Dual discriminator GAN with self-attention

Harsh Goel (251110034)

Kishan Kumar Mishra (251110044)

Singampalli Sri Varshith (251110074)

Thoram Venkata Madhu Sai Kiran (251110082)

Abstract

Our objective in this project is to develop a D2GAN model with self-attention, combining the mode-diversity benefits of dual discriminators with the global coherence provided by attention mechanism. We evaluate four models in a progressive manner: a baseline DCGAN, a standard D2GAN, and self-attention–augmented variants of both. While self-attention improves feature consistency in convolutional GANs, our experiments show that its benefits are limited on low-resolution datasets such as CIFAR-10. The standard D2GAN achieves the best overall performance, surpassing both DCGAN and its attention-based extensions. Attempts to combine D2GAN with self-attention do not yield further improvements and instead reduce performance, likely due to the interaction between attention modules, dataset resolution, and D2GAN's KL-based training dynamics. These findings demonstrate that self-attention is not universally advantageous and that D2GAN remains the most effective architecture for this setting.

Introduction

Generative Adversarial Networks (GANs) are a class of generative models that learn to synthesize realistic data by imitating an underlying distribution. A GAN consists of two neural networks trained adversarially: a **generator** $G(z)$, which maps a noise vector $z \sim p_z(z)$ to a synthetic output, and a **discriminator** $D(x)$, which predicts whether its input is real or generated.

The training objective follows the classical minimax formulation:

$$\min_G \max_D \; V(G,D) = \mathbb{E}_{x\sim p_{data}}[\log D(x)] + \mathbb{E}_{z\sim p_z}[\log(1 - D(G(z)))].$$

This leads to the discriminator loss

$$J_D = -\frac{1}{m}\sum_{i=1}^{m}\Big[\log D(x_i) + \log(1 - D(G(z_i)))\Big],$$

and the non-saturating generator loss

$$J_G = -\frac{1}{m}\sum_{i=1}^{m} \log D(G(z_i)).$$

Despite their success across image synthesis, art, and data augmentation, GANs remain difficult to train. A major difficulty is **mode collapse**, where the generator produces limited or repetitive samples instead of capturing the full diversity of the true data distribution.

Extensive research has explored strategies to address mode collapse and the broader challenge of GAN instability. These approaches generally fall into three categories: architectural modifications, alternative loss functions, and improved training dynamics. On the architectural side, models such as D2GAN employ dual discriminators to provide richer and more balanced feedback, helping the generator cover a broader set of modes. Another significant advancement is the integration of self-attention mechanisms, as demonstrated in SAGAN, which allows the model to capture long-range spatial dependencies. This capability greatly enhances feature diversity, sharpness, and global coherence, particularly for complex or high-resolution images.

In terms of loss functions, the Wasserstein GAN (WGAN) introduced the Earth Mover's (Wasserstein-1) distance as a more stable and meaningful optimization objective, yielding smoother gradients compared to the classical adversarial loss. Further stability improvements have been achieved through enhanced training dynamics and regularization techniques, most notably in WGAN-GP (Gradient Penalty) and WGAN-WP (Weight Penalty), which enforce the Lipschitz constraint necessary for reliable Wasserstein training. Collectively, these developments reflect the substantial effort devoted to overcoming mode collapse and achieving stable, high-quality generative modeling.

In this project, we aim to build a **self-attention-augmented D2GAN** that leverages the complementary strengths of dual discriminators and attention-based feature modeling.

We begin with the **DCGAN** architecture, which serves as the foundational baseline due to its stable convolutional design and widespread adoption in image generation tasks. Building on this, we explore the **D2GAN** framework, which introduces dual discriminators to improve mode coverage and reduce collapse. To examine the effect of long-range feature dependencies, we incorporate self-attention into both architectures, creating **DCGAN + Self-Attention** and **D2GAN + Self-Attention**. The final model, D2GAN with self-attention, integrates the strengths of convolutional GANs, dual-discriminator training, and attention-based feature modeling.

DCGAN: Baseline Architecture

Introduction

The **Deep Convolutional Generative Adversarial Network (DCGAN)**[1] is one of the most influential and stable architectures derived from the original GAN framework. It replaces fully connected layers with convolutional and transposed convolutional blocks, enabling the generator and discriminator to learn hierarchical spatial features more effectively. DCGAN serves as the foundational baseline in this project due to its strong performance on natural images, stable training behavior, and clear architectural conventions.

Architecture Overview

DCGAN consists of two deep convolutional networks trained adversarially:

- **Generator:** Maps a noise vector $z \sim p_z(z)$ to an image through a sequence of transposed convolutional layers, progressively increasing spatial resolution and shaping visual structure.
- **Discriminator:** Receives an image and processes it through strided convolutions to determine whether it is real or generated, learning discriminative hierarchical features in the process.

Design Principles

DCGAN follows several architectural guidelines that contribute to its stability and performance:

- **Convolutional structure:** Pooling layers are removed and replaced with strided convolutions in the discriminator and transposed convolutions in the generator, allowing the model to learn optimal spatial transformations.
- **Batch Normalization:** Applied in both networks (excluding the generator output and discriminator input layers) to stabilize training and prevent internal covariate shift.
- **Activation Functions:** The generator uses **ReLU** activations in all layers except the output, which uses **Tanh** to produce pixel values in the range $[-1, 1]$. The discriminator employs **LeakyReLU** to maintain gradient flow for negative activations.
- **Removal of Fully Connected Layers:** Enhances spatial learning by allowing deeper convolutional hierarchies and reducing unnecessary parameters.

These design constraints make DCGAN significantly more stable than early GAN implementations, producing coherent textures and structured images even with relatively simple datasets.

Transposed Convolution in the Generator

The generator begins with a low-dimensional latent vector that encodes abstract variations of the image space. This vector is projected into a compact 4×4 feature map and then expanded spatially through a series of transposed convolutional layers. Each layer increases resolution while adding semantic structure, progressing from coarse shapes to fine-grained visual details. ReLU activations introduce nonlinearity, and a final Tanh activation ensures that the generated image matches normalized dataset ranges.

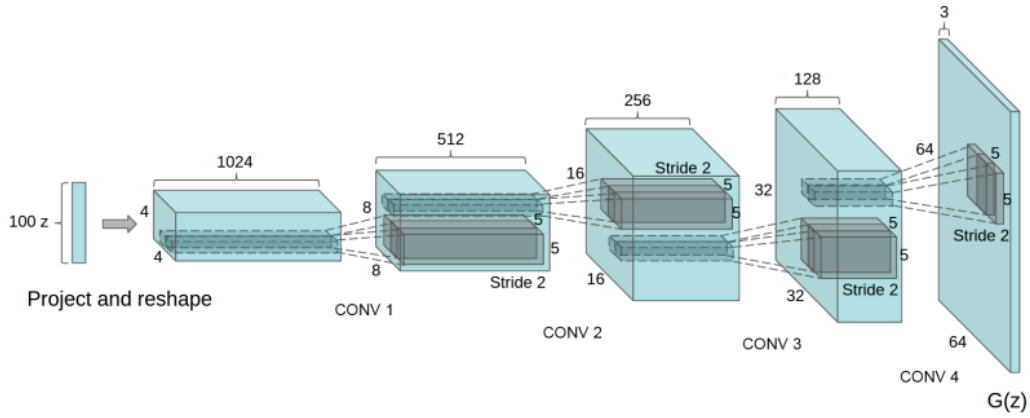

Figure 1: DCGAN generator architecture. A latent vector is projected into a small spatial representation, which is upsampled through transposed convolutions to synthesize a full-resolution image.

Dual Discriminator Generative Adversarial Network (D2GAN)

Introduction

The **Dual Discriminator Generative Adversarial Network (D2GAN)**[2] is a GAN variant specifically designed to alleviate **mode collapse**. Traditional GANs rely on a single discriminator, whose feedback can unintentionally guide the generator toward producing samples from only a few dominant modes. D2GAN addresses this limitation by introducing a **three-player adversarial game** consisting of a generator and two discriminators with complementary objectives. This dual-feedback mechanism enables the model to balance output diversity and realism, contributing to more stable training on complex multi-modal datasets.

Architecture

D2GAN consists of one generator and two discriminators with opposing roles:

- **Generator (G):** Maps noise $z \sim p_z(z)$ to synthetic samples $G(z)$.
- **Discriminator D_1:** Rewards real samples and penalizes generated ones, encouraging the generator to cover all modes of the data distribution.
- **Discriminator D_2:** Rewards generated samples and penalizes real ones, promoting high-quality, realistic outputs.

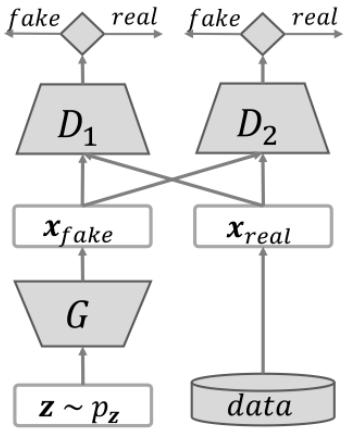

Figure 2: Overview of the D2GAN framework with one generator and two discriminators.

Together, the discriminators provide richer and more balanced gradients than standard GANs, enhancing both sample fidelity and diversity.

Objective Function

The joint minimax objective of D2GAN is:

$$\min_{G} \max_{D_1, D_2} J(G, D_1, D_2) = \alpha\, \mathbb{E}_{x \sim p_{data}}[\log D_1(x)] - \mathbb{E}_{z \sim p_z}[D_1(G(z))] - \mathbb{E}_{x \sim p_{data}}[D_2(x)] + \beta\, \mathbb{E}_{z \sim p_z}[\log D_2(G(z))],$$

where coefficients α and β control the relative influence of each discriminator. This formulation allows the model to simultaneously encourage mode coverage and sample realism.

Theoretical Insight

For a fixed generator, the optimal discriminators are:

$$D_1^*(x) = \frac{\alpha\, p_{data}(x)}{p_G(x)}, \qquad D_2^*(x) = \frac{\beta\, p_G(x)}{p_{data}(x)}.$$

Substituting these into the objective yields:

$$J(G) = \alpha\, D_{KL}(p_{data} \| p_G) + \beta\, D_{KL}(p_G \| p_{data}) + C.$$

Thus, D2GAN jointly minimizes the **forward** and **reverse** KL divergences.

- The forward KL term penalizes missing modes (improves diversity).
- The reverse KL term penalizes unrealistic samples (improves quality).

The optimum is achieved when $p_G = p_{data}$, where both discriminators converge to constant outputs.

Practical Advantages

D2GAN is shown to:

- reduce mode collapse on synthetic and real datasets,
- improve coverage of multi-modal distributions,
- provide richer training signals than single-discriminator GANs,
- scale more effectively to high-resolution or large-scale datasets.

These characteristics make D2GAN an effective foundation for architectures that seek to enhance both sample quality and mode diversity—motivating its integration with self-attention in this project.

Self-Attention in Generative Adversarial Networks

Convolutional neural networks (CNNs) effectively capture local spatial patterns but struggle to model long-range dependencies due to their limited receptive fields. As a result, traditional GANs may generate images with inconsistent textures or incoherent global structures. To address this limitation, the **Self-Attention GAN (SAGAN)**[3] incorporates self-attention modules into both the generator and discriminator, enabling each spatial location to attend to all others. This mechanism allows the model to integrate global contextual information, improving coherence across distant regions of an image.

Given convolutional feature maps $x \in \mathbb{R}^{C\times H\times W}$, SAGAN computes three learned transformations:

$$f(x) = W_f x, \qquad g(x) = W_g x, \qquad h(x) = W_h x,$$

representing query, key, and value features. These are used to form an attention map:

$$\beta_{ij} = \frac{\exp(f(x_i)^T g(x_j))}{\sum_{j=1}^{N} \exp(f(x_i)^T g(x_j))},$$

where β_{ij} captures the influence of position j on position i and $N = H \times W$. The self-attended output is computed as:

$$o_i = \sum_{j=1}^{N} \beta_{ij} h(x_j),$$

which aggregates contextual information across all spatial positions. A learnable scalar balances the contribution of attention relative to convolutional features.

By enabling long-range information flow, self-attention improves global consistency and structural coherence in generated images, especially for datasets with large or complex objects. SAGAN further stabilizes training through **spectral normalization**, which constrains the Lipschitz constant of network layers. The introduction of self-attention marked a major advancement in GAN architecture design and influenced subsequent high-performance models such as BigGAN.

Training Setup

This section outlines the architectural configurations and hyperparameters used to train the different GAN variants evaluated in this work. All models were trained on the CIFAR-10 dataset under consistent preprocessing, optimization settings, and training conditions to ensure fair comparison across architectures.

Table 1: DCGAN architecture and training hyperparameters

Network	**Layer**	**Output Size**
Generator	Input $z \sim \mathcal{N}(0, I)$	$100 \times 1 \times 1$
	ConvT $100 \to 512$, 4×4, s=1 + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 512$, 3×3 + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 256$, 4×4, s=2 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 256$, 3×3 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 128$, 4×4, s=2 + BN + ReLU	$128 \times 16 \times 16$
	ConvT $128 \to 128$, 3×3 + BN + ReLU	$128 \times 16 \times 16$
	ConvT $128 \to 3$, 4×4, s=2 + Tanh	$3 \times 32 \times 32$
Discriminator	Input image	$3 \times 32 \times 32$
	Conv $3 \to 128$, 4×4, s=2 + LReLU	$128 \times 16 \times 16$
	Conv $128 \to 256$, 4×4, s=2 + BN + LReLU	$256 \times 8 \times 8$
	Conv $256 \to 512$, 4×4, s=2 + BN + LReLU	$512 \times 4 \times 4$
	Conv $512 \to 1$, 4×4 + Sigmoid	$1 \times 1 \times 1$
	Output probability	scalar
Training Params	Dataset: CIFAR-10 (50k)	
	Image size: 32×32	
	Normalization: $[-1, 1]$	
	Latent dim: 100	
	Batch size: 64	
	Optimizer: Adam	
	LR: 2×10^{-4}	
	Betas: $(0.5, 0.999)$	
	Loss: BCE (non-saturating G)	
	Epochs: 100; Checkpoint: 5	

DCGAN was trained using the Adam optimizer with a learning rate of 2×10^{-4} and $\beta_1 = 0.5$, following the standard DCGAN training protocol. The generator used the non-saturating loss to avoid vanishing gradients, while the discriminator optimized the standard BCE objective. CIFAR-10 images were resized and normalized to $[-1, 1]$. Training was performed for 100 epochs with checkpointing every 5 epochs. A fixed latent vector was used throughout training for visual comparison, and loss curves were monitored to ensure stable convergence.

Table 2: D2GAN architecture and training hyperparameters

Network	**Layer Description**	**Output Size**
Generator	Input $z \sim \mathcal{N}(0, I)$	$100 \times 1 \times 1$
	ConvT $100 \to 512$, 4×4, s=1 + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 512$, 3×3, s=1 + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 256$, 4×4, s=2 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 256$, 3×3, s=1 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 128$, 4×4, s=2 + BN + ReLU	$128 \times 16 \times 16$
	ConvT $128 \to 128$, 3×3, s=1 + BN + ReLU	$128 \times 16 \times 16$
	ConvT $128 \to 3$, 4×4, s=2 + Tanh	$3 \times 32 \times 32$
D1, D2 (shared)	Input image	$3 \times 32 \times 32$
	Conv $3 \to 128$, 4×4, s=2 + LeakyReLU	$128 \times 16 \times 16$
	Conv $128 \to 256$, 4×4, s=2 + BN + LeakyReLU	$256 \times 8 \times 8$
	Conv $256 \to 512$, 4×4, s=2 + BN + LeakyReLU	$512 \times 4 \times 4$
	Conv $512 \to 1$, 4×4, s=1 + Softplus	$1 \times 1 \times 1$
	Final score (positive real)	scalar
Training Params	Dataset: CIFAR-10 (50k)	
	Normalization: $[-1, 1]$	
	Latent dim: 100	
	Batch size: 64	
	Optimizer: Adam	
	LR: 2×10^{-4}	
	Betas: (0.5, 0.999)	
	Divergence weights: $\alpha = 0.01$, $\beta = 0.20$	
	Loss: D2GAN (forward + reverse KL)	
	D_1: minimize $-\alpha \log D_1(x) + D_1(G(z))$	
	D_2: minimize $D_2(x) - \beta \log D_2(G(z))$	
	G: minimize $-D_1(G(z)) + \beta \log D_2(G(z))$	
	Epochs: 150	
	Checkpointing every 5 epochs	

D2GAN employs two discriminators that optimize forward and reverse KL divergence terms, controlled by hyperparameters α and β. Both discriminators were trained using Adam with a learning rate of 2×10^{-4} and $\beta_1 = 0.5$, using Softplus outputs as required by the D2GAN formulation. Each discriminator was updated before a single generator update. CIFAR-10 preprocessing matched the DCGAN setup, and checkpoints were saved every 5 epochs. A fixed noise vector was used to track generator evolution and for consistent quantitative evaluation.

Table 3: DCGAN with Self-Attention (DCGAN-SA) architecture and training hyperparameters

Network	**Layer Description**	**Output Size**
Generator	Input $z \sim \mathcal{N}(0, I)$	$100 \times 1 \times 1$
	SN ConvT $100 \to 512$, 4×4 + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 512$, 3×3 + BN + ReLU	$512 \times 4 \times 4$
	SN ConvT $512 \to 256$, 4×4, s=2 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 256$, 3×3 + BN + ReLU	$256 \times 8 \times 8$
	SN ConvT $256 \to 128$, 4×4, s=2 + BN + ReLU	$128 \times 16 \times 16$
	Self-Attention block	$128 \times 16 \times 16$
	SN ConvT $128 \to 3$, 4×4, s=2	$3 \times 32 \times 32$
	Tanh	$3 \times 32 \times 32$
Discriminator	SN Conv $3 \to 128$, 4×4, s=2 + LeakyReLU	$128 \times 16 \times 16$
	Self-Attention block	$128 \times 16 \times 16$
	SN Conv $128 \to 256$, 4×4, s=2 + BN + LeakyReLU	$256 \times 8 \times 8$
	SN Conv $256 \to 512$, 4×4, s=2 + BN + LeakyReLU	$512 \times 4 \times 4$
	SN Conv $512 \to 1$, 4×4	$1 \times 1 \times 1$
	Output logit (hinge loss)	scalar
Training Params	Loss: hinge loss	
	Spectral Norm: all conv layers	
	Latent dim: 100	
	Batch size: 64	
	Optimizer: Adam	
	LR: 2×10^{-4} (G), 4×10^{-4} (D)	
	Betas: $(0.5, 0.999)$	
	Dataset: CIFAR-10	
	Epochs: 150	

Self-attention layers were added to both the generator and discriminator to capture long-range spatial dependencies. All convolutional layers used spectral normalization to stabilize training with the hinge loss objective. The discriminator used a higher learning rate (4×10^{-4}) than the generator. Adam with $\beta_1 = 0.5$ was used for optimization. Fixed noise vectors were maintained for qualitative evaluation, and periodic checkpoints ensured reproducible training.

Table 4: D2GAN with Self-Attention (D2GAN-SA) architecture and training hyperparameters

Network	**Layer Description**	**Output Size**
Generator	Input $z \sim \mathcal{N}(0, I)$	$100 \times 1 \times 1$
	SN ConvT $100 \to 512$ + BN + ReLU	$512 \times 4 \times 4$
	ConvT $512 \to 512$ + BN + ReLU	$512 \times 4 \times 4$
	SN ConvT $512 \to 256$, s=2 + BN + ReLU	$256 \times 8 \times 8$
	ConvT $256 \to 256$ + BN + ReLU	$256 \times 8 \times 8$
	SN ConvT $256 \to 128$, s=2 + BN + ReLU	$128 \times 16 \times 16$
	Self-Attention block	$128 \times 16 \times 16$
	SN ConvT $128 \to 3$ + Tanh	$3 \times 32 \times 32$
Discriminators	SN Conv $3 \to 128$, 4×4, s=2 + LeakyReLU	$128 \times 16 \times 16$
	Self-Attention block	$128 \times 16 \times 16$
	SN Conv $128 \to 256$, 4×4, s=2 + BN + LeakyReLU	$256 \times 8 \times 8$
	SN Conv $256 \to 512$, 4×4, s=2 + BN + LeakyReLU	$512 \times 4 \times 4$
	SN Conv $512 \to 1$ + Softplus	$1 \times 1 \times 1$
	Output score (positive)	scalar
Training Params	Loss: D2GAN (forward + reverse KL)	
	D_1: minimize $-\alpha \log D_1(x) + D_1(G(z))$	
	D_2: minimize $D_2(x) - \beta \log D_2(G(z))$	
	G: minimize $-D_1(G(z)) + \beta \log D_2(G(z))$	
	$\alpha = 0.01$, $\beta = 0.20$	
	Self-Attention in G, D1, D2	
	Spectral Norm: all conv layers	
	Latent dim: 100	
	Batch size: 64	
	Optimizer: Adam (2×10^{-4})	
	Epochs: 100–150	

D2GAN-SA extends D2GAN by incorporating self-attention modules into both discriminators and the generator, together with spectral normalization across all convolutional layers for improved training stability. The KL-based objectives used the same weighting parameters ($\alpha = 0.01$, $\beta = 0.2$), and all networks were optimized using Adam with a 2×10^{-4} learning rate. Both discriminators were updated independently before each generator step. Checkpoints stored random-state information, fixed noise vectors, and dataloader seeds to ensure full reproducibility, and attention-related improvements were monitored through visual and quantitative evaluations.

Experiments and Results

This section provides a comprehensive experimental evaluation of the four GAN models introduced previously.

Experiments with DCGAN

Effect of Feature-Map Depth (DCGAN)

We tested DCGAN with discriminator feature-map depths $ndf \in \{64, 128, 256\}$, using the same value for the generator ($ngf = ndf$) to maintain architectural balance.

Table 5: Effect of feature-map depth on DCGAN performance.

ngf/ndf	Inception Score	Stability	Remarks
64	1.30	Unstable	Generator/Discriminator too weak
128	**6.17**	Stable	Best performance
256	6.00	Stable but slower	No improvement over 128

The results show that shallow feature maps (64) make DCGAN unstable and severely degrade image diversity. Increasing to 128 dramatically improves both fidelity and stability. Increasing further to 256 does not enhance results and increases compute cost. **Thus,** $ngf = ndf = 128$ **is selected for the final DCGAN model.**

Effect of Refining Layers in the Generator (DCGAN)

We compare DCGAN with and without additional 3×3 refining layers inserted between upsampling layers.

Table 6: Effect of refining layers on DCGAN image quality.

Configuration	Inception Score
Without refining layers	5.90
With refining layers	**6.17**

Adding refining layers improves texture sharpness and global coherence, raising the Inception Score from 5.90 to 6.17.

These results confirm that the final DCGAN configuration used in our work—**ngf = ndf = 128 with refining layers**—is the optimal architecture for stable, high-quality image generation.

Experiments with D2GAN

Tuning α and β

We performed a grid search over $\alpha, \beta \in \{0.01, 0.05, 0.10, 0.20\}$ and computed the Fréchet Inception Distance (FID) for each combination. The results are summarised in Fig. 3, with darker regions indicating lower FID and therefore better generative quality.

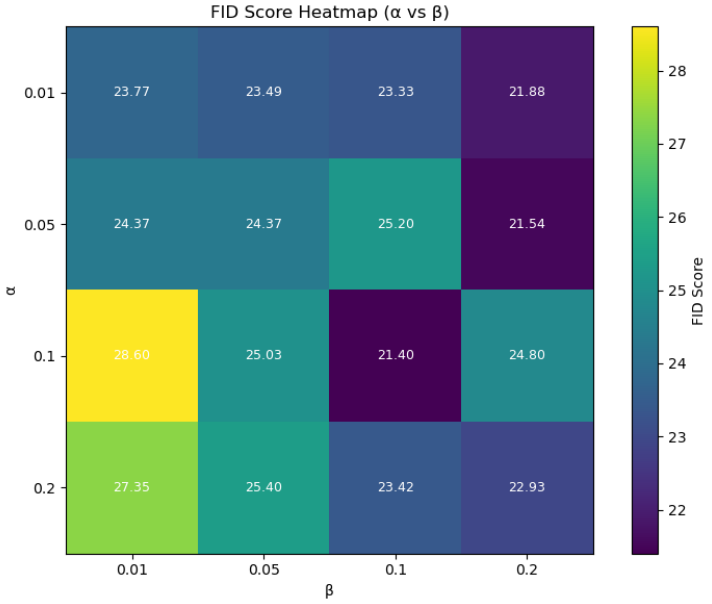

Figure 3: FID scores for different (α, β) combinations. Darker color = lower FID = better performance.

Effect of α: We observe that *lower* values of α consistently produce better FID scores than larger values. Since α controls the weight of the forward-KL component (enforcing mode coverage), large α encourages aggressive diversity, which destabilises training and leads to degraded image quality. In contrast, small α (e.g., $\alpha = 0.01$) provides a balanced level of diversity without overwhelming the generator. Overall, the model performs best in the regime of low α.

Effect of β: When α is fixed, increasing β generally improves performance. Because β scales the reverse-KL term (which rewards realistic, sharp samples), higher β pushes the generator towards higher-fidelity images. This trend is visible across almost all rows of the heatmap: columns with larger β values have lower FID compared to those with smaller β.

Best-performing region: The optimal setting in our sweep occurs at $(\alpha, \beta) = (0.10, 0.10)$, yielding a minimum FID of 21.40. However, the overall trend still favours *low* α and *moderately high* β. This indicates that encouraging realism (via larger β) is beneficial, while excessive pressure for diversity (via large α) is harmful.

Experiments with DCGAN-SA

Table 7: Experimental results for DCGAN-SA variants

ID	Model Variant	IS
1	Baseline DCGAN-SA: self-attention included, spectral norm enabled, refining layers enabled	5.77
2	Spectral normalization removed from the generator	5.40
3	Refining layers removed (no spectral norm in G)	5.30
4	Same as above but discriminator made slower (fewer updates)	5.30
5	Same as above but self-attention moved to 16×16 resolution	5.75
6	Same as above but trained with hinge loss	5.99
7	Same as above but generator uses the same learning rate as the discriminator	6.05
8	Same as above but discriminator made faster (more updates per generator step)	6.23
9	Same as above but two refining layers added	6.34
10	Same as (9) but an extra refining layer added before the self-attention block	5.74
11	Same as (9) but a refining layer added after the self-attention block	6.13
12	Same as (9) but discriminator made 4× faster and generator learning rate reduced to 0.0001	6.19
13	Same as (9) but batch normalization removed from the discriminator	5.10

Observations

- **Spectral normalization** in the generator contributes significantly to stability and performance. Removing it causes a drop from 5.77 to 5.40.
- **Attention placement** matters. Moving attention to 16×16 resolution provides a meaningful boost (5.30 → 5.75).
- The **discriminator–generator update ratio** matters. A faster discriminator improves performance (6.05 → 6.23), but extremely fast updates (4×) yield diminishing returns (6.19).
- **Hinge loss** performs better than BCE loss.
- Removing **batch normalization** from the discriminator severely degrades performance (IS = 5.10).
- **Refining layers** play a crucial role in sharp texture generation. Removing them reduces IS

($5.40 \rightarrow 5.30$), while adding two refining layers(model no. 9) results in the best performance (IS = 6.34).

Conclusion

The best-performing configuration is the model with:

- self-attention at 16×16,
- hinge loss,
- a moderately fast discriminator,
- two refining layers.

Experiments with D2GAN-SA

We combined the best-performing DCGAN-SA configuration with the strongest D2GAN variant to assess whether their complementary strengths could yield further improvements. Surprisingly, the hybrid model underperformed relative to the DCGAN-SA baseline, obtaining an Inception Score of 6.09, compared to the baseline score of 6.17.

A key reason for this decline is the nature of the CIFAR-10 dataset: at a resolution of 32×32, long-range spatial dependencies are limited, which reduces the usefulness of self-attention. As a result, adding attention primarily increases model capacity without providing substantial benefit. This aligns with our DCGAN-SA results, where improvements from attention and refinement layers were modest (from 6.17 to 6.34).

Another possible contributing factor is a mismatch between D2GAN's dual-discriminator KL-based training dynamics and the architectural or normalization choices used in DCGAN-SA. When combined, these components may introduce conflicting optimization signals, making it difficult for the hybrid model to effectively leverage the strengths of either approach.

Although larger datasets like ImageNet would better showcase the benefits of attention, training GANs at that scale was infeasible due to time and compute limitations. Self-attention and multi-discriminator models become significantly more expensive at higher resolutions, so experiments were restricted to CIFAR-10.

We now present a unified comparison of all models using both Inception Score (IS) and FID, summarised in the table below.

Table 8: Summary of final model performance on CIFAR-10 (FID ↓, IS ↑)

Model	**FID (↓)**	**IS (↑)**
DCGAN (baseline)	25.66	6.17
D2GAN	21.40	6.43
DCGAN-SA	23.62	6.34
D2GAN-SA	27.55	6.09

Qualitative Results from the Best Model

To complement the quantitative metrics, we now present sample images generated by our best-performing model, **D2GAN**. The labels give on images is the classification output of Inception network.

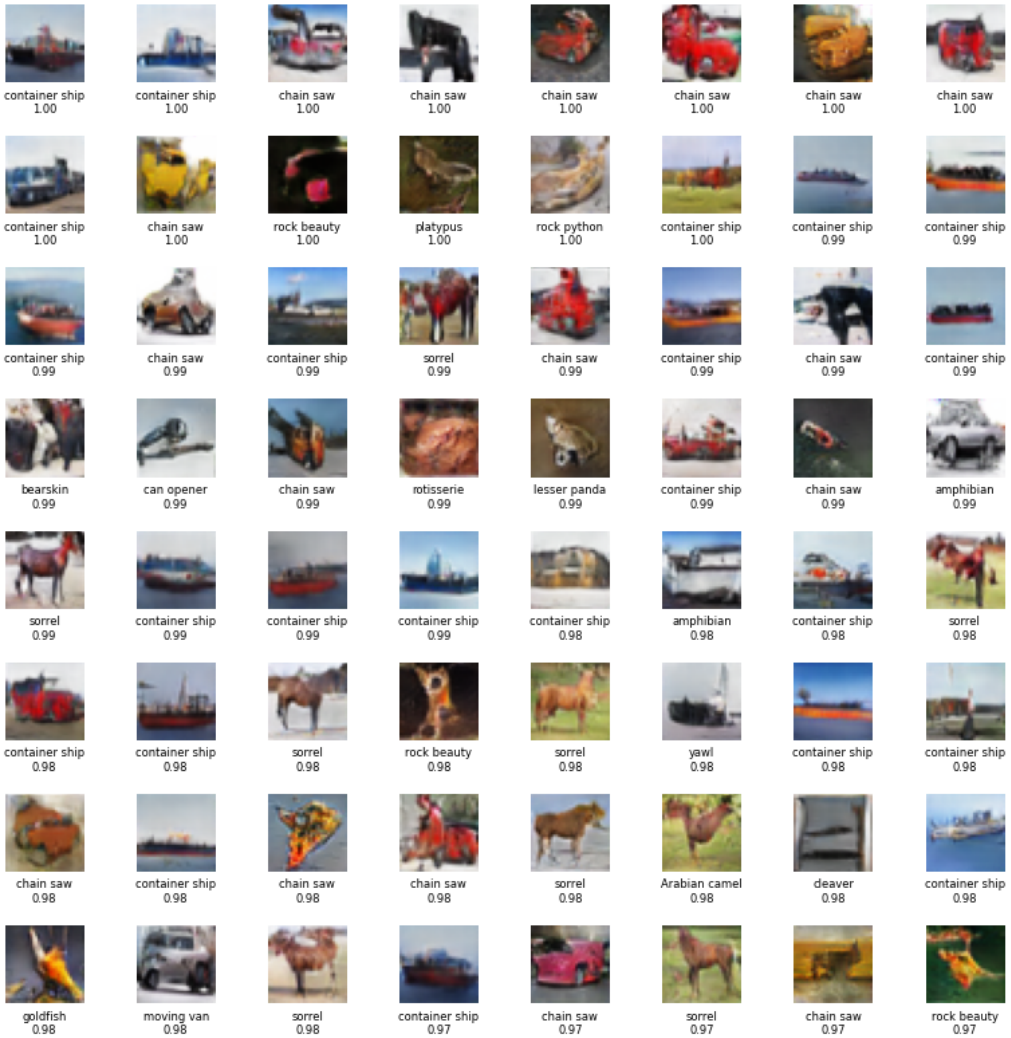

Figure 4: Samples generated by the best-performing model (D2GAN).

Summary

In this work, we explored a progression of GAN architectures beginning with the baseline DCGAN, extending to D2GAN, and finally incorporating self-attention mechanisms into both frameworks. Our experiments show that **D2GAN** achieves the best overall generative performance on CIFAR-10, obtaining the lowest FID (21.40). The dual-discriminator setup effectively improves mode coverage and sample realism compared to single-discriminator GANs.

Integrating self-attention into DCGAN provided modest improvements (FID: 23.62); however, applying the same modification to D2GAN led to a decline in performance (FID: 27.55). This behavior can be attributed to two main factors: (i) attention mechanisms offer limited benefit at the low spatial resolution of CIFAR-10, and (ii) the KL-driven training dynamics of D2GAN may conflict with attention-based feature modeling, resulting in unstable optimization. Additionally, our study underscores the critical role of spectral normalization, discriminator update frequency, and the strategic placement of refining layers in stabilizing training and enhancing generative quality.

Overall, the results suggest that self-attention is not particularly effective for low-resolution datasets like CIFAR-10, where long-range dependencies are limited. While attention improves feature coherence slightly in DCGAN variants, it does not translate into meaningful gains for D2GAN. In contrast, the standard D2GAN—without attention—remains the strongest model, offering the best balance between diversity and fidelity.

Future Work

A natural direction for future work is to extend these experiments to higher-resolution datasets such as ImageNet, CelebA-HQ, or LSUN. Self-attention and multi-discriminator architectures typically show their full potential at larger spatial scales, where long-range dependencies become more meaningful. Training on higher resolutions would allow a more reliable assessment of the benefits of attention and refinement layers, and may reveal improvements that are not apparent on CIFAR-10 due to its limited image size.

References

[1] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *4th International Conference on Learning Representations (ICLR)*, 2016. [Online]. Available: https://arxiv.org/abs/1511.06434

[2] T. D. Nguyen, T. Le, T.-T. Vu, and D. Phung, "Dual discriminator generative adversarial nets," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2017, pp. 2670–2680. [Online]. Available: https://arxiv.org/abs/1709.03831

[3] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *36th International Conference on Machine Learning (ICML)*, 2019, pp. 7354–7363. [Online]. Available: https://arxiv.org/abs/1805.08318