Linearly Controllable GAN: Unsupervised Feature Categorization and Decomposition for Image Generation and Manipulation

Sehyung Lee, Mijung Kim, Yeongnam Chae, and Björn Stenger

Rakuten Institute of Technology, Rakuten Group, Inc.
{sehyung.lee, mijung.a.kim, yeongnam.chae, bjorn.stenger}@rakuten.com

Abstract. This paper introduces an approach to linearly controllable generative adversarial networks (LC-GAN) driven by unsupervised learning. Departing from traditional methods relying on supervision signals or post-processing for latent feature disentanglement, our proposed technique enables unsupervised learning using only image data through contrastive feature categorization and spectral regularization. In our framework, the discriminator constructs geometry- and appearance-related feature spaces using a combination of image augmentation and contrastive representation learning. Leveraging these feature spaces, the generator autonomously categorizes input latent codes into geometry- and appearance-related features. Subsequently, the categorized features undergo projection into a subspace via our proposed spectral regularization, with each component controlling a distinct aspect of the generated image. Beyond providing fine-grained control over the generative model, our approach achieves state-of-the-art image generation quality on benchmark datasets, including FFHQ, CelebA-HQ, and AFHQ-V2.

Keywords: Controllable GAN · Unsupervised learning · Generative model

1 Introduction

Generative Adversarial Networks (GAN) have emerged as powerful tools for image generation, with the StyleGAN series standing out as prominent models in this domain [16–18]. These models leverage latent codes to shape the generative process; however, interpreting these codes remains a challenging task. Numerous studies have delved into the StyleGAN latent space to enhance controllability. Yet, many controllable StyleGAN-based models necessitate pre-trained classifiers [32], supervision information [31], or reliance on 3D morphable face models (3DMM) [7, 34]. Such dependencies can limit generalizability and introduce additional labeling efforts when applied to novel datasets.

An alternative approach involves examining the trained model parameters using subspace projection techniques like Principal Component Analysis (PCA) or integrating additional regularization terms during neural network training,

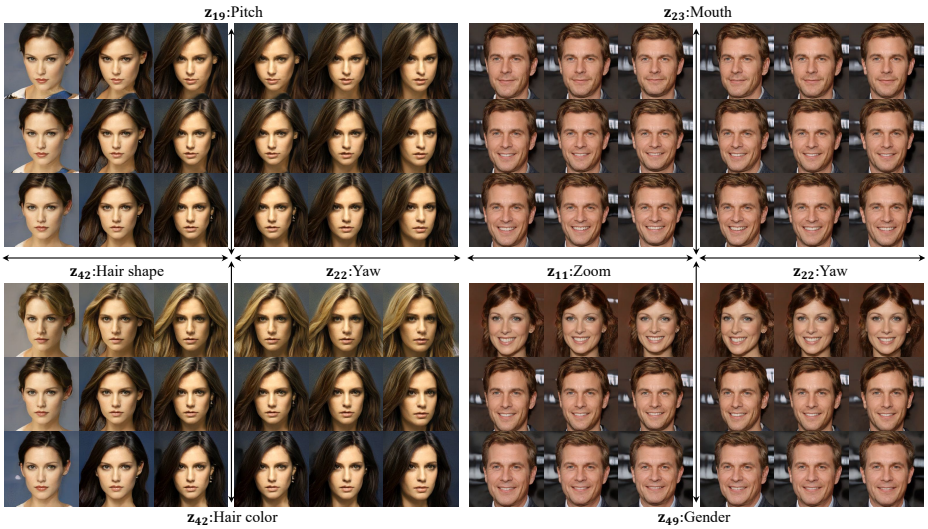

Fig. 1: Illustrative outcomes from LC-GAN trained on CelebA-HQ dataset. The method autonomously learns significant semantics and relates them to input latent codes in a fully unsupervised manner. Leveraging linear control over latent codes, the model generates diverse and customizable outputs with fine-grained adjustments.

such as Hessian penalty [25] and OroJaR [36], to encourage feature disentanglement. However, subspace analysis-based methods [10, 29, 30] often require preprocessing steps for projecting model parameters during image generation and post-processing steps for identifying interpretable directions after model training. This multi-step process can be cumbersome and time-consuming and potentially lead to a distortion of the original distribution of the training data. Moreover, orthogonal regularizer-based methods employ a stochastic estimator to approximate the first- or second-order finite difference, which necessarily requires many samples for accurate estimation. This increased demand for samples poses computational challenges, especially in scenarios with very high-dimensional latent codes, potentially affecting the scalability of these regularization techniques and leading to instability in GAN training. Another drawback of approaches such as [11, 25, 33, 36], jointly optimizing orthogonal constraints with GAN, is the potential generation of less diverse and natural images, reflected in much higher Fréchet Inception Distance (FID) scores.

To overcome these limitations, we introduce Linearly Controllable GAN (LC-GAN), a novel approach that employs end-to-end learning for high-quality image synthesis, eliminating the need for pre-trained classifiers, supervision information, or stochastic estimations of first-/second-order finite differences. Our method begins by decomposing the input noise vector, sampled from a Gaussian distribution, into distinct geometry and appearance codes. Initially, the discriminator is trained to construct geometry and appearance embedding feature spaces by clustering training and augmented images through a combination of image

augmentation and contrastive feature clustering. Subsequently, the generator utilizes these embedding feature spaces to ensure the consistency of its conditionally generated images by partially resampling the noise, facilitating geometry or appearance feature changes. Additionally, the features, initially separated into geometry and appearance codes, undergo further decomposition into more fine-grained subcategories through a feature selection mechanism. This process directly maps semantic changes in the generated images to changes in the input noise vector, enabling each subcategory to linearly and independently activate semantically meaningful features. This fine-grained control provides the ability to make specific changes, such as variations in appearance or viewpoint, while preserving other essential properties.

In this work, we make the following contributions to the field of controllable image generation: (1) We introduce an unsupervised method that categorizes input latent codes into distinct geometry and appearance codes. (2) We present a mechanism for further decomposing features separated into geometry and appearance codes into more fine-grained subcategories. This approach not only enables control over semantically meaningful features during the image generation process but also enhances the interpretability of the input latent code. (3) We demonstrate higher quality of the generated images compared to existing state-of-the-art (SOTA) generative models. Our evaluation results show that LC-GAN outperforms other models in terms of both visual quality and feature controllability. We believe that these contributions advance the field of controllable image generation and provide a useful tool for various applications, such as image editing and synthesis. The source code can be found at `https://github.com/rakutentech/lcgan`.

2 Related work

GAN [9] is widely used for image generation, but achieving precise control over specific image features remains challenging. To address this limitation, recent approaches have introduced additional data as supervision signals, including segmentation maps [31], image attributes [7,21,28], and text descriptions [22,27, 42]. The primary goal of these methods is to disentangle latent features, enabling more accurate control over the generated images. For instance, [31] uses semantic segmentation masks to guide the generation process, controlling object class and location in the resulting images.

Another approach to generating images involves the use of pre-trained models or synthetic data instead of supervision signals. For instance, [2,32] present GAN models that leverage pre-trained classifiers to verify if the generated images are correctly labeled into the target class. Attempts have also been made to learn 3D pose information from 2D GANs by disentangling pose in the latent space, but these efforts require additional 3D supervision, such as synthetic face datasets [21] or 3D morphable models [7, 34, 40].

On the other hand, several post-processing techniques [4, 10, 29, 30, 37] have been proposed to discover the semantic information encoded in the trained

model's latent space through subspace projections. These methods show that latent codes can be disentangled without supervision, but the learned subspace still requires interpretation by visual inspection after training. However, these methods have some limitations, including the inconvenience of the post-processing and pre-processing steps required to identify the important directions and subspace projection. Moreover, this process may also result in some images being ignored due to being deemed unimportant through the subspace projections, potentially leading to a distortion of the original distribution of the training data.

Orthogonal regularizers such as HessianPenalty [25] and OroJaR [36] have been proposed as a means to encourage feature disentanglement and control in the training of neural networks. These regularizers operate by penalizing feature changes based on orthogonal constraints, which can facilitate the learning of more interpretable and independent features. The regularization term encourages the model to map input variations in specific directions, promoting the separation of different factors of variation in the latent space. Despite their potential benefits, orthogonal regularizers have limitations. Estimating first- or second-order finite differences, a key component of these methods, can incur a higher computational cost. This computational demand may impose constraints on training models with high-resolution images or large-scale architectures, limiting the applicability of these regularizers in such scenarios.

3 Linearly Controllable Generative Adversarial Network

The proposed method, LC-GAN, adopts an unsupervised approach to acquire an embedding feature space by integrating conditional feature clustering with real/fake image classification tasks. The discriminator network D outputs both real/fake classification scores and cluster assignments, facilitating the aggregation of features originating from diverse image augmentations. Throughout training, a contrastive loss is employed to encourage closer proximity within the same cluster and greater separation between different clusters, enabling the discriminator to learn a feature space that encapsulates desired image properties, thus enabling controllable image generation through input vector manipulation.

3.1 Embedding Feature Space for Feature Categorization

In this approach, the feature space is constructed by combining basic image augmentations with feature clustering techniques. The discriminator is extended with two projection heads, h^g and h^a, which map images into distinct embedding spaces: geometric change and appearance change. These feature spaces validate the consistency of generated images with intended feature changes. Figure 2 provides an overview of the training procedure. To construct individual feature spaces, a set of augmentations is applied, including random perspective transformation and random image erasing/color jittering. Augmented images simulating variations in geometry and appearance serve as positive and negative samples in corresponding and opposite projection heads to capture desired image changes.

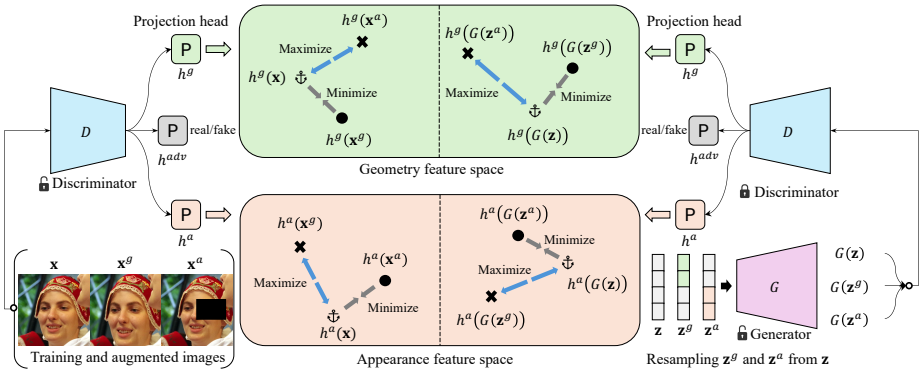

Fig. 2: Overview of geometry and appearance feature space construction and latent code categorization. The discriminator learns embedding feature spaces through contrastive feature clustering with samples generated from geometric and appearance augmentations. By clustering these features, the discriminator associates anchor and positive samples while separating negative samples in the feature space. The learned feature space is used to confirm the relationship between generated images $G(\mathbf{z})$ and re-generated images $G(\mathbf{z}^g)$ and $G(\mathbf{z}^a)$, created by partially resampling the noise of the drawn sample $\mathbf{z}$. This allows the generator to understand the desired feature changes with specific latent code resampling.

For instance, in the geometry feature space, positive samples are generated by applying the viewpoint change augmentation to an anchor image, while negative samples undergo appearance augmentation, introducing alterations unrelated to viewpoint changes. Subsequently, both the anchor image and augmented images undergo projection onto an L_2 normalized feature space using the corresponding projection head. The projection head associated with viewpoint change augmentation extracts features capturing changes in viewpoint. To enhance feature space learning, the discriminator employs contrastive feature clustering, aiming to bring positive samples of the same change type closer together in the feature space.

The contrastive loss for each training image is calculated using the following equation:

$$\mathcal{L}_{cl}^{D} = C(h^g(\mathbf{x}), h^g(\mathbf{x}^g), h^g(\mathbf{x}^a)) + C(h^a(\mathbf{x}), h^a(\mathbf{x}^a), h^a(\mathbf{x}^g)). \tag{1}$$

Here, h^g and h^a are the projection heads projecting the images into the geometry and appearance feature spaces, and $\mathbf{x}$, $\mathbf{x}^g$, and $\mathbf{x}^a$ are the training and augmented images by the geometry and appearance changes. The contrastive loss quantifies the dissimilarity between the feature representation of the anchor image and the positively augmented image, while considering the similarity between the feature representation of the anchor image and the negatively augmented image. The negative logarithm of the fraction is used as the loss function to be minimized during training.

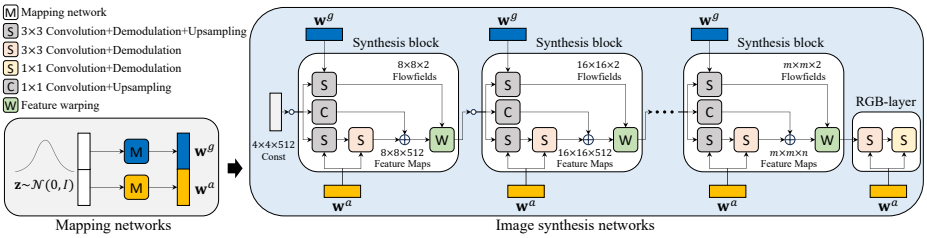

Fig. 3: Generator architecture. The generator comprises mapping networks and image synthesis networks. It begins with input vectors sampled from a Gaussian distribution, which are then divided into geometry and appearance latent codes, $\mathbf{w}^g$ and $\mathbf{w}^a$, by the corresponding mapping networks. These latent codes are subsequently used to produce images at an $m \times m$ resolution. The synthesis networks utilize these latent codes to create flowfields and generate feature maps, enabling precise control over the image synthesis process.

$$C(\mathbf{f}, \mathbf{f}^+, \mathbf{f}^-) = -\log \frac{\exp(\mathbf{f}^T\mathbf{f}^+/\tau)}{\exp(\mathbf{f}^T\mathbf{f}^+/\tau) + \exp(\mathbf{f}^T\mathbf{f}^-/\tau)}, \tag{2}$$

where $\mathbf{f}$, $\mathbf{f}^+$, and $\mathbf{f}^-$ are the feature vectors of the training image, and positively and negatively augmented images, calculated by a projection head h of the discriminator, and the temperature parameter $\tau = 0.05$ controls the strength of the penalties applied to the positive and negative samples during the contrastive loss calculation.

The discriminator is trained using both contrastive loss and standard adversarial loss. Note that for the real *vs.* fake image classification, it does not utilize the augmented samples that are employed to construct the embedding feature space. The adversarial loss is computed as follows:

$$\mathcal{L}^D_{adv} = \mathop{\mathbb{E}}_{\mathbf{x}\sim p_{data}(\mathbf{x})} [\log D(\mathbf{x})] + \mathop{\mathbb{E}}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})} [\log(1 - D(G(\mathbf{z})))] \tag{3}$$

where p_{data} and $p_{\mathbf{z}}$ are the distributions of training data and noise samples, respectively, and $\mathbf{z}$ denotes random noise drawn from a Gaussian distribution with zero mean and unit variance, serving as the latent codes input to the generator network G. Additionally, we incorporate R_1 regularization [23] during discriminator training, defined as:

$$\mathcal{L}^D_{R_1} = \|\nabla D(\mathbf{x})\|^2 . \tag{4}$$

This term encourages the discriminator to be more consistent in its predictions for real images, resulting in more stable training and better performance. The complete loss function for training the discriminator is the sum of the adversarial loss, the contrastive loss, and the regularization term: $\mathcal{L}^D = \mathcal{L}^D_{adv} + \lambda_{cl}\mathcal{L}^D_{cl} + \lambda_{R_1}\mathcal{L}^D_{R_1}$, where $\lambda_{cl} = 0.5$ and $\lambda_{R_1} = 10$ are experimentally set hyperparameters to balance the relative importance of different loss functions.

3.2 Linearly Controllable Generator

In Figure 3, we present an overview of the generator architecture, characterized by two fundamental components: the mapping and synthesis networks. In this architecture, the generator strategically divides the input latent code $\mathbf{z}$ into two distinctive vectors. These vectors then undergo a linear mapping process independently through dedicated networks, resulting in intermediate latent codes. Subsequently, these codes play a crucial role within the image synthesis block to linearly control the desired output. Here, the geometry latent code takes the lead in orchestrating the creation of flowfields, dynamically shaping the feature maps generated based on the appearance latent code. The generator's training loss function is intricately composed of three essential terms: 1) a penalty for adversarial loss, 2) a emphasis on latent code categorization into geometry and appearance codes, and 3) decomposition of categorized features into more refined semantics.

Latent Code Categorization: Firstly, the adversarial loss, defined as

$$\mathcal{L}_{adv}^{G} = \mathop{\mathbb{E}}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})} \left[\log D(G(\mathbf{z}))\right], \tag{5}$$

penalizes the generator for failing to effectively deceive the discriminator. In addition, a conditional resampling is employed to enhance the generator's understanding of input vector alterations. As shown in figure 2, the generator's input $\mathbf{z}$ undergoes conditional resampling to yield $\mathbf{z}^g$ and $\mathbf{z}^a$, where the noise in the designated segment is resampled while retaining the other segments unchanged. This approach enables controlled modifications within specific feature spaces of the generated images. Notably, $\mathbf{z}^g$ is derived by resampling the segment associated with geometry augmentation, while $\mathbf{z}^a$ is obtained by resampling the segment correlated with appearance augmentation. Such conditioning establishes a direct relationship between the generated images and the alterations in the latent code. The generated images $G(\mathbf{z})$, $G(\mathbf{z}^g)$, and $G(\mathbf{z}^a)$ are evaluated as following

$$\mathcal{L}_{cl}^{G} = C(h^g(G(\mathbf{z})), h^g(G(\mathbf{z}^g)), h^g(G(\mathbf{z}^a))) + C(h^a(G(\mathbf{z})), h^a(G(\mathbf{z}^a)), h^a(G(\mathbf{z}^g))). \tag{6}$$

This loss function validates whether the generated output contains the desired property changes by conditionally modifying the sampled latent vector based on the constructed embedding feature spaces of the discriminator.

Spectral Regularization for Feature Selection and Mapping: Drawing upon the formulation in Equation 6, the generator gains an understanding of the semantics encoded within the latent code segment. To enable more fine-grained control by disentangling the appearance and geometry latent codes into finer semantics in a fully unsupervised manner, we introduce a spectral regularization approach leveraging covariance matrix parameterization.

Our approach focuses on learning an anisotropic Gaussian distribution by parameterizing the covariance matrix. By doing so, we enable the model to automatically select important features while minimizing the number of representation dimensions through the application of L_1 regularization on its eigen-values.

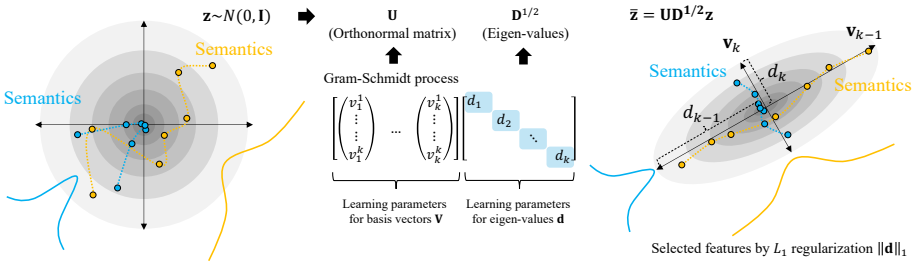

Fig. 4: Spectral regularization. To enable fine-grained control, the geometry and appearance features, categorized through contrastive feature learning, undergo further analysis and decomposition via a feature selection mechanism. This involves applying L_1 regularization to the eigen-values of the covariance matrix, allowing for the automatic selection of important features and achieving more semantic control over the generative process.

We draw input noise from a Gaussian distribution $\mathcal{N}(0, \mathbf{\Sigma})$ with a learnable covariance matrix $\mathbf{\Sigma}$. This adaptive covariance matrix, being symmetric, can be decomposed into $\mathbf{\Sigma} = \mathbf{U}\mathbf{D}\mathbf{U}^T$ according to the eigen-decomposition, where $\mathbf{U}$ consists of orthonormal vectors and $\mathbf{D}$ is a diagonal matrix containing the eigen-values of the covariance matrix.

To learn these matrices, as shown in figure 4, we initialize learning parameters determining the basis vectors and lengths of the axes, denoted as $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_k\}$, where $\mathbf{v}_k = \{v_k^1, ..., v_k^k\}$, and $\mathbf{d} = \{d_1, ..., d_k\}$ at the first layer of the mapping network. These parameters are then transformed into square and diagonal matrices, $\mathbf{V} \in \mathbb{R}^{k \times k}$ and $\mathbf{D}^{1/2} \in \mathbb{R}^{k \times k}$. Subsequently, $\mathbf{V}$ is converted into $\mathbf{U}$ using the Gram–Schmidt process. The transformed vectors are input into mapping networks consisting of 12 fully connected layers, generating the geometry and appearance latent codes. It is noteworthy that no non-linear activation functions are applied in the mapping networks to maintain the linearity of the Gaussian distribution.

The spectral regularizer is defined as the L_1 regularization of the eigen-values of the covariance matrix, aiming to suppress behavior associated with small eigen-values:

$$\mathcal{L}_s^G = ||\mathbf{d}^g||_1 + ||\mathbf{d}^a||_1. \tag{7}$$

Here, $||\mathbf{d}^g||_1$ and $||\mathbf{d}^a||_1$ represent the L_1 norms of the vectors corresponding to the mapping networks for geometry and appearance latent codes. Applying L_1 regularization to the eigen-values of the covariance matrix encourages some eigen-values to become zero, promoting sparsity in the representation. This can lead to simpler and more interpretable representation learning in each geometry and appearance latent code by reducing the number of non-zero eigen-values, and consequently, the important features are autonomously selected.

The entire training loss function for the generator is calculated as the combination of the explained three terms: $\mathcal{L}^G = \mathcal{L}_{adv}^G + \lambda_{cl}\mathcal{L}_{cl}^G + \lambda_s\mathcal{L}_s^G$. Here, $\lambda_s = 1e^{-7}$ is experimentally set to control the importance of the sparsity term in the overall

loss function. This value is determined through empirical testing and tuning to achieve the desired balance between adversarial, contrastive, and spectral regularization losses during training.

4 Experiments

Datasets: We trained our model using three public datasets: FFHQ (FF) [17], CelebA-HQ (CA) [14], and AFHQ-V2 (AF) [5]. FF contains 70K high-quality human faces with variations in pose, expression, and lighting. CA includes 30K celebrity faces with diverse appearances, and AF comprises 15,803 animal faces with a wide range of visual characteristics. To adapt to different image sizes, we used eight, seven, and six residual blocks for 1024^2, 512^2, and 256^2 resolutions, respectively. The datasets were resized as follows: AF images to 512^2 and 256^2, and FFHQ and CelebA-HQ images to all three resolutions.
Training and Implementation Details: The model was trained using the PyTorch framework, with Albumentations used to apply image augmentations [3]. The Adam optimizer [20] was used with a learning rate of 0.002 for the 256^2 and 512^2 resolution datasets, and 0.001 for the 1024^2 resolution dataset, with $\beta_1 = 0.0$ and $\beta_2 = 0.99$. A batch size of $B = 32$ was used, and the model was trained for 450K, 700K, and 900K epochs for the AF, CA, and FF datasets. To improve the quality of generated images, the exponential moving average of the generator parameters [39] was employed with a decay rate of 0.9999, starting from the 5K-th training iteration. Additionally, FreezeD [24], which freezes the weight parameters of the discriminator layers until a spatial resolution of 64^2, was applied from the 150K-th iteration for the AF dataset, and from the 300K-th and 500K-th iteration for the CA and FF datasets. The projection heads of the discriminator consist of three fully connected layers, producing a 256-dimensional output vector. The length of the input latent vector $\mathbf{z}$ is 128, which is divided into two 64-dimensional vectors. Each latent code is then transformed into the geometry and appearance latent codes, $\mathbf{w}^g$ and $\mathbf{w}^a$, using mapping networks composed of subspace transformation and fully connected layers. To accelerate training, we employed a lazy regularization strategy, applying contrastive loss and spectral regularization every other iteration, and only adversarial loss otherwise. This approach sped up training while maintaining image quality. We trained the networks on 256^2, 512^2, and 1024^2 resolution images using four, four, and eight NVIDIA-H100 GPUs, respectively, with training times of approximately 8, 17, and 19 hours for 100K iterations.

4.1 High-quality Image Generation: Comparison with SOTAs

We compared the performance of our method with SOTA image generation methods using the FF, CA, and AF datasets. The quality of the generated images was evaluated using the FID metric [12], which measures the similarity between the distribution of real and generated images in the feature space of an Inception-v3 network. Lower FID scores indicate better image quality. To ensure

Table 1: FID evaluation results on FFHQ (FF), CelebA-HQ (CA), and AFHQ-V2 (AF) datasets at different image resolutions (256^2:L, 512^2:M, and 1024^2:H).

Method	FF-L	CA-L	AF-L	FF-M	CA-M	AF-M	FF-H	CA-H	controllability
StyleGAN2 [15]	-	-	-	-	-	4.62	2.70	-	✗
StyleGAN3 [16]	-	-	-	-	-	4.40	3.07	-	✗
SWAGAN-Bi [8]	5.22	-	-	-	-	-	4.06	-	✗
StyleNAT [35]	2.05	-	-	-	-	-	4.17	-	✗
MSG-GAN [13]	-	-	-	-	-	-	5.80	6.37	✗
CIPS [1]	4.38	-	-	6.18	-	-	10.07	-	✗
StyleSwin [41]	2.81	3.25	-	-	-	-	5.07	4.43	✗
HiT-B [43]	2.95	3.39	-	-	-	-	6.37	8.83	✗
WaveDiff [26]	-	5.94	-	-	6.40	-	-	-	✗
DDGAN [38]	-	7.64	-	-	8.43	-	-	-	✗
SeFa [30]	6.87	6.43	9.48	4.21	4.52	5.98	9.36	5.83	✓
EigenGAN [11]	13.90	14.05	-	13.99	11.07	-	18.51	16.31	✓
LC-GAN	3.65	3.72	5.77	3.36	3.49	4.74	3.32	3.46	✓

Fig. 5: Examples of images generated by the trained models. From left to right, the image sets show generated images by models trained on the FFHQ, CelebA-HQ, and AFHQ-V2 datasets at different image resolutions.

a fair comparison, we followed the testing protocols used in other papers, such as [1, 8, 13, 15, 16, 26, 35, 38, 41, 43].

For the FF dataset, we randomly sampled 50K images from the training dataset and generated 50K images using our trained generator, then calculated the FID between them. For the CA and AF datasets, we calculated FIDs between 30K and 15,803 generated images, respectively, and the entire training images. The compared algorithms were chosen among the SOTA methods that followed the same FID measurement protocols. Additionally, we trained SeFa [30] and EigenGAN [11] on all training datasets to ensure a comprehensive comparative analysis. However, attempts to apply EigenGAN on the AF dataset with different settings were unsuccessful due to mode collapse.

Table 1 provides the quantitative evaluation results, demonstrating the superiority of our proposed method in terms of FID scores. Particularly noteworthy are the excellent FID scores for FF and CA, comparable to StyleGAN3 and StyleSWIN, which prioritize high-quality and high-resolution image generation without emphasizing control ability. Figure 5 presents a selection of representative images generated by our LC-GAN, illustrating its capability to produce realistic and diverse images.

Fig. 6: Demonstration of semantic control in generated images. The image pairs from top to bottom are generated using LC-GAN, SeFa, and EigenGAN, where the image pairs were created by controlling the same semantic aspect (yaw change).

4.2 Ablation Study of Image Property Controllability

Evaluation Criteria: To evaluate the effectiveness of our proposed method in controlling image properties, we conducted an experiment to generate pairs of images. One image was generated randomly, while the other was generated by resampling one of the latent codes. Our goal was to ensure that the model was properly trained and that the generated image pairs had similar semantics, while the resampled image properties were updated. We used the Arcface-ResNet100 [6] to calculate the feature distance between the pairs of generated images. This measure indicates the semantic similarity between human faces, with a decision boundary trained to distinguish between the same and different identities using a threshold of 1. The feature distance between images of the same person should be less than 1, while the feature distance between images of different people should be greater than 1. In addition, we used MediaPipe's face mesh model [19] to obtain facial landmarks, enabling us to visually assess the geometric consistency of the generated faces across different identities. We calculated the distance between corresponding landmarks of the image pairs to evaluate the consistency of the generated faces. Overall, this experiment allowed us to demonstrate the effectiveness of our proposed method in controlling image properties while maintaining appearance similarity and geometric consistency.
Identity-Preserving Image Generation: In our experiments, we focused on generating human face images while preserving their identity. We generated 1,000 pairs of images using generators trained on the CA and FF datasets by resampling only the viewpoint component. Additionally, we used SeFa and EigenGAN to generate other sets of image pairs, manually exploring the control direction associated with the viewpoint in the latent space. To evaluate the quality of the generated image pairs, we plotted the distance distributions in Figure 7. The distances for our method were consistently lower compared to those obtained using SeFa and EigenGAN. To further assess identity similarity, we tested Arcface on randomly selected image pairs from the training dataset, and no false

Table 2: Perceptual distance between image pairs using Arcface-ResNet100.

Method	CA-L	FF-L	CA-M	FF-M	CA-H	FF-H
Random pairs	1.41	1.41	1.41	1.41	1.41	1.41
SeFa [30]	0.91	0.85	1.01	0.94	1.09	1.13
EigenGAN [11]	0.99	0.98	1.03	0.91	0.97	0.99
LC-GAN	0.71	0.64	0.78	0.74	0.71	0.61

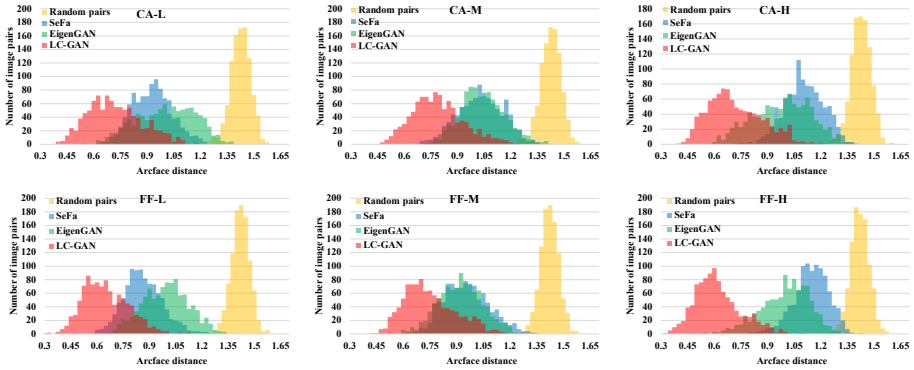

Fig. 7: Image pair similarity using Arcface feature distances. The results show that image pairs generated by our method have lower mean distances.

positive results were observed. This demonstrated the precise measurement of identity similarity by Arcface. Notably, Arcface consistently identified a significant majority of the image pairs generated by our method as depicting the same person, highlighting the robustness of our approach in preserving identity information. In contrast, the image pairs generated by SeFa and EigenGAN exhibited larger distance values, primarily due to variations introduced in both facial pose and appearance through principal direction changes. This impact was particularly notable in the CA-H and FF-H tests, where even slight changes in high-resolution images had a significant effect on the results. The results demonstrate the superior accuracy of our approach in maintaining identity consistency in the generated image pairs.

Viewpoint-Preserving Image Generation: In this experiment, our goal was to generate images that preserve viewpoints by controlling the appearance, resulting in 1,000 pairs of images. To assess the effectiveness of our method, we measured the distance between facial landmarks for each image pair. The results, summarized in Table 3, compare the facial landmark consistency of our approach with random pairs, SeFa, and EigenGAN across datasets with varying resolutions. Our approach consistently outperformed SeFa and EigenGAN, achieving the lowest facial landmark distances across all datasets, demonstrating superior preservation of viewpoints in generated images. SeFa and EigenGAN often compromised viewpoint preservation by slightly altering facial poses when regenerating images with appearance changes. Specifically, LC-GAN excels in learning features related to controlling mouth movements and subtle facial ex-

Fig. 8: Demonstration of semantic control in generated images. Image pairs from top to bottom are generated using LC-GAN, SeFa, and EigenGAN, illustrating controlled changes in the same semantic aspect (identity), along with face landmarks estimated by MediaPipe's face mesh model.

Table 3: Landmark distance between image pairs using MediaPipe's face mesh.

Method	CA-L	FF-L	CA-M	FF-M	CA-H	FF-H
Random pairs	7.63	8.70	15.17	17.08	30.15	34.71
SeFa [30]	2.41	2.64	5.63	6.49	10.42	11.34
EigenGAN [11]	2.97	2.61	5.86	4.93	13.46	13.99
LC-GAN	1.65	1.97	3.66	4.19	5.83	9.09

pressions independently of identity changes. In contrast, SeFa and EigenGAN often exhibit entanglement of these features with changes in identity. Visual examples illustrating the effectiveness of our method in preserving viewpoints are available in Figure 8.

Effectiveness of Contrastive Loss and Spectral Regularization: To validate the effectiveness of contrastive loss (CL) and spectral regularization (SR), we conducted ablation experiments by training the model with and without these terms. Focusing on the CA-L and FF-L datasets, we found that removing both terms led to entangled features solely within the appearance latent code. The top images in Figure 9 demonstrate the lack of disentanglement when both CL and SR are removed. In contrast, applying CL without SR resulted in more uniformly distributed features across dimensions. When controlling a target attribute, it required editing multiple dimensions to achieve similar changes to the full model, as the target features were spread across more dimensions. The middle images in Figure 9 illustrate the increased need for manipulating multiple dimensions to achieve similar variation compared to the full model. Regarding FID scores, no significant differences were observed, although there was a slight degradation when each component was applied individually: 3.54 and 3.66 for CA-L, and 3.30 and 3.45 for FF-L, without CL and SR, and only with CL, respectively.

Visual Analysis: Our exploration of the learned semantics revealed that the proposed method excels in capturing both global features and intricate, specific

Fig. 9: Comparison of ablated models controlling a dimension for yaw changes: without CL and SR (top), with CL only (middle), and full model (bottom).

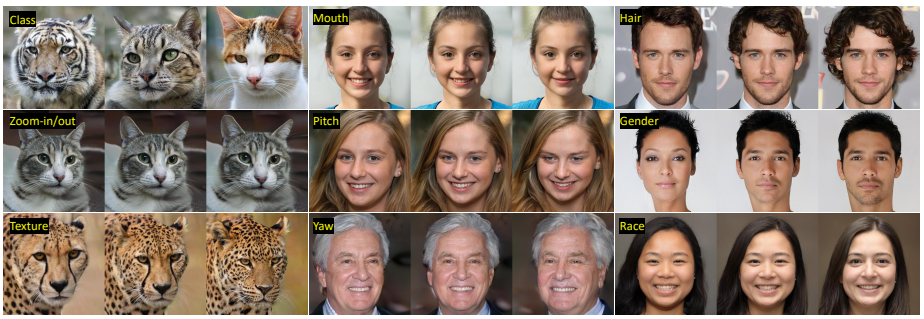

Fig. 10: Exploring the latent space: Our approach facilitates the exploration of semantically meaningful features controlling various image properties.

attributes. Notably, the model can independently manipulate detailed behaviors such as zoom-in/out and mouth movements, as shown in Figure 10. The generated results highlight the model's enhanced controllability over these attributes. For a more comprehensive exploration of the diverse set of discovered semantic attributes, we recommend referring to the provided source code.

5 Conclusion and Limitations

We presented LC-GAN, a novel GAN framework for controllable image generation. By integrating unsupervised disentanglement techniques with the StyleGAN architecture, LC-GAN produces high-quality images with control over image properties. Extensive experiments on multiple datasets, including FF, CA, and AF at different resolutions, demonstrated that LC-GAN outperforms SOTA models in terms of FID scores and controllability in image synthesis. However, LC-GAN has some limitations. The training process is time-consuming and requires larger GPU memory, especially when incorporating contrastive loss. Additionally, achieving precise metric-level control, such as a 20-degree rotation, can be challenging due to the unsupervised nature of the method. Future research should focus on addressing these limitations and improving the efficiency and precision of controllable image generation methods.

References

1. Anokhin, I., Demochkin, K., Khakhulin, T., Sterkin, G., Lempitsky, V., Korzhenkov, D.: Image generators with conditionally-independent pixel synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14278–14287 (2021)
2. Axel Sauer, A.G.: Counterfactual generative networks. In: International Conference on Learning Representations (2021)
3. Buslaev, A., Iglovikov, V.I., Khvedchenya, E., Parinov, A., Druzhinin, M., Kalinin, A.A.: Albumentations: fast and flexible image augmentations. Information **11**(2), 125 (2020)
4. Choi, J., Lee, J., Yoon, C., Park, J.H., Hwang, G., Kang, M.: Do not escape from the manifold: Discovering the local coordinates on the latent space of gans. arXiv preprint arXiv:2106.06959 (2021)
5. Choi, Y., Uh, Y., Yoo, J., Ha, J.W.: Stargan v2: Diverse image synthesis for multiple domains. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
6. Deng, J., Guo, J., Xue, N., Zafeiriou, S.: Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4690–4699 (2019)
7. Deng, Y., Yang, J., Chen, D., Wen, F., Tong, X.: Disentangled and controllable face image generation via 3d imitative-contrastive learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5154–5163 (2020)
8. Gal, R., Hochberg, D.C., Bermano, A., Cohen-Or, D.: Swagan: A style-based wavelet-driven generative model. ACM Transactions on Graphics (TOG) **40**(4), 1–11 (2021)
9. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. Advances in neural information processing systems **27** (2014)
10. Härkönen, E., Hertzmann, A., Lehtinen, J., Paris, S.: Ganspace: Discovering interpretable gan controls. Advances in Neural Information Processing Systems **33**, 9841–9850 (2020)
11. He, Z., Kan, M., Shan, S.: Eigengan: Layer-wise eigen-learning for gans. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 14408–14417 (2021)
12. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
13. Karnewar, A., Wang, O.: Msg-gan: Multi-scale gradients for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7799–7808 (2020)
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of GANs for improved quality, stability, and variation. In: International Conference on Learning Representations (2018), `https://openreview.net/forum?id=Hk99zCeAb`
15. Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., Aila, T.: Training generative adversarial networks with limited data. Advances in neural information processing systems **33**, 12104–12114 (2020)
16. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Advances in Neural Information Processing Systems **34**, 852–863 (2021)

17. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
18. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
19. Kartynnik, Y., Ablavatski, A., Grishchenko, I., Grundmann, M.: Real-time facial surface geometry from monocular video on mobile gpus. arXiv preprint arXiv:1907.06724 (2019)
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: International Conference on Learning Representations (2015)
21. Kowalski, M., Garbin, S.J., Estellers, V., Baltrušaitis, T., Johnson, M., Shotton, J.: Config: Controllable neural face image generation. In: European Conference on Computer Vision. pp. 299–315. Springer (2020)
22. Li, B., Qi, X., Lukasiewicz, T., Torr, P.: Controllable text-to-image generation. Advances in Neural Information Processing Systems **32** (2019)
23. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? In: International conference on machine learning. pp. 3481–3490. PMLR (2018)
24. Mo, S., Cho, M., Shin, J.: Freeze the discriminator: a simple baseline for fine-tuning gans. In: CVPR AI for Content Creation Workshop (2020)
25. Peebles, W., Peebles, J., Zhu, J.Y., Efros, A., Torralba, A.: The hessian penalty: A weak prior for unsupervised disentanglement. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VI 16. pp. 581–597. Springer (2020)
26. Phung, H., Dao, Q., Tran, A.: Wavelet diffusion models are fast and scalable image generators. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
27. Qiao, T., Zhang, J., Xu, D., Tao, D.: Mirrorgan: Learning text-to-image generation by redescription. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1505–1514 (2019)
28. Qiu, H., Yu, B., Gong, D., Li, Z., Liu, W., Tao, D.: Synface: Face recognition with synthetic data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10880–10890 (2021)
29. Shen, Y., Gu, J., Tang, X., Zhou, B.: Interpreting the latent space of gans for semantic face editing. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9243–9252 (2020)
30. Shen, Y., Zhou, B.: Closed-form factorization of latent semantics in gans. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1532–1540 (2021)
31. Shi, Y., Yang, X., Wan, Y., Shen, X.: Semanticstylegan: Learning compositional generative priors for controllable image synthesis and editing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11254–11264 (2022)
32. Shoshan, A., Bhonker, N., Kviatkovsky, I., Medioni, G.: Gan-control: Explicitly controllable gans. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14083–14093 (2021)
33. Song, Y., Sebe, N., Wang, W.: Orthogonal svd covariance conditioning and latent disentanglement. IEEE Transactions on Pattern Analysis and Machine Intelligence (2022)

34. Tewari, A., Elgharib, M., Bharaj, G., Bernard, F., Seidel, H.P., Pérez, P., Zollhofer, M., Theobalt, C.: Stylerig: Rigging stylegan for 3d control over portrait images. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6142–6151 (2020)
35. Walton, S., Hassani, A., Xu, X., Wang, Z., Shi, H.: Stylenat: Giving each head a new perspective. arXiv preprint arXiv:2211.05770 (2022)
36. Wei, Y., Shi, Y., Liu, X., Ji, Z., Gao, Y., Wu, Z., Zuo, W.: Orthogonal jacobian regularization for unsupervised disentanglement in image generation. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6721–6730 (2021)
37. Wu, Z., Lischinski, D., Shechtman, E.: Stylespace analysis: Disentangled controls for stylegan image generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12863–12872 (2021)
38. Xiao, Z., Kreis, K., Vahdat, A.: Tackling the generative learning trilemma with denoising diffusion GANs. In: International Conference on Learning Representations (2022), https://openreview.net/forum?id=JprM0p-q0Co
39. Yaz, Y., Foo, C.S., Winkler, S., Yap, K.H., Piliouras, G., Chandrasekhar, V., et al.: The unusual effectiveness of averaging in gan training. In: International Conference on Learning Representations (2018)
40. Yin, X., Yu, X., Sohn, K., Liu, X., Chandraker, M.: Towards large-pose face frontalization in the wild. In: Proceedings of the IEEE international conference on computer vision. pp. 3990–3999 (2017)
41. Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., Guo, B.: Styleswin: Transformer-based gan for high-resolution image generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11304–11314 (2022)
42. 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. pp. 5907–5915 (2017)
43. Zhao, L., Zhang, Z., Chen, T., Metaxas, D., Zhang, H.: Improved transformer for high-resolution gans. Advances in Neural Information Processing Systems **34**, 18367–18380 (2021)