Generative Trace Attribution Network

Anonymous authors
Paper under double-blind review

Abstract

A deepfake is a digitally created or altered image, video, or audio made using artificial intelligence (AI) that seems real but is intended to deceive or mislead viewers. The rapid rise of generative AI tools like GANs and diffusion models has made it easier to create deepfake images. While traditional methods can detect deepfakes, they often fail to identify which model created them. The process of matching a deepfake image to its generative model is known as deepfake attribution. Deepfake attribution is essential for accountability and preventing misuse of generative models in the creation of deefake. Unfortunately, most existing attribution methods only work well on the specific models they were trained on. They struggle to attribute images generated from unseen generators with different initialization seeds, trained for additional epochs, fine-tuned, retrained, or having slight modifications in loss functions or model architecture. To address these limitations, we propose the Generative Trace Attribution Network (GTA-Net), a generalized attribution network that robustly attributes fake images across diverse generative models with variations, including entirely unseen generative models. GTA-Net works by analyzing hidden patterns in input images using a combination of frequency analysis and latent space analysis to capture training-induced artifacts of target generative models to be attributed. GTA-Net also employs supervised contrastive learning to separate features between different target generative models. Extensive experiments on diverse generative models demonstrate that GTA-Net significantly outperforms existing attribution techniques, offering a more robust and reliable approach for deepfake attribution.

1 Introduction

Deepfake technology, powered by generative models such as Generative Adversarial Networks (GANs) and diffusion models, enables the creation of hyper-realistic digital media that is nearly indistinguishable from real content Karras et al. (2021b); Ho et al. (2020a). While this technology has beneficial applications in entertainment, virtual reality, and education, it poses serious security risks. Deepfakes are increasingly exploited for misinformation campaigns France24, defamation NPR, identity theft CBS News, harassment NBC News, threatening public trust and individual privacy Amezaga & Hajek (2022); Chauhan (2023). Deepfake techniques can generally be classified into two categories: manipulation-based and synthesis-based. In manipulation-based deepfake techniques, the original image is altered using deep learning techniques such as face-swapping Perov et al. (2020), face reenactment Thies et al. (2016), and attribute manipulation Choi et al. (2020). Since these modifications rely on real images, forensic analysis can sometimes identify inconsistencies, such as unnatural facial expressions, blending artifacts, or lighting mismatches Guarnera et al. (2020); Chai et al. (2020); Wang et al. (2020). In contrast, synthesis-based techniques generate entirely new images from scratch Karras et al. (2021b; 2017); Ho et al. (2020a) without requiring any real image, making detection significantly more challenging.

Deepfake detection is generally framed as a binary classification problem, distinguishing between real and fake images. Classification alone is inadequate, as it fails to provide accountable evidence of tampering or image synthesis to the forensic team. Therefore, manipulation-based deepfake detection employs localization techniques Chen & Others (2022); Mazaheri & Roy-Chowdhury (2022) to identify tampered regions by analyzing inconsistencies in the image. However, synthesis-based deepfakes generate images entirely from

scratch, making localization techniques ineffective. Thus, forensic team use model attribution to analyze subtle artifacts, statistical patterns, and architectural fingerprints embedded in generated images to infer the identity of the generative model used Kim et al. (2020); Marra et al. (2019). This process is analogous to ballistic forensics, where microscopic markings left on a bullet are examined to trace it back to the specific firearm that fired it. Moreover, training high-quality generative models requires extensive computational resources, large datasets, and expert knowledge, limiting the availability of open-source models. Thus, many state-of-the-art models remain proprietary and is a valuable assets. Therefore, there must be a copyright protection mechanism for these generative models. In this context, model attribution can be used to validate the ownership of a generative model by extracting and analyzing the model fingerprints from its generated image.

Traditional attribution techniques rely on two aspects for identifying the generated images where the first one involves embedding a unique fingerprint in the training data such that the generated image will contain the same fingerprint Kim et al. (2020); Yu et al. (2020; 2021), while the second approach focuses on identifying the unique patterns left behind by different generative models on generated images Marra et al. (2019); Xuan et al. (2019); Joslin & Hao (2020); Frank et al. (2020); Yu et al. (2019); Ding et al. (2021); Jeon et al. (2020); Li et al. (2024); Sun et al. (2023); Girish et al. (2021); Yang et al. (2023); Zheng et al. (2025). However, these techniques often lack robustness and fail to generalize across models as they are trained on the images from a specific instance of the generative model. One can bypass this technique by fine-tuning the generative model, retraining it on a different dataset, or tweaking hyperparameters, which can create an infinite number of unseen variations, making attribution inconsistent Yu et al. (2019). This highlights the limitations of existing methods, as visually demonstrated in Figure 1. These challenges point to the need for a more generalized and robust attribution technique that remains effective.

We propose *GTA-Net* (Generative Trace Attribution Network), a robust deepfake attribution technique that generalizes across retrained, fine-tuned, modified or unseen generative models. Unlike prior methods that identify specific model instances, *GTA-Net* attributes images based on architectural-level traces and training pipeline characteristics. *GTA-Net* has three key components: an Abstractor Network, a Feature Extraction Network, and a Multi-class Classification Network. The Abstractor Network first processes each input image using a denoising autoencoder and wavelet transform to extract residual and frequency-based features that capture subtle traces left by the generative model, while minimizing the influence of image content. These extracted residual and frequency-based features are combined into a unified representation and then refined using a multi-head self-attention mechanism, which helps the network focus on the most informative patterns corresponding to generative models to be attributed. Next, the Feature Extraction Network uses contrastive learning, a self-supervised learning technique, given a pair of images whose objective is to push the input images closer or further apart depending on whether the images are from the same or different generative models. This helps the network create precise and meaningful feature representations that capture the unique characteristics of each generative model. Finally, the Classification Network uses these features to accurately identify which generative model produced the image. Our experimental results demonstrate that the sequential architecture of *GTA-Net* facilitates the extraction of model-specific patterns, enabling robust attribution even in challenging scenarios, such as changes in seeds, datasets, training duration, loss functions, minor architectural variations or encountering unknown generative models. In summary, we make the following contributions:

- We proposed *GTA-Net*, which aims to attribute fake images to their source generative model, irrespective of whether the models producing those images have been retrained with an alternative seed, fine-tuned for additional epochs, trained on different datasets, use slightly different loss functions or architecture.
- *GTA-Net* facilitates attribution of unseen generative models (different generative models whose generated data is not used in training) by identifying the closest matching generative model using supervised contrastive learning, frequency-based analysis, and latent space analysis.
- We conducted comprehensive evaluations of *GTA-Net* on various generative models to demonstrate the accuracy and robustness of our technique.

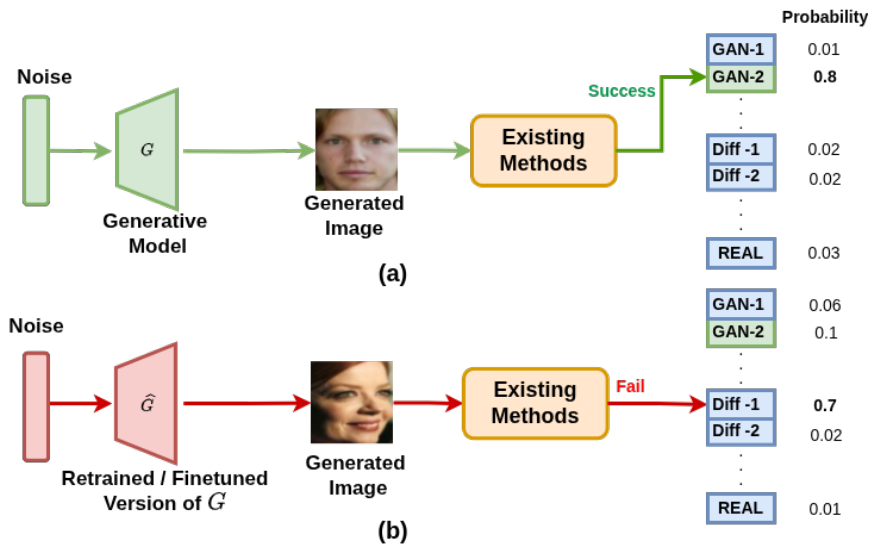

Figure 1: Key drawbacks of existing methods. In (a), attribution is correctly performed on the original instance of the generative model. However, as shown in (b), when the generative model is fine-tuned or retrained, the existing technique fails to attribute correctly. The green box is the ground truth label.

2 Related Work

Various deepfake detection techniques have been developed to differentiate real images from fake images Guarnera et al. (2020); Zhao et al. (2021); Asnani et al. (2023); Jeon et al. (2020); Durall et al. (2020); Wang et al. (2020). Simply identifying a generated image as fake is not sufficient, as it does not provide any insights into its origin. Forensic reliability is significantly improved when attribution techniques not only detect a fake but also attribute it back to the generative model. This deeper attribution helps distinguish between different generative models, enabling better accountability, forensic analysis, and mitigation of misuse. The attribution of generated images falls into two main categories: artificial fingerprint embedding and model fingerprinting. The first category of artificial fingerprint embeds unique fingerprints into the training data so that any generated image retains identifiable fingerprints Kim et al. (2020); Yu et al. (2020; 2021). These fingerprints can then be extracted from generated images to attribute them to a specific generative model. However, this technique requires white-box access to the model, which limits its applicability in real-world scenarios. Additionally, it does not work for pre-trained models that were not trained with artificial fingerprint-embedded datasets, and retraining existing models to incorporate fingerprints is computationally expensive and time-consuming. In contrast, model fingerprinting identifies unique patterns and artifacts naturally left by different generative models in their generated images without altering the training process. These patterns arise from architectural differences and training procedures, which remain consistent even when models are trained on different datasets. This technique operates in a black-box setting, making it more suitable for real-world applications.

The model fingerprinting is performed using both statistical and deep learning-based techniques. In statistical techniques, the focus is on detecting residual noise in generated images using various filters Marra et al. (2019), handcrafted features Li et al. (2024) or performing frequency analysis through techniques like the Discrete Cosine Transform (DCT) Frank et al. (2020) and Discrete Fourier Transform (DFT) Joslin & Hao (2020). Deep learning-based techniques, on the other hand, utilize image transformation techniques, filters, residual images, and custom loss functions to capture subtle, architectural and training-specific features. These features are then passed into a classifier for final attribution Xuan et al. (2019); Yu et al. (2019); Yang et al. (2022). However, existing techniques perform attribution on seen models, meaning both the training and testing data are generated from the same trained generative model. These techniques fail when tested on a retrained or fine-tuned version of the same generative model. Few techniques Yang et al. (2022) attempts to improve robustness via patch-wise contrastive learning. While effective in some cases, it is sensitive to

resolution changes, making it less suitable for diffusion models that produce high-resolution, low-artifact images. Recent works Sun et al. (2023); Girish et al. (2021); Yang et al. (2023); Zheng et al. (2025) explore open-world attribution, where test models may be unseen during training. While promising, these methods often depend on distributional assumptions or thresholds that struggle with fine-tuned variants and fail to generalize to realistic, artifact-free images from advanced generative models. A technique Mu et al. (2025) also tried to attribute AI-generated ex-regulatory images by extracting model fingerprint features using the CLIP network, but it limits the attribution coverage of unseen real-world distributions.

To address the limitations of both patch dependent and open-world attribution methods, we propose *GTA-Net*, a robust and resolution-independent model attribution framework. Unlike *DNA-DET*, which relies on patch-wise contrastive learning and is sensitive to changes in image resolution and structure, *GTA-Net* operates holistically without decomposing images into patches. It begins by extracting content-dependent features, allowing the network to isolate and amplify generative model-specific traces that persist across different datasets, fine-tuning, and retraining. This makes *GTA-Net* inherently more robust to model modifications and domain shifts. Crucially, *GTA-Net* is designed to capture deep, intrinsic fingerprints linked to the architecture and training dynamics of generative models features that are stable and generalize well to both seen and unseen models. This contrasts with existing open-world attribution methods, which often depend on handcrafted thresholds or distributional assumptions, making them vulnerable to minor perturbations or high-fidelity outputs from modern diffusion models. By learning discriminative, architecture-level signatures, *GTA-Net* achieves superior performance in both closed-world and open-world settings, offering reliable attribution even under black-box constraints and with photorealistic inputs.

3 Proposed Methodology

3.1 Problem Formulation

The growing popularity of generative models, particularly those capable of creating realistic images from scratch, has sparked significant interest in their applications across various domains. These include the generation of avatars, virtual environments, *AI*-assisted content creation, synthetic medical scans, artwork, and more. As a result, a wide range of generative models with high-quality image generation capabilities has been developed. However, with the growing prevalence of generative models, attributing generated images to their source is essential for ensuring reliability, preventing misuse, maintaining accountability, and fostering transparency in *AI*-generated content. To address this, we define attribution as the task of determining which source generative model produced a given fake image. This can be framed as a multi-class classification problem, where the goal is to attribute each image to its generative model or identify it as real. Given an image X^y with source $y \in \{real, G_1, G_2, \ldots, G_N\}$, where $\{G_1, G_2, ..., G_N\}$, are different generative models. Our goal is to learn a mapping $f(X^y) \rightarrow y$.

3.2 Key Challenges

Formulating a multiclass classification problem for generative model attribution may appear straightforward. However, the slightest change, such as fine-tuning, training for additional epochs or retraining, will lead to different instances of the original generative model, which will fail the multiclass classifier. One can even think of defining distinct classes for each of those model instances. In that case, the key challenge is to determine what constitutes a distinct generative model. For instance, does modifying architectural components, such as adding or removing layers or altering the loss function, fundamentally constitute a distinct model? Furthermore, variations introduced through different initial seeds, datasets, training epochs, optimization strategies, or fine-tuning can lead to substantially different model instances, raising the question of how many such variations are necessary to construct a robust classifier that generalizes across all possible instances of the target generative models. Adding to that, in real-world scenarios, we are more likely to encounter previously unseen models or variants. Therefore, the core challenge is to distinguish among known variants but also attribute outputs to the correct generative models based on underlying architecture and training pipelines.

To address these challenges, *GTA-Net* is designed to attribute images to the closest matching generative model rather than assigning them to a generic unknown class, enabling a more precise inference of the underlying architectural family and training paradigm. For previously unseen models that are not part of the training set, *GTA-Net* performs attribution based on architectural similarities, training pipeline, and loss functions. This approach preserves structural similarity information, improves interpretability, and allows finer-grained forensic analysis even when the exact generator was unseen during training. To enhance generalization and mitigate model-specific biases, multiple instances of each generative model are trained while varying key hyperparameters such as learning rate, batch size, and training steps. This ensures that *GTA-Net* learns intrinsic architectural and training-related characteristics rather than memorizing specific generative model instances.

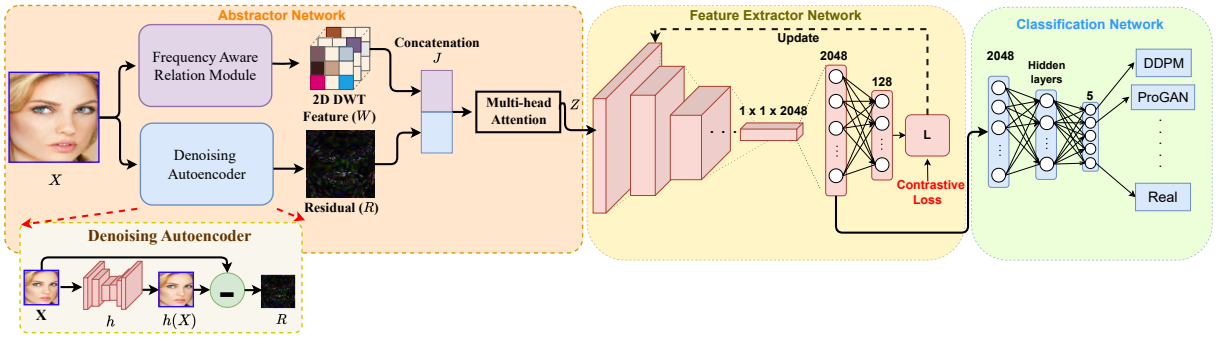

Figure 2: GTA-Net architecture for generative model attribution. It consists of three modules: Abstractor Network for extracting residual and frequency features using a denoising autoencoder and wavelet transform; Feature Extraction Network, which refines representations via contrastive learning; Multi-Class Classification Network, to perform final attribution for identifying the source generative model.

3.3 Overview

The technique of *GTA-Net* can be seen in Fig. 2. *GTA-Net* contains three sub-networks: the first one is an abstractor network, the second one is a feature extraction network, and the third one is a multi-class classification network. The abstractor network, feature extraction network, and multi-class classification network are trained as a pipeline. First, the abstractor network is trained on the fake images from different generative models and real images such that it can extract data-independent traces from images which focuses more on the architectural-level traces and the training pipeline rather than the type of data generated by the model. Feature extraction network is trained on these traces to extract the feature embedding of the generative models. These features essentially represent the fingerprints of the architecture and training pipeline. The multi-class classification network is trained using the features obtained from the feature extraction network for final attribution. The details of each network are discussed in the following section.

3.4 Abstractor Network

The objective of the abstractor network is to reduce data dependency and focus more on the architectural-level traces and the training pipeline of the generative model. It consists of the following two modules:

3.4.1 Denoising Autoencoder

The aim of this module is to extract high-level features that are independent of image content and can serve as a fingerprint for attributing generative models Marra et al. (2019); Frank et al. (2020). To achieve this, we first train a denoising autoencoder on real images. Once trained, the autoencoder takes an input image, whether real or generated and reconstructs it. Next, we compute the residual image by subtracting the reconstructed image from the original input image. These residuals capture subtle differences that are partly dependent on the image content but can also contain unique traces left by the generative model. The task

of the denoising autoencoder can be mathematically formulated as:

$$R = abs(X - h(X)) \tag{1}$$

where X is the input image, h represents the trained denoising autoencoder which reconstructs X, R is the calculated residual image, and abs represents absolute value.

3.4.2 Frequency Aware Relation Module

The objective of this model is to explicitly incorporate frequency-domain features to enhance attribution accuracy by reducing the dependency on the content of the image. To achieve this, we leverage wavelet transform Mallat (1989), a powerful tool for frequency analysis, specifically employing 2D Discrete Wavelet Transform ($2D\ DWT$) to decompose input images into multiple frequency components. The conventional $2D\ DWT$ employs two filters: L (low-pass filter), which retains low-frequency components, and H (high-pass filter), which captures high-frequency details. Recent findings Miao et al. (2023) suggest that high-frequency features play a more critical role in face forgery detection than low-frequency ones. Therefore, we selectively retain only the LH(low-high), HL (high-low), and HH (high-high) sub-bands for further processing, ensuring the extraction of discriminative frequency features. The feature extracted by $2D\ DWT$ can be formulated as:

$$W = DWT(X; LH, HL, HH) \tag{2}$$

where X is the input image, W is the extracted $2D\ DWT$ features which contain LH, HL and HH sub bands. If the size of the input image is $M \times N$, the size of each subband is $\frac{M}{2} \times \frac{N}{2}$ and each subband has the same number of channels as the input image (for RGB image it is 3).

Multi-Head Self-Attention Mechanism: Once we have the R (from denoising autoencoder) and W (from frequency aware relation module), we upsample LH, HL and HH sub bands of W using bicubic interpolation to match the dimension of R. Subsequently, R is concatenated with upsampled LH, HL and HH sub bands of W to form a representation given by:

$$J = \text{Concat}(R, LH, HL, HH) \in \mathbb{R}^{M \times N \times 4C} \tag{3}$$

where $M \times N$ is the spatial resolution and C represents the channel dimension (here $C = 3$). We apply multi-head self-attention with four heads on the input feature map J to refine the extracted features. The multi-head attention mechanism gives a refined feature map Z, which is a representation that is specific to a given architecture and training pipeline. The output Z maintains the same spatial dimensions as J but with enhanced global contextual information.

3.5 Feature Extraction Network

For attribution, a standard deep learning-based classifier can be trained on images generated from different generative models Yu et al. (2020). The main drawback of using a simple classifier is the drop in accuracy due to the presence of overlapping images as shown in Fig. 3 (a). While real images also exhibit overlapping features, classifiers can still perform well because they learn discriminative features from diverse datasets. However, in the case of generative model attribution, the feature overlap from structural and statistical similarities introduced by the generation process, leading to less distinctive patterns and making it harder for the classifier to learn decision boundaries.

To overcome this limitation, we incorporate a feature extraction network, which leverages supervised contrastive learning to enhance generalization and improve attribution performance. The feature extraction network is designed to generate discriminative embeddings that effectively capture the subtle traces left by different generative models. It is built upon a convolutional neural network (CNN) backbone, followed by a two-branch projection: a 128-dimensional projection head for contrastive loss computation and a 2048-dimensional classification head for final prediction. To train this network, we employ supervised contrastive learning Khosla et al. (2020), which improves feature robustness by explicitly modelling inter-class and intra-class relationships. In contrast to traditional cross-entropy loss, which only considers class labels for final classification, supervised contrastive learning forms contrastive pairs using label supervision during training.

Given a batch of samples, the model pulls together representations of samples with the same class label (positive pairs) and pushes apart those from different classes (negative pairs). This is achieved via a contrastive loss function that operates on the cosine similarity between normalized feature embeddings.

We train feature extraction network with the feature map Z, extracted through the multi-head self-attention mechanism. Feature extraction network produces a 2048-dimensional feature embedding, which is subsequently reduced to a 128-dimensional feature embedding using a deep neural network. The 2048-dimensional feature embedding is called the classification head, which is used to train the multi-class classification network for attribution. The 128-dimensional feature embedding is called the projection head, which is used to calculate the supervised contrastive loss for training the feature extraction network. Using supervised contrastive loss results in a well-structured feature space where the same models are clustered together and dissimilar ones are distinctly separated, as shown in Fig. 3(b)

3.6 Multi-class Classification Network:

Our multi-class classification network is a deep neural network with fully connected layers. We trained this network using the 2048-dimensional feature embeddings generated from the classification head of feature extraction network. This multi-class classification network makes the final attribution of the generative models.

3.7 Loss Functions:

To optimize the entire pipeline, we integrate three loss functions:

- Multi-Head Attention Loss (L_{MHA}) promotes effective contextual feature extraction by aligning the input (J_i) and refined (Z_i) feature maps for batch of N samples:

$$L_{MHA} = \frac{1}{N} \sum_{i=1}^{N} ||J_i - Z_i||^2 \tag{4}$$

- Supervised Contrastive Loss (L_{SCL}) improves feature separability by bringing embeddings from the same generative model closer and pushing apart those from different ones:

$$L_{SCL} = \sum_{i \in I} \left(-\frac{1}{|P(i)|} \sum_{p \in P(i)} \log \left(\frac{\exp(f_i \cdot f_p / \tau)}{\sum_{a \in A(i)} \exp(f_i \cdot f_a / \tau)} \right) \right) \tag{5}$$

  Here, I is the set of all images in a mini-batch. $i \in I$ represents the index of an image in the mini-batch. $P(i) \equiv p \in A(i) : \tilde{y}_p = \tilde{y}_i$ denotes the set of positive samples (sample generated by the same generative model) excluding i itself. $|P(i)|$ is the number of such positive samples. $A(i) \equiv I \setminus \{i\}$ is the set of all other images in the batch. f_i is the feature embedding and τ is the temperature scaling factor. The 128-dimensional projection head is utilized for contrastive loss computation during supervised training. Meanwhile, the 2048-dimensional feature embedding, referred to as the classification head, is used for multi-class classification.

- Cross-Entropy Loss (L_{CE}) guides the classification of the 2048-dimensional embedding using cross-entropy loss with *Adam* optimizer and a learning rate of 0.00003:

$$L_{CE} = -\sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{6}$$

  where y_i and $\hat{y}_i$ are the ground truth and predicted labels, respectively.

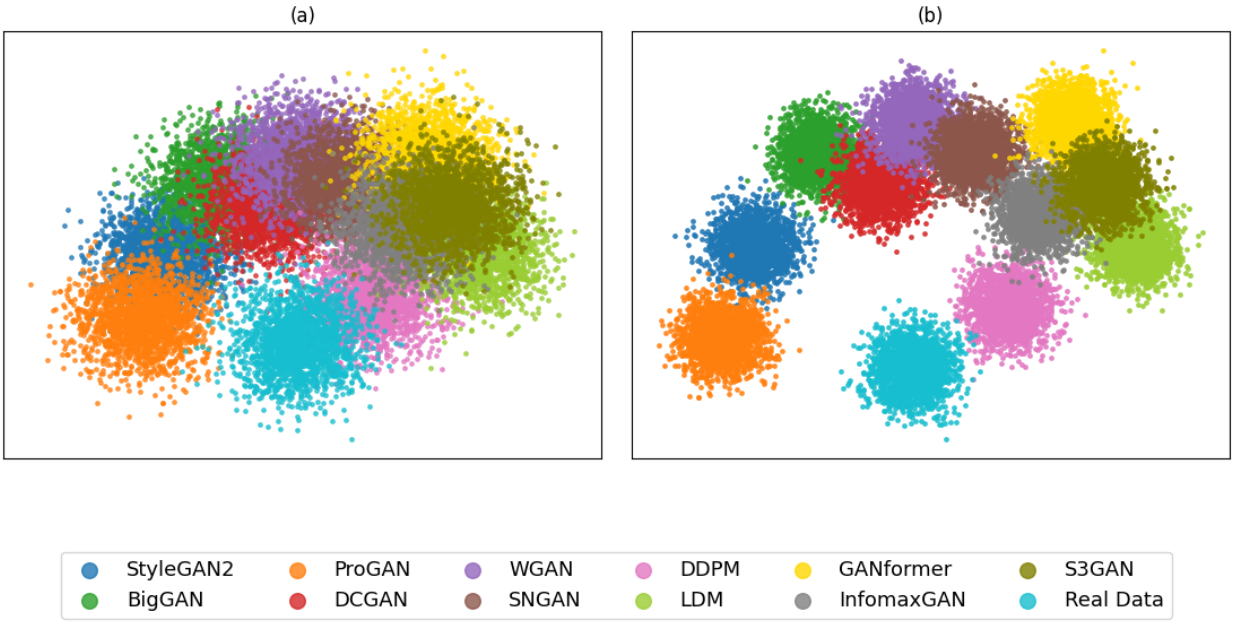

Figure 3: $t-SNE$ plot of feature embeddings of different generative models. In Fig. (a) the t-SNE plot is generated by passing images directly through the feature extraction network and the extracted features are visualized in the embedding space. In Fig. (b), the images are first processed by an abstractor network, which transforms them before being passed through the feature extraction network. The extracted features are then used for visualization.

The overall loss is a weighted sum:

$$L_{Total} = \lambda_1 L_{MHA} + \lambda_2 L_{SCL} + \lambda_3 L_{CE} \tag{7}$$

This combination ensures that $GTA\text{-}Net$ learns both discriminative and context-aware representations for reliable attribution of generated images.

4 Experimental Setup

In this section, we demonstrate the effectiveness of $GTA\text{-}Net$ through comprehensive experiments. We performed all our experiments using the machine with 14-$core\ Intel\ i9\ 10940X\ CPU$, 32 $GB\ RAM$, and $Nvidia\ RTX$-5000 GPU with 16 $GB\ VRAM$. Training $GTA\text{-}Net$ requires a diverse set of pre-trained generative models that can generate high-quality fake images. These models help us build a rich and diverse dataset to train and evaluate our network. We use eleven generative models in total, eight GANs, 2 diffusion models, and one transformer based which generate high-fidelity images for attribution tasks (detailed in Section 4.2). We selected these models because they collectively span the major families of modern generative modelling. These models differ in objective functions, sampling mechanisms and representation spaces, ensuring exposure to a wide spectrum of generative artifacts and improving the robustness and generalization of the classifier.

4.1 Dataset and Model Architecture

We used $CelebA$ Liu et al. (2015), $LSUN\text{-}Cat$ Yu et al. (2015), and $MS\text{-}COCO$ Lin et al. (2014) datasets to train all the eleven generative models, which are to be attributed based on their generated data. The GAN models used for attribution are $StyleGAN2$ Karras et al. (2020), $BigGAN$ Brock et al. (2019), $ProGAN$ Karras et al. (2017), $DCGAN$ Radford et al. (2015), $WGAN$ Arjovsky et al. (2017), $InfomaxGAN$ Lee et al. (2021b), $S3GAN$ Lučić et al. (2019), and $SNGAN$ Miyato et al. (2018). For diffusion model, we

use $DDPM$ Ho et al. (2020b), and LDM Rombach et al. (2022). Atlast for transformer based we have used $GANformer$ Hudson & Zitnick (2021). We selected these generative models spanning the dominant paradigms of modern image synthesis to ensure architectural, objective-level, and representational diversity within the training space of GTA-Net. These models collectively represent distinct training objectives (min–max adversarial optimization vs. iterative denoising likelihood maximization), sampling strategies (single-pass generation vs. multi-step refinement), and feature inductive biases (convolutional locality, spectral normalization, style modulation, and attention-based global modeling). By including representative architectures from each major generative family rather than multiple minor variants within a single family, we ensure that GTA-Net learns objective-level and architecture-level fingerprints rather than overfitting to superficial implementation details. This design choice provides maximal coverage of the generative modeling landscape while maintaining computational feasibility, and establishes a structured embedding space that supports robust nearest-class attribution for unseen models belonging to related generative families.

The apparent numerical imbalance between GAN-based and diffusion-based models in the training set is intentional rather than incidental. GANs represent a broader design space with significant architectural and objective-level variability (e.g., foundational CNN-GANs, spectral-normalized variants, large-scale class-conditional models, and style-modulated architectures), each introducing distinct inductive biases in feature synthesis and discriminator feedback dynamics. In contrast, diffusion-based models share a common probabilistic denoising formulation, where architectural variations primarily affect backbone implementation (e.g., U-Net vs. latent U-Net) rather than the underlying generative objective. Consequently, fewer diffusion representatives are sufficient to capture the diffusion-family fingerprint, while multiple GAN variants are required to span the adversarial design space. Similarly, GANformer is included as a representative of transformer-augmented generative modeling, where attention replaces convolutional locality while preserving adversarial optimization. Thus, the training set is balanced at the level of generative paradigms rather than raw model counts, ensuring coverage of adversarial, likelihood-based, and attention-driven synthesis mechanisms without over-representing redundant objective formulations.

GTA-Net is trained on the data generated by these generative models. The denoising autoencoder present in the abstractor network is a pre-trained model that is trained on the combination of $CelebA$ and MS-$COCO$ datasets. Its architecture is based on CNN and consists of both an encoder and a decoder, each comprising 5 convolutional layers, excluding pooling and normalization layers. For feature extraction network, we adopt the encoder network and projection network used in Khosla et al. (2020). The encoder network, which utilizes $ResNet$-101 He et al. (2015) as its backbone, maps all the input samples into 2048-dimensional feature embeddings, ensuring robust representation learning. The projection network transforms feature embedding into a size of 128-dimension feature embedding by using a multi-layer perceptron with a single hidden layer of size 2048. The multi-class classification network consists of four fully connected layers with 1024, 512, 256, and 12 neurons, respectively. $ReLu$ activation is used in the intermediate layers, and $Softmax$ is in the final layer.

4.2 Training Details

To improve generalization, we train three different instances of each generative model on $CelebA$ by varying seeds and hyperparameters like learning rate, batch size, training steps, and initial model parameters. This introduce diversity while preserving the fundamental structure of each generative model. We choose three instances to strike a balance between diversity and computational efficiency, ensuring sufficient variability in training while keeping the dataset size manageable for effective learning. Once trained, each instance generates $5,000$ images, leading to $135,000$ fake images. We also include $15,000$ real $CelebA$ images, forming a dataset of $150,000$ samples ($120,000$ for training, $30,000$ for validation).

In the abstractor network, we have a pre-trained denoising autoencoder, which is trained on a combination of $CelebA$ and MS-$COCO$ datasets. We use mean absolute error as loss metric, $Adam$ optimizer, and set the learning rate at 0.0001. We resize the images to 128×128 for training and apply augmentation to the images (rotation, blurring, flipping, centre crop etc.). Training GTA-Net involves training of abstractor network, feature extraction network and multi-class classification network in the pipeline. The training begins by passing the image through the denoising autoencoder and frequency-aware relation module to obtain the residual image R and 2D DWT feature W. Both R and W are concatenated and passed through a multi-

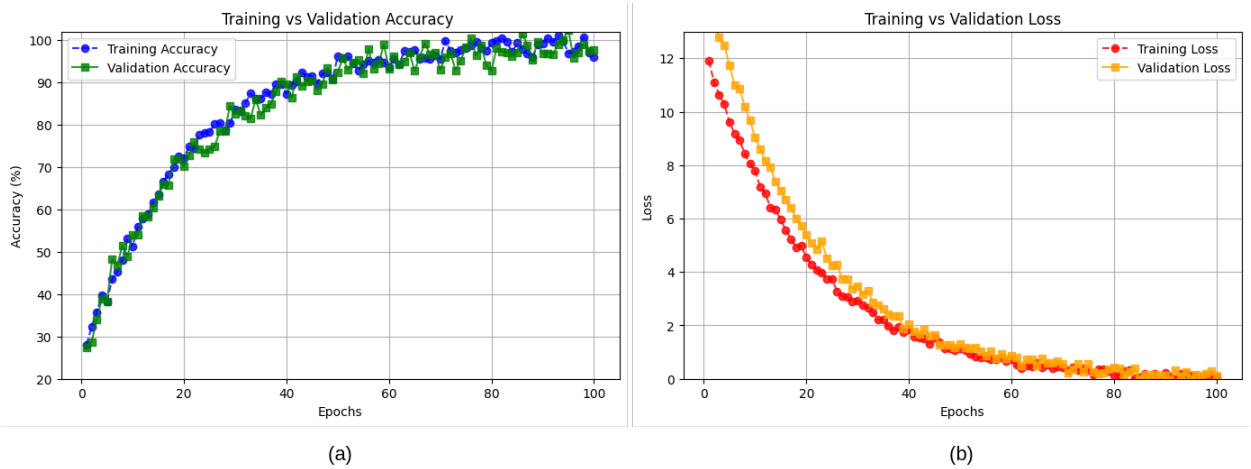

Figure 4: Training and validation accuracy of GTA-Net is shown in (a) and training and validation loss of GTA-Net is shown in (b)

head self-attention mechanism to obtain the refined feature map Z, as mentioned in the previous section. We compute multiple attention "heads" in parallel, each learning to focus on different types of patterns. The outputs of these attention heads are then merged into a single refined feature representation Z, which is used to train feature extraction network. We train the feature extraction network using a supervised contrastive loss, and pass the 2048-dimensional embedding into a multi-class classification network, enabling end-to-end training of GTA-Net.

5 Results

To assess the effectiveness of the proposed GTA-Net technique, we evaluate its performance in attribution.

5.1 Training and Validation Accuracy

To evaluate the effectiveness of GTA-Net, we use classification accuracy as the primary performance metric. The model demonstrates stable training, with loss convergence observed within 100 epochs, ensuring reliable optimization. On the validation set, GTA-Net achieves an impressive 98.78% average classification accuracy, highlighting its ability to accurately attribute generated images to their respective generative models. A comparative analysis of training and validation accuracy is illustrated in Fig. 4, showcasing the model's consistent performance across different stages of training. Additionally, we analyze the classification accuracy for individual generative models, providing a detailed breakdown in Table 1, further validating the robustness of our approach.

5.2 Feature Consistency and Separability

The abstractor network and supervised contrastive loss (L_{SCL}) in the feature extraction network enhance feature separability by ensuring that embeddings from the same generative model remain closely clustered while maintaining clear distinctions between different models. This facilitates more precise attribution. The t-SNE visualization in Fig. 3 demonstrates this effect. In Fig. 3 (a), overlapping clusters indicate weaker separation, whereas in Fig. 3 (b), the clusters are more distinct, reflecting improved feature representation. These results highlight the ability of GTA-Net to effectively capture architectural and training characteristics, leading to robust and accurate attribution.

Table 1: Classification accuracy of GTA-Net for different generative models.

Generative Model	Train Accuracy (%)	Validation Accuracy (%)
StyleGAN2	99.8	99.6
BigGAN	99.7	99.6
ProGAN	99.9	99.7
DCGAN	99.1	98.7
WGAN	99.1	98.9
InfomaxGAN	99.2	98.7
S3GAN	99.3	98.3
SNGAN	99.3	99.1
GANformer	98.7	98.1
DDPM	98.6	97.8
LDM	99.3	98.2
Real Data	99.1	98.7
Average Accuracy	99.25	98.78

5.3 Effect of Cross-seed and Cross-dataset:

So far, we have evaluated the performance of $GTA\text{-}Net$ on images generated from the three instances of the generative models used for training. To assess the generalizability of $GTA\text{-}Net$, we trained the fourth instance for each generative model on the $CelebA$ dataset using different hyperparameters and used it for cross-seed testing. We generated $5,000$ images from the fourth instance of each generative model and tested $GTA\text{-}Net$ for attribution. Previous work attempted a similar test but was only successful in attributing cross-seed data generated from ProGAN Khosla et al. (2020). In contrast, our technique successfully attributed images from all generative models, as shown in the third column of Table 2.

Similarly, for cross-dataset testing, we trained a fifth instance for each generative model using the $LSUN\text{-}Cat$ dataset instead of $CelebA$. Since our technique captures architectural and training-level traces, it is capable of correctly attributing images even when the dataset changes. We generated $5,000$ images from the fifth instance of each generative model to perform attribution. The results, presented in the fourth column of Table 2, further validate the robustness of our technique.

Table 2: Attribution performance of GTA-Net under different settings: cross-seed, cross-dataset, and fine-tuning which is compared to the baseline model.

Generative Model	Baseline	Cross-Seed	Cross-Dataset	Normal Fine-Tune	Cross Fine-Tune
StyleGAN2	99.6	99.8	99.6	99.7	99.5
BigGAN	99.6	99.4	99.2	99.3	99.0
ProGAN	99.7	99.5	99.3	99.4	99.1
DCGAN	98.7	98.6	98.3	98.5	98.0
WGAN	98.9	98.3	98.0	98.2	97.8
InfomaxGAN	98.7	97.1	97.2	97.1	97.0
S3GAN	98.3	97.8	97.4	97.6	97.1
SNGAN	99.1	98.8	98.6	98.7	98.3
GANformer	98.1	97.5	97.2	97.4	97.0
DDPM	97.8	98.0	97.8	97.9	97.5
LDM	98.2	97.2	96.9	97.1	96.7
Real Data	98.7	100	100	100	100
Average Accuracy	98.8	98.3	98.0	98.2	97.9

5.4 Effect of Fine-tuning

Fine-tuning a generative model for additional epochs or steps alters its generation capability, leading to variations in the images produced for the same noise input. A robust attribution network should remain unaffected by fine-tuning as long as the underlying architecture and training procedure remains unchanged. To evaluate whether fine-tuning impacts *GTA-Net*'s attribution capability, we conducted the following experiment:

- We fine-tuned the fourth and fifth instances of each generative model on the *CelebA* dataset for 20 additional epochs.
- The fine-tuned fourth instance was labelled normal-fine-tune, while the fine-tuned fifth instance was labelled cross-fine-tune.
- We then generated $5,000$ images from each fine-tuned model and tested *GTA-Net* for attribution.

As shown in the last two columns of Table 2, *GTA-Net* achieved satisfactory results, demonstrating its robustness against fine-tuning.

5.5 Effect of Conditions on Generative Models

To further evaluate the attribution performance of *GTA-Net*, we trained conditional versions of *BigGAN*, *ProGAN*, *S3GAN*, *StyleGAN2*, *GANformer* , and *LDM* on the *MS-COCO* dataset. During the training of *GTA-Net*, we used images generated from their unconditional counterparts. Since the underlying architectures remain unchanged, with only minor modifications in the training procedure to accommodate conditioning, this experiment allows us to assess whether *GTA-Net* is robust to such variations. To conduct this evaluation, we generated $5,000$ images from each conditional generative model and tested their attribution using *GTA-Net*. If *GTA-Net* successfully attributes these images, it indicates that the technique primarily captures the architectural-level traces and the training pipeline rather than being overly sensitive to slight changes in training methodology. The results of this evaluation are presented in Table 3.

Table 3: Performance of *GTA-Net* on conditional generative models. We compared our results to a baseline model trained on an unconditional generative model.

Generative Model	**Baseline (%)**	**Conditional Model (%)**
BigGAN	99.6	97.3
ProGAN	99.7	96.4
S3GAN	98.3	95.2
StyleGAN2	99.6	96.9
GANformer	98.1	95.7
LDM	97.5	93.0
Average Accuracy	98.9	95.5

5.6 Evaluation on Generative Model in the Wild

To assess the generalization capability of *GTA-Net* in the wild, we perform attribution on generative models that were not included in the training phase. Rather than assigning images from unknown generative models that are not included in *GTA-Net*'s training (also do not belong to real image class) to a separate class, we instead attribute them to the closest class in the *GTA-Net* adapting a nearest-class attribution strategy. This technique ensures that even unseen generative models are mapped to the most similar model within the existing options and the ambiguity around threshold sensitivity associated with open-set rejection can be avoided. We evaluate attribution performance on three different categories of generative models. The first category includes generative models that synthesize images directly from noise, optionally conditioned on text prompts,

without requiring an input image. This category covers *BEGAN* Berthelot et al. (2017), *CramerGAN* Bellemare et al. (2017), *MMDGAN* Li et al. (2017), *ProjectedGAN* Sauer et al. (2021), *StyleGAN* Karras (2019), *StyleGAN3* Karras et al. (2021a), *TransGAN* Jiang et al. (2021), *ViTGAN* Lee et al. (2021a), *Stable Diffusion* Rombach et al. (2022), *GLIDE* Nichol et al. (2021), *Taming Transformer* Esser et al. (2021), *Self-Attention GAN* (*SAGAN*) Zhang et al. (2019), *Latent Consistency Model* (*LCM*) Bai et al. (2023), *CIPS* Anokhin et al. (2021), *DiffusionGAN* Wang et al. (2022), *Denoising Diffusion GAN* Xiao et al. (2021), *DiT* Peebles & Xie (2023), *U-ViT* Bao et al. (2023), *VQ-Diffusion* Gu et al. (2022), and *Imagen* Saharia et al. (2022b). The second category consists of generative models that require an input image, mask, or semantic layout as conditioning information. These models perform image-to-image translation, attribute manipulation, semantic synthesis, or inpainting. This includes *Pix2Pix* Isola et al. (2017), *StarGAN* Choi et al. (2018), *Palette* Saharia et al. (2022a) , *CycleGAN* Zhu et al. (2017), *Pix2Pixhd* Wang et al. (2018), *LaMa* Suvorov et al. (2022), *MAT* Li et al. (2022), *GauGAN* Park et al. (2019), *AttGAN* He et al. (2019), *STGAN* Liu et al. (2019), and *RelGAN* Wu et al. (2019). The third category consists of modified variants of *ProGAN*, *StyleGAN*, *SNGAN*, *GANformer* and *DDPM*, ensuring that *GTA-Net* is tested on a wide range of architectures, as presented in Table 4.

Table 4: Architectural and Loss Function Modifications in Generative Models

Model	**Architectural Modifications**	**Loss Function Modifications**
ProGAN	Progressive growing of layers; modified upsampling strategy	Hinge loss
StyleGAN	Mapping network; style modulation (AdaIN); noise injection	R1 regularization; perceptual loss
SNGAN	Spectral normalization applied to discriminator layers	Hinge or Wasserstein adversarial loss
GANformer	Transformer-based self-attention layers in generator and discriminator	Hinge loss with attention regularization
DDPM	U-Net backbone with residual blocks and time-step embeddings	Noise prediction loss (L2 objective)

These models span both *GAN*-based, transformer-based and diffusion-based architectures, ensuring a diverse evaluation set. We generate a total of 2000 images from each model to perform attribution analysis. *GTA-Net* is trained to attribute images to their corresponding architecture and training procedure. It successfully attributes images from unseen generative models to the closest trained counterpart, as shown in Table 5, Table 6, and Table 7 . When performing attribution on known models, the confidence score is generally above 90%. However, for unseen generative models, we observe a lower average confidence score. Despite this, *GTA-Net* maintains high attribution accuracy, effectively mapping unseen models to their most similar trained counterparts.

From Table 5, it is found that:

- **GAN Models Cluster According to Training Objective and Architectural Design:** Unseen CNN-based *GAN* variants are consistently attributed to foundational adversarial models with moderate confidence and relatively high accuracy. *GTA-Net* captures similarities as variations in loss functions primarily affect optimization dynamics but preserve low-level convolutional inductive biases for spatial feature synthesis, which remain detectable in generated samples.
- **Transformer-Based GANs Show Reduced Confidence but Stable Attribution:** Transformer-GAN models exhibit noticeably lower confidence compared to CNN-GANs, yet maintain reasonable attribution accuracy. This behavior suggests transformer-based generators deviate from convolutional inductive biases. The adversarial min-max optimization introduces shared high-level distribution alignment, but the architectural shift from convolution to attention weakens low-level artifact similarity.
- **Diffusion Models Exhibit Strongest and Most Consistent Attribution Signals:** Latent diffusion models achieve high confidence and high accuracy, significantly outperforming most GAN-based counterparts. This likely occurs because diffusion models rely on iterative denoising pro-

Table 5: Nearest-Class Attribution on Unseen Noise-Based / Full-Synthesis Models

Model Type	**Model Name**	**Attributed Model**	**Confidence (%)**	**Accuracy (%)**
CNN-GAN	BEGAN	DCGAN	72.3	79.4
CNN-GAN	CramerGAN	WGAN	75.7	83.1
CNN-GAN	MMDGAN	WGAN	74.2	81.6
Stabilized GAN	ProjectedGAN	SNGAN	78.9	86.4
Style-Based GAN	StyleGAN	StyleGAN2	83.5	90.8
Style-Based GAN	StyleGAN3	StyleGAN2	84.1	91.3
Self-Att GAN	SAGAN	BIGGAN	89.4	76.7
Transformer-GAN	TransGAN	GANformer	51.6	78.2
Transformer-GAN	ViTGAN	GANformer	50.8	77.5
Implicit Generator	CIPS	StyleGAN2	39.8	66.9
Pixel Diff	GLIDE	DDPM	78.7	85.3
Latent Diff	Stable Diffusion	LDM	85.2	92.6
Latent Diff	LCM	LDM	83.1	90.4
Transformer Diff	DiT	LDM	61.9	88.7
Transformer Diff	U-ViT	LDM	60.5	86.8
Discrete Diff	VQ-Diffusion	LDM	77.3	84.1
Text Diff	Imagen	LDM	82.4	89.2
Hybrid Diff-GAN	DiffusionGAN	DDPM	76.2	83.5
Hybrid Diff-GAN	Denoising Diffusion GAN	DDPM	75.4	82.7
VQ + Transformer	Taming Transformer	GANformer	78.6	85.9

cesses, which introduce structured residual noise patterns. The forward-noise and reverse-denoise cycle leaves measurable statistical traces across frequency bands. These combined effects create stronger and more stable generative fingerprints than single-pass adversarial generation. Notably, Transformer diffusion models are correctly mapped to LDM despite architectural differences. This indicates that the training paradigm (diffusion process) dominates over backbone architecture in determining attribution similarity. Thus, the generative objective (diffusion vs adversarial) appears to exert stronger influence on learned artifacts than the specific network backbone (CNN vs Transformer).

- **Implicit Generators Exhibit Weakest Attribution Confidence:** $CIPS$ shows the lowest confidence and reduced accuracy. This is expected because $CIPS$ generates pixels directly from coordinate-conditioned MLPs. It lacks convolutional feature maps or diffusion process. This indicates that $GTA\text{-}Net$ relies partially on convolutional or denoising artifacts, which are less pronounced in coordinate-based implicit generators.

Table 6 highlights that conditional and image-to-image generative models exhibit comparatively lower attribution confidence and accuracy than full-synthesis models, suggesting that conditioning signals partially obscure intrinsic generative artifacts. Most adversarial translation frameworks are consistently mapped to $SNGAN$ with moderate confidence and accuracy, indicating that despite task-specific conditioning mechanisms, their core adversarial training pipeline and convolutional generator–discriminator structure dominate the learned artifact space. $CycleGAN$ aligns with $DCGAN$, reflecting its foundational CNN-based adversarial formulation without advanced normalization or style modulation. $Pix2PixHD$ maps to $ProGAN$, likely due to its progressive high-resolution synthesis strategy, while $GauGAN$ aligns with $StyleGAN2$, suggesting that semantic layout–conditioned generation shares structural similarities with style-modulated feature injection. Diffusion-based conditional models demonstrate clearer separation: Palette is attributed to $DDPM$, indicating that pixel-space diffusion artifacts persist even under inpainting tasks, whereas $LaMa$ achieves substantially higher confidence and accuracy when mapped to LDM, implying that latent-space

Table 6: Nearest-Class Attribution on Unseen Conditional / Image-to-Image Models

Tested Model	**Attributed Model**	**Confidence (%)**	**Accuracy (%)**
Pix2Pix	SNGAN	57.5	64.9
StarGAN	SNGAN	58.1	65.7
CycleGAN	DCGAN	53.4	60.2
Pix2PixHD	ProGAN	60.3	67.4
GauGAN	StyleGAN2	62.7	69.6
AttGAN	SNGAN	56.9	63.8
STGAN	SNGAN	55.3	62.4
RelGAN	SNGAN	57.6	64.5
Palette	DDPM	59.6	66.8
LaMa	LDM	81.2	88.1
MAT	GANformer	59.8	66.5

compression and denoising introduce strong, consistent signatures that survive conditional masking. MAT, which integrates transformer-based components, is attributed to $GANformer$ with moderate performance, suggesting that attention mechanisms influence artifact similarity but remain secondary to the dominant training objective. Overall, these results indicate that in conditional generation settings, the conditioning pathway introduces variability that attenuates artifact strength, yet the underlying generative objective remains the primary factor governing nearest-class attribution behavior in $GTA\text{-}Net$.

Table 7: Nearest-Class Attribution on Architectural Variants (Category 3)

Tested Model	**Attributed Model**	**Confidence (%)**	**Accuracy (%)**
ProGAN (modified)	ProGAN	96.1	93.8
StyleGAN (modified)	StyleGAN2	95.4	92.9
SNGAN (modified)	SNGAN	94.6	91.5
GANformer (modified)	GANformer	96.8	94.3
DDPM (modified)	DDPM	95.7	92.4

Table 7 demonstrates that $GTA\text{-}Net$ exhibits strong robustness to architectural and loss-function perturbations within the same generative family. All modified variants are attributed to their corresponding base architectures with very high confidence and accuracy, indicating that minor structural changes, normalization strategies, or objective refinements do not significantly alter the underlying generative fingerprint. This suggests that $GGMA-$Net captures stable architecture-level priors rather than superficial implementation details. In particular, the high attribution consistency for modified $ProGAN$ and $StyleGAN$ variants implies that progressive growing and style-modulated synthesis impose persistent structural biases in feature statistics. Similarly, the strong alignment of modified $SNGAN$ and $GANformer$ variants confirms that spectral normalization and transformer-based attention introduce family-specific artifact patterns that remain invariant under moderate architectural adjustments. The diffusion variant also preserves its attribution signature, indicating that iterative denoising dynamics generate stable, process-driven artifacts resistant to architectural tuning. Overall, these results validate that $GTA\text{-}Net$ learns generative-family representations that are invariant to intra-family modifications, reinforcing its suitability for robust real-world attribution where exact implementation details may vary.

These results collectively demonstrate that both training objectives and architectural design choices fundamentally shape the attribution behavior of $GTA\text{-}Net$. Across all categories, diffusion-based models exhibit the strongest and most consistent attribution confidence, indicating that iterative denoising and latent compression introduce stable, process-driven artifacts. In contrast, adversarial CNN-based GANs cluster according to shared inductive biases and optimization objectives, while transformer-based and implicit generators show comparatively lower confidence due to architectural divergence from convolutional priors. Conditional image-to-image models further reveal that conditioning pathways attenuate artifact strength, yet the under-

lying generative objective remains the dominant factor governing attribution similarity. The confidence score therefore serves not only as a measure of classification certainty but also as an indicator of generative-family alignment in the learned embedding space. High confidence reflects strong architectural or objective-level consistency with known model families, whereas lower confidence may signal architectural novelty, hybrid training paradigms, or weakened artifact persistence. Importantly, the high intra-family attribution accuracy observed for architectural variants confirms that *GTA-Net* captures stable generative priors rather than overfitting to specific implementations. Overall, these findings suggest that *GTA-Net* generalizes beyond closed-set recognition and instead learns structured, objective-level generative representations, enabling robust nearest-class attribution for unseen models in realistic forensic scenarios.

5.7 Comparison

We compared our *GTA-Net* with existing deepfake attribution techniques of *LeveFreq* Frank et al. (2020), *AttNet* Yu et al. (2019), *DNA-Det* Yang et al. (2022), *Openworld-GAN* Girish et al. (2021), *POSE* Yang et al. (2023), *CPL* Sun et al. (2023), Handcrafted Li et al. (2024), ExDa Mu et al. (2025), Decoupled attention Zheng et al. (2025) and Frequency Zhang et al. (2025). Among these, *DNA-Det* and *POSE* are designed for cross-seed attribution, but it has been tested exclusively on *ProGAN*. The techniques, *LeveFreq*, *AttNet*, and Frequency focus on *GAN* attribution for models seen during training, meaning both training and testing data originate from the same generative model. *Openworld-GAN*, *POSE*, *CPL*, and Handcrafted Li et al. (2024) are used for open-set model attribution but achieves accuracy of around 70%. To ensure a fair comparison, we trained all the models using the same dataset used for training *GTA-Net*, following their respective experimental setups. The test results are shown in Table 8. From these results, it is clear that the existing attribution models except *LeveFreq* performed well on closed-set data. The cross-seed attribution results were not entirely satisfactory, as the attribution accuracy for *ProGAN* and *WGAN* was only slightly above 85%. Furthermore, a significant drop in accuracy was observed for *AttNet*, *LeveFreq*, *DNA-Det*, *Openworld-GAN*, *POSE*, Handcrafted, and Frequency when tested on the cross-seed data of other generative models. We also tested the existing techniques with the cross-dataset and fine-tuned data. The accuracy of the existing techniques dropped significantly, as shown in the last three columns of Table 8. This implies that *GTA-Net* outperforms the existing techniques in attribution under cross-seed, cross-dataset and fine-tuning scenarios.

Table 8: Comparison of *GTA-Net* with existing deepfake attribution techniques in different attribution scenarios. The table shows accuracies obtained under closed-set, cross-seed, cross-dataset, and fine-tuned scenarios.

Technique	**Closed-Set**	**Cross-Seed**	**Cross-Dataset**	**Normal Fine-Tune**	**Cross Fine-Tune**
LeveFreq	72.1	52.4	48.6	61.3	42.7
AttNet	84.8	61.9	37.2	64.1	35.4
DNA-Det	89.1	72.5	67.1	79.5	70.2
Openworld-GAN	81.8	68.7	59.2	70.3	58.1
POSE	90.3	68.7	84.2	86.7	82.4
CPL	94.1	84.5	80.1	89.4	79.2
Handcrafted	97.3	86.2	78.2	81.3	78.1
ExDa	77.8	73.2	61.1	82.6	68.7
Decoupled Attention	95.25	69.7	88.2	81.3	78.1
Frequency	84.6	79.9	76.7	92.8	81.4
GTA-Net	**98.8**	**98.3**	**98.0**	**98.2**	**97.9**

5.8 Ablation Study

To quantify the importance of each module within *GTA-Net*, we conduct an ablation study by removing different components and training the model accordingly. The results, summarized in Table 9, reveal that:

- Removing the Multi-Head Self-Attention module decreases accuracy by 19%, confirming its role in effective feature refinement. Without this module, the model struggles to refine distinguishing features, leading to increased misclassifications, especially in cross-dataset and fine-tuned scenarios.
- Excluding the Supervised Contrastive Learning module reduces feature separability, as evidenced by overlapping clusters in the *t-SNE* visualization. This suggests that contrastive learning plays a key role in enhancing representation learning, ensuring robustness even when models undergo retraining with different seeds or datasets.
- Without the Denoising Autoencoder, model performance degrades by 18%, highlighting its contribution to feature extraction. The autoencoder helps in mitigating variations introduced by different training conditions, ensuring that *GTA-Net* remains effective in identifying the underlying generative architecture despite changes in model initialization, loss functions, or dataset variations.

Table 9: Ablation study results for GTA-Net.

Model Configuration	**Accuracy (%)**
Full GTA-Net	98.8
Without Multi-Head Attention	81.2
Without Supervised Contrastive Loss	67.4
Without Denoising Autoencoder	82.3

To further evaluate the sensitivity of *GTA-Net* to input resolution, we trained the attribution pipeline at multiple image scales (128×128, 256×256 , 512×512, and 1024×1024) as shown in Table 10. While higher resolutions provide increased spatial detail, the attribution performance exhibits only marginal improvement beyond 128×128. This suggests that generative fingerprints exploited by *GTA-Net* are primarily encoded in mid- to low-frequency statistical patterns rather than high-frequency pixel-level detail. Diffusion artifacts, convolutional inductive biases, and adversarial texture inconsistencies appear to be resolution-invariant once sufficient spatial structure is preserved. Consequently, scaling the input resolution increases computational cost (memory footprint and training time grow quadratically with spatial dimension) without yielding statistically significant gains in attribution accuracy. These findings indicate that 128×128 provides an optimal trade-off between computational efficiency and forensic discriminability, confirming that *GTA-Net* captures architecture-level generative priors rather than resolution-dependent visual cues.

Table 10: Effect of Input Resolution on GTA-Net Attribution Performance

Resolution	**Accuracy (%)**	**Confidence (%)**	**Cost**
128×128	98.8	88.4	$1\times$
256×256	98.9	89.1	$4\times$
512×512	97.3	89.5	$16\times$
1024×1024	98.9	89.7	$64\times$

6 Discussion

Unlike synthetic images generated directly from a trained model, real-world manipulations sometimes involve edits such as face swaps, deepfake augmentations, and attribute manipulation techniques that do not follow a standard generative process. To validate the generalizability of our method in real-world scenarios, we evaluated *GTA-Net* on the *ForgeryNet* dataset He et al. (2021), which comprises 15 distinct manipulation techniques based on generative models. We randomly sampled $2,000$ manipulated images from each technique and performed attribution. Unlike traditional generative models that synthesize images from scratch, *ForgeryNet* primarily involves manipulation of existing images, such as identity swapping, expression transfer, and facial reenactment. As a result, the confidence scores for attribution were slightly lower due to the subtler generative traces. Nonetheless, the attribution accuracy remained strong, despite *GTA-Net* being trained exclusively on images fully generated by *GAN*s or diffusion models, without exposure to manipulated

images during training which can be seen in Table 11. Many face manipulation pipelines involve multiple modules such as encoders, facial landmark extractors, motion transfer networks, and domain adaptation layers. However, our attribution focuses on the generative component, such as the backbone of the technique, the impact of the loss function, preprocessing/postprocessing modules, and training strategies, which jointly influence the final output. These factors may obscure the dominant generative signature, making precise attribution more challenging despite the model's overall strong performance. Still, this demonstrates the robustness of *GTA-Net* in detecting generative fingerprints even in complex, post-processed manipulations.

Table 11: Attribution results of GTA-Net on ForgeryNet.

Manipulation Technique	**Attributed Model**	**Confidence**	**Accuracy**
FaceShifter	StyleGAN2	71.2%	86.5%
FS-GAN	ProGAN	59.8%	80.2%
DeepFakes	DCGAN	54.1%	77.0%
BlendFace	StyleGAN2	68.9%	85.0%
MMReplacement	WGAN	51.6%	75.3%
DeepFakes-StarGAN	SNGAN	48.3%	78.9%
Talking Head Video	DDPM	60.4%	80.5%
ATVG-Net	DDPM	62.7%	82.0%
StarGAN-BlendFace	SNGAN	46.2%	84.1%
First Order Motion	DDPM	55.9%	80.6%
StyleGAN2	StyleGAN2	91.7%	90.2%
MaskGAN	DCGAN	56.0%	76.5%
StarGAN2	SNGAN	67.4%	85.1%
SC-FEGAN	StyleGAN2	70.6%	86.9%
DiscoFaceGAN	ProGAN	52.7%	78.0%

7 Conclusion

In this paper, we proposed *GTA-Net* for attributing the generative model-based images to their original source generative model. To achieve this, we have developed a technique that can extract traces of model architecture and training pipeline from the generated images. *GTA-Net* filters out data-dependent features, allowing the feature extraction network to focus or concentrate on characteristics unique to the generative model such as its architecture and training process. To demonstrate the generalization capability of *GTA-Net*, we evaluated it across six different *GAN*s and three different diffusion models. Our results show that *GTA-Net* successfully attributes generated images to their corresponding generative models, even when the images originate from models that were unseen during training. We also compared our approach with existing attribution methods to demonstrate the effectiveness of *GTA-Net*. Looking ahead, we plan to expand our technique to attribute images that are modified through real-world editing with higher accuracy. This will enable *GTA-Net* to attribute both synthesized and manipulated images to their respective sources.

References

Naroa Amezaga and Jeremy Hajek. Availability of voice deepfake technology and its impact for good and evil. In *Proceedings of the 23rd Annual Conference on Information Technology Education*, pp. 23–28, 2022.

Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. Image generators with conditionally-independent pixel synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14278–14287, 2021.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *ICML*, 2017.

Vishal Asnani, Xi Yin, Tal Hassner, and Xiaoming Liu. Reverse engineering of generative models: Inferring model hyperparameters from generated images. *IEEE TPAMI*, 2023.

Yutong Bai, Shitong Zhang, Zhuohan Zhou, Zi Lin, Ji Geng, Zhiqian Wang, Shang Xu, Zhen Wang, Chunyuan Yang, Kurt Keutzer, et al. Latent consistency models: Synthesizing high-resolution images with few-step inference. *preprint arXiv:2305.14090*, 2023.

Fan Bao, Shen Nie, Kaiwen Xue, Yue Cao, Chongxuan Li, Hang Su, and Jun Zhu. All are worth words: A vit backbone for diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 22669–22679, 2023.

Marc G Bellemare, Ivo Danihelka, Will Dabney, Shakir Mohamed, Balaji Lakshminarayanan, Stephan Hoyer, and Rémi Munos. The cramer distance as a solution to biased wasserstein gradients. *arXiv preprint arXiv:1705.10743*, 2017.

David Berthelot, Thomas Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *ICLR*, 2019.

CBS News. Pope francis' fake puffer jacket photo reveals the power and peril of ai. `https://www.cbsnews.com/news/pope-francis-puffer-jacket-fake-photos-deepfake-power-peril-of-ai/`, 2024. Accessed: 2025-01-09.

Lucy Chai, David Bau, Ser-Nam Lim, and Phillip Isola. What makes fake images detectable? understanding properties that generalize. In *ECCV*, pp. 103–120. Springer, 2020.

Ashish Chauhan. Deepfakes strike deep in gujarat .. *Times of India*, 2023. URL `https://timesofindia.indiatimes.com/city/ahmedabad/deepfakes-strike-deep-in-gujarat-too/articleshow/105136316.cms`.

Zhiwei Chen and Others. Mmnet: Multi-collaboration framework for robust deepfake detection. *IEEE/CVF CVPR*, 2022.

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

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE/CVF CVPR*, pp. 8188–8197, 2020.

Yuzhen Ding, Nupur Thakur, and Baoxin Li. Does a gan leave distinct model-specific fingerprints. In *BMVC*, 2021.

Ricard Durall, Margret Keuper, and Janis Keuper. Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In *IEEE/CVF CVPR*, pp. 7890–7899, 2020.

Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

France24. France 24 journalist impersonated in new deepfake video. `https://www.france24.com/en/tv-shows/truth-or-fake/20240215-france-24-journalist-impersonated-in-new-deepfake-video`, 2024. Accessed: 2025-01-09.

Joel Frank, Thorsten Eisenhofer, Lea Schönherr, Asja Fischer, Dorothea Kolossa, and Thorsten Holz. Leveraging frequency analysis for deep fake image recognition. In *ICML*, pp. 3247–3258. PMLR, 2020.

Sharath Girish, Saksham Suri, Sai Saketh Rambhatla, and Abhinav Shrivastava. Towards discovery and attribution of open-world gan generated images. In *IEEE/CVF ICCV*, pp. 14094–14103, 2021.

Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10696–10706, 2022.

Luca Guarnera, Oliver Giudice, and Sebastiano Battiato. Deepfake detection by analyzing convolutional traces. In *IEEE/CVF CVPR workshops*, pp. 666–667, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition.e-prints. *preprint arXiv:1512.03385*, 10, 2015.

Yinan He, Bei Gan, Siyu Chen, Yichun Zhou, Guojun Yin, Luchuan Song, Lu Sheng, Jing Shao, and Ziwei Liu. Forgerynet: A versatile benchmark for comprehensive forgery analysis. In *IEEE/CVF CVPR*, pp. 4360–4369, 2021.

Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020a.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 2020b.

Drew A Hudson and Larry Zitnick. Generative adversarial transformers. In *International conference on machine learning*, pp. 4487–4499. PMLR, 2021.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *IEEE/CVF CVPR*, pp. 1125–1134, 2017.

Hyeonseong Jeon, Youngoh Bang, Junyaup Kim, and Simon S Woo. T-gd: Transferable gan-generated images detection framework. *preprint arXiv:2008.04115*, 2020.

Yifan Jiang, Shiyu Chang, and Zhangyang Wang. Transgan: Two pure transformers can make one strong gan, and that can scale up. *Advances in Neural Information Processing Systems*, 34:14745–14758, 2021.

Matthew Joslin and Shuang Hao. Attributing and detecting fake images generated by known gans. In *IEEE Security and Privacy Workshops*, pp. 8–14, 2020.

Tero Karras. A style-based generator architecture for generative adversarial networks. *preprint arXiv:1812.04948*, 2019.

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *preprint arXiv:1710.10196*, 2017.

Tero Karras, Samuli Laine, and Timo Aila. Analyzing and improving the image quality of stylegan. *CVPR*, 2020.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *Advances in neural information processing systems*, 34:852–863, 2021a.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Alias-free generative adversarial networks. *NeurIPS*, 34:852–863, 2021b.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *NeurIPS*, 33:18661–18673, 2020.

Changhoon Kim, Yi Ren, and Yezhou Yang. Decentralized attribution of generative models. *preprint arXiv:2010.13974*, 2020.

Kwonjoon Lee, Huiwen Chang, Lu Jiang, Han Zhang, Zhuowen Tu, and Ce Liu. Vitgan: Training gans with vision transformers. *arXiv preprint arXiv:2107.04589*, 2021a.

Kwot Sin Lee, Ngoc-Trung Tran, and Ngai-Man Cheung. Infomax-gan: Improved adversarial image generation via information maximization and contrastive learning. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 3942–3952, 2021b.

Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.

Jialiang Li, Haoyue Wang, Sheng Li, Zhenxing Qian, Xinpeng Zhang, and Athanasios V Vasilakos. Are handcrafted filters helpful for attributing ai-generated images? In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 10698–10706, 2024.

Wenbo Li, Zhe Lin, Kun Zhou, Lu Qi, Yi Wang, and Jiaya Jia. Mat: Mask-aware transformer for large hole image inpainting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10758–10768, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. *ECCV*, 2014.

Ming Liu, Yukang Ding, Min Xia, Xiao Liu, Errui Ding, Wangmeng Zuo, and Shilei Wen. Stgan: A unified selective transfer network for arbitrary image attribute editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3673–3682, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of ICCV*, 2015.

Mario Lučić, Michael Tschannen, Marvin Ritter, Xiaohua Zhai, Olivier Bachem, and Sylvain Gelly. High-fidelity image generation with fewer labels. In *International conference on machine learning*, pp. 4183–4192. PMLR, 2019.

Stephane G Mallat. A theory for multiresolution signal decomposition: the wavelet representation. *IEEE TPAMI*, 11(7):674–693, 1989.

Francesco Marra, Diego Gragnaniello, Luisa Verdoliva, and Giovanni Poggi. Do gans leave artificial fingerprints? In *IEEE conference on multimedia information processing and retrieval*, pp. 506–511, 2019.

Ghazal Mazaheri and Amit K Roy-Chowdhury. Detection and localization of facial expression manipulations. In *IEEE/CVF WACV*, pp. 1035–1045, 2022.

Changtao Miao, Zichang Tan, Qi Chu, Huan Liu, Honggang Hu, and Nenghai Yu. F 2 trans: High-frequency fine-grained transformer for face forgery detection. *IEEE TIFS*, 18:1039–1051, 2023.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *ICLR*, 2018.

Wenpeng Mu, Zheng Li, Qiang Xu, Xinghao Jiang, and Tanfeng Sun. Exda: Towards universal detection and plug-and-play attribution of ai-generated ex-regulatory images. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 11512–11521, 2025.

NBC News. More ai-generated child sex abuse material is being posted online. `https://www.nbcnews.com/tech/security/ai-generated-child-sex-abuse-material-posted-online-rcna162169`. Accessed: 2024-10-4.

Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021.

NPR. Deepfake video of zelenskyy highlights experts' concerns about war manipulation. `https://www.npr.org/2022/03/16/1087062648/deepfake-video-zelenskyy-experts-war-manipulation-ukraine-russia`, 2022. Accessed: 2025-01-09.

Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2337–2346, 2019.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Ivan Perov, Daiheng Gao, Nikolay Chervoniy, Kunlin Liu, Sugasa Marangonda, Chris Umé, Carl Shift Facenheim, Luis RP, Jian Jiang, Sheng Zhang, et al. Deepfacelab: Integrated, flexible and extensible face-swapping framework. *preprint arXiv:2005.05535*, 2020.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *preprint arXiv:1511.06434*, 2015.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. *IEEE/CVF CVPR*, 2022.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022a.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Ghasemipour, Raphael Gontijo-Lopes, Yilun Karagol, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022b.

Axel Sauer, Kashyap Chitta, Jens Müller, and Andreas Geiger. Projected gans converge faster. *Advances in Neural Information Processing Systems*, 34:17480–17492, 2021.

Zhimin Sun, Shen Chen, Taiping Yao, Bangjie Yin, Ran Yi, Shouhong Ding, and Lizhuang Ma. Contrastive pseudo learning for open-world deepfake attribution. In *IEEE/CVF ICCV*, pp. 20882–20892, 2023.

Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 2149–2159, 2022.

Justus Thies, Michael Zollhofer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *IEEE/CVF CVPR*, pp. 2387–2395, 2016.

Sheng-Yu Wang, Oliver Wang, Richard Zhang, Andrew Owens, and Alexei A Efros. Cnn-generated images are surprisingly easy to spot... for now. In *IEEE/CVF CVPR*, pp. 8695–8704, 2020.

Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018.

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

Po-Wei Wu, Yu-Jing Lin, Che-Han Chang, Edward Y Chang, and Shih-Wei Liao. Relgan: Multi-domain image-to-image translation via relative attributes. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 5914–5922, 2019.

Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Xinsheng Xuan, Bo Peng, Wei Wang, and Jing Dong. Scalable fine-grained generated image classification based on deep metric learning. *preprint arXiv:1912.11082*, 2019.

Tianyun Yang, Ziyao Huang, Juan Cao, Lei Li, and Xirong Li. Deepfake network architecture attribution. In *AAAI*, volume 36, pp. 4662–4670, 2022.

Tianyun Yang, Danding Wang, Fan Tang, Xinying Zhao, Juan Cao, and Sheng Tang. Progressive open space expansion for open-set model attribution. In *IEEE/CVF CVPR*, pp. 15856–15865, 2023.

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

Ning Yu, Larry S Davis, and Mario Fritz. Attributing fake images to gans: Learning and analyzing gan fingerprints. In *IEEE/CVF ICCV*, pp. 7556–7566, 2019.

Ning Yu, Vladislav Skripniuk, Dingfan Chen, Larry Davis, and Mario Fritz. Responsible disclosure of generative models using scalable fingerprinting. *preprint arXiv:2012.08726*, 2020.

Ning Yu, Vladislav Skripniuk, Sahar Abdelnabi, and Mario Fritz. Artificial fingerprinting for generative models: Rooting deepfake attribution in training data. In *IEEE/CVF ICCV*, pp. 14448–14457, 2021.

Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *ICML*, pp. 7354–7363, 2019.

Junbin Zhang, Yixiao Wang, Hamid Reza Tohidypour, and Panos Nasiopoulos. An efficient frequency domain based attribution and detection network. *IEEE Access*, 2025.

Hanqing Zhao, Wenbo Zhou, Dongdong Chen, Tianyi Wei, Weiming Zhang, and Nenghai Yu. Multi-attentional deepfake detection. In *IEEE/CVF CVPR*, pp. 2185–2194, 2021.

Yu Zheng, Boyang Gong, Fanye Kong, Yueqi Duan, Bingyao Yu, Wenzhao Zheng, Lei Chen, Jiwen Lu, and Jie Zhou. Learning counterfactually decoupled attention for open-world model attribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 122–132, 2025.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE/CVF ICCV*, pp. 2223–2232, 2017.