Multi-angle Consistent Generative NeRF with Additive Angular Margin Momentum Contrastive Learning

Hang Zou[1], Hui Zhang[2], Yuan Zhang[1,3]*, Hui Ma[4], Dexin Zhao[1], Qi Zhang[1], Qi Li[5]
[1]China Telecom Research Institute (CTRI), China;
[2]Tianjin University of Science & Technology, China;
[3]Zhejiang University, China; [4]M.U.S.T, Macau; [5]MAIS, CASIA, China
[1]{zouh3,zhangy666}@chinatelecom.cn

Abstract

The NeRF and GAN-based GIRAFFE algorithm has drawn a lot of attention because of its controllable image production capacity. However, the consistency of GIRAFFE rendering results from different perspectives of the same object is not stable. The reasons are twofold: First, the optimization goal of GIRAFFE is only concerned with whether the generated image resembles the real image or not. Second, GIRAFFE could learn knowledge implicitly to complement the feature deformation of large camera angle change, which may introduce uncontrollable generation mode resulting in low consistency of the 3D object. This limits its application in fields such as digital person generation and biometric identity. In this paper, We introduce an additional Encoder to form a momentum-based Contrastive Learning with the Discriminator of GAN. In addition, we propose an AamNCE loss to train our model which introduces an additive angular margin to the positive sample pairs. In brief, the proposed framework could be regarded as a new paradigm of GAN and Contrastive Learning. The Contrastive Learning improves the characteristic expression ability of the model, and the AamNCE loss makes the category boundaries of the generated images more explicit. The experimental results demonstrate that our method maintains the consistency of face identity well in the multi-angle rotation of the face dataset.

1. Introduction

The introduction of NeRF [28] has made implicit 3D reconstruction a popular area of research. Unlike traditional 3D reconstruction methods, its output is direct images rendered through a neural rendering network. This approach simplifies the process of 3D reconstruction. Since the proposal of NeRF, several works have emerged to improve its limitations. For example, Deng et al.[4] introduced SFM to address the slow training speed of a single scene in NeRF, achieving faster training speeds. Meanwhile, Li et al.[16] proposed Neural Scene Flow Fields to enable the modeling of dynamic scenes, addressing NeRF's inability to render such scenes. However, the ability to generate images is also a crucial aspect of computer vision, while those improvements have largely focused on the theme of reconstruction.

Figure 1. The figure shows faces generated and rendered with large-angle by our method. It has clear and realistic rendering results with good variety.

Generative models have proven to be highly effective in generating images, as seen in GAN [9], VAE [15], Diffusion Model [12], among others. However, compared with regular images, 3D shapes contain a greater amount of information, which requires more complex computations for their generation. Recently, researchers have attempted to incorporate 3D representations into generative models to explore the generation of 3D shapes based on GANs, as demonstrated by works such as [11] and [29]. Despite successfully achieving this goal, the generated results lack sufficient fine-grained details, which is likely due to the models' limited ability to effectively learn features from 3D representations such as voxels. Unlike traditional methods, NeRF implicitly learns 3D shapes through a neural rendering network and directly outputs rendered images. This approach simplifies the process of 3D reconstruction and avoids the use of 3D representation methods such as vox-

*Corresponding author.

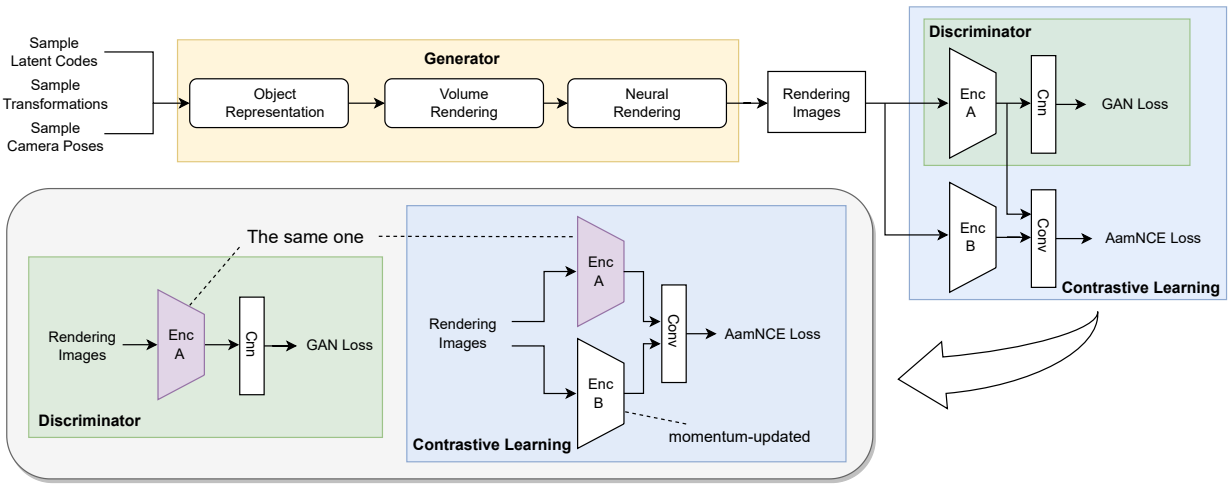


Figure 2. The figure presents the overall framework of our method, which comprises three main components: Generator, Discriminator, and Contrastive Learning. The Discriminator is composed of the $EncA$ block and the Cnn block. Simultaneously, the Contrastive Learning is composed of the $EncA$ block, $EncB$ module, and $Conv$ layer. During training, we add up the loss of GAN and the loss of Contrastive Learning, jointly train and update all three components.

els and primitives. Therefore, the combination of GAN and NeRF, which is called GIRAFFE [30], utilizes the powerful representation capabilities of neural rendering networks to achieve highly refined 3D shape generation and image rendering. It introduces the NeRF technique into the generation field and provides a new approach to 3D shape generation.

The composition of GIRAFFE is similar to GANs, consisting of a Generator and a Discriminator. The Generator in GIRAFFE is a decoder responsible for converting randomly sample code from the latent space into implicit 3D objects, which are rendered into multi-view images using neural rendering blocks. The Discriminator is designed to distinguish between "real" images from the data and "fake" images rendered by the Generator. The Generator and Discriminator engage in an ongoing adversarial training process, which combines a joint update mechanism that enables the Generator to generate implicit 3D objects smoothly and render multi-view images with various perspectives.

Similar to GANs, GIRAFFE is proficient at image rendering, and its loss function could guide it to render images that resemble the input dataset. However, since we aim to learn how to generate realistic 3D objects, merely constraining the image rendering to be realistic is inadequate. In other words, if the Discriminator only judges whether the rendered images are real or fake, the Generator would only make the rendered images as "real" as possible, without considering whether the rendered images of the same 3D object are consistent. Due to the powerful learning and memory capabilities of the neural rendering block, this could lead to differences between images rendered from different viewpoints of the same object, such as changes in gender or identity in larger rendering viewpoints of a human face, as shown in 4. This is a question that requires careful consideration, which means that if the rendered images from the same 3D object are not consistent enough, the Generator would render independent "real images" rather than learning the "real 3D structure". This raises concerns about whether GIRAFFE has truly learned how to generate reasonable "3D structures".

To address this issue, we introduce additional constraints to restrict the GIRAFFE, making it not only consider whether the rendered image is realistic but also maintain consistency among different rendering views of the same object to achieve the generation of realistic 3D shapes. We use contrastive learning to constrain the training process of GIRAFFE, due to the SimCLR [2], MoCo [10], SwAV [1] are effective in feature extraction and classification tasks. The contrastive learning methods focus solely on data contrast in the latent feature space, which makes them easier to optimize and generalize. In this work, we follow the MoCo to form contrastive learning in GIRAFFE. Specifically, we regard the Discriminator in GIRAFFE as one encoder and introduce another encoder to construct a contrastive learning framework.

For the loss function, MoCo uses InfoNCE loss, which aims to reduce intra-class distance and increase inter-class distance, making class boundaries clearer and improving classification performance. In this work, we consider images rendered from the same object as intra-class relations and images from different objects as inter-class relations. This means that our loss function could be constructed by InfoNCE. On this basis, we aim to propose a loss function

that enhances the distinguishing ability of class boundaries. And there are many effective loss functions [33] [26] [3] have demonstrated that adding a margin could improve classification results. Inspired by the Additive Angular Margin proposed by Arcface [3], we propose the AamNCE loss based on the InfoNCE loss. It uses the cosine angle to describe the similarity between sample pairs and applies an additional angle penalty to positive sample pairs to enhance intra-class compactness and inter-class differences, thereby improving intra-class similarity and inter-class separability. Fig. 1 shows some faces generated and rendered with large-angle by our method.

To validate our method, we focus on the generation and rendering of faces because of the rich details and high distinguishability. We compare the identity of rendered images and quantitatively evaluate them by the face recognition system LightCNN [35]. Experimental results demonstrate that our method is effective in maintaining the consistency of the same object. Our method is effective for any dataset. However, other datasets cannot evaluate whether the rendered images of the same object are consistent, we could only judge them by the naked eye. Thus, we only tested on face datasets in this work.

In this work, we make the following contributions:

- We propose a model that incorporates contrastive learning to optimize GIRAFFE and enforce consistency constraints on the shape and appearance of objects.
- We introduce an AamNCE loss, which enhances both intra-class compactness and inter-class differences, leading to improved intra-class similarity and inter-class separability.

2. Related Work

Generative Models. In recent years, generative models have become a popular topic in the field of computer vision, enabling the creation of new data by learning the distribution of input datasets. There are several mainstream frameworks for generative models, including GANs [9], VAEs [15], and diffusion models [12]. They have different approaches, but all demonstrate strong generative capabilities. GANs have been effectively used in numerous fields, such as CycleGAN [40], which introduces Cycle-Consistent loss to perform style transfer between unpaired images. The diffusion models leverage the diffusion process to learn from the dataset. The representative models include DALL-E 2 [31] and Stable Diffusion [32], etc.

Neural Rendering. Neural rendering has become a hot topic in the field of 3D modeling, since the proposal of NeRF [28]. Its ability to directly render multi-angle images through implicit 3D modeling avoids the tediousness of traditional 3D modeling methods. Due to the model's need to calculate each pixel of each image during the rendering process, the training speed of NeRF is slow. Depth-supervised NeRF [4] and FastNeRF [8] have proposed some improvements to address this issue. PixelNeRF [36] is proposed to solve NeRF's high requirements on the number of input views. It implements reconstruction under a small number of view inputs by improving the generalization of the model. And there are many other methods and directions for NeRF such as its generative ability.

Contrastive Learning. Contrastive learning models aim to train a feature-extracting encoder that could be utilized in downstream tasks. It has amazing effects in the field of feature extraction and classification and even outperforms supervised learning on computer vision tasks. Contrastive learning has some representative works, such as SimCLR [2] which generates new samples by data augmentation to form positive and negative sample pairs. MoCo [10] introduces two encoders to reduce computational requirements and sets one of them with momentum updates to alleviate the gap between the two encoders.

Face-Anti Spoofing. Face Anti-Spoofing (FAS) aims at protecting the Face Recognition system (FR) from various presentation attacks, ranging from print-attack [38, 39], replay-attack [7, 20] and mask-attack [6, 23]. Based on these datasets, the recent series of FAS competitions [5, 18, 19, 22, 37] have driven the development of this community. CMA-FAS [21] consists of a Modality Translation Network and a Modality Assistance Network to close the visible gap between different modalities via a generative model. MA-ViT [17] enables flexible testing of any given modal samples with a Modality-Agnostic Transformer Block (MATB). FM-ViT [24] retains a specific branch for each modality to capture different modal information and introduces the CMTB to guide each modal branch to mine potential features. MDIL [34] not only learns knowledge well from the new domain but also maintains the performance of previous domains stably. CFPL-FAS [25] is the first work to explore Domain Generalization FAS via textual prompt learning.

3. Method

In this work, we propose a generative neural rendering framework based on GIRAFFE [30]. We enhance the feature extraction capability of the Discriminator by building momentum contrastive learning. The introduction of AamNCE loss makes the class boundaries clearer when the Generator generates objects. The joint improvement of the Discriminator and Generator strengthens the consistency constraints on the shape and appearance of the generated objects, which is equivalent to an implicit improvement in the requirement for generating reasonable 3D shapes with individual consistency.

In this section, we provide a detailed description of our proposed framework. First, in Sec. 3.1, we analyze the fundamental implementation of GIRAFFE, which is based on implicit 3D object generation and rendering. Next,

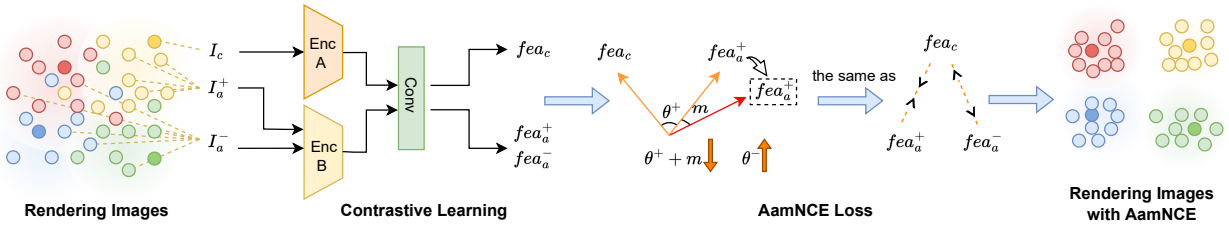


Figure 3. This figure shows some details of the AamNCE loss. We divide the rendered images of the Generator into $I_c, I_a^+, I_a^- \in \hat{I}$, and obtain their features fea_c, fea_a^+, and fea_a^- through $EncA$ and $EncB$. We define the cosine angle between fea_c and fea_a^+ as θ^+ and the cosine angle between fea_c and fea_a^- as θ^-, and the m is an additional penalty on θ^+. For AamNCE, the optimization goal is to reduce the $\theta^+ + m$ and increase the θ^-, which is equivalent to making fea_c similar to fea_a^+ and different from fea_a^-. The AamNCE could improve intra-class similarity and inter-class separability.

in Sec. 3.2, we explain how to add momentum-updated encoder to GIRAFFE to incorporate contrastive learning. We also propose a novel loss function, AamNCE, for generating objects with clearer class boundaries. Finally, in Sec. 3.3, we present the overall framework of our method.

3.1. Neural Feature Generating and Rendering

The GIRAFFE [30] is similar in the overall structure to vanilla GAN [9]. It consists primarily of a Generator and a Discriminator. For the Generator of GANs, its inputs are latent codes sampled from a predefined distribution, and its outputs are the target data. In GIRAFFE, the Generator also takes samples from the latent space as input, and the sampled codes are processed through Object Representation, Volume Rendering, and Neural Rendering to output multi-angle rendered images of the implicitly 3D-generated objects.

The sampling process of the Generator is divided into three parts. The first part is the sampling for the image content, which is similar to the sampling process in vanilla GAN. The second part is the sampling of affine transformations. The third part is the sampling for the camera pose. They could be expressed as follows:

$$z_s, z_a \sim N(0, I); T \sim p_T; \boldsymbol{\xi} \sim p_{\boldsymbol{\xi}} \tag{1}$$

In this function, z_s and z_a are samples from predefined latent distributions $N(0, I)$. The model generates image content based on these latent codes. T represents object-level affine transformation, which is used to guarantee the reliability of rendering images during random transformations. The $\boldsymbol{\xi}$ is the camera pose. And the GIRAFFE defines p_T and $p_{\boldsymbol{\xi}}$ as uniform distributions over the dataset-dependent valid object transformations and camera elevation angles, respectively.

After obtaining the sampling information, the Object Representation block processes the camera parameters $\boldsymbol{\xi}$ by Ray Casting and 3D point Sampling algorithms and gets the 3D point $x \in \mathbb{R}^3$ and viewing direction $d \in \mathbb{S}^2$. After that, the model represents the implicit 3D object from (x, d) through an MLP network called h_θ. Finally, we obtain the density σ and feature f of the object.

$$\begin{aligned} \boldsymbol{\xi} &\mapsto (\gamma(x), \gamma(d)) \\ (\sigma, \mathbf{f}) &= h_\theta\left(\gamma\left(k^{-1}\right)(\mathbf{x})\right), \gamma\left(k^{-1}(\mathbf{d})\right), \mathbf{z}_s, \mathbf{z}_a \end{aligned} \tag{2}$$

The function $k(x)$ is derived from T. GIRAFFE [30] provides further information about this function. And the γ is defined as follows:

$$\begin{aligned} \gamma(t, L) = (&\sin\left(2^0 t\pi\right), \cos\left(2^0 t\pi\right), \sin\left(2^1 t\pi\right), \cos\left(2^1 t\pi\right), \\ &\ldots, \sin\left(2^L t\pi\right), \cos\left(2^L t\pi\right)) \end{aligned} \tag{3}$$

After getting $(\sigma, \mathbf{f})$, we put them into the Volume Rendering block [13], denoted as π_{vol}, to obtain the feature map. We express this process as $I_v = \pi_{vol}(\sigma, \mathbf{f})$. Finally, we put I_v into the final part of the Generator - the Neural Rendering block which is denoted as π_θ^{neural}, to obtain rendering images I. We express this process as $\hat{I} = \pi_\theta^{neural}(I_v)$, and $\hat{I}$ is the final output of the Generator.

So for the Generator G_θ, it could be described by:

$$\begin{aligned} &\hat{I} = G_\theta(z_s, z_a, T, \boldsymbol{\xi}); \\ &z_s, z_a \sim N(0, I); T \sim p_T; \boldsymbol{\xi} \sim p_{\boldsymbol{\xi}} \end{aligned} \tag{4}$$

We have introduced the Generator component. In GANs, the other component is the Discriminator. The Discriminator takes the images generated by the Generator and the images from the input dataset, and its role is to distinguish between the generated images and the images from the dataset, also known as "real" or "fake" classification. It assigns low scores to the generated images, letting the Generator-generate more realistic images. GIRAFFE also includes a Discriminator, similar to GANs, denoted as D_ϕ. It is used to distinguish between the rendering images rendered by the Generator and the images from the input

dataset. It assigns lower scores to the rendering images. Therefore, for GIRAFFE, its overall loss function could be expressed as follows:

$$\begin{aligned}&\mathcal{J}(\theta,\phi)_{GIRAFFE}=\mathcal{J}(\theta,\phi)_{GAN}+\mathcal{J}(\phi)_{penalty}\\&=\mathbb{E}_{z_s,z_a\sim\mathcal{N},T\sim p_T,\boldsymbol{\xi}\sim p_\xi}\left[f\left(D_\phi\left(G_\theta\left(\{z_s,z_a,T\},\boldsymbol{\xi}\right)\right)\right)\right]\\&+\mathbb{E}_{I\sim p_D}\left[f\left(-D_\phi(I)\right)-\lambda\left\|\nabla D_\phi(I)\right\|^2\right]\end{aligned} \tag{5}$$

This function consists of two parts: the GAN loss function and the R_1 gradient penalty [27]. In addition, $f(t) = -\log(1+\exp(-t))$, $\lambda = 10$, and $p_{\mathcal{D}}$ is the data distribution.

3.2. Contrastive Learning

In this section, we describe our main contribution which adding an extra encoder with momentum updates to GIRAFFE to construct contrastive learning. And we will also introduce the AamNCE loss in this section.

In Sec. 3.1, we introduce the Discriminator of GIRAFFE. Like the Discriminator in GANs, it is an encoder block. In simple terms, the Discriminator of GIRAFFE is a simple binary classification model. One class is images from the input dataset, and the other one is rendering images rendered from the Generator. According to its loss Eq. (5), we find that the GIRAFFE indeed treats all generated images as the same class, i.e. "fake images".

This means that during training, the optimization goal is to enable the Generator G_θ to generate real images, without considering the relation between images rendered from the same object. If the model only judges whether an image is "real" or "fake", the optimization goal of the model will shift from learning how to generate a real 3D object to an easier task of generating some realistic 2D images. That makes the original intention of GIRAFFE's design not fulfilled. The rendering results show that GIRAFFE cannot achieve good consistency when rendering different views of the same object, especially when rendering from larger viewpoints. As shown in Fig. 4.

To address this issue, we aim to constrain the optimization objective of the model, such that it not only judges a rendered image as "real" or "fake", but also ensures that images rendered from different views of the same object have homogeneity. We consider images rendered from the same object as the same class and aim to make them as similar as possible. Meanwhile, we aim to make the images from different objects as different as possible to ensure the clarity of class boundaries.

To achieve this goal, we draw inspiration from the MoCo [10] method in contrastive learning, aiming to train a more powerful Discriminator to provide stronger constraints on the optimal direction of the Generator. The MoCo method constructs two encoders, with one being set for momentum updates, to achieve excellent unsupervised feature extraction. In GIRAFFE, the Discriminator is an encoder block that extracts features from the rendered images of the Generator and performs classification to determine whether it is generated or not, thus computing the loss value. Therefore, to build a framework similar to MoCo, we add encoder in GIRAFFE, which together with the Discriminator forms the basis of contrastive learning, as shown in Fig. 2.

The same as MoCo, the additional encoder is set up for momentum updates to reduce the model's computational cost. For ease of distinction, we call the encoder serving as the Discriminator D_ϕ as $EncA$, and the momentum-based encoder as $EncB$. It should be noted that D_ϕ is not equivalent to $EncA$, as we have constructed a CNN block to classify the features extracted by $EncA$. The CNN block takes the features extracted by $EncA$ as input and outputs a score for the features, which is used to compute the GAN loss. Therefore, $EncA$ and CNN block together form D_ϕ. And, during the iterative optimization of contrastive learning, $EncA$ and $EncB$ will eventually become very similar, so either of them could serve as the Discriminator. In this work, we choose $EncA$ as the Discriminator because it updates before $EncB$, but it does not mean that only $EncA$ could serve as the discriminator. And the $EncA$ and $EncB$ are constructed with the same architecture. To make the features extracted by $EncA$ and $EncB$ as similar as possible, we connect a shared single convolutional layer named $Conv$ at the end of their last layer, to avoid errors caused by inconsistent feature representation.

To effectively train the $EncA$ and $EncB$, we adopt the InfoNCE loss proposed in MoCo. The function is:

$$\mathcal{L}_{InfoNCE}=-\log\frac{\exp\left(q\cdot k^+/\tau\right)}{\exp\left(q\cdot k^+/\tau\right)+\sum_{k^-}\exp\left(q\cdot k^-/\tau\right)} \tag{6}$$

In this function, q, k^+, and k^- are the features extracted from the encoder of MoCo. q is a query, k^+ is a positive key, and k^- is a negative key. τ is a temperature hyperparameter. q and k^+ are different augmentations from the same data, forming a positive pair. q and k^- are from different data, forming a negative pair. The purpose of InfoNCE is to make the query q as similar as possible to its positive key k^+, and as far as possible from its negative key k^-. In other words, it reduces the intra-class distance and increases the inter-class distance.

In this work, we define positive and negative sample pairs differently from MoCo. Unlike MoCo, where the model is provided with an image dataset, we use images generated by the Generator for contrastive learning. The rendered images of Generator G_theta are denoted as $\hat{I}$, which are divided into three types: $I_c, I_a^+, I_a^- \in \hat{I}$. During the rendering process, the image rendered by the cam-

era straight to the object is set as I_c, while images rendered from other camera angles of the same object are named I_a^+. Images rendered from all camera angles of other objects are named I_a^-. MoCo regards each image in the dataset as a separate class, and positive samples are defined as the different augmentation results of the same image. In this work, we define positive samples as the rendered images of the same object from different camera angles as (I_c, I_a^+). Our definition of negative samples is similar to MoCo. We consider all the rendering results of different objects, which have different face identities, as negative samples (I_c, I_a^-).

Figure 4. The figure shows the rendering images of GIRAFFE, and each row renders from the same object with different view angles. It could be observed that it does not render satisfactory images when the angle is large, with unclear edges and unnatural artifacts. In addition, there are also cases of inconsistent identity.

Due to the difference in the definition of positive and negative sample pairs from MoCo, although InfoNCE loss can be used to train our method, there is still room for improvement. Inspired by ArcFace [3], we improved the InfoNCE loss by adding an additive angular margin. Specifically, in InfoNCE loss, the similarity calculation between positive and negative sample pairs is achieved by multiplying their features, as shown in Eq. (6). However, the similarity between the two features could also be measured by their cosine distance. Therefore, we use the cosine distance to implement InfoNCE loss for the convenience of subsequent calculation. We let $sim(a,b) = a * b/||a|| * ||b||$,

$$\begin{aligned} cos\theta^+ &= sim(q,k^+) = sim(fea_c, fea_a^+) \\ cos\theta^- &= sim(q,k^-) = sim(fea_c, fea_a^-) \end{aligned} \tag{7}$$

In this function, the fea_c is the feature of I_c extracted by $EncA$ with $Conv$. The fea_a^+ and fea_a^- are the features of I_a^+ and I_a^- extracted by $EncB$ with $Conv$.

The InfoNCE loss using cosine distance could be formulated as follows:

$$\mathcal{L} = -\log \frac{exp(\cos\theta^+/\tau)}{exp(\cos\theta^+/\tau) + \sum_{\theta^-} exp(\cos\theta^-)/\tau} \tag{8}$$

The optimization objectives of Eq. (8) and Eq. (6) are similar, but they use different ways to measure the similarity between features. Eq. (8) is used in our work to apply an additional penalty to the cosine angle between intra-class features, which adds an additive angular margin, to enhance the intra-class compactness and inter-class discrepancy, and then improves intra-class similarity and inter-class separability. More details are shown in Fig. 3. And the AamNCE loss is:

$$\mathcal{L}_{AamNCE} = -\log \frac{exp(\cos\left(\theta^+ + m\right)/\tau)}{exp(\cos\left(\theta^+ + m\right)/\tau) + \sum_{\theta^-} exp(\cos\theta^-)/\tau} \tag{9}$$

where m represents the additional penalty on the cosine angle between intra-class features, which we set to 0.5.

3.3. Overall framework

According to the above introduction, we will present the complete framework of our method in this section. Its loss function consists of three parts: GAN loss, gradient penalty loss, and AamNCE loss for contrastive learning, as follows:

$$\mathcal{L}(\theta,\phi) = \mathcal{L}(\theta,\phi)_{GAN} + \mathcal{L}(\phi)_{penalty} + \alpha\mathcal{L}(\phi)_{AamNCE} \tag{10}$$

The parameter α controls the weight of AamNCE loss. In this work, we set $\alpha = 0.1$.

4. Experiments

Database and Experimental Setup. We select the CelebA-HQ [14] dataset, which consists of 30,000 high-resolution (256x256) face images, as our training dataset without performing any additional preprocessing. The reason for choosing a face dataset is that faces are highly recognizable and could more intuitively demonstrate the consistency of our rendering images of the same object. Moreover, we need a quantifiable evaluation system to demonstrate that our method's rendering images have better consistency with multi-angle than vanilla GIRAFFE. Therefore, we chose the CelebA-HQ as our training set, and our model learns to generate face objects and render them into multi-view images. And we measure the effectiveness of the model by face recognition performance with LightCNN [35].

Here are some experimental details. We form a collection of images rendered by Generator G_θ as $\hat{I}$ and divide the rendering images into three types which $I_c, I_a^+, I_a^- \in \hat{I}$. During the rendering process, the image rendered by the camera straight to the object is set as I_c, while images rendered from other camera angles of the same object are named I_a^+. Images rendered from all camera angles of other objects are named I_a^-. To evaluate the performance of our method, we conducted experiments with our method(with InfoNCE), our method(with AamNCE), and GIRAFFE to generate 1000 objects each for testing. To test the accuracy of our model for face identification, we rendered 9 different angles images(including 1 label image) for each object. The 9 different angles used for rendering the images are left 60 degrees, left 45 degrees, left 30 degrees, left 15 degrees, label(straight), right 15 degrees, right 30 degrees, right 45

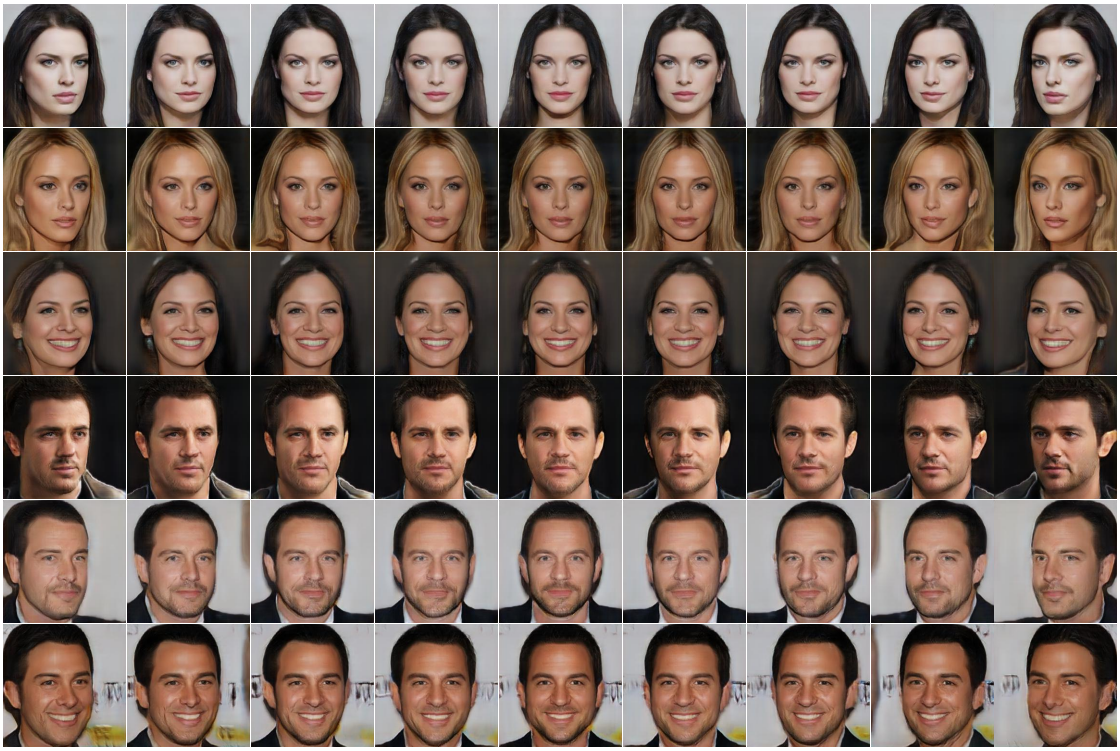


Figure 5. This figure shows the rendering images of our method (AamNCE). Each row represents the rendering images of a generated object with 9 different angles (including the label images that are rendered by the camera straight to the object). It shows that our method produces high-quality rendering images, and maintains high consistency during the rendering process of the same object.

degrees, and right 60 degrees. So there are 9000 rendering images(including 1000 label images) for each model, and 1-1 comparison identification is conducted to draw the ROC curve with LightCNN [35]. To avoid errors caused by the randomness of the generative model, we fixed the random seed to ensure the reproducibility of the testing results. The results are shown in Fig. 7.

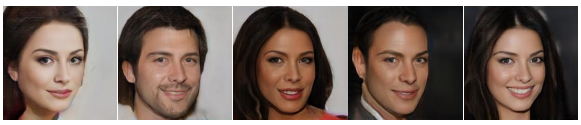
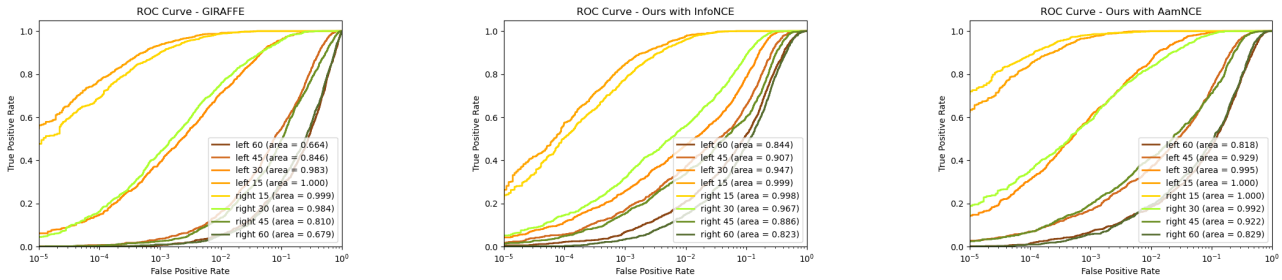


Figure 6. The figure shows some rendering images with a 60-degree angle by our method (AamNCE), where the objects' eyes are directly facing the camera, caused by the bias of the dataset.

Recognition Results of GIRAFFE. Fig. 7a shows the identity consistency of GIRAFFE's rendering images across multiple angles by ROC curve, and Tab. 1 presents its AUC values at each angle. Based on the experimental results, it is found that GIRAFFE performs well in maintaining identity consistency when the deviation angle is small, which is consistent with what we observed from the rendered images with GIRAFFE in Fig. 4. However, when the deviation angle exceeds 30 degrees, the identity consistency of GIRAFFE significantly decreases. This is because the model has not learned a good implicit 3D representation. When the angle is too large, the model relies on the strong memory of the neural network to correct the image, resulting in severe inconsistencies in the rendering of images at large angles. Moreover, the images even exhibit unrealistic artifacts and blurring under a deviation angle of 60 degrees.

Recognition Results of Ours. Fig. 7b and Fig. 7c illustrate the identity consistency of Our methods across multiple angles by ROC curve. Fig. 7b shows the experimental results using InfoNCE loss when introducing contrastive learning into GIRAFFE. Fig. 7c shows the experimental results using our proposed AamNCE loss when introducing contrastive learning into GIRAFFE. And Tab. 1 compares the face recognition performance, assessed by AUC.

For our method (InfoNCE), according to the experimental results, we found that after introducing contrastive learning, the rendered images of the same object maintain good consistency at various angles, and high AUC can also be achieved at large angles such as 45 degrees and 60 degrees. This is because we enhance the Discriminator's ability by introducing contrastive learning. At the same time, the introduction of InfoNCE improves the clarity of the generated



(a) GIRAFFE (b) Ours(InfoNCE) (c) Ours(AamNCE)

Figure 7. Quantitative Comparison - ROC curves

class boundaries by the Generator. However, at small deviation angles, the introduction of contrastive learning makes the convergence of the training more challenging, resulting in results slightly worse than the vanilla GIRAFFE.

For our method(AamNCE), according to the experimental results, we found that with the introduction of both contrastive learning and AamNCE loss, the consistency of rendering images of the same object at various angles has reached a higher level. The model not only achieves high AUC at large angle rotations, such as 45 and 60 degrees but also shows excellent performance at small angle deviations. This is because applying an additional penalty on the cosine angle of the intra-class features, i.e., adding an Additive Angular Margin, enhances the intra-class compactness and inter-class discrepancy, and then improves intra-class similarity and inter-class separability. In other words, for Our method(AamNCE), the construction of contrastive learning enhances the discriminative ability of the Discriminator, while the AamNCE loss improves the generative ability of the Generator. The combination of the two balances the game between the two modules of the GANs, resulting in significant improvements in the model's ability to generate objects and the consistency of rendered images.

Additional Analysis. Fig. 4 shows the images rendered by the GIRAFFE [30]. Fig. 5 shows the effects of objects generated by our method (AamNCE) and images rendered with multiple angles. It demonstrates that our methods achieve significant improvement in intra-class consistency, with almost no unrealistic artifacts or blurring cases. It is worth mentioning that we used the same face dataset CelebA-HQ as GIRAFFE. It is observed that the human eyes in Fig. 4 are almost all looking at the camera, which is not natural. This phenomenon is due to data set bias, that the images in the dataset are primarily looking at the camera. This phenomenon is severe in the GIRAFFE because its Discriminator considers such images more realistic. In our methods, this issue is greatly alleviated. As the rendering results illustrated in Fig. 5, most eyes face forward naturally at each rendering image. However, there are still some cases of unnatural sight directions, for example, some rendering images with a 60-degree rotation as shown in Fig. 6 with the eyes looking directly into the camera.

Angle	AUC		
	GIRAFFE	Ours(InfoNCE)	Ours(AamNCE)
left 60	0.664	**0.844**	0.818
left 45	0.846	0.907	**0.929**
left 30	0.983	0.947	**0.995**
left 15	**1.000**	0.999	**1.000**
right 15	0.999	0.998	**1.000**
right 30	0.984	0.967	**0.992**
right 45	0.810	0.886	**0.922**
right 60	0.679	0.823	**0.829**

Table 1. Quantitative Comparison - AUC values

5. Conclusion and Future Work

In this work, we introduce an additional momentum-based encoder into GIRAFFE to form a Contrastive Learning framework that enhances the Discriminator's feature extraction ability. Additionally, we propose an AamNCE loss to improve InfoNCE by introducing an angular margin, which increases the intra-class similarity and inter-class separability for the Generator. Experimental results show that our method maintains high consistency in rendering images from the same object. With its powerful generation ability, our method has promising applications in generating virtual faces and other related fields. In future work, we will focus on exploring controllable generation methods.

6. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant No.62076240), in part by the Beijing Municipal Natural Science Foundation (Grant No.4222054), in part by the Youth Innovation Promotion Association CAS (Grant No.Y2023143), and the Beijing Nova Program (Z211100002121113).

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020. 2

[2] 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. 2, 3

[3] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4690–4699, 2019. 3, 6

[4] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised nerf: Fewer views and faster training for free. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12882–12891, 2022. 1, 3

[5] Hao Fang, Ajian Liu, Jun Wan, Sergio Escalera, Hugo Jair Escalante, and Zhen Lei. Surveillance face presentation attack detection challenge. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6361–6371. IEEE, 2023. 3

[6] Hao Fang, Ajian Liu, Jun Wan, Sergio Escalera, Chenxu Zhao, Xu Zhang, Stan Z Li, and Zhen Lei. Surveillance face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 2023. 3

[7] Hao Fang, Ajian Liu, Haocheng Yuan, Junze Zheng, Dingheng Zeng, Yanhong Liu, Jiankang Deng, Sergio Escalera, Xiaoming Liu, Jun Wan, et al. Unified physical-digital face attack detection. *arXiv preprint arXiv:2401.17699*, 2024. 3

[8] Stephan J Garbin, Marek Kowalski, Matthew Johnson, Jamie Shotton, and Julien Valentin. Fastnerf: High-fidelity neural rendering at 200fps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14346–14355, 2021. 3

[9] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 1, 3, 4

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 2, 3, 5

[11] Philipp Henzler, Niloy J Mitra, and Tobias Ritschel. Escaping plato's cave: 3d shape from adversarial rendering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9984–9993, 2019. 1

[12] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 1, 3

[13] James T Kajiya and Brian P Von Herzen. Ray tracing volume densities. *ACM SIGGRAPH computer graphics*, 18(3):165–174, 1984. 4

[14] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 6

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 1, 3

[16] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6498–6508, 2021. 1

[17] Ajian Liu and Yanyan Liang. Ma-vit: Modality-agnostic vision transformers for face anti-spoofing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1180–1186, 2022. 3

[18] Ajian Liu, Jun Wan, Sergio Escalera, Hugo Jair Escalante, Zichang Tan, Qi Yuan, Kai Wang, Chi Lin, Guodong Guo, Isabelle Guyon, et al. Multi-modal face anti-spoofing attack detection challenge at cvpr2019. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 3

[19] Ajian Liu, Xuan Li, Jun Wan, Yanyan Liang, Sergio Escalera, Hugo Jair Escalante, Meysam Madadi, Yi Jin, Zhuoyuan Wu, Xiaogang Yu, et al. Cross-ethnicity face anti-spoofing recognition challenge: A review. *IET Biometrics*, 10(1):24–43, 2021. 3

[20] Ajian Liu, Zichang Tan, Jun Wan, Sergio Escalera, Guodong Guo, and Stan Z Li. Casia-surf cefa: A benchmark for multi-modal cross-ethnicity face anti-spoofing. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1179–1187, 2021. 3

[21] Ajian Liu, Zichang Tan, Jun Wan, Yanyan Liang, Zhen Lei, Guodong Guo, and Stan Z Li. Face anti-spoofing via adversarial cross-modality translation. *IEEE Transactions on Information Forensics and Security*, 16:2759–2772, 2021. 3

[22] Ajian Liu, Chenxu Zhao, Zitong Yu, Anyang Su, Xing Liu, Zijian Kong, Jun Wan, Sergio Escalera, Hugo Jair Escalante, Zhen Lei, et al. 3d high-fidelity mask face presentation attack detection challenge. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 814–823, 2021. 3

[23] Ajian Liu, Chenxu Zhao, Zitong Yu, Jun Wan, Anyang Su, Xing Liu, Zichang Tan, Sergio Escalera, Junliang Xing, Yanyan Liang, et al. Contrastive context-aware learning for 3d high-fidelity mask face presentation attack detection. *IEEE Transactions on Information Forensics and Security*, 17:2497–2507, 2022. 3

[24] Ajian Liu, Zichang Tan, Zitong Yu, Chenxu Zhao, Jun Wan, Yanyan Liang Zhen Lei, Du Zhang, Stan Z Li, and Guodong Guo. Fm-vit: Flexible modal vision transformers for face anti-spoofing. *IEEE Transactions on Information Forensics and Security*, 2023. 3

[25] Ajian Liu, Shuai Xue, Jianwen Gan, Jun Wan, Yanyan Liang, Jiankang Deng, Sergio Escalera, and Zhen Lei. Cfpl-fas: Class free prompt learning for generalizable face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[26] Weiyang Liu, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song. Sphereface: Deep hypersphere embedding for face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 212–220, 2017. 3

[27] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 5

[28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 3

[29] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7588–7597, 2019. 1

[30] Michael Niemeyer and Andreas Geiger. Giraffe: Representing scenes as compositional generative neural feature fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11453–11464, 2021. 2, 3, 4, 8

[31] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 3

[32] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10684–10695, 2022. 3

[33] Hao Wang, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu. Cosface: Large margin cosine loss for deep face recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5265–5274, 2018. 3

[34] Keyao Wang, Guosheng Zhang, Haixiao Yue, Ajian Liu, Gang Zhang, Haocheng Feng, Junyu Han, Errui Ding, and Jingdong Wang. Multi-domain incremental learning for face presentation attack detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 5499–5507, 2024. 3

[35] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11): 2884–2896, 2018. 3, 6, 7

[36] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2021. 3

[37] Haocheng Yuan, Ajian Liu, Junze Zheng, Jun Wan, Jiankang Deng, Sergio Escalera, Hugo Jair Escalante, Isabelle Guyon, and Zhen Lei. Unified physical-digital attack detection challenge, 2024. 3

[38] Shifeng Zhang, Xiaobo Wang, Ajian Liu, Chenxu Zhao, Jun Wan, Sergio Escalera, Hailin Shi, Zezheng Wang, and Stan Z Li. A dataset and benchmark for large-scale multi-modal face anti-spoofing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 919–928, 2019. 3

[39] Shifeng Zhang, Ajian Liu, Jun Wan, Yanyan Liang, Guodong Guo, Sergio Escalera, Hugo Jair Escalante, and Stan Z Li. Casia-surf: A large-scale multi-modal benchmark for face anti-spoofing. *IEEE Transactions on Biometrics, Behavior, and Identity Science*, 2(2):182–193, 2020. 3

[40] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 3