Contents

1 Comparative Analysis: Convolutional VAE vs DCGAN on CIFAR-10

Authors: Adil Akhmetov, Perizat Yessenova
Course: PhD Generative Models
Date: November 2025

1.1 Abstract

This report presents a comparative analysis of two generative models—Convolutional Variational Autoencoder (VAE) and Deep Convolutional GAN (DCGAN)—trained on CIFAR-10. We evaluate both models on reconstruction quality, representation learning, and sample generation under limited compute constraints (M1 Mac). Our experiments reveal fundamental trade-offs: GANs achieve superior visual quality while VAEs provide stable training and better representations (37.69% linear probe accuracy vs. 10% random). We identify two critical failure modes for each model and propose principled mitigations based on recent literature.

1.2 1. Introduction and Methodology

1.2.1 1.1 Models and Objectives

We compare two prominent generative approaches: Variational Autoencoders (probabilistic latent variable models) and Generative Adversarial Networks (adversarial training frameworks). Our objectives are to: (1) implement both models on CIFAR-10, (2) compare training dynamics, (3) evaluate representation vs. sample quality trade-offs, and (4) identify and mitigate failure modes.

1.2.2 1.2 Experimental Setup

Dataset: CIFAR-10 with 10k training subset (32×32 RGB, normalized to [-1,1]) and 10k test images.

Architectures: - **VAE**: Encoder (Conv32→64→128) → latent z → Decoder (ConvT64→32→3). Loss: L = MSE(x,x̂) + · KL(q||p) with =1.0 - **DCGAN**: Generator z [1] →[4×4×128]→32×32×3 and Discriminator with standard adversarial loss

Training: VAE (20 epochs, Adam lr=1e-3), DCGAN (40 epochs, Adam lr=2e-4), batch size 64, seed 42. Hardware: M1 Mac with MPS acceleration. Training times: VAE ~22 min, GAN ~45 min.

Evaluation: VAE metrics (ELBO, linear probe on frozen latents), DCGAN metrics (loss curves, visual quality), qualitative comparison.

1.3 2. Results and Analysis

1.3.1 2.1 Quantitative Results

VAE Performance: - Final ELBO: **200.82**, Reconstruction Loss: **139.56**, KL Divergence: **61.25** - Linear Probe Accuracy: **37.69%** (test) vs. 10% random baseline - Training: Smooth monotonic convergence, no posterior collapse (healthy KL>60)

DCGAN Performance: - Final Generator Loss: **3.866**, Discriminator Loss: **0.159** - Loss ratio 24:1 indicates stable adversarial equilibrium - Training: Oscillating losses (expected), balanced D/G after ~30 epochs

Key Finding: The KL divergence of 61.25 demonstrates active latent usage. Linear probe accuracy of 37.69% (3.7× random) shows the VAE learned semantically meaningful representations without supervision. DCGAN achieved stable equilibrium without mode collapse, evidenced by diverse generated samples.

1.3.2 2.2 Qualitative Analysis

VAE Reconstructions (Figure 1): Top row shows original CIFAR-10 images; bottom row shows reconstructions. Observations: (1) global structure and colors preserved, (2) notable blurriness in fine details (expected with MSE loss), (3) object identity maintained. **Latent Interpolations** (Figure 2): Smooth transitions between images demonstrate continuous, structured latent space—crucial for representation learning applications.

DCGAN Samples (Figure 3): Generated samples show sharp textures and recognizable objects. Good diversity across the batch with no obvious mode collapse. Visual quality superior to VAE but some artifacts remain at 40 epochs.

Training Curves (Figure 4-5): VAE losses decrease smoothly with stable KL term. GAN losses oscillate (characteristic of adversarial training) but maintain equilibrium.

1.3.3 2.3 Why Each Model Behaved As Observed

VAE: Blurry reconstructions result from MSE loss minimizing pixel-wise error—averaging over ambiguous outputs rather than committing to specific details. The Gaussian prior (KL regularization) creates smooth latent spaces enabling interpolation but limits detail capacity. Good linear

separability emerges because the encoder must compress information effectively for accurate reconstruction while regularization prevents overfitting.

DCGAN: Sharp samples arise from adversarial loss optimizing perceptual realism, not pixel accuracy—the discriminator penalizes unrealistic features including blur. Oscillating losses reflect the minimax game where networks continuously adapt to each other. This is expected behavior, not failure. The longer training time (40 vs. 20 epochs) reflects the need for both networks to co-evolve.

1.4 3. Failure Modes and Mitigations

1.4.1 3.1 VAE Failure Modes

Failure Mode 1: Posterior Collapse

Description: Model ignores latent code, decoding from prior alone. Symptoms: KL 0, averaged outputs, no latent influence.

Why it happens: KL penalty discourages encoder from using z. If decoder is powerful enough to generate from prior alone, model takes this "easy route."

Proposed mitigation: **Cyclical -annealing** [Fu et al., 2019]. Gradually increase over multiple cycles ((t) = min(1.0, 2 · (t mod cycle)/(cycle))). This allows reconstruction to dominate initially, forcing the model to use z before heavy regularization.

Justification: The model learns meaningful latent representations before KL penalty restricts them. Multiple cycles provide "fresh starts." Expected improvement: KL>10, improved linear probe accuracy by 5-10%.

Note: Our experiment (KL=61.25) shows no collapse, validating our =1.0 choice.

Failure Mode 2: Blurry Reconstructions

Description: Low MSE but poor perceptual quality. Textures and edges averaged.

Why it happens: MSE loss is minimized by averaging likely outputs. For ambiguous regions, blurring is safer than committing to potentially incorrect details.

Proposed mitigation: **Perceptual loss with VGG features** [Johnson et al., 2016]. Augment loss with L_perceptual = MSE(VGG(x), VGG($\hat{x}$)) where VGG extracts conv4_3 features. Combined loss: L = L_MSE + _p · L_perceptual + · L_KL with _p 0.1.

Justification: VGG features capture high-level semantic and textural content. Matching features encourages perceptually similar reconstructions while maintaining structure. Widely validated in super-resolution and style transfer.

Expected improvement: Sharper textures, better high-frequency detail preservation, slight increase in MSE (acceptable trade-off).

1.4.2 3.2 DCGAN Failure Modes

Failure Mode 1: Mode Collapse

Description: Generator produces limited variety, failing to cover data distribution. Symptoms: similar samples, missing classes, low diversity.

Why it happens: Generator finds "easy wins" that consistently fool discriminator. Rather than exploring full distribution, it exploits these modes. Discriminator evaluates samples individually, missing diversity issues.

Proposed mitigation: **Minibatch discrimination** [Salimans et al., 2016]. Extend discriminator to see batch statistics via minibatch layer computing L1 distances between sample features. This allows D to detect when G produces similar samples within a batch.

Justification: Discriminator can now penalize lack of diversity. Generator must produce varied samples to avoid detection. Adds minimal computation (~100 features). Expected improvement: increased diversity, 20-30% better coverage.

Note: Our experiment showed good diversity, indicating no severe collapse.

Failure Mode 2: Training Instability

Description: Oscillating losses without convergence, gradient explosion/vanishing. Symptoms: erratic curves, poor sample quality, D accuracy→100% or 50%.

Why it happens: Adversarial game lacks inherent stability. If D becomes too strong, vanishing gradients to G. If too weak, no learning signal. Equilibrium is fragile.

Proposed mitigation: **Spectral normalization + TTUR** [Miyato et al., 2018; Heusel et al., 2017]. Apply spectral normalization to D layers (constrains Lipschitz constant, prevents gradient issues). Use two-timescale learning rates: lr_D = 4×lr_G (e.g., lr_G=1e-4, lr_D=4e-4).

Justification: Spectral norm bounds discriminator's Lipschitz constant, ensuring smooth gradients. TTUR allows D to stay ahead without overwhelming G, maintaining productive learning signal. Expected improvement: smooth training curves, consistent convergence.

1.5 4. Comparative Discussion

1.5.1 4.1 Trade-offs

Metric	VAE	DCGAN	Analysis
Sample Sharpness			GAN's adversarial loss targets perceptual quality
Training Stability			VAE's ELBO provides stable objective
Representation Quality	37.69%	N/A	VAE latents semantically meaningful
Training Time	22 min	45 min	GAN requires longer co-evolution
Convergence	Monotonic	Oscillating	Reflects optimization structure

Fundamental insight: The trade-off between stability and quality is not an engineering problem

but reflects fundamental differences in objectives. VAEs optimize evidence lower bound (principled probabilistic framework), while GANs optimize adversarial minimax game (better perceptual quality but unstable equilibrium).

1.5.2 4.2 Use Case Recommendations

Use VAE for: (1) Representation learning and downstream tasks (classification, clustering), (2) applications requiring stable, predictable training, (3) interpretable latent spaces, (4) likelihood estimation or uncertainty quantification, (5) limited compute budgets.

Use DCGAN for: (1) High-quality sample generation, (2) visual content creation, (3) applications prioritizing perceptual quality over training ease, (4) when expertise available for careful tuning.

1.6 5. Conclusion

This study demonstrates fundamental trade-offs in generative modeling through controlled comparison on CIFAR-10. Key findings:

1. **Training Dynamics**: VAEs converge smoothly via ELBO maximization (22 min, monotonic losses); GANs require careful adversarial balancing (45 min, oscillating losses).
2. **Quality Trade-offs**: GANs achieve superior perceptual quality with sharp samples; VAEs produce blurrier outputs but more reliable, diverse generation with structured latent spaces.
3. **Representation Learning**: VAE's 37.69% linear probe accuracy (vs. 10% random) demonstrates semantically meaningful unsupervised representations suitable for downstream tasks—a key advantage not available in GANs.
4. **Failure Modes**: Both models have characteristic failure modes (posterior collapse for VAE, mode collapse for GAN) addressable through principled modifications grounded in recent literature, not ad-hoc engineering.
5. **Practical Implications**: M1 Mac hardware proves sufficient for generative model research with appropriate optimizations (MPS acceleration, batch tuning, subset sampling), democratizing access to this field.

Future directions include hybrid architectures (VAE-GAN combining advantages), advanced objectives (WGAN-GP for stability, -VAE for disentanglement), and improved metrics (precision/recall decomposition beyond FID).

The choice between VAE and GAN depends on application requirements. Our experiments provide empirical guidance: representation learning and stability favor VAEs; high-fidelity generation favors GANs. Understanding these trade-offs enables informed model selection and appropriate mitigation strategies.

1.7 References

1. Kingma, D. P., & Welling, M. (2014). Auto-encoding variational bayes. *ICLR*.
2. Radford, A., Metz, L., & Chintala, S. (2016). Unsupervised representation learning with DCGAN. *ICLR*.

3. Heusel, M., et al. (2017). GANs trained by a two time-scale update rule. *NeurIPS*.
4. Higgins, I., et al. (2017). -VAE: Learning basic visual concepts. *ICLR*.
5. Miyato, T., et al. (2018). Spectral normalization for GANs. *ICLR*.
6. Salimans, T., et al. (2016). Improved techniques for training GANs. *NeurIPS*.
7. Fu, H., et al. (2019). Cyclical annealing schedule for KL vanishing. *NAACL*.
8. Johnson, J., et al. (2016). Perceptual losses for style transfer. *ECCV*.

Reproducibility: All experiments use seed 42, deterministic backends. Training config: VAE (latent_dim=64, =1.0, lr=1e-3, batch=64, epochs=20), GAN (z_dim=100, lr=2e-4, batch=64, epochs=40). Complete code, results, and checkpoints available. Single command `./reproduce.sh` runs full pipeline.

Word count: ~2,100 | Page count: ~4 pages