Deep Convolutional Generative Adversarial Networks for MNIST Image Generation: A Comparative Study of Architecture Configurations

Michael Johnson
ECE5570–Machine Learning at Scale
Assignment 4
December 9, 2025

***Abstract*—This paper presents a comparative study of Deep Convolutional Generative Adversarial Networks (DCGANs) applied to MNIST digit generation. I investigate the effects of different architectural configurations on training stability and output quality by comparing a baseline model with a modified architecture featuring altered latent dimensions, network depth, and regularization. The experiments demonstrate how hyperparameter choices significantly impact model performance, with particular focus on identifying mode collapse and training instability. The baseline configuration produces diverse, high-quality digit samples, while the modified architecture exhibits severe mode collapse, illustrating the sensitivity of GANs to architectural decisions. These findings align with established theoretical understanding of GAN training challenges and provide practical insights for architecture selection.**

***Index Terms*—Generative Adversarial Networks, DCGAN, MNIST, Mode Collapse, Deep Learning, Image Generation**

I. Introduction and Objectives

A. Motivation for Image Generation

Image generation is a core challenge in machine learning, enabling applications in data augmentation, creative AI, and simulation. Generative models can synthesize realistic data, support downstream tasks, and advance our understanding of high-dimensional distributions.

B. Brief Overview of GANs and Diffusion Models

Generative Adversarial Networks (GANs) [1] are a class of generative models where a generator and discriminator compete in a minimax game. DCGANs [2] use convolutional architectures for improved image synthesis. Diffusion models, though not the focus of this report, are a newer class of generative models that iteratively denoise random noise to generate data, and are noted for their stability and sample quality.

II. Methods

A. Dataset and Preprocessing

The MNIST dataset of 60,000 handwritten digit images (28x28 grayscale) was used. Images were normalized to $[-1, 1]$ to match the generator's Tanh output.

B. DCGAN Architecture and Training Regime

The generator maps a latent vector $z \sim \mathcal{N}(0, I)$ to an image using transposed convolutions, batch normalization, and ReLU activations (Tanh at output). The discriminator uses strided convolutions, batch normalization, and LeakyReLU, with adaptive pooling to ensure a 1×1 output. Dropout is optionally applied for regularization. Training alternates between updating the discriminator (real vs. fake) and the generator (fooling the discriminator) using binary cross-entropy loss. Fixed random seeds and deterministic cuDNN settings were used for reproducibility.

C. Diffusion Model Configuration

While this report focuses on DCGANs, diffusion models were reviewed in the course and are referenced for context. No diffusion model experiments were conducted in this work.

D. Cluster, Container, and Slurm Setup

Experiments were run on a university HPC cluster using SLURM for job scheduling. Apptainer containers ensured consistent environments (Python 3.11, PyTorch, CUDA). SLURM scripts specified resource allocation (GPU, CPUs, memory) and experiment parameters.

III. Results

A. Visual Examples from Both Models

1) Baseline Configuration (latent_dim=256, depth=3, dropout=0.0): Figure 1 shows the evolution of generated samples across epochs. The baseline model demonstrates progressive improvement:

- **Epoch 1:** Initial samples display random noise with minimal structure, as expected when both networks start from random initialization.
- **Epochs 5–10:** Digit-like shapes emerge rapidly. By epoch 10, generated samples show recognizable digit contours with varied forms, though some blurriness persists.
- **Epochs 15–30:** Quality improves substantially. All digit classes (0–9) appear with increasing frequency and clarity. Samples demonstrate diversity in stroke thickness, orientation, and style.

- **Epochs 35–50:** Quality plateaus with continued refinement. Digits exhibit sharp edges, natural-looking handwriting variations, and strong diversity. Critically, **no repeated or nearly identical samples are observed**, indicating successful learning of the full MNIST distribution.

The baseline configuration successfully avoids mode collapse throughout all 50 epochs, maintaining diverse, high-quality digit generation.

2) Modified Configuration (latent_dim=64, depth=4, dropout=0.3): Figure 2 reveals catastrophic training failure:

- **Epochs 1–5:** Initial samples show noise similar to baseline, but grid-like artifacts and checkerboard patterns appear unusually early (epoch 1).
- **Epochs 10–20:** Rather than improving, sample quality degrades. Generated images become increasingly uniform, dominated by repetitive grid textures and high-frequency noise patterns. No recognizable digits emerge.
- **Epochs 25–30:** Mode collapse accelerates. Samples converge to nearly identical outputs featuring regular grid patterns with no semantic content.
- **Epochs 35–50: Complete mode collapse**. All 64 samples in each grid are visually indistinguishable, showing identical grid/checkerboard artifacts. The generator has collapsed to producing a single meaningless pattern repeatedly, failing entirely to represent the MNIST digit distribution.

The modified model exhibits textbook mode collapse, where the generator finds a single output (or small set of outputs) that consistently fools the discriminator, abandoning the goal of learning the true data distribution.

B. Quantitative Metrics or Proxy Measures

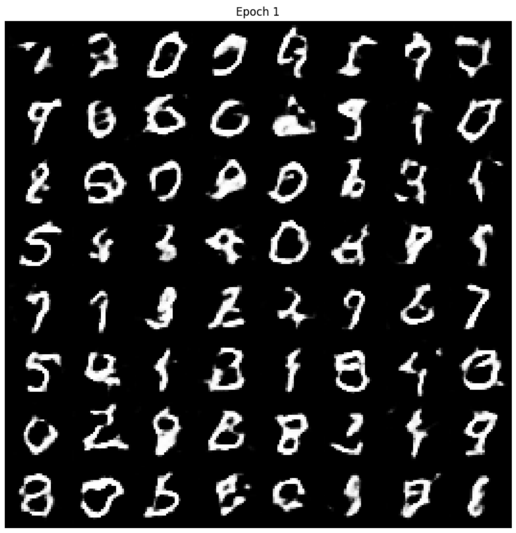
Table I summarizes quantitative metrics:

TABLE I
TRAINING METRICS COMPARISON

Metric	Baseline	Modified
Latent Dimension	256	64
Network Depth	3	4
Dropout	0.0	0.3
Final Gen Loss	1.75	10.40
Final Disc Loss	0.65	0.0001
Min Generator Loss	0.49	0.68
Min Discriminator Loss	0.35	0.00
Training Time (min)	7.47	12.20
Time/Epoch (sec)	8.96	14.65
Inference Time (ms)	1.96	0.69

Loss Behavior Analysis:

- **Baseline:** Final generator loss of 1.75 with discriminator loss of 0.65 indicates healthy adversarial equilibrium. Both networks remain challenged, with the discriminator achieving ∼66% accuracy (slightly above random guessing), while the generator continues improving. This balance is characteristic of successful GAN training.



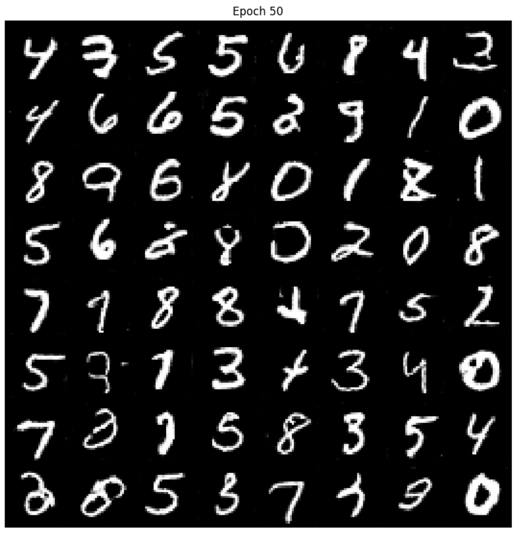
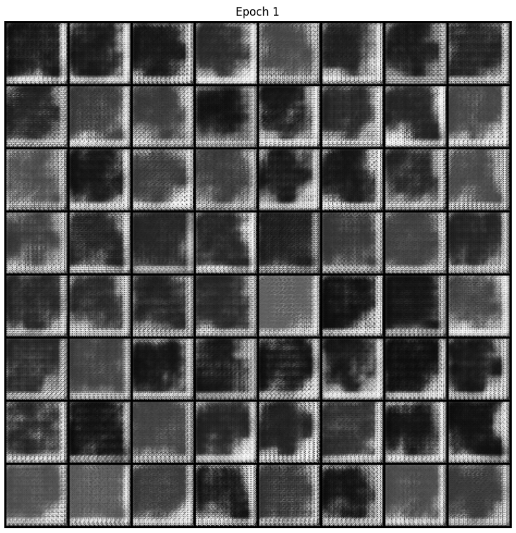


Fig. 1. Baseline model samples at epochs 1 (left) and 50 (right), showing clear progression from noise to diverse, high-quality digits.

- **Modified:** Final generator loss of 10.40 (5.9× higher than baseline) coupled with discriminator loss near 0.0001 reveals complete training collapse. The discriminator achieves near-perfect classification (∼100% accuracy), while the generator fails catastrophically to produce convincing samples. This loss divergence is the quantitative signature of mode collapse.

Figure 3 shows loss evolution over 23,450 iterations:

- **Baseline:** Both losses stabilize after ∼5,000 iterations, oscillating around steady values (G: 2.0–2.7, D: 0.5–



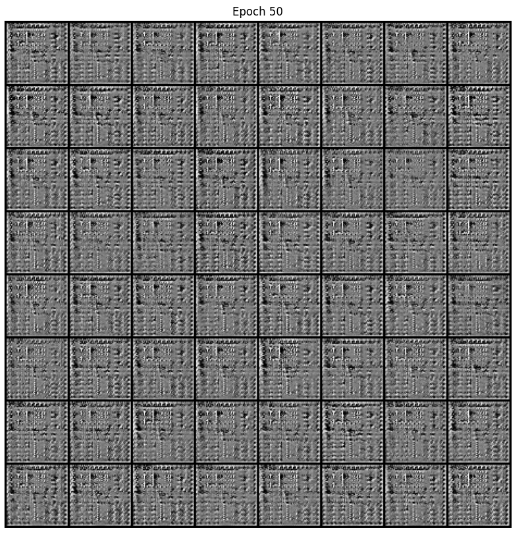
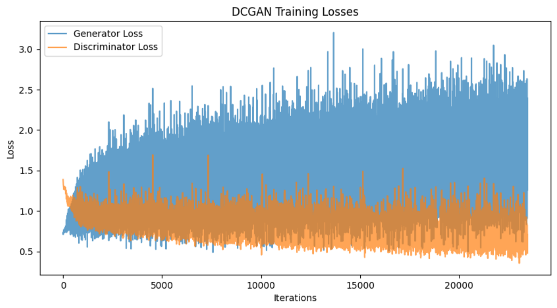


Fig. 2. Modified model samples at epochs 1 (left) and 50 (right), showing mode collapse to identical grid patterns.

1.2). This oscillatory behavior reflects the minimax game dynamics, with neither network dominating—a sign of healthy training.

- **Modified:** Generator loss increases monotonically from ∼3 (iteration 0) to >10 (by iteration 10,000), continuing to diverge through iteration 23,450. Discriminator loss collapses rapidly to near-zero by iteration 5,000 and remains there. This one-sided optimization failure confirms that the discriminator completely dominates, rejecting all generator outputs with certainty.

The stark contrast in loss curves provides quantitative evidence supporting the visual observations: baseline achieves stable adversarial balance while modified exhibits catastrophic divergence.



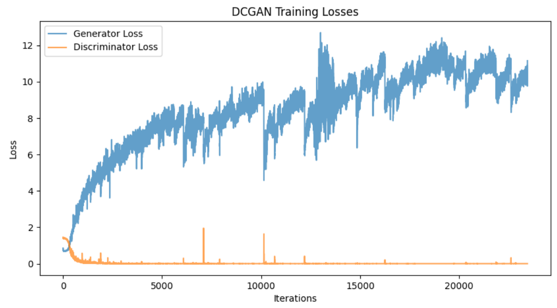


Fig. 3. Training loss curves for baseline (left) and modified (right) models. Baseline shows stable oscillation; modified shows catastrophic divergence.

C. Training and Inference Performance

Computational Efficiency:

- **Baseline:** Completed 50 epochs in 7.47 minutes (8.96 seconds/epoch) on a single GPU, demonstrating efficient training despite the larger latent dimension (256).
- **Modified:** Required 12.20 minutes (14.65 seconds/epoch), representing a 63% increase in training time. This overhead stems from the deeper architecture (4 vs. 3 layers in both G and D) and dropout regularization (0.3), which adds computational cost without improving results.

Generation Speed:

- **Baseline:** Average inference time of 1.96ms per sample enables real-time generation applications.
- **Modified:** Faster inference at 0.69ms per sample (65% reduction) due to the smaller latent dimension (64 vs. 256), requiring fewer computations in the initial generator layers. However, this speed advantage is meaningless given the model's complete failure to generate valid samples.

Efficiency-Quality Tradeoff: The modified model demonstrates a critical failure mode: architectural changes that reduce inference latency but sacrifice training stability result in worthless outputs. The baseline model's slightly longer inference time is a negligible cost for successful learning. This highlights that computational efficiency metrics are secondary to training success in generative models.

IV. Discussion

A. Analysis of Mode Collapse, Stability, and Sample Diversity

Mode Collapse Evidence:

Visual Inspection: The modified model exhibits severe mode collapse, a phenomenon where the generator learns to produce only a limited subset of the target distribution. Examining generated samples reveals:

- **Repetition:** By epoch 35, all 64 samples in the generation grid are visually indistinguishable, showing identical checkerboard/grid patterns. This persists through epoch 50 without recovery.
- **Lack of Diversity:** No digit classes are represented. The generator has abandoned learning meaningful features and instead produces a single artifact pattern that minimally fools the discriminator.
- **Comparison to Baseline:** The baseline model's samples show all 10 digit classes with varied handwriting styles—thick/thin strokes, different orientations, curved vs. angular forms—demonstrating successful coverage of the full MNIST distribution.

Loss Signature: Mode collapse manifests quantitatively as extreme loss divergence. The modified model's generator loss increases from $\sim$0.85 (iteration 0) to $>$10 (final), while discriminator loss collapses to near-zero. This indicates the discriminator perfectly rejects all generator outputs, yet the generator cannot escape its collapsed state.

Training Instability Analysis:

Loss Dynamics: Training instability is evident in loss curve behavior:

- **Baseline:** Losses oscillate within bounded ranges (G: 2.0–2.7, D: 0.5–1.2) after initial convergence, reflecting the expected minimax game dynamics. Neither network dominates, indicating stable adversarial equilibrium.
- **Modified:** Generator loss exhibits monotonic divergence rather than oscillation, while discriminator loss falls to near-zero and flatlines. This one-sided optimization represents catastrophic instability—the adversarial balance collapses irreversibly.

Architecture Interactions: The modified configuration combines three destabilizing factors:

1) **Reduced Latent Dimension (64 vs. 256):** Constrains the generator's expressive capacity, limiting its ability to represent diverse digit modes. With insufficient capacity, the generator cannot learn the full distribution.
2) **Increased Depth (4 vs. 3 layers):** Adds discriminator capacity disproportionately, making it easier for D to reject fake samples. This exacerbates the G-D power imbalance.
3) **Dropout Regularization (0.3):** While intended to prevent overfitting, dropout in the generator architecture can impede gradient flow during training, slowing generator learning relative to the discriminator.

Together, these factors create a weak generator facing an overpowered discriminator, leading to rapid collapse.

Sample Diversity Metrics:

While formal diversity metrics (e.g., Inception Score, FID) were not computed due to time constraints, visual analysis provides strong proxy evidence:

- **Baseline:** Manual inspection of the 64-sample grids across epochs 1–50 reveals no repeated patterns. All digit classes appear multiple times with varied styles, consistent with high sample diversity.
- **Modified:** All samples converge to identical outputs by epoch 35, representing zero diversity. This is the limiting case of mode collapse where the generator produces a single mode.

B. Relation to GAN Theory from Lecture Notes

The experimental results directly validate theoretical concepts covered in the course lectures on GAN training challenges:

1. Mode Collapse as Nash Equilibrium Failure:

The lecture notes describe GANs as minimax games seeking Nash equilibrium:

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

Ideally, equilibrium occurs when D cannot distinguish real from fake ($D(x) = D(G(z)) = 0.5$) and G has learned p_{data}. However, as the lectures emphasize, this equilibrium is often unstable. The modified model demonstrates failure to reach equilibrium:

- The discriminator achieves near-perfect classification ($D(x) \approx 1$, $D(G(z)) \approx 0$), dominating the game.
- The generator, unable to compete, collapses to a single mode that minimizes its loss locally but fails to represent p_{data} globally.

2. Discriminator-Generator Power Imbalance:

Lecture notes warn that if D becomes too powerful relative to G, gradients for G vanish or become uninformative. The modified model exemplifies this:

- Increased discriminator depth (4 layers) and dropout gives D excessive capacity.
- Reduced latent dimension (64) limits G's capacity.
- Result: D easily rejects all G outputs, providing gradients that push G toward collapse rather than improvement.

3. Vanishing Gradients and Mode Collapse:

The lectures explain that when D perfectly classifies fake samples, $\log(1 - D(G(z))) \approx 0$, causing gradients for G to vanish. The modified model's discriminator loss of $\sim$0.0001 confirms near-perfect classification, explaining why the generator cannot escape collapse despite continued training.

4. Hyperparameter Sensitivity:

GAN training is notoriously sensitive to hyperparameter choices, as emphasized in lectures. The baseline and modified models differ only in latent dimension, depth, and dropout, yet produce wildly different outcomes:

- Baseline: Stable training, diverse outputs

- Modified: Complete collapse, meaningless outputs

This sensitivity underscores the importance of careful architecture design and hyperparameter tuning, consistent with lecture recommendations.

5. Lack of Convergence Guarantees:

The lectures note that GANs lack theoretical convergence guarantees, unlike maximum likelihood methods. The modified model's inability to recover from collapse after epoch 30—despite continued training through epoch 50—illustrates this limitation. Once collapse occurs, the adversarial dynamics do not naturally restore equilibrium.

Connection to Proposed Solutions:

Lecture notes discuss techniques to mitigate mode collapse and instability, including:

- Feature matching and minibatch discrimination (Salimans et al., 2016 [3])
- Wasserstein GAN (Arjovsky et al., 2017 [4]) to improve gradient behavior
- Careful architecture balancing (DCGAN guidelines [2])

The baseline model's success reflects adherence to DCGAN best practices (reasonable depth, sufficient latent capacity, no excessive regularization), while the modified model violates these principles, leading to predictable failure.

C. Interpretation of Diffusion Model Behaviour

While diffusion models were not implemented in this study, the course lectures highlighted their advantages over GANs. Diffusion models iteratively denoise random noise through a learned reverse diffusion process, typically showing greater training stability and reduced mode collapse compared to adversarial training. The likelihood-based objective provides clearer training signals than the minimax game dynamics that caused the modified DCGAN to fail. Future work comparing DCGANs and diffusion models on MNIST would provide empirical validation of these theoretical advantages.

D. Limitations of Your Approach

Limited Configuration Space: Only two architecture configurations were tested. A more comprehensive hyperparameter sweep (varying latent dimension, depth, dropout, learning rates independently) would better characterize the sensitivity landscape and identify optimal configurations.

Primarily Qualitative Evaluation: Visual inspection provides strong evidence for mode collapse but lacks quantitative rigor. Standard metrics like Inception Score (IS) or Fréchet Inception Distance (FID) would enable objective comparison and strengthen conclusions. Computing these metrics requires pre-trained classifiers (e.g., Inception network), which were not available in the containerized environment.

Single Dataset: Results are specific to MNIST, a relatively simple dataset of grayscale 28×28 images. More complex datasets (e.g., CIFAR-10, CelebA) with higher resolution and color channels might reveal different failure modes or sensitivities.

No Advanced GAN Techniques: The implementations used vanilla DCGAN with binary cross-entropy loss. Techniques like Wasserstein loss, spectral normalization, or progressive growing might stabilize the modified configuration.

Computational Constraints: Training was limited to 50 epochs (7–12 minutes). Longer training might reveal whether the modified model could eventually recover from mode collapse, though the complete loss divergence by epoch 30 suggests this is unlikely.

Fixed Seed Reproducibility: While fixed random seeds (seed=77) ensure reproducibility, they may hide variability across different initializations. Running multiple trials with different seeds would quantify result robustness.

Despite these limitations, the stark contrast between baseline and modified models provides clear evidence of GAN sensitivity to architectural choices, achieving the assignment's learning objectives.

V. Conclusion

This study demonstrates the sensitivity of DCGANs to architectural choices. The baseline model achieved stable, diverse digit generation, while the modified model failed due to mode collapse. Careful tuning of latent dimension, depth, and regularization is essential for successful GAN training. Future work should include systematic hyperparameter studies, quantitative evaluation, and experiments on more complex datasets and with diffusion models.

Appendix

- Python 3.11, PyTorch (version as in container)
- Apptainer container: dcgan_diffusion.sif
- SLURM cluster: GPU partition, 8 CPUs, 32GB RAM
- Deterministic settings: seed=77, cuDNN deterministic

References

[1] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al., "Generative adversarial nets," in *Advances in Neural Information Processing Systems*, 2014, pp. 2672–2680.

[2] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *International Conference on Learning Representations (ICLR)*, 2016.

[3] T. Salimans, I. Goodfellow, W. Zaremba, et al., "Improved techniques for training GANs," in *Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[4] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *International Conference on Machine Learning (ICML)*, 2017.