sensors

MDPI

Article

GammaGAN: Gamma-Scaled Class Embeddings for Conditional Video Generation

Minjae Kang [1] **and Yong Seok Heo** [1,2,*]

[1] Department of Electrical and Computer Engineering, Ajou University, Suwon 16499, Republic of Korea; dreamer29@ajou.ac.kr

[2] Department of Artificial Intelligence, Ajou University, Suwon 16499, Republic of Korea

* Correspondence: ysheo@ajou.ac.kr

Abstract: In this paper, we propose a new model for conditional video generation (GammaGAN). Generally, it is challenging to generate a plausible video from a single image with a class label as a condition. Traditional methods based on conditional generative adversarial networks (cGANs) often encounter difficulties in effectively utilizing a class label, typically by concatenating a class label to the input or hidden layer. In contrast, the proposed GammaGAN adopts the projection method to effectively utilize a class label and proposes scaling class embeddings and normalizing outputs. Concretely, our proposed architecture consists of two streams: a class embedding stream and a data stream. In the class embedding stream, class embeddings are scaled to effectively emphasize class-specific differences. Meanwhile, the outputs in the data stream are normalized. Our normalization technique balances the outputs of both streams, ensuring a balance between the importance of feature vectors and class embeddings during training. This results in enhanced video quality. We evaluated the proposed method using the MUG facial expression dataset, which consists of six facial expressions. Compared with the prior conditional video generation model, ImaGINator, our model yielded relative improvements of 1.61%, 1.66%, and 0.36% in terms of PSNR, SSIM, and LPIPS, respectively. These results suggest potential for further advancements in conditional video generation.

Keywords: class embeddings; conditional generative adversarial networks; conditional video generation; GammaGAN; generative adversarial networks; projection discriminator; video generation

check for updates

Citation: Kang, M.; Heo, Y.S. GammaGAN: Gamma-Scaled Class Embeddings for Conditional Video Generation. *Sensors* **2023**, *23*, 8103. https://doi.org/10.3390/s23198103

Academic Editor: Alexander Wong

Received: 26 August 2023
Revised: 23 September 2023
Accepted: 25 September 2023
Published: 27 September 2023

1. Introduction

Deep learning models currently dominate video generation tasks because they can create high-quality and realistic videos. The challenge of conditional video generation, specifically the generation of plausible videos from a single image with a class label, has prompted intensive research. Various deep learning methods using generative adversarial networks (GANs) [1], transformer models [2–5], and diffusion models [6–9] have been extensively explored. Although transformer and diffusion models have outperformed GANs in certain aspects [9,10], GANs are efficient in video generation due to their simplicity and relatively shorter inference time. However, traditional video generation methods using conditional generative adversarial networks (cGANs) encounter difficulties in effectively utilizing the class information as a condition. Typically, these methods concatenate class labels to feature maps in a generator and a discriminator through simple concatenation [11], which makes it difficult to utilize conditional information [12]. In particular, the role of the discriminator is to distinguish between the distribution of the generator and that of real data on the set of data samples and conditional class labels. The performance of the discriminator is crucial for improving generative quality and stability of the cGANs.

Given these considerations, increased research has been conducted on methods for providing class labels to discriminators. Techniques for utilizing class information in discriminators can generally be categorized into two types: injecting a class label directly and employing auxiliary classifiers. The former is commonly achieved by concatenating [11,13–18] or by projecting the class information using class embeddings [12,19–21]. On the other hand, auxiliary classifiers [22–26] aim to use class information effectively through classification.

As an alternative to the concatenation method for providing the conditional class information, the projection discriminator [12] can be applied to video generation [27], which offers the benefit of learning the relationship between feature vectors and class embeddings by applying the inner product between them. However, when a projection discriminator is employed for video generation, it tends to diminish class differences, resulting in similar results despite different classes. This is because the projection discriminator is vulnerable to overfitting and mode collapse [28,29].

To address the limitations of current projection discriminators in video generation, in this paper, we propose a novel technique, GammaGAN, to amplify the differences between classes and improve the quality of videos. This is achieved by adaptively scaling the class embeddings and normalizing the outputs, as shown in Figure 1. Scaled class embeddings emphasize class information, and normalized outputs automatically learn to effectively balance the feature vectors and class embeddings. Consequently, our discriminator can provide proper feedback to the generator during training by effectively distinguishing class differences, which results in enhanced video quality. Our experiments on the MUG facial expression dataset [30] demonstrated the effectiveness of our approach, with quantitative and qualitative analyses showing improved quality compared to prior video generation models such as VGAN [31], MoCoGAN [32], and ImaGINator [33]. In particular, our approach led to relative improvements of 1.61%, 1.66%, and 0.36% in terms of PSNR, SSIM, and LPIPS, respectively, on the MUG dataset compared to ImaGINator [33].

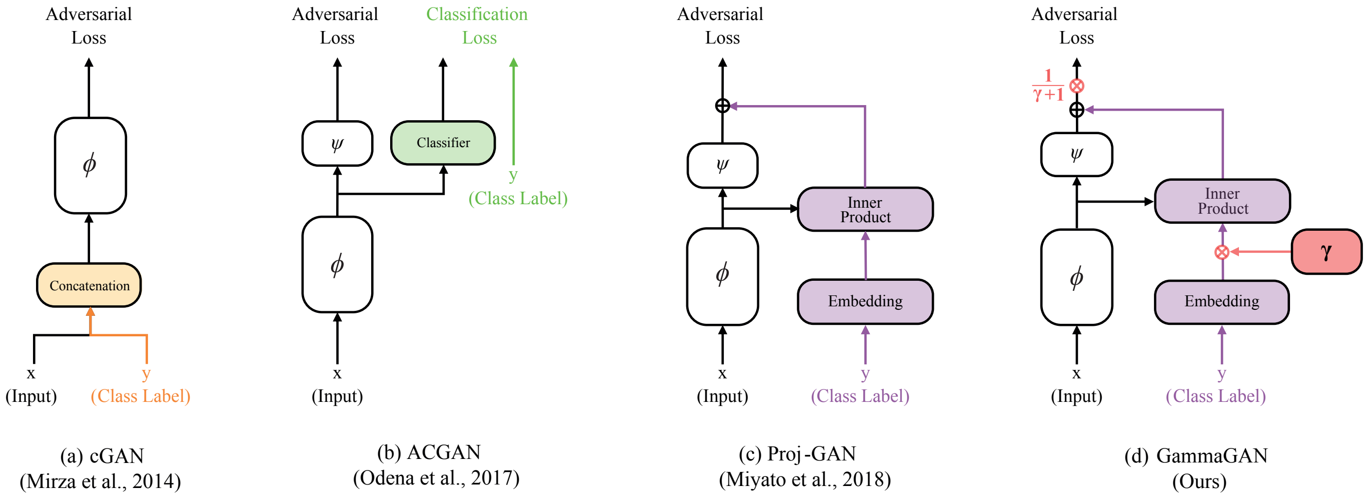


Figure 1. Various class label utilization methods of cGANs. (**a**) cGAN [11] uses the concatenation method (orange), and (**b**) ACGAN [22] employs classification loss and a classifier (green) to utilize conditional information. (**c**) Proj-GAN [12] and (**d**) GammaGAN use the projection method (purple) with class embeddings. In GammaGAN (our proposed method), we define a network consisting of two streams: the data stream (**left**) and the class embedding stream (**right**). The class embeddings are scaled by γ (red) to emphasize class conditional information, and outputs are normalized by $\frac{1}{\gamma+1}$ (red) to balance the outputs.

To provide an overview, our main contributions are summarized as follows:

- We propose GammaGAN, an enhanced video discriminator network for conditional video generation that incorporates two novel techniques: scaling class embeddings and normalizing outputs.
- Scaled class embeddings emphasize class conditional information, thereby enhancing the distinction between different classes.

- Our technique for normalizing outputs balances the outputs of the model. This enables the prioritization between feature vectors and class embeddings during training, leading to improved video quality.

2. Related Work

2.1. Video Generation

The goal of video generation is to produce realistic videos. Video generation is a challenging problem due to the high dimensionality of video data and the difficulties in creating effective feature vectors [34]. Various generative models have recently been proposed for video generation. Generative adversarial networks (GANs), initially introduced in the image domain [1,14,35–42], have been adapted for the video domain [27,31,32,43,44]. In addition, advanced models, such as diffusion models [45,46] and transformer models [47–49], have been introduced for video generation tasks. These models have demonstrated impressive performance, often surpassing the results achieved by traditional GANs [9,10]. However, these advanced models incur higher computational costs and longer inference times.

2.2. Conditional Generative Adversarial Networks

Generative adversarial networks (GANs) are models designed to produce realistic images. Conditional generative adversarial networks (cGANs) allow us to provide conditional information to a model during training using a class label. In the early stages of cGANs research, class conditional information was provided to a model by concatenation [11]. This is typically done by directly injecting it into the input [13] or by injecting it into a hidden layer [14–18]. However, projection-based methods [12,19–21] have been developed to effectively utilize class information for cGANs. This method involves generating two types of embeddings (feature vectors and class embeddings) and then calculating the inner products between them to measure their similarities. Nevertheless, when the projection method is applied to the task of video generation [27], it tends to decrease class differences during training because it still suffers from overfitting and mode collapse [28,29]. To address this problem, we propose a novel technique that focuses on enhancing class differences and produces videos with improved quality. This is achieved by scaling the class embeddings and normalizing the outputs of the model. Consequently, we demonstrate that our proposed method can produce videos with more distinct class differences and improved quality.

3. Method

In this section, we propose GammaGAN, an enhanced video discriminator network, featuring two novel techniques: scaling class embeddings and normalizing outputs. The proposed GammaGAN architecture, as illustrated in Figure 2, consists of a generator G, an image discriminator D_I, and a video discriminator D_V. The generator G uses a single image, noise, and a class label to generate a realistic video. The image discriminator D_I randomly selects a frame from either a real or fake video and determines the appearance of the sample rather than its motion. Given a video and class label, the video discriminator D_V determines whether the motion in the video is appropriate.

In particular, the methods used to inject the class label differed among the networks. Class labels are injected into the feature map in generator G by concatenation. In contrast, the video discriminator D_V uses the proposed projection method to inject a class label, as shown in Figure 2.

We employed the ImaGINator architecture [33], which shares the same generator and image discriminator as our network, as a backbone to evaluate the effectiveness of our video discriminator. The ImaGINator network uses only the concatenation method to inject class information into networks.

In the following sections, we introduce details of GammaGAN by providing mathematical descriptions of our method and explaining its application to our model. In addition, we describe the objective function of the training process.

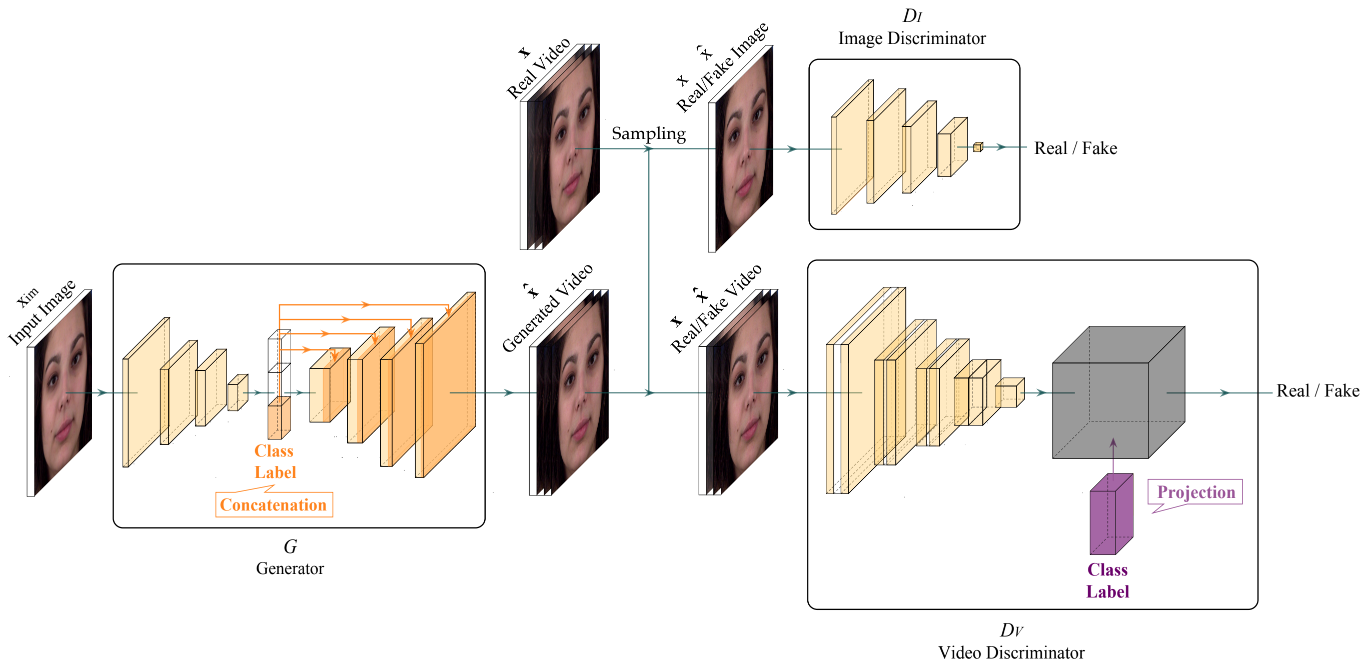


Figure 2. GammaGAN architecture which is visualised using PlotNeuralNet [50]. Our proposed model consists of three networks: generator G, image discriminator D_I, and video discriminator D_V. We adopt ImaGINator [33] as our backbone, sharing the same generator and image discriminator. The utilization of class labels varies between networks: the generator G uses the concatenation method [11] (orange), whereas the video discriminator D_V employs the projection method [12] (purple). The details of our proposed method are encapsulated within the black box in the video discriminator D_V, which are described in detail in Figure 3.

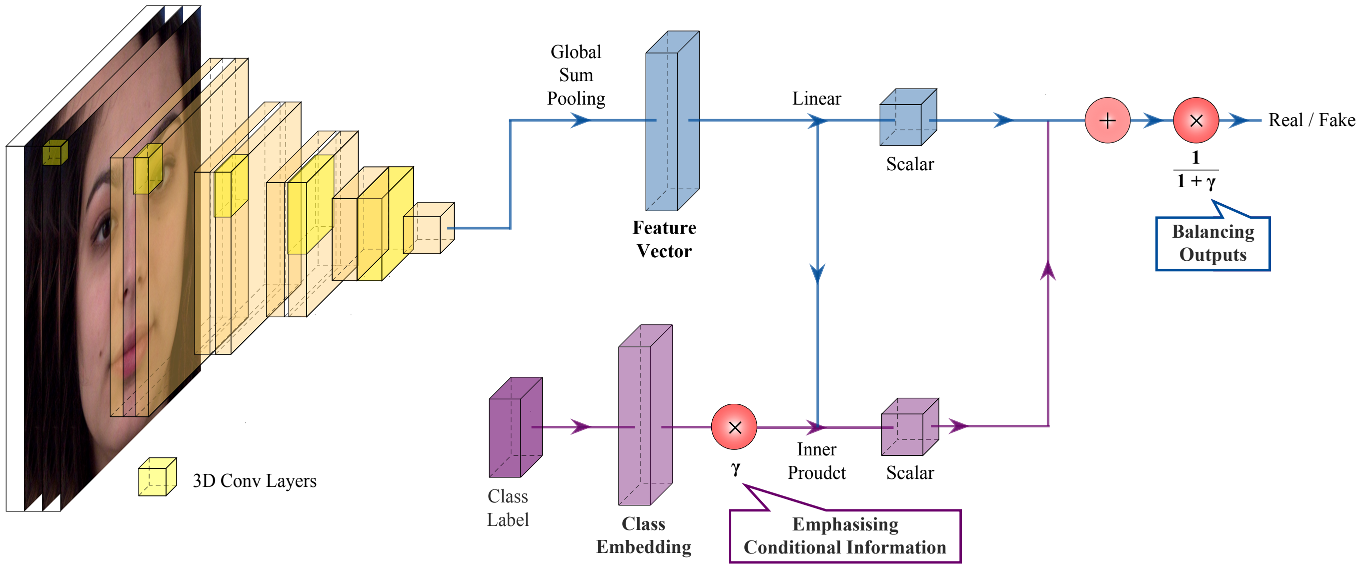


Figure 3. Proposed video discriminator. Our proposed model, the video discriminator in GammaGAN, consists of two streams: the data stream (blue) and the class embedding stream (purple). First, the data stream is designed to extract feature vectors and normalize the outputs for balance using $\frac{1}{\gamma+1}$, leading to improved video quality. Second, the class embedding stream utilizes a novel learnable parameter, γ, to generate scaled class embeddings. These scaled embeddings emphasize class information, enabling our model to generate more distinguishable videos between classes.

3.1. GammaGAN Video Discriminator

3.1.1. Mathematical Description

Our goal is to generate a realistic video from a single image and a class label while preserving the class differences and enhancing the quality of the video. To this end, we propose an enhanced projection method for conditional video generation, GammaGAN. Rather than using conventional concatenation [11] to inject a class label into the discriminator, we apply our new approach based on the projection method [12].

The projection discriminator [12] is an alternative to the concatenation method [11,13,14,17,18] for injecting class information into the discriminator. The operation of the projection discriminator is based on two fundamental principles. First, it uses embeddings to train the model, specifically class embeddings and feature vectors. Second, it calculates the inner products of the class embeddings and feature vectors. This allows the network to understand the relationships and similarities between them. Therefore, the projection discriminator leverages class information more effectively than the concatenation methods.

Discriminators in cGANs learn to distinguish between real and fake samples from each distribution, given conditioning information: a class label [1,11]. The objective function of the cGANs' discriminators [11,12,29] can be written as:

$$\begin{aligned}\mathcal{L}(D) &= -\mathbb{E}_{y\sim p(y)}\Big[\mathbb{E}_{\mathbf{x}\sim p(\mathbf{x}|y)}[\log(D(\mathbf{x},y))]\Big] \\ &\quad - \mathbb{E}_{y\sim q(y)}\Big[\mathbb{E}_{\mathbf{x}\sim q(\mathbf{x}|y)}[\log(1-D(\mathbf{x},y))]\Big] \\ &= -\int \log(D(\mathbf{x},y))p(\mathbf{x},y)\,d\mathbf{x}\,dy \\ &\quad -\int \log(1-D(\mathbf{x},y))q(\mathbf{x},y)\,d\mathbf{x}\,dy,\end{aligned} \tag{1}$$

where $\mathbf{x} \in \mathcal{X}$ represents the input data and $y = \{1,\ldots,c\} \in \mathcal{Y}$ is a class label (class information). $p(y)$ and $q(y)$ are the true and fake label marginal distributions, respectively. $p(\mathbf{x}|y)$ and $q(\mathbf{x}|y)$ are the true and fake data distributions conditional on y, respectively. $p(\mathbf{x},y)$ and $q(\mathbf{x},y)$ are the true and fake joint distributions of $\mathbf{x}$ and y, respectively [29].

Gamma-exponentiated conditional probability. To propose our method, we denote $\mathbf{x}$ as the input vector and $\mathbf{y}$ as class information (i.e., label; we assume that the label $\mathbf{y}$ is a one-hot vector). In cGANs' adversarial loss (1), $D(\mathbf{x},\mathbf{y})$ represents the probability that the discriminator estimates the pair of input data $\mathbf{x}$ and class label $\mathbf{y}$ as real. The joint probability can be expressed using the conditional probability and the marginal probability,

$$D(\mathbf{x},\mathbf{y}) = D(\mathbf{y}|\mathbf{x})D(\mathbf{x}) = \mathcal{A}(f(\mathbf{x},\mathbf{y})), \tag{2}$$

where $\mathcal{A}$ is the activation function. A sigmoid function (logistic function) is applied to our task. In this context, $f(\mathbf{x},\mathbf{y})$ denotes the logit, which is the output value generated by the model before the activation function is applied.

Our goal is to increase class differences during training. To formulate our idea, we define D_γ by exponentiating the conditional probability as follows:

$$D_\gamma(\mathbf{x},\mathbf{y}) := (D(\mathbf{y}|\mathbf{x}))^\gamma D(\mathbf{x}) = \mathcal{A}(f_\gamma(\mathbf{x},\mathbf{y})), \tag{3}$$

where γ is a real number used to weigh the importance of the conditional information in our model.

To express the logit itself, we can take the inverse function of the activation function on both sides of the equation as follows:

$$f_\gamma(\mathbf{x},\mathbf{y}) = \mathcal{A}^{-1}(D_\gamma(\mathbf{x},\mathbf{y})). \tag{4}$$

Now, (4) can simply be expanded by the definition of logit as follows:

$$\begin{aligned} f_\gamma(\mathbf{x},\mathbf{y}) &= \mathcal{A}^{-1}(D_\gamma(\mathbf{x},\mathbf{y})) \\ &= \mathcal{A}^{-1}((D(\mathbf{y}|\mathbf{x}))^\gamma D(\mathbf{x})) \\ &= \log\left(\frac{p(\mathbf{y}|\mathbf{x})^\gamma p(\mathbf{x})}{q(\mathbf{y}|\mathbf{x})^\gamma q(\mathbf{x})}\right) \\ &= \log\left(\frac{p(\mathbf{y}|\mathbf{x})^\gamma}{q(\mathbf{y}|\mathbf{x})^\gamma}\right) + \log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right) \\ &= \gamma\left(\underbrace{\log\left(\frac{p(\mathbf{y}|\mathbf{x})}{q(\mathbf{y}|\mathbf{x})}\right)}_{\text{Label Matching}}\right) + \left(\underbrace{\log\left(\frac{p(\mathbf{x})}{q(\mathbf{x})}\right)}_{\text{Marginal Matching}}\right) \\ &:= r(\mathbf{y}|\mathbf{x}) + r(\mathbf{x}), \end{aligned} \tag{5}$$

where $r(\mathbf{y}|\mathbf{x})$ and $r(\mathbf{x})$ are the log-likelihood ratios, respectively [12,20].

Gamma-scaled class embeddings. Our method can be achieved by weighting the class embeddings to increase class differences. First, the logit of the projection discriminator can be derived as follows [12]:

$$f(\mathbf{x},\mathbf{y};\theta) = \mathbf{y}^T V\phi(\mathbf{x};\theta_\Phi) + \psi(\phi(\mathbf{x};\theta_\Phi);\theta_\Psi), \tag{6}$$

where V is the embedding matrix, which includes the embedding vectors for all the classes, $\phi(\cdot,\theta_\Phi)$ is the feature vector of $\mathbf{x}$, and $\psi(\cdot,\theta_\Psi)$ is a scalar function that is denoted for the normalization constant [12]. Equation (6) can be written when $y = c$ as follows:

$$f(\mathbf{x}, y = c) = \mathbf{v}_c^T\phi(\mathbf{x}) + \psi(\phi(\mathbf{x})), \tag{7}$$

where $\mathbf{v}_c^T$ is the class embedding of $\mathbf{y}$ [12].

Now, we can derive the proposed logit in (5) using (7),

$$\begin{aligned} f_\gamma(\mathbf{x}, y = c) &:= \gamma(\mathbf{v}_c^T\phi(\mathbf{x})) + \psi(\phi(\mathbf{x})) \\ &= (\gamma\mathbf{v}_c^T)\phi(\mathbf{x}) + \psi(\phi(\mathbf{x})), \end{aligned} \tag{8}$$

where γ represents the weight of the class embedding of $y = c$, demonstrating that our method can increase the class differences by simply scaling the class embeddings. Note that the scale γ is a learnable parameter.

Normalized outputs. We introduce the normalization technique in GammaGAN. This constrains output (logit) growth, which can occur when the learnable parameter γ diverges, by normalizing the two terms. In addition, this allows our network to be trained by balancing the importance of the two terms and observing their relative significance. Our proposed method, GammaGAN, can be expressed as follows:

$$\begin{aligned} f_\gamma(\mathbf{x}, y = c) &:= \frac{\gamma}{\gamma+1}(\mathbf{v}_c^T\phi(\mathbf{x})) + \frac{1}{\gamma+1}(\psi(\phi(\mathbf{x}))) \\ &= \frac{1}{\gamma+1}((\gamma\mathbf{v}_c^T)\phi(\mathbf{x}) + \psi(\phi(\mathbf{x}))). \end{aligned} \tag{9}$$

Furthermore, (9) can be generalized because it represents the specific case when $y = c$. By generalizing (9), we derive our method as follows:

$$\begin{aligned} f_{\gamma}(\mathbf{x}, \mathbf{y}) &= \frac{\gamma}{\gamma+1}(\mathbf{y}^T V \phi(\mathbf{x})) + \frac{1}{\gamma+1}(\psi(\phi(\mathbf{x}))) \\ &= \frac{1}{\gamma+1}((\gamma \mathbf{y}^T V)\phi(\mathbf{x}) + \psi(\phi(\mathbf{x}))), \end{aligned} \tag{10}$$

where V is the class embedding matrix.

3.1.2. Architecture

Figure 3 illustrates our proposed video discriminator, which is composed of two streams: the data stream (blue) and the class embedding stream (purple). The first and second terms in (10) represent the class embedding stream and the data stream, respectively.

First, the data stream aims to extract valuable feature vectors from the video data using 3D convolutional layers. Once the feature vectors are obtained, the data stream splits into two branches: one continuing as the data stream for unconditional marginal matching and the other feeding into the class embedding stream for conditional label matching. The branch that continues as the data stream transforms the feature vectors into a scalar using a linear layer, whereas the branch directed into the class embedding stream uses feature vectors to perform the inner product with class embeddings.

Second, the class embedding stream focuses on obtaining class embeddings and emphasizing the class information. The one-hot encoded class labels transform their dimensions through an embedding layer, which enables the class embeddings to perform inner products with the feature vectors obtained from the data stream. The class information is emphasized through multiplication by our proposed constant, denoted as γ. After obtaining the inner product, resulting in a scalar, the scalar is added to the scalar from the data stream. Subsequently, the scalars from both the data and class embedding streams are balanced using our proposed method, denoted as $\frac{1}{\gamma+1}$. It is crucial to note that the normalization technique gains its significance only when the class embeddings are scaled, given that the scaling factor γ of class embeddings is a learnable parameter.

Therefore, the data stream is designed to extract feature vectors and normalize outputs; the model can balance outputs, resulting in improved video quality. On the other hand, the class embedding stream is designed to obtain class embeddings and scale these class embeddings. Class information can be emphasized by scaling class embeddings, enabling the model to generate more distinguishable videos with different classes.

3.2. Objective Function

In this section, the objective function of the training process is introduced. First, we describe the full objective function, followed by the objective functions of the generator G, the image discriminator D_I, and the video discriminator D_V. Since we use ImaGINator [33] as our backbone, the losses for the generator and discriminators are similar to those of the ImaGINator.

3.2.1. Full Objective Function

The objective function for training our model is described in the following way. Note that real images and videos are denoted by x and $\mathbf{x}$, respectively, and fake images and videos are denoted by $\hat{x}$ and $\hat{\mathbf{x}}$. Since we aim to generate a video from noise z, an input image x_{im}, and a class label $\mathbf{y}$, the process can be represented as follows:

$$G : \{z, x_{im}, \mathbf{y}\} \rightarrow \hat{\mathbf{x}}, \tag{11}$$

where $\hat{\mathbf{x}}$ represents the generated video output.

Generator G aims to generate a realistic video using noise z, an input image x_{im}, and a class label $\mathbf{y}$. In contrast, the video discriminator D_V differentiates between real videos $\mathbf{x}$ and fake videos $\hat{\mathbf{x}}$. Similarly, the image discriminator D_I distinguishes between sampled frames from real videos and fake videos (x and $\hat{x}$). The objective function during training is defined as follows:

$$\arg\min_{G}\max_{D_I,D_V}\mathcal{L}(G,D_I,D_V). \tag{12}$$

We introduce our full objective function, which consists of two losses: the adversarial loss $\mathcal{L}_{adv}$ and the reconstruction loss $\mathcal{L}_{rec}$,

$$\mathcal{L}(G,D_I,D_V)=\mathcal{L}_{adv}(G,D_I,D_V)+\lambda\mathcal{L}_{rec}(G), \tag{13}$$

where the parameter λ is used to stabilize and ensure the balance between these two losses ($\mathcal{L}_{adv}$ and $\mathcal{L}_{rec}$) during the training [33]. In the next step, the objective functions of the generator G, the image discriminator D_I, and the video discriminator D_V are described, in that order.

3.2.2. Generator Loss

The objective function of generator G consists of adversarial loss and reconstruction loss with λ,

$$\mathcal{L}_{\mathrm{G}}=\mathcal{L}_{adv}+\lambda\mathcal{L}_{rec}. \tag{14}$$

The adversarial loss of the generator is defined as follows:

$$\mathcal{L}_{\mathrm{adv}}=\mathcal{L}^{I}_{adv}+\mathcal{L}^{V}_{adv}, \tag{15}$$

where $\mathcal{L}^{I}_{adv}$ and $\mathcal{L}^{V}_{adv}$ are the image adversarial loss and the video adversarial loss, respectively. The full adversarial loss $\mathcal{L}_{\mathrm{adv}}$ is the sum of the image adversarial loss $\mathcal{L}^{I}_{adv}$ and the video adversarial loss $\mathcal{L}^{V}_{adv}$.

The image adversarial loss is defined as:

$$\begin{aligned}\mathcal{L}^{I}_{adv}&=\mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_I(\hat{x})]\\&=\mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_I(G(z,x_{im},\mathbf{y}))'],\end{aligned} \tag{16}$$

where the generator attempts to minimize $\mathcal{L}^{I}_{adv}$ by penalizing the distance between the distribution of the generated image samples $\hat{x}$ and real image samples x [41]. $\hat{x}=G(z,x_{im},\mathbf{y})'$ denotes a random image sampled from the generated video, $\hat{\mathbf{x}}=G(z,x_{im},\mathbf{y})$.

The video adversarial loss is similarly defined as:

$$\begin{aligned}\mathcal{L}^{V}_{adv}&=\mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_V(\hat{\mathbf{x}},\mathbf{y})]\\&=\mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_V(G(z,x_{im},\mathbf{y}),\mathbf{y})],\end{aligned} \tag{17}$$

where the generator attempts to minimize $\mathcal{L}^{V}_{adv}$ by penalizing the distance between the distribution of generated video samples $\hat{\mathbf{x}}$ and real video samples $\mathbf{x}$ [41].

The reconstruction loss is utilized for the generated video's coherence and authenticity, which allows the generator to generate realistic and plausible videos:

$$\mathcal{L}_{\mathrm{rec}}=\mathbb{E}[|\mathbf{x}-\hat{\mathbf{x}}|_1]=\mathbb{E}[|\mathbf{x}-G(z,x_{im},\mathbf{y})|_1], \tag{18}$$

where $\mathbf{x}$ is the real video, and $\hat{\mathbf{x}}=G(z,x_{im},\mathbf{y})$ is the generated video from the generator [33].

3.2.3. Image Discriminator Loss

The image discriminator D_I learns to distinguish whether a sampled image (x or $\hat{x}$) is from the real distribution or the fake distribution, which attempts to maximize the image discriminator loss. The loss function of the image discriminator is expressed as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{I}} &= \mathbb{E}_{x\sim p_{\mathrm{data}}}[\log D_I(x)] \\ &\quad + \mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_I(\hat{x})] \\ &= \mathbb{E}_{x\sim p_{\mathrm{data}}}[\log D_I(x)] \\ &\quad + \mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_I(G(z,x_{im},\mathbf{y}))'],\end{aligned} \tag{19}$$

where x and $\hat{x}$ represent a randomly sampled frame from real videos $\mathbf{x}$ and generated videos $\hat{\mathbf{x}}$, respectively, for the image discriminator [33].

3.2.4. Video Discriminator Loss

Similarly, the video discriminator D_V learns to determine whether a sampled video ($\mathbf{x}$ or $\hat{\mathbf{x}}$) is from the real distribution or the fake distribution, which attempts to maximize the video discriminator loss. The loss function of the video discriminator is as follows:

$$\begin{aligned}\mathcal{L}_{\mathrm{V}} &= \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}},y}[\log D_V(\mathbf{x},\mathbf{y})] \\ &\quad + \mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_V(\hat{\mathbf{x}},\mathbf{y})] \\ &= \mathbb{E}_{\mathbf{x}\sim p_{\mathrm{data}},y}[\log D_V(\mathbf{x},\mathbf{y})] \\ &\quad + \mathbb{E}_{z\sim p_z(z),x_{im},\mathbf{y}}[1-\log D_V(G(z,x_{im},\mathbf{y}),\mathbf{y})],\end{aligned} \tag{20}$$

where $\mathbf{x}$ represents the real video, and z is the latent variable, which signifies noise [33].

4. Experimental Results

In this section, we present our experimental results and quantitatively and qualitatively evaluate the proposed method.

4.1. Experimental Setup

4.1.1. Dataset

Our experiments utilized the MUG facial expression database provided by the Multimedia Understanding Group [30]. MUG is a facial expression dataset that consists of seven labels: happiness, sadness, surprise, anger, disgust, fear, and neutral. For the experiment, we used six labels corresponding to happiness, sadness, surprise, anger, disgust, and fear. The neutral expression was used as the initial frame to generate videos. Each video has a resolution of 896 × 896 pixels and contains between 50 and 160 frames. There are 931 videos and 52 subjects in total. The intensity of facial expressions varies from frame to frame. Each video initially starts with a neutral facial expression, progresses to the most expressive point around half of the frames, and returns to the neutral facial expression again.

4.1.2. Implementation Details

We conducted end-to-end training on a single A100 NVIDIA GPU using PyTorch [51]. We utilized ImaGINator [33] as our backbone, sharing the same generator and image discriminator architecture. Our experiment focused on evaluating the effectiveness of the proposed video discriminator. For comparison, we selected approximately 32 frames from the first half of each video because the videos typically reached the peak of facial expression intensity around their midpoints. These frames progressively increased the intensity of the facial expressions from neutral to the maximum. To ensure a fair comparison, we used the Adam optimizer [52] with parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$ for all three networks: the generator, the image discriminator, and the video discriminator, matching the values used in ImaGINator [33]. We set the learning rate for all parameters, including the proposed weight parameter γ, to 2×10^{-4} , with the exception of λ, which was set to 1×10^{-4}. The λ

was used to balance the adversarial and reconstruction losses, matching the value used in our backbone architecture. The batch size was set to 64 during training.

4.1.3. Evaluation Metrics

For qualitative evaluation, we compared the generated videos with the ground truth using three metrics: PSNR, SSIM, and LPIPS.

The Peak Signal-to-Noise Ratio (PSNR) measures image quality based on pixel-level differences. A higher PSNR value indicates better quality of the generated images.

The Structural Similarity Index Measure (SSIM) [53] measures the structural similarity aligned with human perception. A higher SSIM value indicates better image quality.

The Learned Perceptual Image Patch Similarity (LPIPS) [54] is a novel metric that provides a more perceptual evaluation between two images compared to other metrics. LPIPS calculates the similarities between feature vectors extracted from a pre-trained VGG [55] network, enabling more perceptual similarity evaluation between the two images. A lower LPIPS value indicates better image quality.

4.1.4. Evaluation Method

To ensure fairness and precision, we applied our evaluation method to the pre-trained ImaGINator model [33], which Y. Wang et al. shared. We chose 10 subjects not part of the training set out of the 52 for video generation. The neutral facial expressions of these 10 subjects were utilized as the input for the generator, producing six unique facial expressions for each subject. After the videos were generated, they were compared frame-by-frame with the ground truth corresponding to each facial expression.

4.2. Ablation Study

In this section, we present the results of our ablation study, evaluating the essential elements of our proposed method, scaling class embeddings, and normalizing the outputs. We compared the performance of four different models according to their use of projection, scaling, and normalization, as shown in Table 1. First, we applied the projection method [12] to our conditional video generation task, assessing its effectiveness in the video domain. Second, we evaluated our method without the normalization technique to scrutinize the impact of scaled class embeddings. Lastly, we assessed our proposed method, GammaGAN, to demonstrate its enhanced performance.

Table 1. Ablation study on GammaGAN. The results from our models are shown in bold, and the best-performing models are underlined.

Model	Method			Metric		
	Projection	Scaling	Normalization	PSNR ↑	SSIM ↑	LPIPS ↓
ImaGINator [33] [1]				24.8779	0.8214	0.1119
Proj-GAN [12]	✓			25.2851	0.8344	<u>0.1070</u>
GammaGAN w/o normalization	✓	✓		25.1176	0.8288	0.1179
GammaGAN w/ normalization (Ours)	✓	✓	✓	**<u>25.2917</u>**	**<u>0.8346</u>**	**0.1112**

[1] We re-evaluated the pre-trained ImaGINator.

4.2.1. Effectiveness of Normalization

GammaGAN employs the normalization technique as outlined in (10), balancing the outputs of the two streams: the data stream and the class embedding stream. An ablation study was conducted to evaluate the effectiveness of the normalization technique in our proposed method, GammaGAN, both qualitatively and quantitatively.

To begin with, Figure 4 represents the variations in the weight parameter γ throughout training when γ is initialized with a value of 1.0. The observed fluctuation in the weight parameter γ demonstrates that our method adaptively and effectively balances the data stream and the class embedding stream. This suggests that our model automatically learns what elements to give more attention to between feature vectors and class embeddings during training.

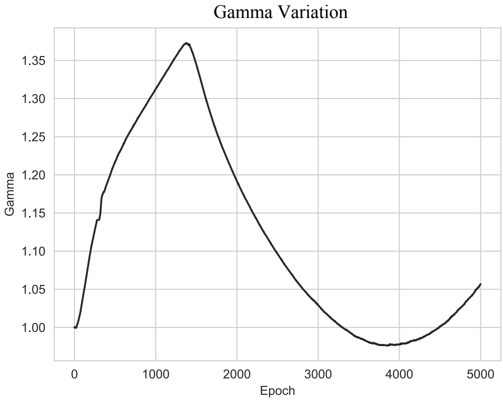


Figure 4. Gamma variation during training. The graph shows the variation of the proposed learnable parameter, γ, during training from epoch 0 to epoch 5000.

For qualitative comparison, Figure 5 illustrates the effectiveness of our normalization technique by comparing GammaGAN with and without the normalization method. The second row, without normalization, and the third row, representing GammaGAN with normalization, clearly show differences in video quality, indicating that GammaGAN generates improvement in videos when using normalization. This suggests that our normalization technique enables the model to automatically learn to balance feature vectors in the data stream and class embeddings in the class embedding stream, giving adaptive attention to feature vectors during training.

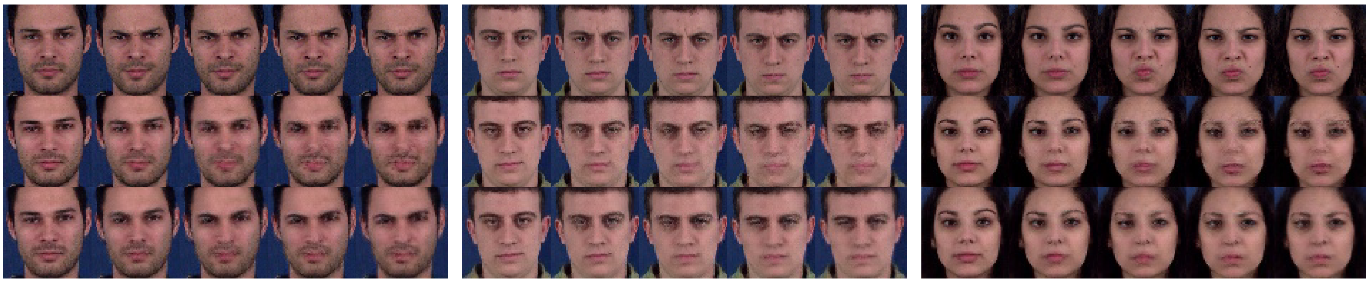


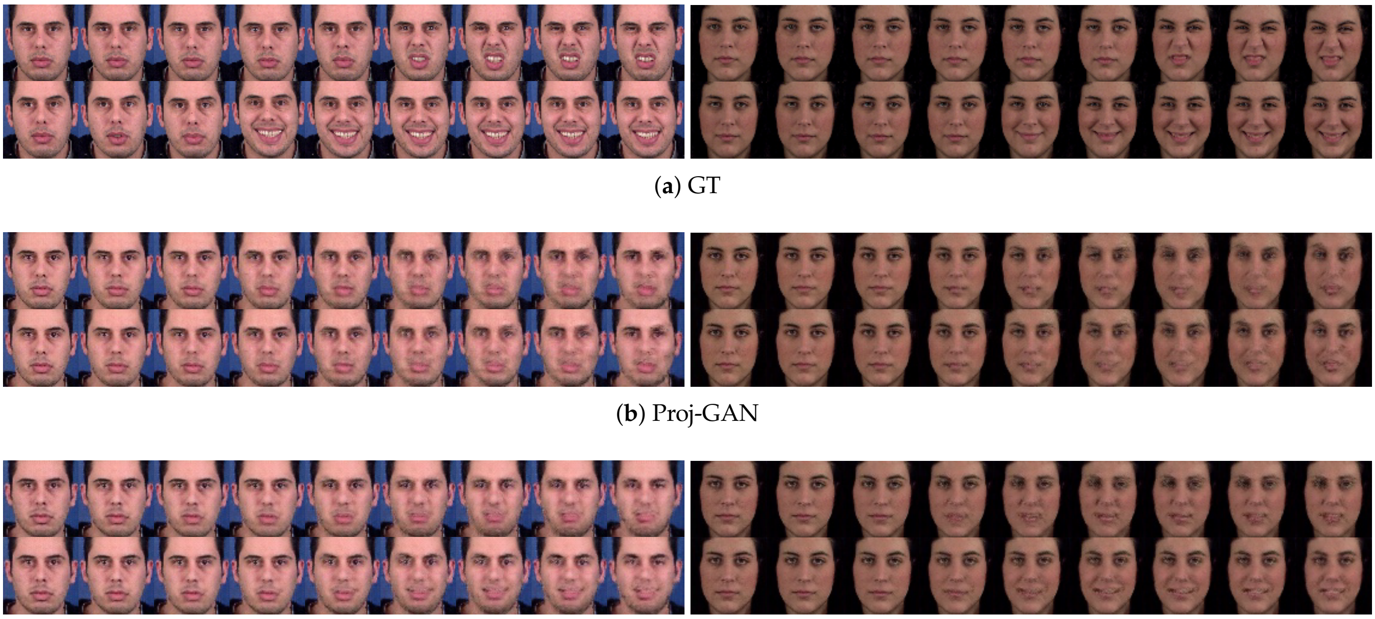
Figure 5. Effectiveness of normalization. An ablation study was conducted to demonstrate the effects of normalization. All the results presented have the label 'anger'. The first row represents the ground truth. The second row represents the results without the normalization technique in GammaGAN, yielding lower-quality videos, whereas the third row represents the improved results of GammaGAN with normalization.

A quantitative comparison was conducted to evaluate the performance of the proposed method, as presented in Table 1. Our proposed method, GammaGAN with the normalization method, performed better than GammaGAN without the normalization method. This demonstrates that our normalization method can effectively enhance the quality of videos generated by the model.

4.2.2. Effectiveness of Scaling Class Embeddings

In this section, we discuss the effectiveness of scaling class embeddings in our proposed method, GammaGAN. As shown in Table 1, our method without normalization yielded results inferior to those of Proj-GAN [12], suggesting that the utilization of both scaling and normalization methods is essential for optimal performance.

We conducted an experiment comparing our method, GammaGAN, with Proj-GAN [12] to demonstrate the effectiveness of scaling class embeddings and the normalization technique, factors that differentiate it from Proj-GAN. As illustrated in Figure 6, our method produces videos that more effectively distinguish between classes than those generated by Proj-GAN, particularly between the labels 'disgust' and 'happiness'. These results demonstrate that our proposed method effectively differentiates between different classes when class embeddings are emphasized by our proposed γ, and that our method generates enhanced video quality simultaneously using the normalization technique.



(**a**) GT

(**b**) Proj-GAN

(**c**) GammaGAN (Ours)

Figure 6. Comparison between Proj-GAN and GammaGAN. (**a**) represents the ground truth of generated videos. (**b**) represents the generated videos when Proj-GAN [12] is applied to the video discriminator. (**c**) shows the results of GammaGAN. The first rows represent 'disgust' (**top**), and the second rows represent 'happiness' (**bottom**) in each of (**a**–**c**).

For the quantitative comparison, our proposed method, GammaGAN with a normalization technique, outperformed other experiments in terms of PSNR and SSIM, as shown in Table 1. However, our method underperforms in the LPIPS metric. A lower LPIPS score is observed with the model when only the projection method is applied. This stems from our training emphasis on improving PSNR and SSIM metrics without including additional loss. Even though the LPIPS metric seems better with the projection method, qualitative evaluations in Figure 6 highlight the efficacy of our method, specifically in distinguishing class-specific differences and improving video quality. This demonstrates the effectiveness of our proposed method, particularly when using the two essential techniques: scaling and normalization. As a result, GammaGAN generates more realistic videos by effectively differentiating facial expressions between various classes, achieved by scaling class embeddings. In addition, our method enhances video quality by balancing feature vectors and class embeddings during training.

4.3. Comparative Results

4.3.1. Quantitative Evaluation

For quantitative comparison, we compared the results of the proposed GammaGAN with those of VGAN [31], MoCoGAN [32], and ImaGINator [33]. The results are presented in Table 2. Our method led to relative improvements of 1.61%, 1.66%, and 0.36% in terms of PSNR, SSIM, and LPIPS, respectively, compared with ImaGINator [33] on the MUG facial expression dataset [30]. These improvements indicate that our method enhances the quality of videos by scaling class embeddings using our proposed learnable constant, γ, and normalizing the outputs.

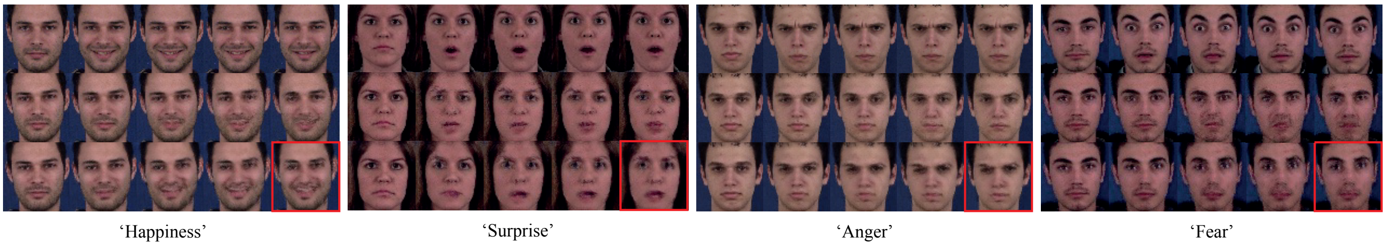
Table 2. Comparison of PSNR, SSIM, and LPIPS among different video generation methods. The results from our models are shown in bold, and the best-performing models are underlined.

Method	PSNR ↑	SSIM ↑	LPIPS ↓
VGAN [31]	14.54	0.28	-
MoCoGAN [32]	18.16	0.58	-
ImaGINator [33]	22.63	0.75	-
ImaGINator [33] [1]	24.8779	0.8214	0.1119
GammaGAN (Ours)	**25.2917**	**0.8346**	**0.1112**

[1] We re-evaluated the pre-trained ImaGINator [33] with our evaluation method.

4.3.2. Qualitative Evaluation

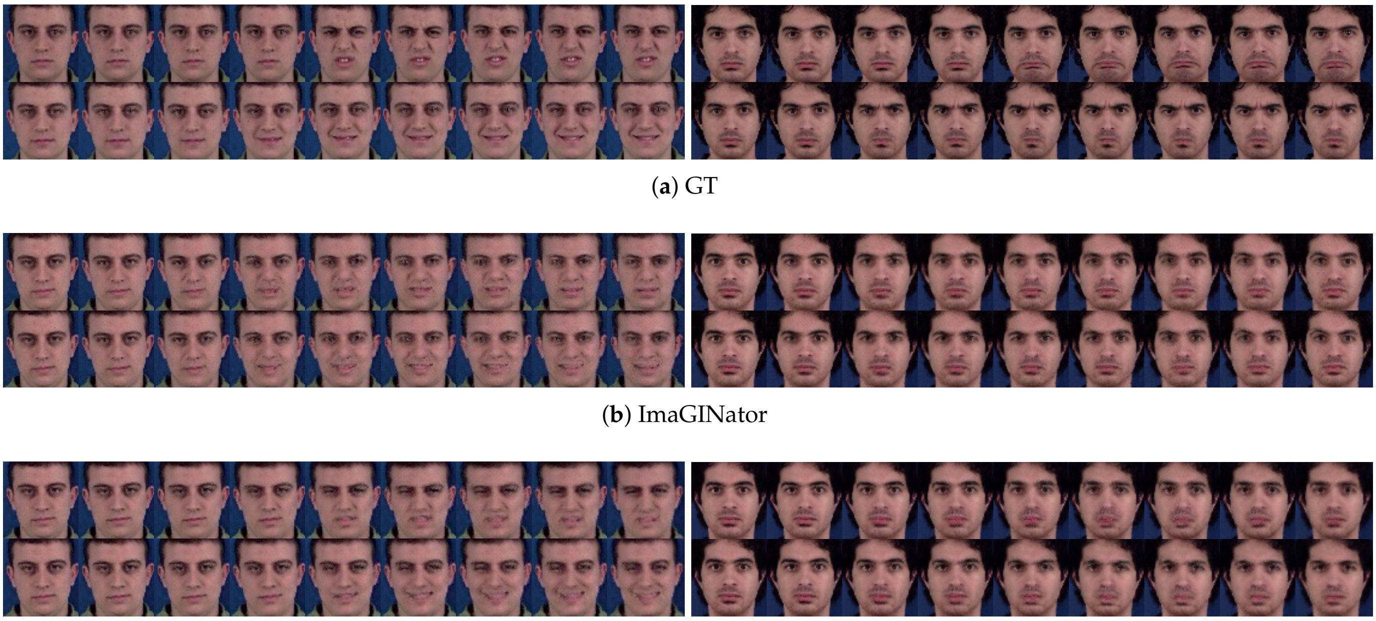
Figure 7 shows the qualitative results of our method, GammaGAN, compared with a previous video generation method, ImaGINator [33]. As seen in Figure 7, our method produces more distinguishable videos between each class label. This distinction is due to the different utilization of class labels and our proposed method. Whereas ImaGINator simply concatenates a class label in the video discriminator, our method, GammaGAN, employs the class label not only in a projection method but also in assigning γ weights to the class embeddings.



'Happiness' 'Surprise' 'Anger' 'Fear'

Figure 7. Comparative results between ImaGINator and GammaGAN. Each figure represents the output video results for the labels 'happiness', 'surprise', 'anger', and 'fear'. Within each figure, the first row represents the ground truth, the second row shows the results of ImaGINator, and the third row shows the results from GammaGAN (Ours). The red boxes highlight the last frames in each video, where the intensity of facial expression is assumed to be at its peak. GammaGAN produces more plausible videos than the previous method, ImaGINator [33].

Figure 8 demonstrates the enhanced ability of the proposed method to distinguish between different facial expression classes. Whereas distinctions between 'disgust' and 'happiness' and between 'sadness' and 'anger' from the previous method are somewhat subtle, there is an improvement in the differentiation of classes when our method is applied. This improvement is due to the enhancement of the class embeddings by the proposed method. In addition, as shown in Figure 8, our method improves the quality of frames and videos, resulting in fewer artifacts compared with the previous method because of our normalization technique.



(**a**) GT

(**b**) ImaGINator

(**c**) GammaGAN (Ours)

Figure 8. Comparison between different classes using ImaGINator and GammaGAN. (**a**) represents the ground truth, (**b**) represents the results of ImaGINator [33], and (**c**) represents the results of GammaGAN. Each result in (**a**–**c**) illustrates generated videos labeled as 'disgust' (**top left**), 'happiness' (**bottom left**), 'sadness' (**top right**), and 'anger' (**bottom right**). GammaGAN generates more distinctive video results between different classes.

5. Conclusions

We introduced GammaGAN, a novel network designed for conditional video generation using a single image and a class label. Our method successfully increased the differences between classes and enhanced the quality of the generated videos using two essential methods: scaling class embeddings and normalizing outputs. Our approach enhances the differences between classes by scaling class embeddings using a learnable parameter, γ, effectively emphasizing conditional information. Furthermore, our model balances the data stream and the class embedding stream by normalizing the outputs, leading to improved quality of videos. This suggests that our approach has the potential for further advancement in conditional video generation. Nonetheless, our method has shown a limitation, particularly with the LPIPS metric. One of the suggestions is to include additional loss terms to improve the performance of our model. This should be done in future work.

Author Contributions: Conceptualization, M.K. and Y.S.H.; methodology, M.K. and Y.S.H.; software, M.K.; validation, Y.S.H.; formal analysis, M.K.; investigation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, Y.S.H.; visualization, M.K.; supervision, Y.S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Institute of Information & communications Technology Planning & Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00255968) grant funded by the Korea government (MSIT); in part by the Brain Korea 21 (BK21) FOUR Program of the National Research Foundation of Korea through the Ministry of Education under Grant NRF5199991014091; and in part by the Ministry of Science and Information and Communications Technology (MSIT), South Korea, through the Information Technology Research Center (ITRC) Support Program, supervised by the Institute for Information and Communications Technology Promotion (IITP), under Grant IITP-2023-2018-0-01424.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GANs	Generative Adversarial Networks
cGANs	Conditional Generative Adversarial Networks
MUG	Multimedia Understanding Group
VGG	Visual Geometry Group
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure
LPIPS	Learned Perceptual Image Patch Similarity

References

1. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Nets. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 8–13 December 2014; pp. 2672–2680.
2. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Advances in Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 5998–6008.
3. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021; pp. 1–21.
4. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
5. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin Transformer V2: Scaling up Capacity and Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–23 June 2022; pp. 12009–12019.
6. Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; Volume 37, pp. 2256–2265.
7. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; pp. 1–12.
8. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021; pp. 1–20.
9. Dhariwal, P.; Nichol, A. Diffusion models beat GANs on image synthesis. In Proceedings of the Advances in Neural Information Processing Systems, Virtual Event, 6–14 December 2021; pp. 8780–8794.
10. Lee, K.; Chang, H.; Jiang, L.; Zhang, H.; Tu, Z.; Liu, C. ViTGAN: Training GANs with Vision Transformers. In Proceedings of the International Conference on Learning Representations, Virtual Event, 25–29 April 2022; pp. 1–18.
11. Mirza, M.; Osindero, S. Conditional Generative Adversarial Nets. *arXiv* **2014**, arXiv:1411.1784.
12. Miyato, T.; Koyama, M. cGANs with Projection Discriminator. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–21.
13. Denton, E.L.; Chintala, S.; Szlam, A.; Fergus, R. Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks. In Proceedings of the Advances in Neural Information Processing Systems, Montreal, QC, Canada, 7–12 December 2015; Volume 28, pp. 1486–1494.
14. Reed, S.; Akata, Z.; Yan, X.; Logeswaran, L.; Schiele, B.; Lee, H. Generative Adversarial Text to Image Synthesis. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; Volume 48, pp. 1060–1069.
15. Zhang, H.; Xu, T.; Li, H.; Zhang, S.; Wang, X.; Huang, X.; Metaxas, D.N. StackGAN: Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 5907–5915.
16. Perarnau, G.; van de Weijer, J.; Raducanu, B.; Álvarez, J.M. Invertible Conditional GANs for image editing. *arXiv* **2016**, arXiv:1611.06355.
17. Dumoulin, V.; Belghazi, I.; Poole, B.; Lamb, A.; Arjovsky, M.; Mastropietro, O.; Courville, A. Adversarially Learned Inference. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017; pp. 1–18.
18. Sricharan, K.; Bala, R.; Shreve, M.; Ding, H.; Saketh, K.; Sun, J. Semi-supervised Conditional GANs. *arXiv* **2017**, arXiv:1708.05789.
19. Miyato, T.; Kataoka, T.; Koyama, M.; Yoshida, Y. Spectral Normalization for Generative Adversarial Networks. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–26.

20. Han, L.; Min, M.R.; Stathopoulos, A.; Tian, Y.; Gao, R.; Kadav, A.; Metaxas, D.N. Dual Projection Generative Adversarial Networks for Conditional Image Generation. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 14438–14447.
21. Han, S.; Lee, T.B.; Heo, Y.S. Semantic-Aware Face Deblurring with Pixel-Wise Projection Discriminator. *IEEE Access* **2023**, *11*, 11587–11600. [CrossRef]
22. Odena, A.; Olah, C.; Shlens, J. Conditional Image Synthesis with Auxiliary Classifier GANs. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; Volume 70, pp. 2642–2651.
23. Nguyen, A.; Clune, J.; Bengio, Y.; Dosovitskiy, A.; Yosinski, J. Plug & play generative networks: Conditional iterative generation of images in latent space. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 3510–3520.
24. Gong, M.; Xu, Y.; Li, C.; Zhang, K.; Batmanghelich, K. Twin Auxilary Classifiers GAN. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 1328–1337.
25. Kang, M.; Shim, W.; Cho, M.; Park, J. Rebooting ACGAN: Auxiliary Classifier GANs with Stable Training. In Proceedings of the Advances in Neural Information Processing Systems, Virtual, 6–14 December 2021; pp. 23505–23518.
26. Hou, L.; Cao, Q.; Shen, H.; Pan, S.; Li, X.; Cheng, X. Conditional GANs with Auxiliary Discriminative Classifier. In Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022; Volume 162, pp. 8888–8902.
27. Clark, A.; Donahue, J.; Simonyan, K. Adversarial Video Generation on Complex Datasets. *arXiv* **2019**, arXiv:1907.06571.
28. Kang, M.; Park, J. ContraGAN: Contrastive Learning for Conditional Image Generation. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 6–12 December 2020; pp. 1–13.
29. Ding, X.; Wang, Y.; Xu, Z.; Welch, W.J.; Wang, Z.J. CcGAN: Continuous Conditional Generative Adversarial Networks for Image Generation. In Proceedings of the International Conference on Learning Representations, Virtual Event, Austria, 3–7 May 2021; pp. 1–30.
30. Aifanti, N.; Papachristou, C.; Delopoulos, A. The MUG facial expression database. In Proceedings of the 11th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Desenzano, Italy, 12–14 April 2010; pp. 1–4.
31. Vondrick, C.; Pirsiavash, H.; Torralba, A. Generating videos with scene dynamics. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 613–621.
32. Tulyakov, S.; Liu, M.Y.; Yang, X.; Kautz, J. MoCoGAN: Decomposing Motion and Content for Video Generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 1526–1535.
33. WANG, Y.; Bilinski, P.; Bremond, F.; Dantcheva, A. ImaGINator: Conditional spatio-temporal GAN for video generation. In Proceedings of the IEEE Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 1160–1169.
34. Haim, H.; Feinstein, B.; Granot, N.; Shocher, A.; Bagon, S.; Dekel, T.; Irani, M. Diverse generation from a single video made possible. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp. 491–509.
35. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X. Improved Techniques for Training GANs. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016 pp. 2226–2234.
36. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 105–114.
37. Zhu, J.Y.; Krähenbühl, P.; Shechtman, E.; Efros, A.A. Generative Visual Manipulation on the Natural Image Manifold. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp 597–613.
38. Gatys, L.A.; Ecker, A.S.; Bethge, M. A Neural Algorithm of Artistic Style. *arXiv* **2015**, arXiv:1508.06576.
39. Karras, T.; Laine, S.; Aila, T. A Style-Based Generator Architecture for Generative Adversarial Networks. *arXiv* **2018**, arXiv:1812.04948.
40. Zhu, J.Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired Image-To-Image Translation Using Cycle-Consistent Adversarial Networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2242–2251.
41. Shaham, T.R.; Dekel, T.; Michaeli, T. SinGAN: Learning a Generative Model from a Single Natural Image. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 1–11.
42. Schonfeld, E.; Schiele, B.; Khoreva, A. A U-Net Based Discriminator for Generative Adversarial Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 8207–8216.
43. Saito, M.; Matsumoto, E.; Saito, S. Temporal Generative Adversarial Nets with Singular Value Clipping. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2830–2839.
44. Saito, M.; Saito, S. TGANv2: Efficient Training of Large Models for Video Generation with Multiple Subsampling Layers. *arXiv* **2018**, arXiv:1811.09245.
45. Voleti, V.; Jolicoeur-Martineau, A.; Pal, C. MCVD: Masked Conditional Video Diffusion for Prediction, Generation, and Interpolation. In Proceedings of the Advances in Neural Information Processing Systems, New Orleans, LA, USA, 28 November–9 December 2022; pp. 1–25.
46. Ni, H.; Shi, C.; Li, K.; Huang, S.X.; Min, M.R. Conditional Image-to-Video Generation with Latent Flow Diffusion Models. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 18444–18455.

47. Yu, L.; Cheng, Y.; Sohn, K.; Lezama, J.; Zhang, H.; Chang, H.; Hauptmann, A.G.; Yang, M.H.; Hao, Y.; Essa, I.; et al. MAGVIT: Masked Generative Video Transformer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 10459–10469.
48. Weissenborn, D.; Täckström, O.; Uszkoreit, J. Scaling Autoregressive Video Models. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020; pp. 1–24.
49. Ge, S.; Hayes, T.; Yang, H.; Yin, X.; Pang, G.; Jacobs, D.; Huang, J.B.; Parikh, D. Long Video Generation with Time-Agnostic VQGAN and Time-Sensitive Transformer. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; pp 102–118.
50. Iqbal, H. *HarisIqbal88/PlotNeuralNet v1.0.0*, (v1.0.0). Zenodo: Genève, Switzerland, 8 June 2023. [CrossRef]
51. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Volume 32, pp. 8024–8035.
52. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–15.
53. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]
54. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18 June–23 June 2018; pp. 586–595.
55. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015; pp. 1–14.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.