-C [24], a popular benchmark for evaluating classification robustness. ImageNet-C contains 19 common types of corruption, including various additive noises, blurring and effects, each has 5 different corruption levels. Similar to [24], we only expose 15 transformations to training while the test set is subjected to all possible transformations.

4 Methodology

Synthetic image attribution is a classification problem [56,17,2,22]. In our case, the classes correspond to the GAN architectures from which the images are generated. Unlike semantic classification which relies on discriminative features of

salient objects, the features useful for image attribution are often subtle and may deteriorate due to noise or other image perturbations [56,22]. In order to learn an attribution model robust against (even unseen) semantics and perturbations, we propose RepMix - a simple feature mixing mechanism to synthesize new data from interpolation between existing data points, then learn to predict the mixing ratio. Fig. 2 shows an overview of our approach. Our key technical advancements include (1) the RepMix layer that performs feature mixing between generator classes and (2) the compound loss to predict the mixing ratio for classification.

4.1 Representation Mixing (RepMix) Layer

Suppose we have a training set $\mathcal{X} = \{(\mathbf{x}_i, s_i, y_i), i = 1, 2, ...\}$ where an image $\mathbf{x}_i$ has semantic label $s_i \in \mathcal{S}$ and source label $y_i \in \mathcal{Y}$ (which includes real and a set of GAN source labels). Our goal is to learn mapping $\mathbf{x}_i$ to y_i agnostic to s_i.

Given a training image pair $\mathbf{x}_i$ and $\mathbf{x}_j$ which could either share or differ in source and semantic labels, we first project both images to an intermediate feature space using a nonlinear mapping function f_e:

$$\mathbf{u}_i = f_e(\mathbf{x}_i); \qquad \mathbf{u}_j = f_e(\mathbf{x}_j) \tag{1}$$

where $f_e(.)$ could be the earlier layers of a CNN module. The intermediate representations are input to our RepMix layer:

$$\mathbf{u} = M_\beta(\mathbf{u}_i, \mathbf{u}_j) := \alpha * \mathbf{u}_i + (1 - \alpha) * \mathbf{u}_j \tag{2}$$

with random weight α generated from a certain distribution (here we draw α from a beta distribution[4], $\alpha \sim \text{Beta}(\beta, \beta)$).

Next, the mixed feature map $\mathbf{u}$ is projected into the output via a second mapping function (*e.g.* the later layers of the CNN module):

$$\mathbf{z} = f_l(\mathbf{u}) \in \mathbb{R}^D \tag{3}$$

where D is the output dimension (D=256 in our work). We call $\mathbf{z}$ the embedding space as it directly precedes the objective function (subsec. 4.2).

From an implementation perspective, RepMix is portable and can be inserted anywhere in any existing CNN architecture. Since it has no learnable parameters, it introduces minimal overhead at training time. And since it is used for training only, it can be removed during inference (equivalent to duplicating $\mathbf{x}_i$ to make $\mathbf{x}_j$ with the same semantic and source label). We consider RepMix an extension of MixUp and related work [60,25,59,23] regarding the idea of mixing features. The difference is that existing work performs mixing in the raw image space, while RepMix performs at an intermediate layer. We argue that image attribution relies on subtle artifacts on an image (instead of salient objects) to distinguish real from fake as well as classifying different GAN sources. These useful artifacts could be overwritten or canceled out if images are mixed at pixel level, reducing overall performance (see Sec. 5).

[4] https://en.wikipedia.org/wiki/Beta_distribution

4.2 Compound loss

To attribute an image to its source, existing works [56,17,2] treat the class *real* the same way as other GAN classes prior to modeling classification with a cross-entropy loss. In fact, there is a hierarchical structure in our problem: an image can be either real or fake, if it is fake then it is synthesized from one of the GAN generators. Additionally, real images have a different distribution than GAN synthesized images (see sec. 5.7), therefore should be treated differently. To this end, we proposed a compound loss that takes into account real/fake detection and attribution at the same time.

We first detect the proportion of realness and fakeness scores in the mix up:

$$z_{\text{real}} = \mathbf{W}_{\text{real}}^T \mathbf{z}; \qquad z_{\text{fake}} = \mathbf{W}_{\text{fake}}^T \mathbf{z} \qquad \in \mathbb{R} \tag{4}$$

$$\bar{z}_{\text{real}} = \frac{e^{z_{\text{real}}}}{e^{z_{\text{real}}} + e^{z_{\text{fake}}}}; \qquad \bar{z}_{\text{fake}} = \frac{e^{z_{\text{fake}}}}{e^{z_{\text{real}}} + e^{z_{\text{fake}}}} \tag{5}$$

$$L_{\text{det}} = -\big(\alpha(1-y_i^*) + (1-\alpha)(1-y_j^*)\big)\log(\bar{z}_{\text{real}}) \tag{6}$$

$$- \frac{1}{|\mathcal{Y}|-1}\big(\alpha y_i^* + (1-\alpha) y_j^*\big)\log(\bar{z}_{\text{fake}}) \tag{7}$$

where $\mathbf{W}_{\text{real}}, \mathbf{W}_{\text{fake}} \in \mathbb{R}^{D\times 1}$ are learnable filters, and pseudo label $y_i^* = 0$ if $\mathbf{x}_i$ is real, otherwise 1 (same for y_j^*). This detection loss essentially measures the weighted cross entropy between real and fakeness of each image in the mix. Since there are generally more fake images than real in the training set, the fake term is scaled down by the number of GAN sources accordingly.

The actual attribution task is performed via another cross-entropy loss, taking into account the real/fake-ness score:

$$\mathbf{z}_{\text{attr}} = \mathbf{W}_{\text{attr}}^T \mathbf{z} + \mathbf{b} \in \mathbb{R}^{|\mathcal{Y}|} \tag{8}$$

$$\hat{\mathbf{z}}_{\text{attr}} = \begin{cases} z_{\text{attr}}^{(y_{\text{real}})} * \bar{z}_{\text{real}} \\ z_{\text{attr}}^{(c)} * \bar{z}_{\text{fake}} \qquad \forall c \in \mathcal{Y}\backslash\{y_{\text{real}}\} \end{cases} \tag{9}$$

$$L_{\text{attr}} = -\alpha\log\Big(\frac{e^{\hat{z}_{\text{attr}}^{(y_i)}}}{\sum_k e^{\hat{z}_{\text{attr}}^{(y_k)}}}\Big) - (1-\alpha)\log\Big(\frac{e^{\hat{z}_{\text{attr}}^{(y_j)}}}{\sum_k e^{\hat{z}_{\text{attr}}^{(y_k)}}}\Big) \tag{10}$$

where $\mathbf{W}_{\text{attr}} \in \mathbb{R}^{D\times|\mathcal{Y}|}$ and $\mathbf{b}$ are learnable weight and bias of a fully connected layer to linearly map our embedding $\mathbf{z}$ to the attribution logits. (c) indicates the c-th element of the logit vector. Finally, the total loss is sum of the two above losses $L_{\text{total}} = L_{\text{det}} + L_{\text{attr}}$.

5 Experiments

5.1 Training details

We use the Resnet50 architecture as the backbone for our RepMix model, with the final N-way classification layer replaced by a FC layer producing the 256-D

Table 1. Performance of RepMix and other baselines on a control set that mimics Yu *et al.* [56] settings, and Attribution88 test set. Yu† *et al.* refers to the implementation using the original public code

	1 Sem., Clean			Attribution88		
	Det. Acc. ⇑	Attr. Acc. ⇑	Attr. NMI ⇑	Det. Acc. ⇑	Attr. Acc. ⇑	Attr. NMI ⇑
RepMix	**1.0000**	**0.9994**	**0.9975**	**0.9745**	**0.8207**	**0.6679**
Yu *et al.* [56] (reimp.)	0.9910	0.9838	0.9458	0.9306	0.6784	0.4666
Yu† *et al.* [56]	0.9888	0.9844	0.9455	0.9190	0.6322	0.4028
DCT-CNN [17]	0.9922	0.9838	0.9526	0.9001	0.6447	0.4061
Reverse Eng. [2]	0.9976	0.9960	0.9834	0.8665	0.5637	0.3653
EigenFace [47]	0.8262	0.6538	0.4515	0.7829	0.1515	0.0034
PRNU [38]	0.8544	0.8482	0.7389	0.7845	0.1252	0.0003

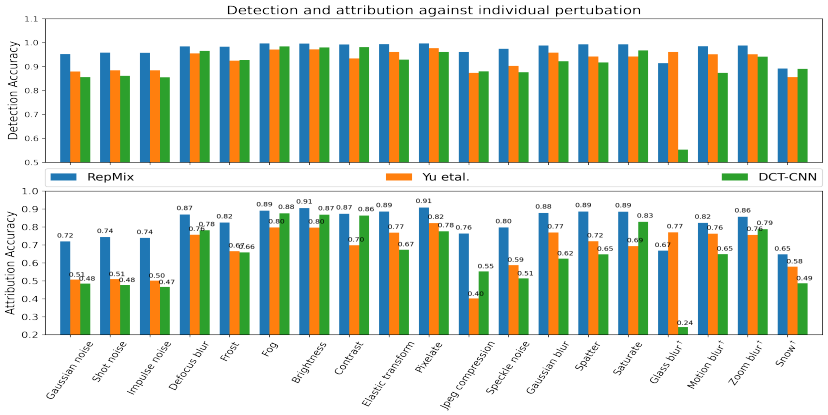


Fig. 3. Detection and attribution performance of our proposed RepMix method vs. two baselines [56,17] in the presence of different benign perturbations of the image.

latent code, followed by our compound loss (subsec. 4.2). Our RepMix layer is inserted at the first FC layer for optimal performance (c.f. subsec. 5.6), with $\beta = 0.4$. Image pairs are randomly sampled from the training data, regardless of generator class and semantics. We do not enforce any constraint on sampling the image pairs to maximize all possible source/semantic combinations. During training we resize images to 256×256 and augment with random crop to 224×224, horizontal flip followed by a random *seen* ImageNet-C perturbation with activation probability of 95%. We train our attribution models for maximum 30 epochs, with Adam optimizer and initial learning rate 1e-4, step decaying with $\gamma = 0.85$ and early stopping based on validation accuracy.

5.2 Baseline comparison

We compare our method with 5 baselines: (i) Yu *et al.* [56] attributes images via a simple fingerprinting CNN model; (ii) DCT-CNN [17] classifies images in the frequency space; (iii) Reverse Engineering [2] models GAN architecture details such as number of layers and loss types to assist attribution; (iv) EigenFace [47] builds an Eigen model for each class and classify an image based on its maximum correlation with each model; (v) PRNU [38] is similar to EigenFace but works

on noise fingerprints of each class instead. The baseline models are trained using public code with the same data augmentation techniques as in the proposed method. We also provide our re-implementation of Yu *et al.*'s approach. More details on the baseline implementation are in the Sup.Mat.

To validate our training of the baselines and the GAN models, we also perform comparison on a replica of Yu *et al.* [56] dataset, denoted as *1 Sem., Clean.* Specifically, we adopt their data cleaning method, use 5 classes (1 real and 4 GANs) as stated in [56] and without any ImageNet-C perturbation. The only difference is that we use our trained GAN models and we apply random crop and horizontal flip as the minimal augmentation during training and test.

Evaluation metrics. We report standard classification accuracy and Normalized Mutual Information (NMI) score [18] that measures the dependence between the prediction and the target. Since *real* is one of the target classes, we are also interested in an auxiliary metric, detection accuracy, which is the proportion of images being correctly classified as *real* or *not-real*.

Tab. 1 compares the performance of RepMix against baselines. The performance on the control set is comparable with existing work [2,17,56], with near-saturated accuracy on the deep learning approaches. Reverse Engineering is the highest scored baseline, next is DCT-CNN [17] which performs slightly better than Yu *et al.* [56]. RepMix achieves perfect detection accuracy and the best attribution accuracy and NMI. However, the baselines underperform on Attribution88. The frequency-based methods (DCT-CNN, Reverse Engineering) under-perform the pixel-based ones (Yu *et al.*). The complexity of our benchmark also causes the shallow methods to either fail completely (PRNU [38]) or just above random prediction (EigenFace [47]). We attribute these changes to the diversity of data (including unseen semantics) and severity of the perturbations. RepMix performs with 4% and 14% higher accuracy than the closest baseline on the detection and attribution scores.

5.3 Robustness against individual perturbation

To analyze the effects of individual perturbation on attribution performance, we evaluate RepMix and the closest competitors, Yu *et al.* [56] and DCT-CNN [17] on Attribution88 with ImageNet-C perturbations applied on test images (Fig. 3). JPEG compression and additive noise hinders the performance most significantly, especially on the two baselines, while other perturbation sources that transform blocks of neighboring pixels but do not replace them (*e.g.* blurring) have less severe effects. DCT-CNN is particularly vulnerable to glass blurring. Performance on seen and unseen perturbations is comparable, indicating generalization of our models when being exposed to a large enough sources of augmentations during training. Additionally, detection performance is more robust than attribution, with detection standard deviation of 2.8% across all perturbations versus 8.0% attribution for RepMix (3.7% vs. 12.1% for Yu *et al.* method; 9.3% vs. 16.8% for DCT-CNN).

Table 2. Attribution errors caused by adversarial attacks on the Attribution88 test set at different levels of max perturbation ϵ. Lower is better

Methods	$\epsilon = 2/255$	$\epsilon = 4/255$	$\epsilon = 8/255$	$\epsilon = 16/255$	$\epsilon = 24/255$	$\epsilon = 32/255$
RepMix	0.1509	0.1952	0.2454	0.3008	0.3333	0.3572
Yu *et al.* [56]	0.2113	0.2709	0.3328	0.3945	0.4303	0.4534
DCT-CNN [17]	0.1545	0.2190	0.2831	0.3375	0.3642	0.3812

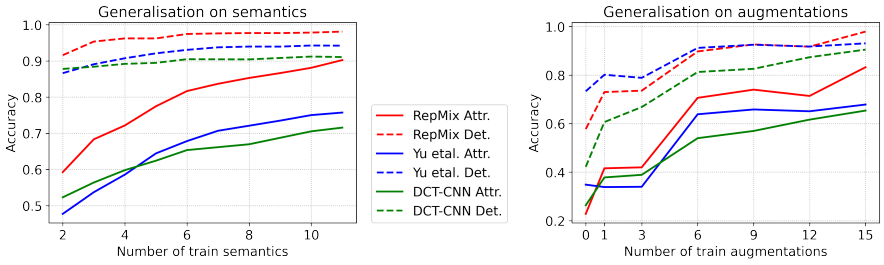


Fig. 4. RepMix performance versus number of (left) semantics and (right) augmentations seen during training.

5.4 Generalization on semantic and perturbation

We evaluate the generalization properties of RepMix, Yu *et al.* and DCT-CNN approaches under the circumstance of limited training data and data augmentation. Fig. 4 (left) depicts detection and attribution performance when the models are exposed to increasing number of semantics during training. We evaluate on the full Attribution88 test set. All 3 detection curves stabilize quite early with RepMix consistently maintaining a 3% gap above other two methods. On attribution performance, the more training data leads to more rewarding results, with RepMix having better generalization capability, scoring from 59% accuracy at 2 seen semantics to 90% when all 11 semantics are exposed during training.

Fig. 4 (right) shows a similar trend as the number of data augmentation methods increases. We fix the number of training semantics at 6, and increase the number of augmentation methods from 0 to 15, and test on a held-out test set of 4 unseen perturbations. The overall trend is a boost in performance when exposing the models to more perturbations during training, with RepMix gaining more generalization power beyond 15 perturbations.

5.5 Robustness against adversarial attacks

Adversarial attacks introduce to an image a subtle layer of noise which is invisible to the naked eye but enough to change the prediction results of a model. Adversarial attacks work by diverting the gradient w.r.t input image toward the most plausible class other than the groundtruth. Repmix enforces a linear inter-class interpolation in the intermediate feature space, therefore is robust to adversarial attacks by design. To verify this, we perform untargeted whitebox attacks on Repmix, Yu *et al.* and DCT-CNN models using the I-FGSM method [20]. We use 20 iterations of I-FGSM for every image in the Attribution88 test

Table 3. Ablation study of RepMix exploring performance at attribution and detection whilst removing different design components, and alternate backbone choices

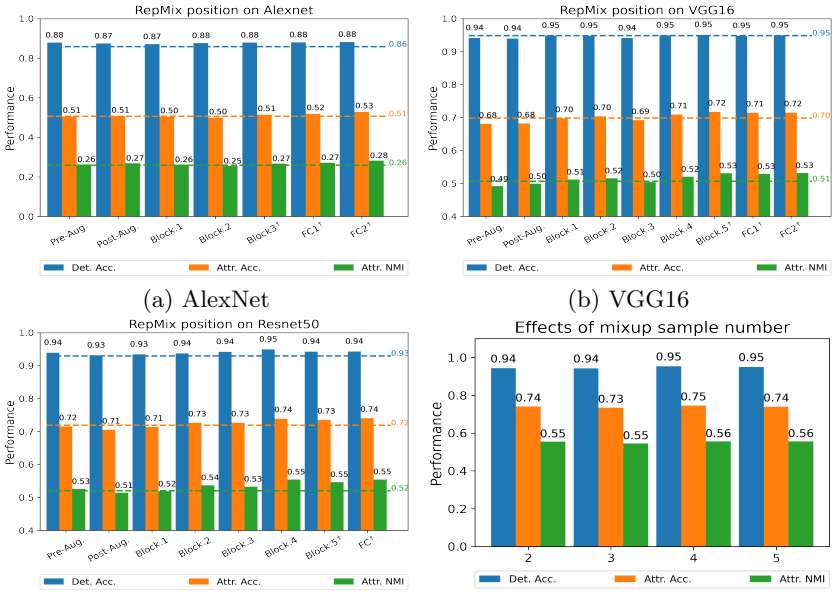
	Detection Acc. ⇑	Attribution Acc. ⇑	Attribution NMI ⇑
All	**0.9426**	**0.7400**	**0.5546**
w/o compound loss	0.9364	0.7204	0.5280
w/o RepMix	0.9296	0.7188	0.5205
w/o RepMix+Compound loss	0.9283	0.7129	0.5167
w/o augmentation	0.7044	0.2762	0.0856
Different backbones			
VGG16	0.9493	0.7150	0.5315
AlexNet	0.8818	0.5280	0.2817

set and stochastic gradient ascend for optimization. Tab. 2 shows the attribution errors, which is the difference in attribution accuracy before and after adversarial attacks, at different noise levels. Although all methods suffer a performance drop and the severity is higher at higher noise tolerant levels (*i.e.* ϵ), RepMix is more robust than the other two approaches. At max perturbation $\epsilon = 32/255$, RepMix accuracy is 2x higher than Yu *et al.* and DCT-CNN (46.35% vs. 22.49% for Yu *et al.*, and 26.34% for DCT-CNN). Interestingly, DCT-CNN [17] has better resistance than Yu *et al.* [56], probably because an images in frequency spectrum are visually more monotonous and alike than in the pixel domain thus would require more efforts (aka. iterations) from I-FGSM for a successful attack.

5.6 Ablation Study

Tab. 3 shows the performance of RepMix when removing one or several of its components or changing the backbone architecture. Without loss of generality we train and test our ablated models on a subset of Attribution88, with all 8 source classes but 2 semantics during training, and test on 4 semantics (2 seen and 2 unseen). Removing either RepMix layer or compound loss or both results in a drop in performance of all metrics. It can be seen that the compound loss does not benefit only the *real* class (small drop in detection accuracy when removing it), but the whole attribution (2% drop). Finally, removing all ImageNet-C perturbations (leave only random crop and horizontal flip as the data augmentation method) significantly decreases the performance, even causes misleading real/fake detection (detection accuracy below random guess). We also replace Resnet50 with AlexNet [30] and VGG16 [46]. AlexNet leads to a significant performance drop, with NMI score reduced by a half. VGG16 has comparable detection accuracy, but 2.5% lower attribution score. More backbone experiments can be found on Sup.Mat.

RepMix position. We experiment with different positions of the RepMix layer in Resnet50, VGG16, and AlexNet. RepMix can be applied to input images at pixel level (equivalent to MixUp [60]), before data augmentation (Pre-Aug.) or after it (Post-Aug.). Within the CNN layers, we insert RepMix after every pooling or FC layer. Fig. 5 shows a similar trend across the three networks. Mixing images at pixel level does not improve performance; meaningful subtle



(a) AlexNet (b) VGG16

(c) Resnet50 (d) Mixup samples on Resnet50

Fig. 5. Effect of RepMix on different layers of (a) AlexNet, (b) VGG16 and (c) Resnet50. Dashed lines refer to baselines without mixing. † indicates the mixing is performed on 1-D feature map (either after Global Average Pooling or FC layer). (d) - The number of mix-up samples have marginal effect on performance of Resnet50.

artifacts are lost. Post-Aug mixing has the worst score since the image is exposed to double corruption. RepMix is more beneficial at the later layers of the networks, benefiting less on 2D feature maps and more on global representation (FC features). This can be seen from Fig. 6, where the attention heatmap covers larger areas. In Fig. 8, semantic clusters appear even at the embedding layer. However, the GAN classification loss ensures semantic features are weaker at the later layers while the GAN class signal is stronger. Thus, mixing representations at later layers is more beneficial.

Number of mixup samples. We test with increasing number of samples to be mixed in RepMix layers. The beta distribution now becomes the Dirichlet distribution to accommodate more than two samples in a mixing group. Fig. 5 (d) shows that increasing number of mixing samples has marginal boost in performance, with 1% improvement at 4 mixing samples at most.

5.7 Further analysis

Real versus other classes. We observe that the detection of real images is fairly robust to training data and perturbations and across various ablation settings (c.f. Sec. 5.2-5.6. This interesting behavior is further demonstrated in Fig. 7, where class real has the highest score and also appears the most consistent across the seen/unseen semantics and perturbations.

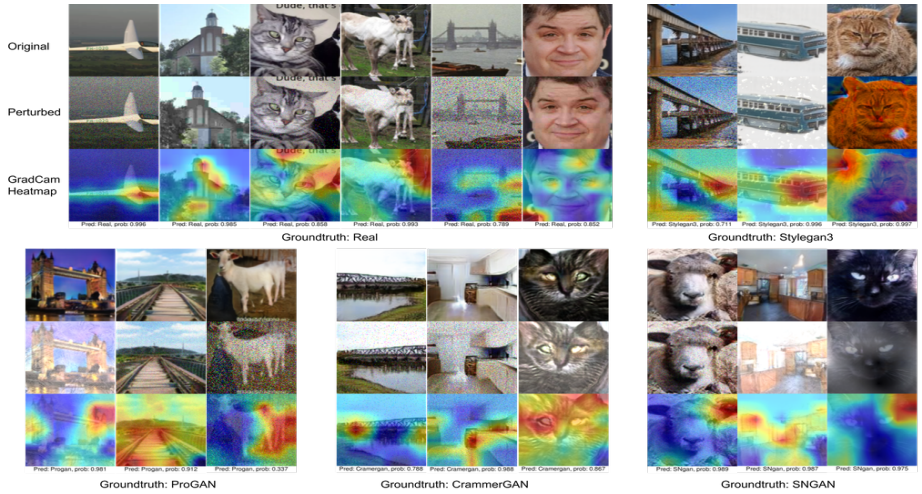


Fig. 6. GradCAM visualization on unseen-semantic test images showing the visual artifacts contributing most signficantly to the GAN classification decision.

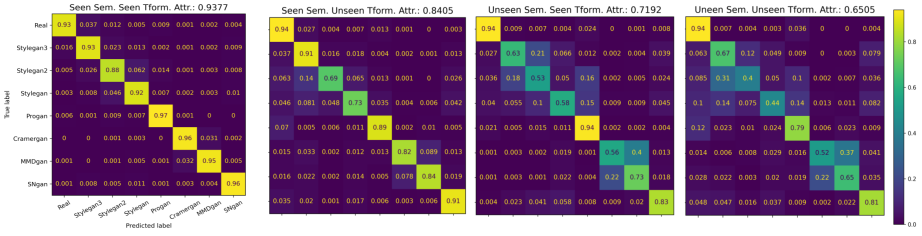


Fig. 7. Confusion matrix of RepMix on seen/unseen semantic classes and on seen/unseen classes of image transformation applied to the test images.

To understand this behavior, we visualize the image regions that contribute the most to the prediction of our model using GradCAM [45]. Fig. 6 shows examples of GradCAM heatmaps for several images of *real* and other GAN classes, from both seen and unseen semantics as well as perturbations. For GAN classes, the heatmaps tend to highlight the edge regions which are often more resilient to perturbation attacks. For real images, GradCAM heatmap also focuses on background objects. We therefore reason that real images have a different distribution from synthesized images particularly because they have vivid background, which often attracts the attention of our attribution model.

t-SNE visualization. We visualize the embedding space $\mathbf{z}$ of RepMix computed on the Attribution88 test set using t-SNE [37] 2D projection, and compare it with Yu *et al*. approach. Fig. 8 shows RepMix has better class separation and semantic fusion than Yu *et al*. Nevertheless, both approaches have a mixed region in the middle of the t-SNE plots where classes are not well separated, which illustrates the challenge of the Attribution88 benchmark.

Limitations. Fig. 9 shows examples where RepMix fails, often due to excessive perturbation that distort finer details of an image, narrowing the gap

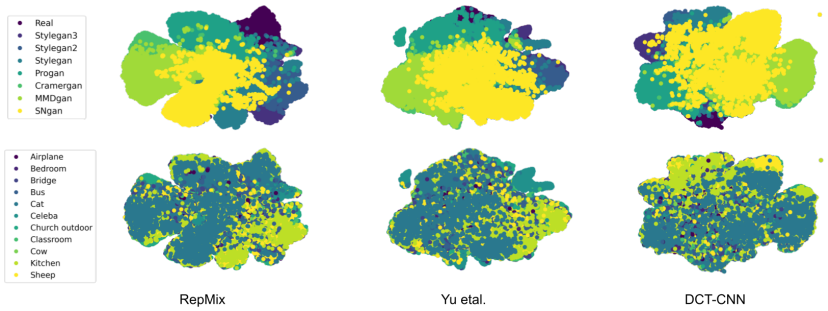


Fig. 8. t-SNE visualization of Attribution88 test set using features extracted from RepMix (left) or Yu *et al.* (middle) and DCT-CNN approach.

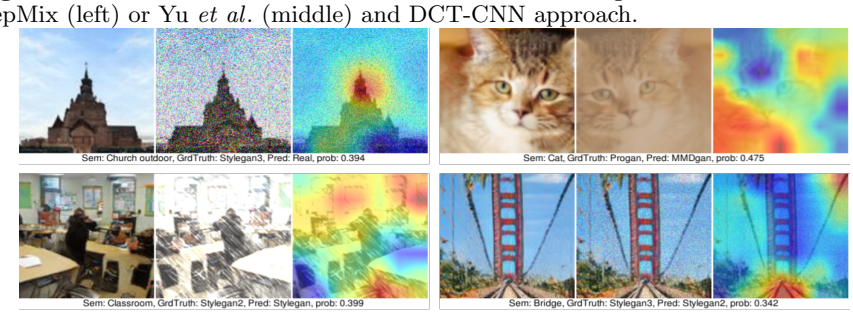


Fig. 9. Examples of attribution failure. For each inset, left: raw image, middle: image after perturbation, right: GradCAM heatmap justifying its (wrong) prediction.

between real/synthesis and between different GAN classes. Another case shown is mis-classification between the three StyleGAN due to architectural similarity.

6 Conclusion

We introduce a challenging image attribution benchmark, Attribution88, for detecting and tracing images to the originating GAN architecture, rather than the GAN model. We present a novel GAN fingerprinting technique that introduces strong zero-shot generalization to unseen semantic classes and unseen transformations, in contrast to prior work that generalizes poorly beyond a single class (*e.g.* faces) even if trained with sight of those classes [56,17]. We demonstrate detection accuracy of 97% and attribution accuracy of 82% on this new benchmark, without introducing any change to the GAN training process (per [58]). Our method is particularly robust to detecting real images, by exploiting an unique feature that current GAN methods have not been able to fabricated. Future work could scale our experiments to even broader classes of GAN including conditional GAN frameworks, although we do not believe such experiments necessary to demonstrate the value of benchmark or contrastive training and mix-up strategy in enabling class generalization for GAN attribution.

Acknowledgement - This work was supported by EPSRC DECaDE Grant Ref EP/T022485/1.

References

1. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: Proc. ICCV. pp. 4561–4569 (2019) 3
2. Asnani, V., Yin, X., Hassner, T., Liu, X.: Reverse engineering of generative models: Inferring model hyperparameters from generated images. arXiv preprint arXiv:2106.07873 (2021) 1, 3, 4, 5, 7, 8, 9
3. Aythora, J., Burke-Agüero, R., Chamayou, A., Clebsch, S., Costa, M., Deutscher, J., Earnshaw, N., Ellis, L., England, P., Fournet, C., et al.: Multi-stakeholder media provenance management to counter synthetic media risks in news publishing. In: Proc. Intl. Broadcasting Convention (IBC) (2020) 3
4. Baba, S., Krekor, L., Arif, T., Shaaban, Z.: Watermarking scheme for copyright protection of digital images. IJCSNS **9**(4) (2019) 3
5. Bellemare, M.G., Danihelka, I., Dabney, W., Mohamed, S., Lakshminarayanan, B., Hoyer, S., Munos, R.: The cramer distance as a solution to biased wasserstein gradients. arXiv preprint arXiv:1705.10743 (2017) 3, 4
6. Bińkowski, M., Sutherland, D.J., Arbel, M., Gretton, A.: Demystifying mmd gans. In: Proc. ICLR (2018) 3, 4
7. Black, A., Bui, T., Jenni, S., Swaminathan, V., Collomosse, J.: Vpn: Video provenance network for robust content attribution. In: Proc. CVMP. pp. 1–10 (2021) 3
8. Black, A., Bui, T., Jin, H., Swaminathan, V., Collomosse, J.: Deep image comparator: Learning to visualize editorial change. In: Proc. CVPR. pp. 972–980 (2021) 2, 3
9. Bui, T., Cooper, D., Collomosse, J., Bell, M., Green, A., Sheridan, J., Higgins, J., Das, A., Keller, J., Thereaux, O., et al.: Archangel: Tamper-proofing video archives using temporal content hashes on the blockchain. In: Proc. CVPR WS. pp. 0–0 (2019) 3
10. Bui, T., Cooper, D., Collomosse, J., Bell, M., Green, A., Sheridan, J., Higgins, J., Das, A., Keller, J.R., Thereaux, O.: Tamper-proofing video with hierarchical attention autoencoder hashing on blockchain. IEEE Trans. Multimedia **22**(11), 2858–2872 (2020) 3
11. (CAI), C.A.I.: Setting the standard for content attribution. Tech. rep., Adobe Inc. (2020) 2, 3
12. Cao, Z., Long, M., Wang, J., Yu, P.S.: Hashnet: Deep learning to hash by continuation. In: Proc. CVPR. pp. 5608–5617 (2017) 3
13. Chen, A., Liu, R., Xie, L., Chen, Z., Su, H., Yu, J.: Sofgan: A portrait image generator with dynamic styling. ACM Trans. Graphics (TOG) **41**(1), 1–26 (2022) 2
14. Devi, P., Venkatesan, M., Duraiswamy, K.: A fragile watermarking scheme for image authentication with tamper localization using integer wavelet transform. J. Computer Science **5**(11), 831–837 (2019) 3
15. Ding, Y., Thakur, N., Li, B.: Does a gan leave distinct model-specific fingerprints? Proc. BMVC (2021) 3, 4
16. Dolhansky, B., Bitton, J., Pflaum, B., Lu, J., Howes, R., Wang, M., Ferrer, C.C.: The deepfake detection challenge (DFDC) dataset. CoRR **abs/2006.07397** (2020), http://arxiv.org/abs/2006.07397 3
17. Frank, J., Eisenhofer, T., Schönherr, L., Fischer, A., Kolossa, D., Holz, T.: Leveraging frequency analysis for deep fake image recognition. In: Proc. ICML. pp. 3247–3258. PMLR (2020) 1, 2, 3, 4, 5, 7, 8, 9, 10, 11, 14
18. Girish, S., Suri, S., Rambhatla, S.S., Shrivastava, A.: Towards discovery and attribution of open-world gan generated images. In: Proc. ICCV. pp. 14094–14103 (2021) 3, 9
19. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. NeurIPS **27** (2014) 2

20. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014) 10
21. Hameed, K., Mumtax, A., Gilani, S.: Digital image watermarking in the wavelet transform domain. WASET **13**, 86–89 (2006) 3
22. He, Y., Yu, N., Keuper, M., Fritz, M.: Beyond the spectrum: Detecting deepfakes via re-synthesis. In: Proc. IJCAI-21. pp. 2534–2541. International Joint Conferences on Artificial Intelligence Organization (2021) 1, 5, 6
23. Hendrycks, D., Basart, S., Mu, N., Kadavath, S., Wang, F., Dorundo, E., Desai, R., Zhu, T., Parajuli, S., Guo, M., et al.: The many faces of robustness: A critical analysis of out-of-distribution generalization. In: Proc. ICCV. pp. 8340–8349 (2021) 6
24. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: Proc. ICLR (2018) 5
25. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: Proc. ICLR (2019) 6
26. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: Proc. ICLR (2018) 3, 4
27. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. NeurIPS **34** (2021) 3, 4
28. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proc. CVPR. pp. 4401–4410 (2019) 3, 4
29. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proc. CVPR. pp. 8110–8119 (2020) 3, 4
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NeurIPS **25** (2012) 11
31. Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proc. CVPR. pp. 4681–4690 (2017) 2
32. Li, W., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. In: Proc. IJCAI. pp. 1711–1717 (2016) 3
33. Li, Y., Ching, M.C., Lyu, S.: In ictu oculi: Exposing ai created fake videos by detecting eye blinking. In: Proc. IEEE WIFS (2018) 3
34. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Proc. CVPR. pp. 2064–2072 (2016) 3
35. Liu, Z., Luo, P., Wang, X., Tang, X.: Deep learning face attributes in the wild. In: Proc. ICCV. pp. 3730–3738 (2015) 4
36. Lu, Y., Wu, S., Tai, Y.W., Tang, C.K.: Image generation from sketch constraint using contextual gan. In: Proc. ECCV. pp. 205–220 (2018) 3
37. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008) 13
38. Marra, F., Gragnaniello, D., Verdoliva, L., Poggi, G.: Do gans leave artificial fingerprints? In: Proc. MIPR. pp. 506–511. IEEE (2019) 8, 9
39. Miyato, T., Kataoka, T., Koyama, M., Yoshida, Y.: Spectral normalization for generative adversarial networks. In: International Conference on Learning Representations (2018) 3, 4
40. Nguyen, E., Bui, T., Swaminathan, V., Collomosse, J.: Oscar-net: Object-centric scene graph attention for image attribution. In: Proc. ICCV. pp. 14499–14508 (2021) 2, 3
41. Nightingale, S.J., Farid, H.: Ai-synthesized faces are indistinguishable from real faces and more trustworthy. Proc. National Academy of Sciences **119**(8) (2022) 1
42. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proc. CVPR. pp. 2337–2346 (2019) 2

43. Perarnau, G., Van De Weijer, J., Raducanu, B., Álvarez, J.M.: Invertible conditional gans for image editing. arXiv preprint arXiv:1611.06355 (2016) 2
44. Profrock, D., Schlauweg, M., Muller, E.: Content-based watermarking by geometric wrapping and feature- based image segmentation. In: Proc. SITIS. pp. 572–581 (2006) 3
45. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: Proc. ICCV. pp. 618–626 (2017) 13
46. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Proc. ICLR (2015) 11
47. Sirovich, L., Kirby, M.: Low-dimensional procedure for the characterization of human faces. Josa a **4**(3), 519–524 (1987) 8, 9
48. Wan, C.H., Chuang, S.P., Lee, H.Y.: Towards audio to scene image synthesis using generative adversarial network. In: Proc. ICASSP. pp. 496–500. IEEE (2019) 3
49. Wang, J., Lan, C., Liu, C., Ouyang, Y., Qin, T., Lu, W., Chen, Y., Zeng, W., Yu, P.: Generalizing to unseen domains: A survey on domain generalization. IEEE Trans. Knowledge and Data Engineering (2022) 3
50. Wang, R., Juefei-Xu, F., Ma, L., Xie, X., Huang, Y., Wang, J., Liu, Y.: Fakespotter: a simple yet robust baseline for spotting ai-synthesized fake faces. In: Proc. IJCAI. pp. 3444–3451 (2021) 1
51. Wang, S.Y., Wang, O., Zhang, R., Owens, A., Efros, A.A.: Cnn-generated images are surprisingly easy to spot... for now. In: Proc. CVPR. pp. 8695–8704 (2020) 1, 2
52. Wu, Y., AbdAlmageed, W., Natarajan, P.: Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In: Proc. CVPR. pp. 9543–9552 (2019) 3
53. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015) 4
54. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proc. CVPR. pp. 5505–5514 (2018) 2
55. Yu, N., Barnes, C., Shechtman, E., Amirghodsi, S., Lukac, M.: Texture mixer: A network for controllable synthesis and interpolation of texture. In: Proc. CVPR. pp. 12164–12173 (2019) 2
56. Yu, N., Davis, L.S., Fritz, M.: Attributing fake images to gans: Learning and analyzing gan fingerprints. In: Proc. ICCV. pp. 7556–7566 (2019) 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 14
57. Yu, N., Skripniuk, V., Abdelnabi, S., Fritz, M.: Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In: Proc. ICCV. pp. 14448–14457 (2021) 3
58. Yu, N., Skripniuk, V., Chen, D., Davis, L., Fritz, M.: Responsible disclosure of generative models using scalable fingerprinting (2022) 3, 14
59. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: Proc. ICCV. pp. 6023–6032 (2019) 6
60. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: Proc. ICLR (2018) 6, 11
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proc. CVPR. pp. 586–595 (2018) 5
62. Zhang, X., Sun, Z.H., Karaman, S., Chang, S.: Discovering image manipulation history by pairwise relation and forensics tools. IEEE J. Selected Topics in Signal Processing. **14**(5), 1012–1023 (2020) 3
63. Zhang, X., Karaman, S., Chang, S.F.: Detecting and simulating artifacts in gan fake images. In: IEEE WIFS. pp. 1–6. IEEE (2019) 2

64. Zhu, H., Long, M., Wang, J., Cao, Y.: Deep hashing network for efficient similarity retrieval. In: Proc. AAAI (2016) 3