Intelligent Multimedia Processing & Communication Systems Journal

IMPCSJ

J IMPCS (2025) 21: 1-11

DOI 10.71856/IMPCS.2025.1215292

Research Paper

A Hybrid GAN Architecture for Stable and High-Quality Image Generation: Integrating AdaptiveMix, DiffAugment, and EIGGAN-Inspired Techniques

Negar Tahvili[1], Shahla Nemati[2]*, Mohammad Ehsan Basiri[3]

1. MSc. Student, Department of Computer Engineering, Shahrekord University, Shahrekord, Iran
2. Associate Professor, Department of Computer Engineering, Shahrekord University, Shahrekord, Iran

**Corresponding Author,* s.nemati@sku.ac.ir

3. Associate Professor, Department of Computer Engineering, Shahrekord University, Shahrekord, Iran

Article Info

Article history:
Received: 28 July 2025
Accepted: 7 Sep 2025

Keywords:
AdaptiveMix,
DiffAugment,
Discriminator Accuracy,
EIGGAN,
Generative Adversarial Networks,
Inception Score.

ABSTRACT

Generative Adversarial Networks (GANs) have become influential tools in unsupervised image generation, significantly impacting fields like computer vision and the creative arts. However, challenges such as training instability and mode collapse often hinder their performance and the quality of generated images. This study introduces a hybrid GAN architecture that combines techniques from AdaptiveMix, which enhances model stability, with insights from EIGGAN, known for its innovative image generation methods. The primary goal is to improve both training stability and the visual quality of generated images.

The generator incorporates differentiable data augmentation (DiffAugment) and Exponential Moving Average (EMA) updates. DiffAugment introduces dynamic transformations to training data, enhancing diversity and robustness, while EMA updates stabilize training by smoothing parameter changes, resulting in more consistent outputs. The discriminator is regularized using the R1 penalty, improving its ability to distinguish between real and generated images, and benefits from feature space shrinkage through AdaptiveMix to maintain compact feature representations.

Trained on the CIFAR-10 dataset for 1000 epochs, the model achieved a peak Inception Score (IS) of 6.80 ± 0.22, indicating significant improvements in generative quality and diversity, along with a best Fréchet Inception Distance (FID) score of 20.90, reflecting high realism in generated samples. The discriminator's accuracy remained stable between 50% and 60%, suggesting a balanced adversarial relationship. These findings demonstrate the effectiveness of the hybrid model.

They also open new directions for improving stability and image quality in GAN training.

Publisher: Islamic Azad University – Zanjan Branch

I. Introduction

Generative Adversarial Networks (GANs), introduced in 2014 by Goodfellow et al., represent a significant breakthrough in the field of unsupervised machine learning, demonstrating remarkable capabilities in generating synthetic data, particularly realistic images [1]. The basic GAN architecture consists of two deep neural networks: a generator, responsible for producing new data samples (e.g., images) from random noise, and a discriminator, tasked with distinguishing real samples from those generated by the generator.

The training process of GANs can be likened to a zero-sum game in which the two components compete with opposing objectives. The generator aims to produce images so realistic that the discriminator cannot differentiate them from real ones. Conversely, the discriminator strives to correctly classify images as either real or fake. This dynamic competition ultimately leads to the simultaneous improvement of both networks: the generator becomes capable of producing highly realistic data, while the discriminator becomes more adept at detecting subtle differences between real and generated data. This adversarial nature makes GANs a powerful tool for tasks such as image generation, image-to-image translation, image super-resolution, and many other applications [1].

However, despite their impressive potential, achieving stable and effective GAN training remains one of the major challenges in deep learning. Instability during training can manifest in various forms, such as vanishing or exploding gradients, which hinder effective learning; mode collapse, where the generator produces only a limited subset of the training data distribution; and oscillatory behavior, where the model fails to converge and alternates between different states. These issues primarily arise from a delicate imbalance between the generator and discriminator as well as the inherently dynamic and unstable nature of the discriminator's data distribution during training [1].

In recent years, extensive research has been devoted to overcoming these challenges and enhancing GAN stability and output quality. Efforts have included developing novel architectures (e.g., DCGAN [1], StyleGAN [2]) introducing more advanced loss functions (e.g., WGAN-GP [5], LSGAN [6]) and employing new regularization [3] and data augmentation strategies [4]. Among these approaches, two recent advancements that are of particular interest in this study are the methods presented in [3] and [4], namely AdaptiveMix and EIGGAN, respectively, both of which propose innovative solutions to improve training stability and the quality of generated images.

The main objective of this research is to implement and evaluate an advanced GAN model that strategically combines the best practices and techniques inspired by AdaptiveMix with selected components of EIGGAN. This combination is designed with the ultimate goal of achieving higher Inception Scores (IS) — a key metric for assessing the quality and diversity of generated images — while also substantially improving training stability.

AdaptiveMix addresses GAN training stability through a novel approach: reducing and compressing the training data regions in the discriminator's feature space. This is achieved by creating hard samples via linear interpolation of pairs of real training images using the Mixup technique [5]. By forcing the discriminator to bring the features of these mixed samples closer to their original samples, the decision boundaries become more stable, preventing excessive overlap between real and generated distributions. This, in turn, improves quality and mitigates mode collapse. In this project's implementation, specific Mixup parameters (e.g., mixup_alpha) are used to control this process.

On the other hand, EIGGAN focuses on enhancing the generator's capabilities to produce higher-quality images by introducing several innovations, including spatial attention in the generator to extract salient information and increase realism, parallel residual operations to capture richer structural details from different network layers, and a composite loss function to balance speed and accuracy in optimization. In the provided code, learning rate adjustments inspired by EIGGAN are applied to accelerate convergence [4]. Additionally, R1 regularization [6] — a gradient penalty that helps stabilize the discriminator — and feature matching [7], which encourages the generator to match the intermediate feature statistics of the discriminator to those of real data, are directly inspired by EIGGAN principles. The inclusion of self-attention further aligns with EIGGAN's objective of capturing long-range dependencies in images.

In addition to integrating AdaptiveMix and EIGGAN techniques, this study employs other advanced methods such as Differentiable Augmentation (DiffAugment) [8] — essential for improving stability and boosting IS— and Exponential Moving Average (EMA) [9] for the generator, which has been empirically shown to produce smoother and higher-quality images, resulting in improved evaluation metrics.

The remainder of this paper first provides a detailed analysis of the theoretical foundations and operational mechanisms of AdaptiveMix and EIGGAN. Next, the exact structure of the implemented code is described, illustrating how these two key approaches, along with other advanced techniques, are integrated into the proposed model. Finally, experimental results from training on the CIFAR-10 dataset including IS progression and discriminator accuracy over time are presented and thoroughly analyzed to assess the effectiveness of the proposed hybrid approach.

II. Background and Literature Review

Generative Adversarial Networks (GANs), introduced in 2014, revolutionized the field of realistic image synthesis [1]. This innovative framework, based on a minimax game between two competing networks —a generator that produces synthetic images and a discriminator that attempts to distinguish real images from fake ones— has demonstrated unprecedented capabilities in synthesizing novel data. However, training these networks has consistently posed significant challenges, including instability in convergence, issues with unstable gradients, and mode collapse, where the generator produces only a limited variety of samples [2].

To address these fundamental problems, numerous studies have proposed solutions such as Wasserstein GAN (WGAN) [2] and WGAN-GP [10], which introduce novel loss functions such as the Wasserstein distance and gradient penalty to improve training stability. Normalization techniques like Batch Normalization [11] and Spectral Normalization [12] also play a crucial role in stabilizing training and controlling gradient behavior. Moreover, to enhance visual quality and the fine details of generated images, methods such as Self-Attention GAN [13], which incorporates attention layers into the architecture, and EMA [9] to smooth generator weights over time, have been employed. DiffAugment [8] has also proven to be an effective approach for stabilizing GAN training, particularly in low-data scenarios.

Building on these advancements, two recent and influential approaches namely AdaptiveMix [3] and EIGGAN [4] have emerged as promising methods for improving GAN stability and output quality.

AdaptiveMix Inspired by research in robust image classification, introduces a simple yet highly effective module for GANs [3]. It stabilizes training and mitigates mode collapse by compressing the training data regions within the discriminator's feature space. This compression is achieved using the Mixup technique [5], which linearly interpolates pairs of real images to create hard samples. AdaptiveMix then reduces the feature-space distance between these hard samples and easy samples in the discriminator's latent space. This approach ensures that the discriminator focuses on learning the real data distribution, providing healthier gradients to the generator and ultimately leading to higher-quality and more diverse outputs. Due to its "plug-and-play" nature, AdaptiveMix can be seamlessly integrated into existing GAN architectures [3].

EIGGAN designed to enhance the generator's ability to produce high quality images [4]. It introduces several architectural and training innovations. It employs a spatial attention mechanism [12] within the generator to capture salient features and improve realism, and integrates parallel residual operations [13] to extract richer structural information from multiple layers. EIGGAN also adopts a composite loss function [14] incorporating regularization techniques such as R1 Regularization [6] and Feature Matching [7]. These regularizes help stabilize training and improve the final image quality, striking a balance between speed and accuracy in generating realistic outputs.

A summary of related studies from recent years is provided in Table I.

Inspired by these advancements, this study aims to intelligently and selectively integrate techniques from AdaptiveMix [3], optimization and regularization strategies proposed in EIGGAN [4], and other effective methods such as DiffAugment [8], EMA [9], R1 Regularization [6], and Feature Matching [7] into a unified GAN architecture. This hybrid approach is designed to produce a stable and efficient GAN capable of achieving higher Inception Scores (IS), reducing Fréchet Inception Distance (FID), and significantly improving training stability.

III. Dataset

In this study, the publicly available and widely used CIFAR-10 dataset [14] was employed to evaluate the performance of the proposed model. This dataset consists of 60,000 color images of size 32×32 pixels, categorized into 10 distinct classes. Each class contains 6,000 images, with 5,000 allocated for training and 1,000 reserved for testing. The classes in CIFAR-10 include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. The selection of CIFAR-10 was motivated by its diverse class composition and relatively small image size, making it a standard and suitable benchmark for assessing the generative capabilities and training stability of GAN models. These characteristics also enable fair comparison with results from previous GAN-related studies.

IV. Data Preprocessing

To train GAN models using the CIFAR-10 dataset, specific preprocessing and data augmentation procedures were applied to enhance training stability and the quality of generated images. These steps are as follows:

A. Image Standardization and Tensor Conversion

CIFAR-10 color images, which are natively 32×32 pixels, were resized to the same dimensions to ensure a standardized data processing pipeline. This ensures that even if images with different resolutions are used in the future, the model consistently receives inputs with the expected dimensions. The images were then converted into PyTorch tensors, the required format for GPU-based computation.

B. Pixel Value Normalization

This step consists of two parts. First, pixel values were normalized from the range [0, 255] to [0, 1], a standard step in deep learning for scaling input data.

Second, to align with the tanh activation function typically used in the generator's output layer (which produces values in the range [−1, 1]), the images were further normalized to the range [−1, 1]. This alignment facilitates more stable convergence by ensuring that the real data distribution matches the generator's output distribution, thus enabling more efficient learning.

C. Differentiable Data Augmentation

This is one of the most important techniques for improving GAN performance, particularly when working with small datasets. Unlike traditional augmentation methods that apply transformations once before training (e.g., cropping or rotation), DiffAugment applies random, differentiable transformations such as slight rotations, color changes, and cropping at every training step. These transformations are implemented in a way that allows gradients to flow through them during backpropagation.

This approach prevents memorization by forcing the discriminator to learn the actual data distribution rather than memorizing specific training samples. It also mitigates overfitting, encouraging the generator to produce more diverse and higher-quality images, thereby improving training stability.

TABLE I: Summary of Related GAN Architectures and Their Evaluation Metrics

No.	YEAR	Method	Description	Dataset(s)
1	2014	Generative Adversarial Networks[1](GANs)	Introduction of the original GAN architecture consisting of a generator and a discriminator	MNIST, CelebA (implicitly)
2	2017	Wasserstein GAN Error! Reference source not found.(WGAN)	Introduction of the Wasserstein distance as a loss function to improve training stability and reduce mode collapse	CIFAR-10, LSUN
3	2023	AdaptiveMix[3]	Compressing training data regions in the discriminator's feature space using the Mixup technique for improved stability	LSUN-Bedroom, CelebA-HQ
4	2024	EIGGAN (Enhanced GAN) Error! Reference source not found.	Incorporates spatial attention, parallel residual operations, and a composite loss function for higher-quality image generation and faster convergence	CIFAR-10, CelebA
5	2020	Differentiable Augmentation (DiffAugment)Error! Reference source not found.	Differentiable data augmentation for efficient GAN training in limited-data scenarios and improved stability	CIFAR-10, Tiny ImageNet
6	2019	Exponential Moving Average (EMA) Error! Reference source not found.	Applies exponential moving average to generator weights for smoother and higher-quality image outputs	CelebA-HQ, FFHQ

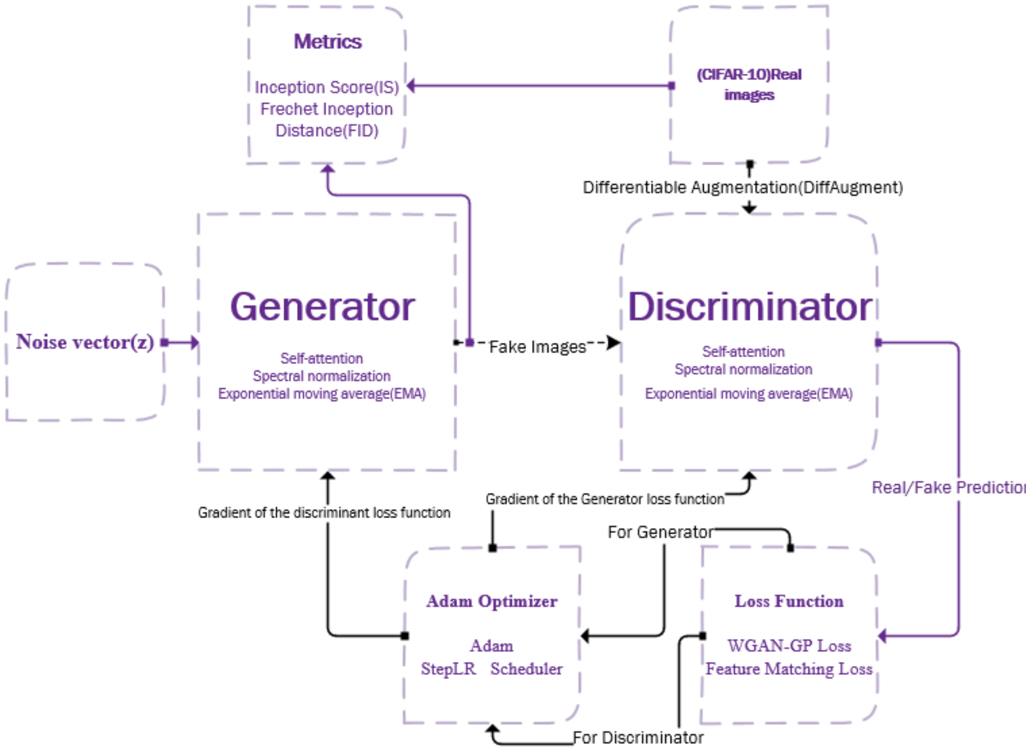

Fig. 1. Architecture of the proposed model.

Overall, these preprocessing steps prepare the input data for effective and efficient GAN training, contributing to improved accuracy and realism in the generated image.

V. Proposed Model Architecture

In this study, a Generative Adversarial Network (GAN) architecture comprising a generator and a discriminator was designed for generating CIFAR-10 images. The generator transforms random noise into an image, while the discriminator distinguishes between real and generated images. This architecture incorporates principles from AdaptiveMix Error! Reference source not found. and selected techniques from EIGGAN Error! Reference source not found. to achieve higher stability and quality. Furthermore, differentiable data augmentation Error! Reference source not found. is applied to the discriminator's input, and an Exponential Moving Average (EMA) Error! Reference source not found. is used for the generator's weights, both of which contribute to improved stability and enhanced image quality. Figure 1 illustrates the proposed model architecture.

The proposed architecture is built upon the foundation of GANs, which consist of two cores, competing components: the generator and the discriminator. The objective of this structure is to generate realistic, high-quality images specifically for the CIFAR-10 dataset, while leveraging novel approaches in GAN training Error! Reference source not found.. The following subsections detail the key components and techniques that shaped the proposed architecture.

A. General GAN Architecture (Generator and Discriminator)

At the heart of the architecture, the generator is responsible for transforming a random noise vector into a synthetic image, while the discriminator differentiates between real images and those generated by the generator. These two networks are trained alternately in an adversarial manner.

The generator is constructed using upsampling layers (e.g., transposed convolutions), and the discriminator is built using downsampling layers (e.g., convolutions). To stabilize training, normalization layers are employed: Batch Normalization Error! Reference source not found. in the generator and Spectral Normalization Error! Reference source not found. in the discriminator.

The primary goal of a GAN is to train both the generator (G) and discriminator (D) in a zero-sum game. The Minimax objective function for GANs is defined as Error! Reference source not found.]:

$$\min_G \max_D V(D,G) = \{E\}_{\{x\sim p_{\{data\}(x)}\}}[\log D(x)] + \{E\}_{\{z\sim p_{z(z)}\}}\left[\log\left(1-D(G(z))\right)\right] \quad (1)$$

$D(x)$: Probability that X is a real sample (discriminator output for real data).
$G(z)$: Sample generated by the generator from noise z.
$D(G(z))$: Probability that the generated sample is real (discriminator output for fake data).
$E_{x\sim pdata(x)}$: Expectation over real samples from the data distribution.
$E_{z\sim pz(z)}$: Expectation over noise vectors from the noise distribution.

B. AdaptiveMix Technique

AdaptiveMix Error! Reference source not found. is a novel approach designed to improve GAN training stability, with a particular focus on the discriminator. This technique compresses the discriminator's feature space so that it does not only focus on clear-cut distinctions between real and fake samples, but also learns fine-grained boundaries in the feature space. Inspired by Mixup Error! Reference source not found., a data augmentation method that linearly combines images and their labels, AdaptiveMix enables the discriminator to learn more precise distinctions in feature-space boundary regions. This helps mitigate mode collapse and encourages the discriminator to provide more stable and informative feedback to the generator.

The standard Mixup formulation for creating mixed samples and labels is given by:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j$$
$$\tilde{y} = \lambda y_i + (1-\lambda)y_j \quad (2)$$

Where x_i and x_j are two input samples from the dataset, y_i and y_j are their corresponding labels, and λ is a mixing coefficient typically drawn from a Beta distribution ($Beta\ (\alpha,\alpha)$).

C. EIGGAN Inspired Technique

The EIGGAN framework [4] introduces several architectural innovations and training strategies designed to improve image quality and training stability in GANs. These include spatial attention modules, parallel residual operations, and composite loss functions.

In our proposed model, we did not employ the complete EIGGAN structure. Instead, we selectively integrated specific techniques inspired by EIGGAN to enhance efficiency and stability while keeping the architecture lightweight.

The components we adopted are:

Self-Attention Mechanism: Incorporated into the generator and discriminator to allow the model to capture long-range dependencies and focus on critical regions of the image.

Feature Matching Loss: Added to the generator's objective to stabilize training and improve the perceptual quality of generated images.

R1 Regularization: Applied to the discriminator to encourage smoother decision boundaries and further stabilize adversarial training.

Learning Rate Scheduling: Inspired by EIGGAN's training refinements, we employ step-based scheduling to ensure stable convergence.

The components not adopted from EIGGAN include full spatial attention blocks, parallel residual operations, and the complete composite loss function. This selective adoption allows our model to remain computationally efficient while benefiting from EIGGAN's most impactful elements.

D. Differentiable Augmentation (DiffAugment)

DiffAugment Error! Reference source not found. is a key technique in modern GAN training, particularly in low-data regimes. Rather than applying augmentation only to real images, DiffAugment applies differentiable transformations (e.g., cropping, rotation, color adjustments—operations for which gradients can be computed) to both real and generated (fake) images before feeding them into the discriminator. This forces the discriminator to learn the genuine characteristics of the data distribution rather than memorizing training examples or augmentation-specific artifacts. As a result, GAN training stability is significantly improved, and the quality of generated images is enhanced.

E. Exponential Moving Average (EMA)

EMA Error! Reference source not found. is a widely used post-training technique for improving the quality of generator outputs in GANs. During training, instead of directly using the generator's weights at each iteration, a shadow copy of the generator's weights is maintained and updated using an exponential moving average of the current and past weights:

$$\theta_{\{(t)\}}^{\{EMA\}} = \alpha \cdot \theta_{\{(t-1)\}}^{\{EMA\}} + (1 - \alpha) \cdot \theta_{\{(t)\}} \tag{3}$$

where $\theta_{\{(t)\}}^{\{EMA\}}$ is the EMA weight at time step t ،$\theta_{\{(t)\}}$ is the current generator weight at t and α is the decay rate, typically close to 1 (e.g., 0.999).

At evaluation time and during final sample generation, the EMA-smoothed generator weights are used, resulting in smoother and visually higher-quality outputs due to reduced fluctuations from early training stages.

F. Wasserstein Loss with Gradient Penalty (WGAN-GP)

To improve training stability and address common issues in early GANs such as mode collapse and unstable gradients, the proposed architecture adopts the Wasserstein GAN with Gradient Penalty (WGAN-GP) Error! Reference source not found.. This method builds upon WGAN Error! Reference source not found. by enforcing the Lipschitz constraint through a gradient penalty applied to the norm of the discriminator's gradient with respect to its inputs. The gradient penalty helps the discriminator learn better cost functions, leading to more stable convergence and higher-quality image generation. This approach is particularly effective when training GANs on complex datasets such as CIFAR-10.

G. Integrated and Synergistic Methods in the Proposed Model

The proposed model in this study involves an intelligent integration and synergy of a set of advanced techniques within a single GAN framework. The objective of this integration is to overcome the common challenges in GAN training and to achieve higher-quality and more stable image generation. This synergy encompasses the competitive interaction between the generator and discriminator, as well as the application of multiple enhancement mechanisms to both model components and their training process.

VI. Evaluation and Comparison Method

To assess the performance of the proposed model in generating high-quality images, particularly for the CIFAR-10 dataset, standard and widely used evaluation metrics in the GAN domain were employed. Unlike classification models, which are evaluated based on prediction accuracy, GANs are assessed based on the visual quality, diversity, and statistical similarity of generated images to real data.

In this study, the model's performance was evaluated using the Inception Score (IS), Fréchet Inception Distance (FID), and discriminator accuracy throughout the training process (up to epoch 1000).

IS evaluates both image quality (how semantically clear and recognizable the generated images are) and the diversity of the generated samples (how well the generator can produce images across different classes). A *higher* IS indicates that the generated images are both semantically meaningful (recognizable by the Inception V3 model) and diverse across generated classes. This metric is widely used for evaluating GAN performance, especially on datasets like CIFAR-10.

FID is a more comprehensive and precise metric for evaluating GAN-generated image quality. FID measures the distance between the feature distribution of real images and that of generated images. *Lower* FID values indicate higher quality and greater similarity to real images. Due to its sensitivity to both quality and diversity, FID is considered one of the most reliable metrics for GAN evaluation.

Discriminator Accuracy indicates the ability of the discriminator to distinguish real images from fake images generated by the generator. In a stable GAN, the discriminator accuracy typically oscillates around 50%–70%, reflecting a healthy balance between the generators and discriminator's competitive abilities. If the discriminator's accuracy approaches 100%, it means it has become too strong and is no longer providing useful gradients to the generator; if it approaches 0%, it indicates a weak discriminator.

VII. Evaluation Results

The proposed model was evaluated using a set of test images that were not seen during training. This evaluation focused on the IS, FID, and discriminator accuracy metrics. The proposed hybrid model, which incorporates AdaptiveMix, DiffAugment, and selectively applied EIGGAN principles, was trained on the CIFAR-10 dataset for 1000 epochs.

Table II presents the specifications of the software and hardware used in this study. Additionally, Table III shows the approximate time required to fully train selected models with an IS lower than 6.80. The experimental results, including IS progression across epochs and the discriminator accuracy chart, are reported in Tables IV and V, demonstrating the gradual improvement in generation quality and training stability of the model.

To rigorously evaluate the performance of GAN-based generative models, three commonly used metrics such as IS, FID, and discriminator accuracy, were employed. The results of the proposed model on the CIFAR-10 dataset were examined over the range of 0 to 1000 epochs, and the trends of these three indicators were analyzed separately.

In Figure 2 discriminator accuracy curve illustrates the discriminator's accuracy during training epochs. As shown in the figure, the discriminator's accuracy fluctuates on average between 50% and 60%. This behavior is expected and indicates relative training stability, as in GAN architectures, a balance must be maintained between the generator and discriminator so that neither dominates the other.

TABLE II: Specifications of the Hardware and Software Used

Category	Specification	Details
Hardware	GPU	NVIDIA Tesla T4 with 16 GB VRAM (Google Colab environment)
	Storage	SSD – temporary storage space allocated by Colab
Software	Operating System	Ubuntu – cloud environment (Google Colab)
	Development Environment	Google Colab, including Jupyter Notebook/Lab
	Python	Python 3.x – default version provided by Colab
	Main Libraries	PyTorch – default Colab version, CUDA-compatible
		torchvision – default in Colab
		scipy – default in Colab

TABLE III: Training and Inference Time

Model Name	Training & Inference Time
VanillaGAN	5-6 hours
DCGAN	6–7 hours
DRAGAN	7–8 hours
WGAN	8–9 hours
MIX + WGAN	9–10 hours
Improved GANs	9–10 hours
DCGAN (with labels)	10–11 hours
ALI	11–12 hours
BEGAN	11–13 hours
WGAN-GP	13–15 hours
WGAN-GP (general)	13–15 hours
SteinGAN	16–18 hours
Proposed Model	22–24 hours

TABLE IV Inception Score (IS) Comparison of the Proposed Model with Baseline GANs on CIFAR-10

Model Name	IS
WGAN Error! Reference source not found.	3.25
MIX + WGAN Error! Reference source not found.	4.04
Improved GANs [18]	4.36
ALI [17]	5.34
BEGANError! Reference source not found.	5.62
WGAN-GP[15]	5.99
DRAGAN Error! Reference source not found.	6.11
Vanilla GANError! Reference source not found.	6.30
SteinGANError! Reference source not found.	6.35
DCGAN [4]	6.37
WGAN-GP (common) [10]	6.40
DCGAN (with labels) [10]	6.58
Proposed Model	**6.80**

Specifically, in the mid-training stages, drops in discriminator accuracy were observed, which can be attributed to the generator's improved performance in producing more realistic-looking samples. In the final epochs, the discriminator's accuracy, while maintaining minor fluctuations, becomes more stable.

In addition, the IS and FID scores were recorded. The highest IS achieved was 6.80 at epoch 875, which is comparable to some of the most advanced models reported in prior works. The best FID obtained was 20.90 at epoch 920, indicating high quality and realism of the generated images. These IS and FID results demonstrate that the proposed model maintained a stable and upward performance trend, confirming the improvement in both quality and diversity of generated images during the training process.

Table IV Comparison of the IS Score of the Proposed model with selected notable models also presents results such as WGAN (IS = 3.25), BEGAN (IS = 5.62), WGAN-GP (~6.4), and DCGAN (IS = 6.16). As can be seen, the proposed model, with an IS close to 6.80, demonstrates competitive and in some cases superior performance compared to baseline models.

TABLE V: Fréchet Inception Distance (FID) Comparison of the Proposed Model with Baseline GANs on CIFAR-10

Model Name	FID
WGAN Error! Reference source not found.	55.96
HingeGAN [19]	42.40
LSGAN Error! Reference source not found.	42.02
DCGAN Error! Reference source not found.	38.56
WGAN-GP Error! Reference source not found.	41.86
Re-implemented WGAN-GPError! Reference source not found.	38.63
Realness GAN-Obj.1 Error! Reference source not found.	36.73
Realness GAN-Obj.2[22]	34.59
Realness GAN-Obj.3 [22]	36.21
AdaptiveMix[3]	30.85
Proposed Model	**20.90**

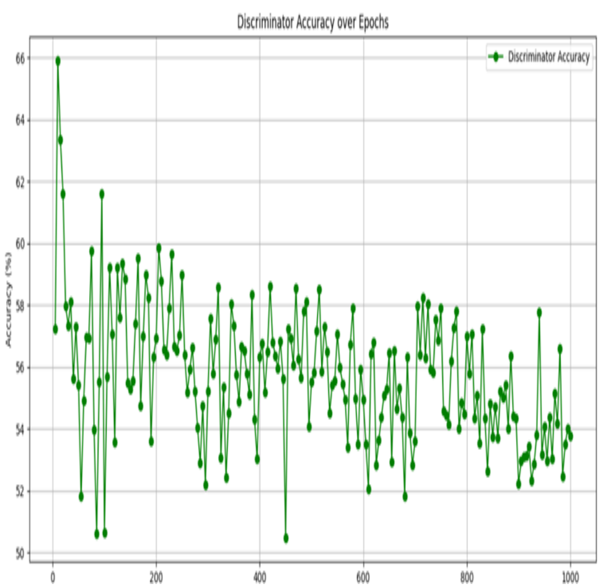

Fig. 2. Discriminator Accuracy Curve

Similarly, Table V Comparison of the FID Score of the proposed model with selected state-of-the-art models shows the results on the CIFAR-10 dataset. As evident, proposed model, with an FID of 20.90, significantly outperforms the methods listed in the table.

This confirms the high realism and diversity of the images generated by the proposed model. These results indicate that the integration of advanced techniques employed in the proposed approach—such as Exponential Moving Average, along with enhancements like DiffAugment and AdaptiveMix—has had a substantial positive impact on the quality of the final outputs.

VIII. Conclusion and Discussion

GANs are among the most advanced frameworks for generating high-quality synthetic data and have been applied in various computer vision domains. In this study, a GAN-based model was designed that, while leveraging the fundamental principles of existing architectures, purposefully incorporated certain optimization and regularization strategies from more advanced models—most notably AdaptiveMix—as well as selected techniques proposed in EIGGAN. It is worth noting that the full EIGGAN architecture was not employed; rather, only specific components—particularly its training enhancement strategies—were selectively and adaptively integrated. This deliberate selection aimed to improve training stability and enhance the quality of the generated outputs.

To evaluate the quality of the generated samples, three key metrics were used: Inception Score (IS) as an indicator of image realism and diversity, Fréchet Inception Distance (FID) as a measure of distributional similarity, and discriminator accuracy as an implicit indicator of balance between the generator and the discriminator. According to the results, the model achieved an IS of 6.80 ± 0.22 at epoch 875, which is competitive and, in many cases, superior to classical models such as WGAN, DCGAN, and BEGAN as reported in previous studies. Furthermore, the best recorded FID was 20.90 at epoch 920, demonstrating strong convergence of the model toward generating realistic images.

The analysis of discriminator accuracy trends throughout training (Figure 2) revealed that this value remained in the 50–60% range during most epochs, indicating the preservation of a dynamic balance between the two GAN components. Maintaining the discriminator's performance within this range prevented dominance by either network and enabled more effective generator training.

In addition, the use of techniques such as Exponential Moving Average (EMA) for stabilizing generator outputs, R1-based regularization, and spectral normalization in the discriminator played a key role in enhancing both the stability and the quality of the final results.

Ultimately, the findings of this research demonstrate that designing hybrid architectures and selectively applying targeted stabilization and augmentation methods can significantly improve the quality and stability of generated samples. These results pave the way for future research into optimizing GAN structures and their practical applications, particularly in areas such as image reconstruction, super-resolution, and synthetic data generation.

References

[1] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2014). Generative adversarial nets. In Advances in Neural

Information Processing Systems (Vol. 27). https://doi.org/10.48550/arXiv.1406.2661
[2] Arjovsky, M., Chintala, S., & Bottou, L. (2017). Wasserstein generative adversarial networks. Proceedings of the 34th International Conference on Machine Learning (PMLR 70), 214–223. https://doi.org/10.48550/arXiv.1701.0
[3] Liu, H., Zhang, W., Li, B., Wu, H., He, N., Huang, Y., … & Zheng, Y. (2023). Improving GAN training via feature space shrinkage. arXiv preprint, arXiv:2303.01559. https://doi.org/10.48550/arXiv.2303.01559
[4] Tian, C., Gao, H., Wang, P., & Zhang, B. (2024). An enhanced GAN for image generation. Computers, Materials & Continua, 80(7), 852097. https://doi.org/10.32604/cmc.2024.052097
[5] Zhang, H., Cisse, M., Dauphin, Y. N., & Lopez-Paz, D. (2017). mixup: Beyond empirical risk minimization. arXiv preprint, arXiv:1710.09412. https://doi.org/10.48550/arXiv.1710.09412
[6] Mescheder, L., Nowozin, S., & Geiger, A. (2018). Which training methods for GANs do actually converge? International Conference on Machine Learning, pp. 3470–3479. PMLR. https://doi.org/10.48550/arXiv.1801.04406
[7] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X. (2016). Improved techniques for training GANs. Advances in Neural Information Processing Systems (NeurIPS), 29, 2234–2242. https://doi.org/10.48550/arXiv.1606.03498
[8] Zhao, S., Liu, D., Lin, C., Zhu, J. Y., & Han, S. (2020). Differentiable augmentation for data-efficient GAN training. Advances in Neural Information Processing Systems, 33, 16251–16263. https://doi.org/10.48550/arXiv.2006.10738
[9] Karras, T., Laine, S., & Aila, T. (2019). A style-based generator architecture for generative adversarial networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4401–4410. https://doi.org/10.1109/CVPR.2019.00453
[10] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved training of Wasserstein GANs. Advances in Neural Information Processing Systems, 30, 5769–5779. https://doi.org/10.48550/arXiv.1704.00028
[11] Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint, arXiv:1502.03167. https://doi.org/10.48550/arXiv.1502.03167
[12] Miyato, T., Kataoka, T., Koyama, M., & Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. International Conference on Learning Representations (ICLR). https://doi.org/10.48550/arXiv.1802.05957
[13] Zhang, H., Goodfellow, I., Metaxas, D., & Odena, A. (2019). Self-attention generative adversarial networks. In International Conference on Machine Learning (pp. 7354-7363). PMLR. https://doi.org/10.48550/arXiv.1805.08318
[14] Krizhevsky, A. (2009). Learning multiple layers of features from tiny images (Technical Report). University of Toronto. Retrieved from https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.
[15] Kodali, N., Abernethy, J., Hays, J., & Kira, Z. (2017). On convergence and stability of GANs. arXiv preprint, arXiv:1705.07215. https://doi.org/10.48550/arXiv.1705.07215
[16] Arora, S., Ge, R., Liang, Y., Ma, T., & Zhang, Y. (2017). Generalization and equilibrium in generative adversarial nets (GANs). https://doi.org/10.48550/arXiv.1703.00573
[17] Dumoulin, V., Belghazi, I., Poole, B., Lamb, A., Arjovsky, M., Mastropietro, O., & Courville, A. (2016). Adversarially learned inference. arXiv preprint, arXiv:1606.00704. https://doi.org/10.48550/arXiv.1606.00704
[18] Berthelot, D., Schumm, T., & Metz, L. (2017). BEGAN: Boundary equilibrium generative adversarial networks. arXiv preprint, arXiv:1703.10717. https://doi.org/10.48550/arXiv.1703.10717
[19] Zhao, J., Mathieu, M., & LeCun, Y. (2017). Energy-based generative adversarial network. In International Conference on Learning Representations. https://doi.org/10.48550/arXiv.1609.03126
[20] Mao, X., Li, Q., Xie, H., Lau, R. Y. K., Wang, Z., & Smolley, S. P. (2017). Least squares generative adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2794–2802). https://doi.org/10.48550/arXiv.1611.04076
[21] Radford, A., Metz, L., & Chintala, S. (2015). Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint, arXiv:1511.06434. https://doi.org/10.48550/arXiv.1511.06434
[22] Xiangli, Y., Deng, Y., Dai, B., Loy, C. C., & Lin, D. (2020). Real or not real, that is the question. International Conference on Learning Representations. https://doi.org/10.48550/arXiv.2002.05512