Cross-GAN Auditing: Unsupervised Identification of Attribute Level Similarities and Differences between Pretrained Generative Models

Matthew L. Olson[1], Shusen Liu[2], Rushil Anirudh[2], Jayaraman J. Thiagarajan[2], Peer-Timo Bremer[2], and Weng-Keen Wong[1]
[1] Oregon State University - EECS, [2]Lawrence Livermore National Laboratory - CASC
{olsomatt,wongwe}@oregonstate.edu, {liu42,anirudh1,jayaramanthi1,bremer5}@llnl.gov

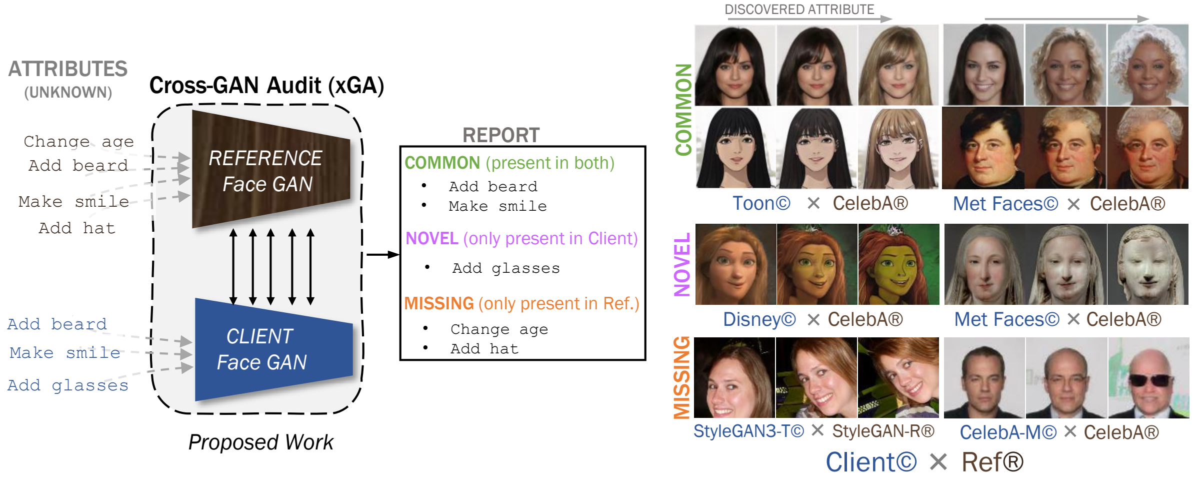

Figure 1. We introduce (**xGA**) an approach for fully unsupervised cross-GAN auditing and validation. Given two pre-trained GANs (Reference & Client), xGA evaluates the client by identifying three types of semantic attributes – (a) Common: those that exist in both models, (b) Novel: those only present in the client and (c) Missing: those that only exist in the reference. On the right, we show results across multiple studies, that among others include notable shifts in distribution between the Reference (CelebA) to Client (Toon, Disney, Met Faces). xGA also lends itself easily to comparing models with different properties on the same dataset as shown on the bottom right for StyleGAN3-T vs. StyleGAN-R. And CelebA-M is a control dataset we create that does not contain glasses, ties and smiles.

Abstract

Generative Adversarial Networks (GANs) are notoriously difficult to train especially for complex distributions and with limited data. This has driven the need for tools to audit trained networks in human intelligible format, for example, to identify biases or ensure fairness. Existing GAN audit tools are restricted to coarse-grained, model-data comparisons based on summary statistics such as FID or recall. In this paper, we propose an alternative approach that compares a newly developed GAN against a prior baseline. To this end, we introduce Cross-GAN Auditing *(xGA) that, given an established "reference" GAN and a newly proposed "client" GAN, jointly identifies intelligible attributes that are either* common *across both GANs,* novel *to the client GAN, or* missing *from the client GAN. This provides both users and model developers an intuitive assessment of similarity and differences between GANs. We introduce novel metrics to evaluate attribute-based GAN auditing approaches and use these metrics to demonstrate quantitatively that xGA outperforms baseline approaches. We also include qualitative results that illustrate the common, novel and missing attributes identified by xGA from GANs trained on a variety of image datasets*[1].

1. Introduction

Generative Adversarial Networks (GANs) [12, 19–21] have become ubiquitous in a range of high impact commercial and scientific applications [5, 7–9, 13]. With this pro-

[1] Source code is available at https://github.com/mattolson93/cross_gan_auditing

lific use comes a growing need for investigative tools that are able to evaluate, characterize and differentiate one GAN model from another, especially since such differences can arise from a wide range of factors – biases in training data, model architectures and hyper parameters used in training etc. In practice, this has been mostly restricted to comparing two or more GAN models against the dataset they were trained on using summary metrics such as Fréchet Inception Distance (FID) [16] and precision/recall [20] scores.

However, in many real world scenarios, different models may not even be trained on the same dataset, thereby making such summary metrics incomparable. More formally, if we define the model comparison problem as one being between a known – and presumably well vetted – *reference* GAN and a newly developed *client* GAN. For example, the reference GANs can correspond to models purchased from public market places such as AWS [2], Azure [3], or GCP [11], or to community-wide standards. Furthermore, there is a critical need for more fine-grained, interpretable, investigative tools in the context of fairness and accountability. Broadly, these class of methods can be studied under the umbrella of AI model *auditing* [1, 6, 32]. Here, the interpretability is used in the context to indicate that the proposed auditing result will involves of human intelligible attributes, rather than summary statistic that do not have explicit association with meaningful semantics.

While auditing classifiers has received much attention in the past [32], GAN auditing is still a relatively new research problem with existing efforts focusing on model-data comparisons, such as identifying how faithfully a GAN recovers the original data distribution [1]. In contrast, we are interested in developing a more general framework that enables a user to visually audit a "client" GAN model with respect the "reference". This framework is expected to support different kinds of auditing tasks: (i) comparing different GAN models trained on the same dataset (e.g. StyleGAN3-Rotation and StyleGAN3-Translate on FFHQ); (ii) comparing models trained on datasets with different biases (e.g., StyleGAN with race imbalance vs StyleGAN with age imbalance); and finally (iii) comparing models trained using datasets that contain challenging distribution shifts (e.g., CelebA vs Toons). Since these tools are primarily intended for human experts and auditors, interpretability is critical. Hence, it is natural to perform auditing in terms of human intelligible attributes. Though there has been encouraging progress in automatically discovering such attributes from a single GAN in the recent years [14, 28, 39, 40, 43] they are not applicable to our setting with multiple GANs.

Proposed work We introduce cross-GAN auditing (xGA), an unsupervised approach for identifying attribute similarities and differences between client GANs and reference models (which could be pre-trained and potentially unrelated). Since the GANs are trained independently, their latent spaces are disparate and encode different attributes, and thus they are not directly comparable. Consequently, discovering attributes is only one part of the solution; we also need to 'align' humanly meaningful and commonly occurring attributes across the individual latent spaces.

Our audit identifies three distinct sets of attributes: (a) common: attributes that exist in both client and reference models; (b) novel: attributes encoded only in the client model; (c) missing: attributes present only in the reference. In order to identify common attributes, xGA exploits the fact that shared attributes should induce similar changes in the resulting images across both the models. On the other hand, to discover novel/missing attributes, xGA leverages the key insight that attribute manipulations unique to one GAN can be viewed as out of distribution (OOD) to the other GAN. Using empirical studies with a variety of StyleGAN models and benchmark datasets, we demonstrate that xGA is effective in providing a fine-grained characterization of generative models.

Contributions (i) We present the first cross-GAN auditing framework that uses an unified, attribute-centric method to automatically discover common, novel, and missing attributes from two or more GANs; (ii) Using an external, robust feature space for optimization, xGA produces high-quality attributes and achieves effective alignment even across challenging distribution shifts; (iii) We introduce novel metrics to evaluate attribute-based GAN auditing approaches; and (iv) We evaluate xGA using StyleGANs trained on CelebA, AFHQ, FFHQ, Toons, Disney and MetFaces, and also provide a suite of controlled experiments to evaluate cross-GAN auditing methods.

2. Related Work

Attribute Discovery Several approaches have been successful in extracting attribute directions in StyleGAN's latent space in the past few years. InterfaceGAN [33] used an external classifier and human annotations to label sampled images in order to build a simple linear model that captures the attribute direction in a GAN's latent space. GANSpace [14] applies PCA to these intermediate representations to find the large factors of variation and then reprojects these directions onto a GAN's latent space. Similarly, SeFa [34] directly captures these directions via matrix factorization of the affine mapping weights in styleGAN, which identify directions of large changes without the need to sample the latent space. An alternative strategy is to directly learn the interpretable directions through a jointly-trained predictive model by assuming that the more predictive variations are more likely to be semantically meaningful [39]– or that using a Hessian penalty [28], or Jacobian [40], in the image space enables learning of directions. LatentCLR [43] used a similar optimization framework, but instead of training a separate predictive model, it leveraged the GAN's internal representation and adopted a contrastive loss [10] for attribute discovery.

Model Auditing With increased awareness of the societal impact of machine learning models, there is an increased interest in characterizing and criticizing model behavior under the broad umbrella of auditing [32, 42]. There has been relatively less work in auditing generative models. For example, [1] introduce a new performance metric for generative models that measures fidelity, diversity, and generalization. Another related work is from Bau et al., [6] who investigate what a GAN cannot generate, whereas our interest is in distinguishing a client GAN from a reference GAN.

Interpretation of Domain Shift Some of the most related work comes from methods that aim for characterizing domain shift [26, 27], but these methods are limited to specific settings: either relying on human intervention [26] or needing a disentangled generator in the input [27]. An indirect way to obtain aligned attributes is via *aligned GANs*– GANs where one is fine-tuned from the other [41], [29]. Here, the attribute direction will be inherent to the children models, eliminating the need to do joint discovery to identify similar attributes. However, obtaining an *aligned GAN* through a separate fine-tuning process for attribute discovery across distributions is neither practical nor feasible.

3. Methods

We approach GAN auditing as performing attribute-level comparison to a reference GAN. For simplicity, we consider the setup where there is a single reference and client model to perform auditing, though xGA can be used even with multiple reference or client models (see experiments). Let us define the reference and client generators as $\mathcal{G}_r : \mathcal{Z}_r \mapsto \mathcal{X}_r$ and $\mathcal{G}_c : \mathcal{Z}_c \mapsto \mathcal{X}_c$ respectively. Here, $\mathcal{Z}_r$ and $\mathcal{Z}_c$ refer to the corresponding latent spaces and the generators are trained to approximate the data distributions $P_r(\mathrm{x})$ and $P_c(\mathrm{x})$. Our formulation encompasses the scenario where $P_r(\mathrm{x}) = P_c(\mathrm{x})$ but the model architectures are different, or the challenging setting of $P_r(\mathrm{x}) \neq P_c(\mathrm{x})$ (e.g., CelebA faces vs Met Faces datasets).

The key idea of xGA is to audit a client model $\mathcal{G}_c$ via attribute (i.e., directions in the latent space) comparison to a reference model, in lieu of computing summary scores (e.g., FID, recall) from the synthesized images. In order to enable a fine-grained, yet interpretable, analysis of GANs, xGA performs automatic discovery and categorization of latent attributes: (i) *common*: attributes that are shared by both the models; (ii) *missing*: attributes that are captured by $\mathcal{G}_r$, but not $\mathcal{G}_c$; (iii) *novel*: attributes that are encoded in $\mathcal{G}_c$ but not observed in the reference model. We express this new categorization scheme in figure 2. Together, these latent attributes can provide a holistic characterization of GANs, while circumventing the need for customized metrics or human-centric analysis.

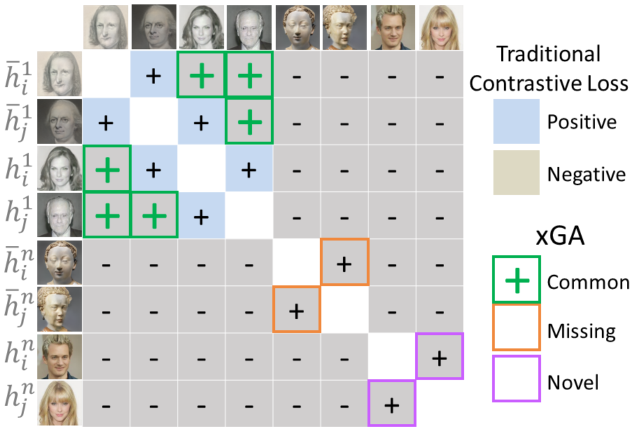

Figure 2. A table showing the proposed xGA modifications to typical contrastive loss with a simple two attribute model.

Latent attributes: Following state-of-the-art approaches such as LatentCLR [43], we define attributes as direction vectors in the latent space of a GAN. For any sample $\mathrm{z} \in \mathcal{Z}_c$ and a direction vector δ_n, we can induce attribute-specific manipulation to the corresponding image as

$$\mathcal{D} : (\mathrm{z}, \delta_n) \rightarrow \mathrm{z} + \alpha \delta_n, \text{ where } \delta_n = \frac{\mathbf{M}_n \mathrm{z}}{\|\mathbf{M}_n \mathrm{z}\|}, \qquad (1)$$

for a scalar α, and a learnable matrix $\mathbf{M}_n$. In other words, we consider the attribute change to be a linear model defined by the learnable direction δ_n. The manipulated image can then be obtained as $\mathcal{G}_c(\mathcal{D}(\mathrm{z}, \delta_n))$, or in shorter notation $\mathcal{G}_c(\mathrm{z}, \delta_n)$. Note that these latent attributes are not prespecified and are discovered as part of the auditing process.

3.1. Common Attribute Discovery

Identifying common attributes between the client and reference GAN models is challenging, since it requires that the latent directions are *aligned*, i.e., the exact same semantic change must be induced in unrelated latent spaces. When distilling from a parent model, i.e., training Toons from Faces, attributes appear to align naturally, even under severe distribution shifts [41]. However, this does not hold true when the two models are trained independently, which requires us to solve the joint problem of identifying the attributes as well as explicitly aligning them.

Formally, for a common attribute, we want the semantic change (in the generated images) induced by manipulating any sample $\mathrm{z} \in \mathcal{Z}_c$ along a direction δ in the client GAN's latent space to match the change in the direction $\bar{\delta}$ from the reference GAN's latent space for any $\bar{\mathrm{z}} \in \mathcal{Z}_r$. In other words, $\mathrm{S}(\mathcal{G}_c(\mathrm{z}, \delta), \mathcal{G}_c(\mathrm{z})) \approx \mathrm{S}(\mathcal{G}_r(\bar{\mathrm{z}}, \bar{\delta}), \mathcal{G}_r(\bar{\mathrm{z}})), \forall\, z \in \mathcal{Z}_c, \bar{\mathrm{z}} \in \mathcal{Z}_r$. Here, S denotes an *oracle* detector (e.g., human subject test) which measures the semantic changes between the original sample and that obtained by manipulating the common attribute.

However, in practice, such a semantic change detector is not accessible and we need to construct a surrogate mechanism to quantify the alignment, i.e.,

$$\min_{\delta_n, \bar{\delta}_n} \mathcal{L}\Big(\mathcal{G}_c(\mathrm{z}, \delta_n), \mathcal{G}_r(\bar{\mathrm{z}}, \bar{\delta}_n)\Big), \forall \mathrm{z} \in \mathcal{Z}_c, \forall \bar{\mathrm{z}} \in \mathcal{Z}_r, \quad (2)$$

for a common attribute pair $(\delta_n, \bar{\delta}_n)$. Any choice of the loss function $\mathcal{L}$ must satisfy two key requirements: (a) identify high-quality, latent directions within each of the latent spaces; (b) encourage cross-GAN alignment such that similar attributes end up being strongly correlated under the loss function. For example, in the case of a single GAN, the LatentCLR [43] approach learns distinct directions using a contrastive objective that defines positive samples as those that have all been perturbed in the same direction, while manipulations in all other directions are considered negative[2]. However, this approach is not suitable for our setting because of a key limitation – alignment requires us to operate in a common feature space so that semantics across the two models are comparable. To address this, we first modify the objective to operate in the latent space of an external, pretrained feature extractor $\mathcal{F}$. In order to support alignment even in the scenario where $P_c(\mathrm{x}) \neq P_r(\mathrm{x})$, we can choose $\mathcal{F}$ that is robust to commonly occurring distributional shifts.

Our approach works on mini-batches of size B samples each, randomly drawn from $\mathcal{Z}_c$ and $\mathcal{Z}_r$ respectively. For the i^{th} sample in a mini-batch from $\mathcal{Z}_c$, let us define the vector h_i^n as the divergence between the output of the GAN before and after perturbing along the n^{th} latent direction, computed in the feature space of $\mathcal{F}$, *i.e.*, $h_i^n = \mathcal{F}(\mathcal{G}_c(\mathrm{z}_i, \delta_n)) - \mathcal{F}(\mathcal{G}_c(\mathrm{z}_i))$. Similarly, we define the divergence $\bar{\mathrm{h}}_j^n = \mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}_j, \bar{\delta}_n)) - \mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}_j))$ for the reference GAN. Next, we measure the semantic similarity between the divergence vectors as $g(\mathrm{h}_i^n, \bar{\mathrm{h}}_j^n) = \exp(\cos(\mathrm{h}_i^n, \bar{\mathrm{h}}_j^n)/\tau)$, where τ is the temperature parameter, and cos refers to cosine similarity. Now, the loss function for inferring a common attribute can be written as

$$\mathcal{L}_{\text{xent}}(\delta_n, \bar{\delta}_n, \lambda_a) = \\ -\log \frac{\sum\limits_{i=1}^{B}\sum\limits_{j \neq i}^{B} g(\mathrm{h}_i^n, \mathrm{h}_j^n) + g(\bar{\mathrm{h}}_i^n, \bar{\mathrm{h}}_j^n) + \lambda_{\mathrm{a}} g(\bar{\mathrm{h}}_i^n, \mathrm{h}_j^n)}{\sum\limits_{\substack{i=1\\j=1}}^{B}\sum\limits_{l=1}^{N} \mathbb{1}_{[l \neq n]}\Big(g(\mathrm{h}_i^l, \mathrm{h}_j^n) + g(\bar{\mathrm{h}}_i^l, \bar{\mathrm{h}}_j^n) + g(\bar{\mathrm{h}}_i^l, \mathrm{h}_j^n)\Big)} \quad (3)$$

Here N denotes the total number of attributes. While the first two terms in the numerator are aimed at identifying distinct attributes from $\mathcal{G}_c$ and $\mathcal{G}_r$, the third term enforces the pair $(\delta_n, \bar{\delta}_n)$ to induce similar semantic change. When the λ_a parameter is set to 0, this optimization reinforces self-similarity of the attributes, without cross-similar semantics. The terms in the denominator are based on the negative pairs (divergences from different latent directions) to enable contrastive training.

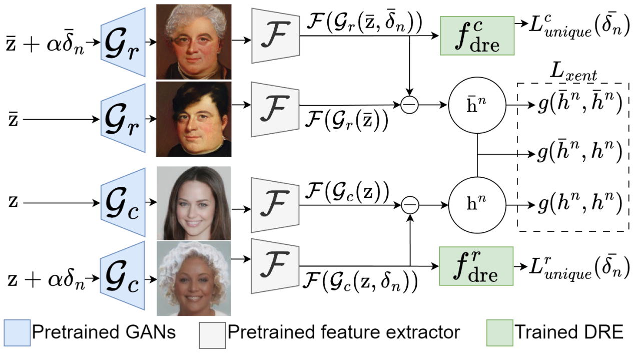
$\bar{\mathrm{z}} + \alpha\bar{\delta}_n$
$\mathcal{G}_r$
$\mathcal{F}$
$\mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}, \bar{\delta}_n))$
f_{dre}^c
$L_{unique}^c(\bar{\delta_n})$
$\bar{\mathrm{z}}$
$\mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}))$
$\bar{\mathrm{h}}^n$
L_{xent}
$g(\bar{h}^n, \bar{h}^n)$
$g(\bar{h}^n, h^n)$
$g(h^n, h^n)$
z
$\mathcal{G}_c$
$\mathcal{F}(\mathcal{G}_c(\mathrm{z}))$
h^n
$\mathrm{z} + \alpha\delta_n$
$\mathcal{F}(\mathcal{G}_c(\mathrm{z}, \delta_n))$
f_{dre}^r
$L_{unique}^r(\bar{\delta_n})$
Pretrained GANs
Pretrained feature extractor
Trained DRE

Figure 3. A diagram of our xGA model. $\mathcal{G}_r$, $\mathcal{G}_c$, and $\mathcal{F}$ are fixed pretrained models. δ_n and $\bar{\delta}_n$ are direction models trained to learn aligned attributes between the two Generators using the features of $\mathcal{F}$, and f_{dre} are regularization models for unique attributes.

3.2. Novel & Missing Attribute Discovery

A key component of our GAN auditing framework is the discovery of interpretable attributes that are unique to or missing from the client GAN's latent space. This allows practitioners to understand the novelty and limitations of a GAN model with respect to a well-established reference GAN. To this end, we exploit the key intuition that images synthesized by manipulating an attribute specific to the client model can manifest as out-of-distribution (OOD) to the reference model (and vice versa).

In order to characterize the OOD nature of such realizations, we define a likelihood score in the feature space from $\mathcal{F}$, which indicates whether a given sample is out of distribution. More specifically, we use the Density Ratio Estimation (DRE) [25, 36] method that seeks to approximate the ratio: $\gamma(\mathrm{x}) = \frac{P(\mathrm{x})}{Q(\mathrm{x})}$ for any sample x. When the ratio is low, it is likely that x is from the distribution Q and hence OOD to P. We choose DRE, specifically the Kullbeck-Liebler Importance Estimation Procedure (KLIEP) [37], over other scoring functions because it is known to be highly effective at accurately detecting outliers [24].

We pre-train two separate DRE models to approximate $\gamma_c(\mathrm{z})$, and $\gamma_r(\bar{\mathrm{z}})$, wherein we treat data from $\mathcal{F}(\mathcal{G}_c(\mathrm{z}))$ as P and $\mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}))$ as Q for the former, and vice versa for the latter. These DRE models are implemented as 2-layer MLP networks, $f_{\text{dre}}^c(.)$, $f_{\text{dre}}^r(.)$, such that

$$\hat{\gamma}_c(\mathrm{z}) = f_{\text{dre}}^c(\mathcal{F}(\mathcal{G}_c(\mathrm{z}))) \text{ and } \hat{\gamma}_r(\bar{\mathrm{z}}) = f_{\text{dre}}^r(\mathcal{F}(\mathcal{G}_r(\bar{\mathrm{z}}))), \quad (4)$$

where $\mathcal{F}$ is the same feature extractor from (3). We pass the

[2] Other single GAN methods could be adapted, but LatentCLR's flexible loss requires less computation without the need to enforce orthogonality at every learning step.

output of the MLPs through a softplus ($\varphi(\mathrm{x}) = \log(1+e^{\mathrm{x}})$) function to ensure non-negativity. As stated previously, we use the KLIEP method to train DRE models. Using Section 4.1 of [24], the KLIEP loss used for training is defined as:

$$\mathcal{L}^{c}_{\text{KLIEP}} = \frac{1}{T_2}\sum_{j=1}^{T_2}\hat{\gamma}_c\left(\bar{\mathrm{z}}_j\right) - \frac{1}{T_1}\sum_{i=1}^{T_1}\ln\hat{\gamma}_c(\mathrm{z}_i), \tag{5}$$

where $\bar{\mathrm{z}}_j$ and z_i are random samples drawn from the latent spaces $\mathcal{Z}_r$ and $\mathcal{Z}_c$ respectively (with T_1 and T_2 total samples). Similarly, we can define the KLIEP loss term for the reference model as:

$$\mathcal{L}^{r}_{\text{KLIEP}} = \frac{1}{T_1}\sum_{i=1}^{T_1}\hat{\gamma}_r\left(\mathrm{z}_i\right) - \frac{1}{T_2}\sum_{j=1}^{T_2}\ln\hat{\gamma}_r(\bar{\mathrm{z}}_j). \tag{6}$$

We also investigated using log-loss functions to train the DRE model, but found it to be consistently inferior to the KLIEP losses (see supplement for details). Finally, we use the pre-trained DRE models from the client and reference GAN data to identify novel and missing attributes, where for a given attribute n in the reference GAN, we can enforce its uniqueness by utilizing the client DRE model to give us $\mathcal{L}^r_{\text{Unique}}(\delta_n) = \hat{\gamma}_{\mathrm{c}}(\mathrm{z}, \delta_n)$ and similarly for the client GAN we can use the reference DRE model $\mathcal{L}^c_{\text{Unique}}(\bar{\delta}_n) = \hat{\gamma}_{\mathrm{r}}(\bar{\mathrm{z}}, \bar{\delta}_n)$ Note, we interpret the novel attributes from the reference GAN as the missing attributes for the client GAN.

3.3. Overall Objective

We now present the overall objective of xGA to identify N_c common, N_n novel and N_m missing attributes simultaneously. Denoting the total number of attributes $N = N_c + \max(N_n, N_m)$, the total loss can be written as:

$$\begin{aligned}\mathcal{L}_{\text{xGA}} = &\sum_{n=1}^{N}\mathcal{L}_{\text{xent}}(\delta_n, \bar{\delta}_n, \mathbb{1}_{[n\leq N_c]}\lambda_a)\\ &+ \lambda_b\left[\sum_{p=N_c+1}^{N_c+N_n}\mathcal{L}^c_{\text{Unique}}(\bar{\delta}_p) + \sum_{q=N_c+1}^{N_c+N_m}\mathcal{L}^r_{\text{Unique}}(\delta_q)\right]\end{aligned}$$

Here, the hyper-parameter λ_b is the penalty for enforcing the attributes between the two latent spaces to be disparate (missing/novel). And we set $g(.,.) = 0$ in L_{xent} if one of the directions vectors does not exist (i.e. when $N_n \neq N_m$).

4. Experiments

In order to systematically evaluate the efficacy of our proposed GAN audit approach, we consider a suite of GAN models trained using several benchmark datasets. In this section, we present both qualitative and quantitative assessments of xGA, and additional results are included in the Supplementary Material.

4.1. Datasets and GAN Models

For most experiments, we used a StyleGANv2 [20] trained on the CelebA [23] dataset as our reference GAN model. This choice is motivated both by its wide-spread use as well as the availability of fine-grained, ground truth attributes for each of the face images in CelebA, and to ensure that this model is fully independent from other client GANs (e.g., ToonGAN is finetuned from FFHQ GAN). In one experiment for the AFHQ dataset, we used a StyleGANv2 trained using only *cat* images from AFHQ as the reference. Also, we considered FFHQ-trained StyleGANv3 [18] and non-StyleGAN architectures such as GANformer [17] for defining the reference (see supplement).

In our empirical study, we constructed a variety of (StyleGANv2) client models and performed xGA: (i) 5 trained with different CelebA subsets constructed by excluding images specific to a chosen attribute (hat, glasses, male, female and beard); (ii) 2 trained with CelebA subsets constructed by excluding images containing any of a chosen set of attributes (beards|hats, smiles|glasses|ties); (iv) 3 transferred GANs for Met Faces, cartoons [4], and Disney images [4] respectively.

4.2. Training Settings

In all our experiments, xGA training is carried out for $10,000$ iterations with random samples drawn from $\mathcal{Z}_c$ and $\mathcal{Z}_r$. We fixed the desired number of attributes to be $N_c = 12$, $N_n = 4$ and $N_m = 4$. Note, this choice was to enable training xGA on a single 15GB Tesla T4 GPU. With the StyleGAN2 models, our optimization takes 4 hours; StyleGAN3 takes 12 hours due to gradient check-pointing. For all latent directions $\{\delta_n\}$ and $\{\bar{\delta}_n\}$, we set $\alpha = 3$ and this controls how far we manipulate each sample in a given direction. In each iteration, the effective batch size was 10, wherein 2 samples were used to construct a positive pair and a subset of 5 directions were randomly chosen for updating (enforced due to memory constraints). We used the Adam [22] optimizer with learning rate 0.001 to update the latent direction parameters. Note, all other model parameters (generators, feature extractor, DRE models) were fixed and never updated. Following common practice with StyleGANs, the attributes are modeled in the style space and the generator's outputs are appropriately resized to fit the size requirements of the chosen feature extractor.

For our optimization objective, we set the hyper-parameter $\lambda_a = 0.1$ in $\mathcal{L}_{\text{xGA}}$. To perform DRE training, we used 2-layer MLPs trained via the Adam optimizer for 1000 iterations to minimize the KLIEP losses specified in (5) and (6). At each step, we constructed batches of 32 samples from both reference and client GANs, and projected them into the feature space of $\mathcal{F}$. Lastly, we set $\lambda_b = 1.0$; we explore tuning this parameter in the supplement, finding it to be relatively insensitive.

4.3. Evaluation: Common Attribute Discovery

We begin by evaluating the ability of xGA in recovering common attributes across reference and client models. As mentioned earlier, for effective alignment, the choice of the feature extractor is critical. More specifically, $\mathcal{F}$ must be sufficiently expressive to uncover aligned attributes from both client and reference models. Furthermore, it is important to handle potential distribution shifts across the datasets used to train the GAN models. Hence, a feature extractor that can be robust to commonly occurring distribution shifts is expected to achieve effective alignment via (3). In fact, we make an interesting observation that performing attribute discovery in such an external feature space leads to improved disentanglement in the inferred latent directions. For all results reported here, we used a robust variant of ResNet that was trained to be adversarially robust to style variations [35]. Please refer to the ablation in Section 4.5 for a comparison of different choices.

Qualitative results In Figure 4, we show several examples of common attributes identified by xGA for different client-reference pairs, we observe that xGA finds non-trivial attributes. For example, the "sketchify" attribute which naturally occurs in Met Faces (a dataset of paintings), is surprisingly encoded even in the reference CelebA GAN (which only consists of photos of people). We also show examples of other interesting attributes such as "orange fur" in the case of dog-GAN × cat-GAN or "blonde hair" in the case of Disney-GAN × CelebA-GAN. These results indicate that our proposed alignment objective, when coupled with a robust feature space, can effectively reveal common semantic directions across the client and reference models. We include several additional examples in the supplement.

Quantitative results To perform more rigorous quantitative comparisons, we setup a controlled experiment using 7 client models corresponding to different CelebA subsets (obtained by excluding images pertinent to specific characteristics). As discussed earlier, we use a standard CelebA StyleGANv2 as the common reference model across all 7 experiments. Next, we introduce a score of merit for common attribute discovery based on the intuition that images perturbed along the same attribute will result in similar prediction changes, when measured through an "oracle" attribute classifier [23].

We first generate a batch of random samples from the latent spaces of client and reference GANs, and manipulate them along a common attribute direction $(\delta_n, \bar{\delta}_n)$ inferred using xGA. In other words, we synthesize pairs of original and attribute-manipulated images from the two GANs and for each pair, we measure the discrepancy in the predictions from an "oracle' attribute classifier. Mathematically, this can be expressed as $\mathrm{a}_i^n = |\mathcal{C}(\mathcal{G}_c(\mathrm{z}_i, \delta_n)) - \mathcal{C}(\mathcal{G}_c(\mathrm{z}_i))|$ and $\bar{\mathrm{a}}_j^n = |\mathcal{C}(\mathcal{G}_r(\bar{\mathrm{z}}_j, \bar{\delta}_n)) - \mathcal{C}(\mathcal{G}_r(\bar{\mathrm{z}}_j))|$, where $\mathcal{C}$ is the attribute classifier trained using the labeled CelebA dataset. Finally, we define an alignment score that compares the expected prediction discrepancy across the two GANs using cosine similarity (higher value indicates alignment).

$$\mathcal{A}_{\text{score}} = \mathbb{E}_n\left[\cos\left(\mathbb{E}_i[\mathrm{a}_i^n], \mathbb{E}_j[\bar{\mathrm{a}}_j^n]\right)\right], \tag{7}$$

where the inner expectations are w.r.t. the batch of samples and the outer expectation is w.r.t. the N_c common attributes.

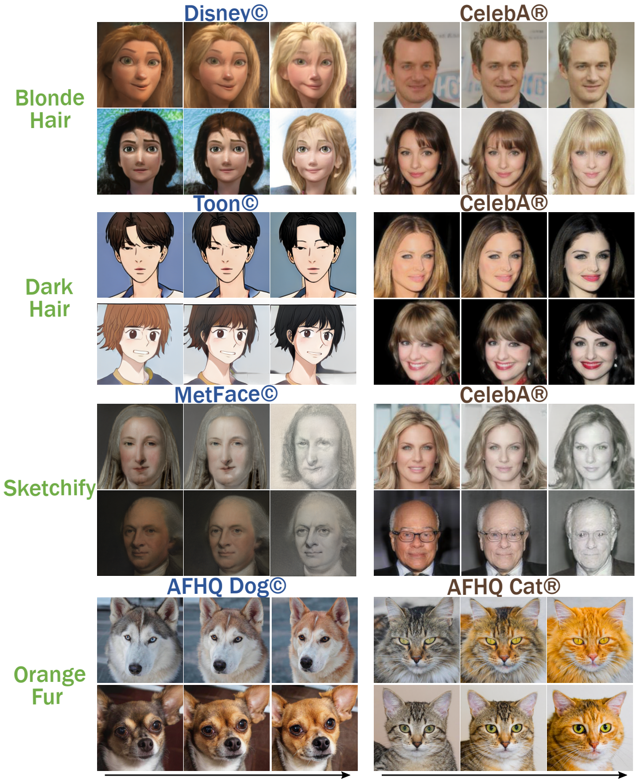

Figure 4. Visualizing common attributes discovered using xGA for different client-reference GAN pairs. For each case, we illustrate one common attribute (indicated by our description in green) with two random samples from the GAN latent space.

We implement 5 baseline approaches that apply state-of-the-art attribute discovery methods to the client and reference GANs (independently), and subsequently peform greedy, post-hoc alignment. In particular, we consider SeFa [34], Voynov [39], LatentCLR [43], Jacobian [40], and Hessian [28] methods for attribute discovery. Given the attributes for the two GANs, we use predictions from the "oracle" attribute classifier to measure the degree of alignment between every pair of directions. For example, the pair with the highest cosine similarity score is selected as the first common attribute. Next, we use the remaining latent directions to greedily pick the next attribute, and this process is repeated until we obtain $N_c = 12$ attributes. We compute the alignment score from (7) for all the methods and

Method	$\mathcal{A}_{\text{score}}$ (↑)	$\mathcal{R}_{\text{score}}$ (↑)
SeFa + G. S	0.382 ± 0.042	0.167 ± 0.165
Voynov + G. S	0.544 ± 0.033	0.254 ± 0.246
LatentCLR + G. S	0.543 ± 0.031	0.297 ± 0.326
Hessian + G. S	0.567 ± 0.065	0.224 ± 0.273
Jacobian + G. S	0.502 ± 0.024	0.233 ± 0.201
xGA	$\mathbf{0.660} \pm 0.147$	$\mathbf{0.411} \pm 0.193$

Table 1. **Common and Missing attribute discovery**. The average alignment scores from the 7 controlled CelebA experiments. We report both the mean and standard deviations (± std) for each case, and "G. S" refers to the greedy strategy we use for alignment.

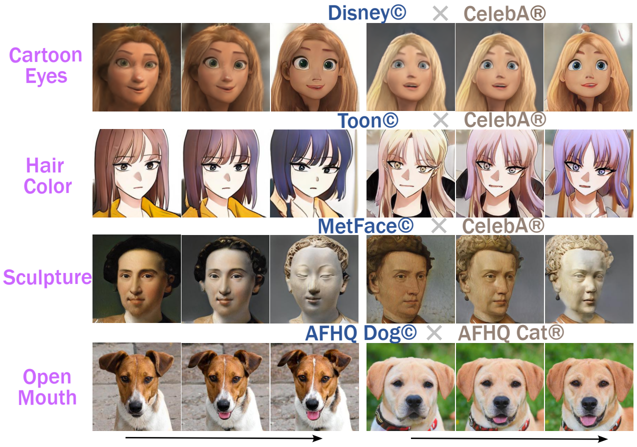

Figure 5. Visualizing novel attributes in different client GANs characterized by challenging distribution shifts with respect to the reference GAN (CelebA or AFHQ Cat-GANs). In each case, we show image manipulation in the attribute direction for two random sample from the latent space.

report results from the 7 controlled experiments in Table 1. Interestingly, we find that, despite using the "oracle" classifier for alignment, the performance of the baseline methods is significantly inferior to xGA. This clearly evidences the efficacy of our optimization strategy.

4.4. Evaluation: Novel/Missing Attribute Discovery

In this section, we study the effectiveness of xGA in discovering novel (only present in the client) and missing (only present in the reference model) attributes.

Qualitative results We first show results for novel attribute discovery for different client GANs in Figure 5. xGA produces highly intuitive results by identifying attributes that are unlikele to occur in the reference GAN. For example, "cartoon eyes" and "sculptures" are found to be unique to Disney and Met Faces GANs, when compared to CelebA. Next, we performed missing attribute discovery from the controlled CelebA experiments, where we know precisely which attribute is not encoded by the client GAN w.r.t the reference (standard CelebA StyleGANv2). As described earlier, the client models are always trained on a subset of data used by the reference model and by design, there are no novel attributes. Figure 6 shows examples for the different missing attributes. We find that xGA successfully reveals each of the missing client attributes, even though the data distributions $P_c(\mathrm{x})$ and $P_r(\mathrm{x})$ are highly similar (except for a specific missing attribute).

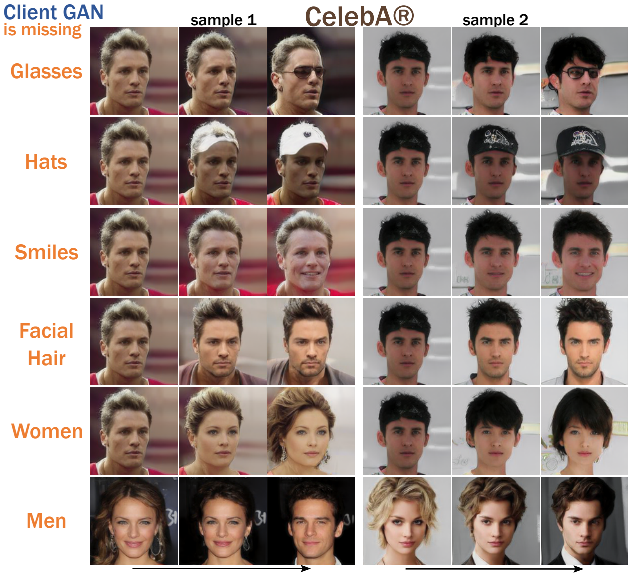

Figure 6. Using multiple clients trained with different subsets of CelebA data (one of the face attributes explicitly dropped), we find that, in all cases, xGA accurately recovers the missing attribute.

Quantitative results To benchmark xGA in missing attribute discovery, we use the 7 controlled CelebA client models and audit with respect to the reference CelebA GAN. We denote the set of attributes (one or more) which are explicitly excluded in each client model by $\mathcal{M}$. In order to evaluate how well xGA identifies the excluded attributes, we introduce a metric based on mean reciprocal rank (MRR) [30,38]. For each of the N_m missing attributes from xGA, we compute the average semantic discrepancy from the "oracle" attribute classifier as,

$$\mathrm{a}^n = \mathbb{E}_i[|\mathcal{C}(\mathcal{G}_c(\mathrm{z}_i, \delta_n)) - \mathcal{C}(\mathcal{G}_c(\mathrm{z}_i))|].$$

Denoting the rank of a missing attribute $m \in \mathcal{M}$ in the difference vector a^n as $\text{rank}(m, \mathrm{a}^n)$, we can define the attribute recovery (for both missing/novelty) score as:

$$\mathcal{R}_{\text{Score}} = \mathbb{E}_m \left[\max_n \left(\frac{1}{\text{rank}(m, \mathrm{a}^n)} \right) \right] \tag{8}$$

In Table 1, we show results for missing attribute discovery based on this score. We observe that xGA significantly outperforms all baselines in identifying the missing attribute across the suite of client GANs.

Method	$\mathcal{H}_{\text{score}}$ (↓)	$\mathcal{D}_{\text{score}}$ (↑)
SeFa [34]	4.006 ± 0.259	1.031 ± 0.077
LatentCLR [43]	2.348 ± 0.203	0.749 ± 0.929
Voynov [39]	2.508 ± 0.069	0.585 ± 0.725
Hessian [28]	2.707 ± 0.145	0.642 ± 0.795
Jacobian [40]	2.675 ± 0.070	0.661 ± 0.826
xGA (ViT)	1.988 ± 0.068	3.072 ± 3.845
xGA (MAE ViT)	2.102 ± 0.035	3.103 ± 3.814
xGA (CLIP ViT)	2.091 ± 0.041	3.135 ± 3.901
xGA (ResNet-50)	1.901 ± 0.060	3.111 ± 3.852
xGA (Clip ResNet-50)	2.033 ± 0.038	3.121 ± 3.863
xGA (advBN ResNet-50)	$\mathbf{1.881} \pm 0.057$	$\mathbf{3.153} \pm 3.904$

Table 2. **Choice of the feature space for attribute discovery**. Using an external feature space is superior to GAN's native style space, in terms of both entropy ($\times 100$) and deviation metrics. In this experiment, we set $\mathcal{G}_r = \mathcal{G}_c$, and aggregate the metrics from the set of controlled CelebA StyleGANs.

4.5. Analysis

In this section, we examine the key components of xGA to understand its behavior better.

Impact of the Choice of $\mathcal{F}$ We start by studying the choice of the external, feature space used to perform attribute discovery. For this analysis, we consider the case where we assume $\mathcal{G}_r = \mathcal{G}_c$, wherein xGA simplifies to the standard setting of attribute discovery with a single GAN model (set $\lambda_b = 0$), such as SeFA and latentCLR. We make an interesting observation that, using a robust latent space leads to improved diversity and disentanglement in the inferred attributes, when compared to the native latent space of StyleGAN. To quantify this behavior we consider two evaluation metrics based on the predictions for a batch of synthesized images $\mathcal{G}_c(\mathrm{z}, \delta_n)$ from the "oracle" attribute classifier. First, for each latent direction δ_n, the average prediction entropy $\mathcal{H}_{\text{score}}$ [23] is defined as:

$$\mathcal{H}_{\text{score}} = \mathbb{E}_n\left[\mathbb{E}_i\left[\texttt{Entropy}(\mathcal{C}(\mathcal{G}_c(\mathrm{z}_i, \delta_n)))\right]\right] \tag{9}$$

Second, the deviation in the predictions across all latent directions $\mathcal{D}_{\text{score}}$ is defined in (10), where K is the total number of attributes in the "oracle" classifier $\mathcal{C}$:

$$\mathcal{D}_{\text{score}} = \sum_{k=1}^{K} \texttt{Variance}\left[\left\{\mathbb{E}_i[\mathcal{C}(\mathcal{G}_c(\mathrm{z}_i, \delta_n)]\right\}_{n=1}^{N}\right]_k \tag{10}$$

When the entropy is low, it indicates that the semantic manipulation is concentrated to a specific attribute, and hence disentangled. On the other hand, when the deviation is high, it is reflective of the high diversity in the inferred latent directions.

For this analysis, we considered the following feature extractors for implementing xGA: (i) vanilla ResNet-50 trained on ImageNet [15]; (ii) robust variant of ResNet-50 trained with advBN [35]; (iii) ResNet-50 trained via CLIP [31]. Table 2 shows the performance of the three feature extractors on attribute discovery with our 7 CelebA GANs trained using different data subsets. Note, we scale all entropy and diversity scores by 100 for ease of readability. We make a striking finding that, in terms both the entropy and deviation scores, performing attribute discovery in an external feature space is significantly superior to carrying out the optimization in the native style space (all baselines). As expected, LatentCLR produces the most disentangled attributes among the baselines, and regardless of the choice of $\mathcal{F}$, xGA leads to significant improvements. More importantly, the key benefit of xGA becomes more apparent from the improvements in the deviation score over the baselines. In the supplement, we include examples for the attributes inferred using all the methods. Finally, among the different choices for $\mathcal{F}$, the advBN ResNet-50 performs the best in terms of both metrics and hence it was used in all our experiments.

5. Discussion

We introduced the first cross-GAN auditing framework, which utilizes a novel optimization technique to jointly infer common, novel and missing attributes for a client GAN w.r.t any reference GAN. Through a large suite of datasets and GAN models, we demonstrate that the proposed method (i) consistently leads to higher quality (disentangled & diverse) attributes, (ii) effectively reveals shared attributes even across challenging distribution shifts, and (iii) accurately identifies the novel/missing attributes in our controlled experiments (i.e., known ground truth).

Limitations First, similar to other optimization-based attribute discovery approaches [39], [43], there is no guarantee that all prevalent factors are captured, though our controlled empirical studies clearly demonstrate the efficacy of xGA over existing approaches. Second, while using an external feature space enhances the performance of attribute discovery, this becomes an additional component that must be tuned. While we found advBN ResNet-50 to be a reasonable choice for a variety of face datasets (and AFHQ), a more systematic solution will expand the utility of our approach to other applications.

Acknowledgements

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The project is directly supported by LDRD 22-ERD-006, and DOE HRRL. The manuscript is reviewed and released under LLNL-PROC-832985.

References

[1] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela van der Schaar. How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models. In *International Conference on Machine Learning*, pages 290–306. PMLR, 2022. 2, 3

[2] AWS. Amazon web services. https://aws.amazon.com/. Accessed: 2023-03-01. 2

[3] Azure. Microsoft azure: Cloud computing services. https://azure.microsoft.com/en-us. Accessed: 2023-03-01. 2

[4] Jihye Back. Fine-tuning stylegan2 for cartoon face generation. *CoRR*, abs/2106.12445, 2021. 5

[5] Viraj Bagal, Rishal Aggarwal, PK Vinod, and U Deva Priyakumar. Molgpt: Molecular generation using a transformer-decoder model. *Journal of Chemical Information and Modeling*, 2021. 1

[6] David Bau, Jun-Yan Zhu, Jonas Wulff, William Peebles, Hendrik Strobelt, Bolei Zhou, and Antonio Torralba. Seeing what a gan cannot generate. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4502–4511, 2019. 2, 3

[7] Brett K Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P Bhavnani, James Brian Byrd, and Casey S Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *Circulation: Cardiovascular Quality and Outcomes*, 12(7):e005122, 2019. 1

[8] Yuemin Bian and Xiang-Qun Xie. Generative chemistry: drug discovery with deep learning generative models. *Journal of Molecular Modeling*, 27(3):1–18, 2021. 1

[9] Jimmy S Chen, Aaron S Coyner, RV Paul Chan, M Elizabeth Hartnett, Darius M Moshfeghi, Leah A Owen, Jayashree Kalpathy-Cramer, Michael F Chiang, and J Peter Campbell. Deepfakes in ophthalmology: Applications and realism of synthetic retinal images from generative adversarial networks. *Ophthalmology Science*, 1(4):100079, 2021. 1

[10] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 3

[11] GCP. Google cloud: Cloud computing services. https://cloud.google.com/. Accessed: 2023-03-01. 2

[12] 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. 1

[13] Harshit Gupta, Thong H Phan, Jaejun Yoo, and Michael Unser. Multi-cryogan: Reconstruction of continuous conformations in cryo-em using generative adversarial networks. In *European Conference on Computer Vision*, pages 429–444. Springer, 2020. 1

[14] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. GANspace: Discovering interpretable GAN controls. *arXiv preprint arXiv:2004.02546*, 2020. 2

[15] 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. 8

[16] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. 2

[17] Drew A Hudson and C. Lawrence Zitnick. Compositional transformers for scene generation. *Advances in Neural Information Processing Systems NeurIPS 2021*, 2021. 5

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 5

[19] 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, 2021. 1

[20] 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. 1, 2, 5

[21] 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. 1

[22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[23] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015. 5, 6, 8

[24] A. Menon and C. S. Ong. Linking losses for density ratio and class-probability estimation. In *Proceedings of the 33rd International Conference on Machine Learning*, page 304–313, 2016. 4, 5

[25] Hyunha Nam and Masashi Sugiyama. Direct density ratio estimation with convolutional neural networks with application in outlier detection. *IEICE Transactions on Information and Systems*, E98.D(5):1073–1079, 2015. 4

[26] Matthew Lyle Olson, Shusen Liu, Rushil Anirudh, Jayaraman J Thiagarajan, Weng-Keen Wong, and Peer-Timo Bremer. Unsupervised attribute alignment for characterizing distribution shift. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021. 3

[27] Matthew L Olson, Thuy-Vy Nguyen, Gaurav Dixit, Neale Ratzlaff, Weng-Keen Wong, and Minsuk Kahng. Contrastive identification of covariate shift in image data. In *2021 IEEE Visualization Conference (VIS)*, pages 36–40. IEEE, 2021. 3

[28] William Peebles, John Peebles, Jun-Yan Zhu, Alexei Efros, and Antonio Torralba. The hessian penalty: A weak prior for unsupervised disentanglement. In *European Conference on Computer Vision*, pages 581–597. Springer, 2020. 2, 6, 8

[29] Justin NM Pinkney and Doron Adler. Resolution dependent gan interpolation for controllable image synthesis between domains. *arXiv preprint arXiv:2010.05334*, 2020. 3

[30] Dragomir R Radev, Hong Qi, Harris Wu, and Weiguo Fan. Evaluating web-based question answering systems. In *LREC*. Citeseer, 2002. 7

[31] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 8

[32] Inioluwa Deborah Raji, Andrew Smart, Rebecca N White, Margaret Mitchell, Timnit Gebru, Ben Hutchinson, Jamila Smith-Loud, Daniel Theron, and Parker Barnes. Closing the ai accountability gap: Defining an end-to-end framework for internal algorithmic auditing. In *Proceedings of the 2020 conference on fairness, accountability, and transparency*, pages 33–44, 2020. 2, 3

[33] Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. InterfaceGAN: Interpreting the disentangled face representation learned by GANs. *IEEE transactions on pattern analysis and machine intelligence*, 2020. 2

[34] Yujun Shen and Bolei Zhou. Closed-form factorization of latent semantics in GANs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1532–1540, 2021. 2, 6, 8

[35] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Encoding robustness to image style via adversarial feature perturbations. *Advances in Neural Information Processing Systems*, 34, 2021. 6, 8

[36] Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012. 4

[37] Masashi Sugiyama, Taiji Suzuki, Shinichi Nakajima, Hisashi Kashima, Paul von Bünau, and Motoaki Kawanabe. Direct importance estimation for covariate shift adaptation. *Annals of the Institute of Statistical Mathematics*, 60(4):699–746, 2008. 4

[38] Ellen M Voorhees et al. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82, 1999. 7

[39] Andrey Voynov and Artem Babenko. Unsupervised discovery of interpretable directions in the gan latent space. In *International conference on machine learning*, pages 9786–9796. PMLR, 2020. 2, 6, 8

[40] Yuxiang Wei, Yupeng Shi, Xiao Liu, Zhilong Ji, Yuan Gao, Zhongqin Wu, and Wangmeng Zuo. Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6721–6730, 2021. 2, 6, 8

[41] Zongze Wu, Yotam Nitzan, Eli Shechtman, and Dani Lischinski. Stylealign: Analysis and applications of aligned stylegan models. *arXiv preprint arXiv:2110.11323*, 2021. 3

[42] Tom Yan and Chicheng Zhang. Active fairness auditing. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 24929–24962. PMLR, 17–23 Jul 2022. 3

[43] Oğuz Kaan Yüksel, Enis Simsar, Ezgi Gülperi Er, and Pinar Yanardag. Latentclr: A contrastive learning approach for unsupervised discovery of interpretable directions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14263–14272, 2021. 2, 3, 4, 6, 8