Short Notes on GAN Training Stability

Subhasish Bandyopadhyay

August 9, 2025

1 Mode Collapse

Let p_{data} denote the true data distribution on some space $\mathscr{X}$, and let p_g denote the distribution induced by a generator $G_\theta(z)$ with $z \sim p(z)$ (e.g., $z \sim \mathcal{N}(0,I)$). Mode collapse means that p_g concentrates on a small subset of the modes (high-density regions) of p_{data}—possibly a single mode—while ignoring others. Samples look sharp and plausible, but they lack diversity: the generator keeps producing variations of the same few patterns.

1.1 Reasons for Mode Collapse

Three interlocking reasons:

1. The generator only learns from its own samples. All generator updates are expectations over $x \sim p_g$. If the generator never visits a region where the real data lives, that region contributes zero gradient to the generator. No matter how wrong the model is there, the generator wont feel it.

2. Support mismatch causes saturated discriminators and vanishing gradients. Early in training, p_g typically lies on a thin manifold that barely overlaps the data manifold. The optimal discriminator saturates near 0 on generated points and near 1 on real points; gradients to the generator become small, especially under the original minimax loss.

3. The adversarial game is non-convex/non-concave and easy to destabilize. Updating a powerful discriminator too aggressively can push the generator toward safe pockets where it consistently fools the discriminator, reinforcing collapse. Capacity limits, regularization choices, and optimization schedules exacerbate the effect.

Mode collapse is not just a bug; it is an emergent property of the objective and sampling scheme. To see why, we need to look at the precise losses, and then contrast them with forward and reverse KL.

2 The Basic GAN Objective and What the Generator Optimizes

The original (minimax) GAN sets up the game

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

If you fix G and solve for the optimal discriminator,

$$D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)},$$

then plugging D^* back yields the value function

$$V(G,D^*) = -\log 4 + 2 \cdot \text{JS}(p_{\text{data}} \| p_g),$$

where JS is JensenShannon divergence:

$$\mathrm{JS}(P\|Q) = \frac{1}{2}\mathrm{KL}\left(P\middle\|\frac{P+Q}{2}\right) + \frac{1}{2}\mathrm{KL}\left(Q\middle\|\frac{P+Q}{2}\right).$$

Two properties of JS matter here:

- Bounded and flat when supports are disjoint. If P and Q have disjoint supports, $\mathrm{JS}(P\|Q) = \log 2$. Around that regime, gradients provide weak guidance because the discriminator saturates: D^* is essentially 0 on fake samples and 1 on real samples.
- Symmetry masks coverage incentives. JS treats missing and extra mass symmetrically, unlike forward KL (which hates missing mass) or reverse KL (which hates extra mass where the target is tiny). That symmetry, plus the generators sampling, reduces direct pressure to cover all modes.

Practitioners rarely use the strict minimax generator loss (which saturates); instead they use the non-saturating generator loss:

$$L_G^{\mathrm{NS}}(\theta) = -\mathbb{E}_{z\sim p(z)}[\log D(G_\theta(z))].$$

Substituting the optimal discriminator:

$$L_G^{\mathrm{NS}}(\theta) = \mathbb{E}_{x\sim p_g}\left[\log\left(1 + \frac{p_g(x)}{p_{\mathrm{data}}(x)}\right)\right].$$

This form is revealing:

- If $p_g(x)$ is too large where $p_{\mathrm{data}}(x)$ is small, the ratio $p_g(x)/p_{\mathrm{data}}(x)$ is large, the log-term is large, and the loss penalizes that behaviorgood.
- But if there is a true data mode where $p_g(x) = 0$, then that region contributes nothing to the expectation (because the expectation is under p_g). There is no direct penalty for not visiting that modebad for coverage.

This is the heart of mode collapse: the loss is blind to regions the generator never samples, so there is no gradient signal to pull probability mass there.

3 Forward vs. Reverse KL: Equations, Properties, and Why They Behave Differently

Let p be the target distribution and q be the model distribution.

3.1 Forward KL (Mass Covering)

$$\mathrm{KL}(p\|q) = \int p(x)\log\frac{p(x)}{q(x)}\,dx = \underbrace{\int p(x)\log p(x)\,dx}_{\text{constant in } q} - \mathbb{E}_{x\sim p}[\log q(x)].$$

Minimizing forward KL with respect to q is the same as maximizing $\mathbb{E}_p \log q(x)$, i.e., maximizing the likelihood of data under q. Two consequences:

- If $q(x) = 0$ on any set where $p(x) > 0$, the objective is infinite because $\log 0 = -\infty$. Hence forward KL strongly punishes missing mass. This is the formal statement of mass covering: to get finite loss, q must put some mass everywhere that p puts mass.
- In practice (e.g., maximum likelihood), q is pushed to assign non-negligible probability across all data modes.

3.2 Reverse KL (Mode Seeking)

$$\mathrm{KL}(q\|p) = \int q(x) \log \frac{q(x)}{p(x)}\, dx = -H(q) - \mathbb{E}_{x\sim q}[\log p(x)].$$

Now the expectation is under q:

- Placing placing mass where p is tiny is heavily penalized (since $\log p(x)$ is very negative).
- But ignoring regions where p has mass costs nothing if q places zero mass there (those regions never get sampled under q). Hence reverse KL tends to concentrate on one or a few high-density modes of p and ignore the restthis is mode seeking.

3.3 Cartoon Example: Two Separated Gaussians

Let the target p be a mixture of two well-separated Gaussians in 1D:

$$p(x) = \frac{1}{2}\mathcal{N}(x \mid -\mu, \sigma^2) + \frac{1}{2}\mathcal{N}(x \mid +\mu, \sigma^2),$$

and suppose the model class q is a single Gaussian $\mathcal{N}(x \mid m, s^2)$.

- Minimizing $\mathrm{KL}(p\|q)$ (forward) yields a solution with $m \approx 0$ and s inflated to cover both lobes. Samples are blurry, but both modes get non-negligible probability.
- Minimizing $\mathrm{KL}(q\|p)$ (reverse) yields a solution with $m \approx +\mu$ or $m \approx -\mu$ and small s: the model locks onto one mode and drops the other.

This simple picture captures the essence of coverage vs. mode seeking and will mirror the behavior we see with different generative training objectives.

4 Coverage Explained by the Likelihood/KL Decomposition

A clean way to express the coverage claim is to write the forward KL between data and a parametric model p_θ:

$$\mathrm{KL}(p_{\text{data}}\|p_\theta) = \underbrace{\int p_{\text{data}}(x) \log p_{\text{data}}(x)\, dx}_{\text{constant wrt } \theta} - \mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)].$$

The first term is a constant (the data entropy). Therefore minimizing the forward KL is equivalent to maximizing the expected log-likelihood $\mathbb{E}_{p_{\text{data}}} \log p_\theta(x)$. Two immediate implications:

1. Because the expectation is under the data p_{data}, every region where the data live contributes to the objective. If $p_\theta(x)$ is (near) zero on a data mode, the term $-\log p_\theta(x)$ is (very) large there. The optimizer is forced to increase $p_\theta(x)$ on that region to reduce loss. Thats coverage.
2. Missing a mode is costly. In the limit $p_\theta(x) \to 0$ on a set of nonzero data measure, the loss diverges. Thus maximum likelihood has an inherent incentive to avoid mode dropping.

This stands in sharp contrast to generator updates that are expectations under p_g, where missing modes vanish from the gradient.

5 Why the GAN Generator Tends to be Mode-Seeking in Practice

Even though the value function with D^* involves $\mathrm{JS}(p_{\text{data}} \| p_g)$, the generator gradient in practical GAN training does not directly minimize JS. It minimizes either the saturating loss

$$L_G^{\text{minimax}}(\theta) = \mathbb{E}_{z\sim p(z)}[\log(1 - D(G_\theta(z)))],$$

or, more commonly, the non-saturating loss

$$L_G^{\text{NS}}(\theta) = -\mathbb{E}_{z\sim p(z)}[\log D(G_\theta(z))].$$

The crucial property for both is:

$$\text{generator gradients} \propto \mathbb{E}_{x\sim p_g}[\cdots],$$

i.e., expectations under the models own distribution. If p_g does not sample a true mode, there is no gradient from that region, which makes it easy for the generator to perfect a subset of modes and ignore the rest. This is a reverse-KL-like bias in the learning signal, even if the minimax formulation involves JS. Lets examine the non-saturating form in more detail. Using $D^*(x) = \frac{p_{\text{data}}(x)}{p_{\text{data}}(x)+p_g(x)}$,

$$L_G^{\text{NS}}(\theta) = -\mathbb{E}_{x\sim p_g}\left[\log \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_g(x)}\right] = \mathbb{E}_{x\sim p_g}\left[\log\left(1 + \frac{p_g(x)}{p_{\text{data}}(x)}\right)\right].$$

- If x is a region where p_g is large and p_{data} is small, the ratio explodes and the term is big: the loss penalizes fake-only regions.
- If x is a region where p_{data} is large and p_g is moderate, the term is smaller: the loss rewards moving mass toward high-density data regionsbut only where the generator already samples.

For a missing mode $A \subset \mathscr{X}$ with $p_{\text{data}}(A) > 0$ and $p_g(A) = 0$, we have

$$\int_A \log\left(1 + \frac{p_g(x)}{p_{\text{data}}(x)}\right) p_g(x)\, dx = 0,$$

so A contributes nothing to the loss or its gradient. The generator thus lacks a direct learning signal to discover A, which is exactly why mode collapse is persistent once it starts.

6 The Role of Support Mismatch, Saturation, and Gradients

When the supports of p_g and p_{data} barely overlap, the optimal discriminator saturates: $D^*(x) \approx 0$ for generator samples and $D^*(x) \approx 1$ for real samples. Under the minimax loss, the generator gradient is roughly proportional to $\nabla_\theta \log(1 - D(G_\theta(z)))$, and when D is near 0 on generated samples, $\log(1-D)$ is near 0 and changes slowlythe gradient is small. This is the classic vanishing-gradient issue. The non-saturating loss was introduced precisely to avoid that: when $D(G(z)) \approx 0$, $-\log D(G(z))$ is large, giving a stronger learning signal. However, this does not fix the sample-myopia effect: if the generator does not sample a mode, there is still no gradient pointing to it. A common mitigation is to increase support overlap early in training, e.g., by adding instance noise (Gaussian noise) to both real and fake samples or by using data augmentation on images. This thickens the manifolds so that D cannot cleanly separate them, preventing immediate saturation and allowing the generator to receive more informative gradients.

7 A Unifying View with f-Divergences and Density-Ratio Estimation

Many adversarial training formulations can be seen through an f-divergence lens:

$$D_f(p\|q) = \int q(x) f\left(\frac{p(x)}{q(x)}\right) dx$$

for a convex function f with $f(1) = 0$. Various choices of f reproduce KL, reverse KL, Pearson χ^2, JS, etc. Adversarial methods often learn a critic $T_\phi(x)$ that estimates the density ratio or its monotone transform and train the generator against that critic. Even in this general setting, generator updates are typically expectations under p_g, because we can only sample from the model and from data. When the objective is arranged so that the generators gradient depends on $\mathbb{E}_{x\sim p_g}[\cdots]$, unseen modes vanish from the gradient. Thus, mode dropping is not unique to the original GAN; it is a common failure mode for adversarial objectives whose learning signal is model-sampleweighted. This is precisely why maximum likelihood (forward KL) has such different behavior: the expectation is over dataunseen modes cannot hide.

8 Comparing GANs, MLE Models, WGANs, and Diffusion/Score Models on Coverage

It helps to place GANs in a wider map of generative training paradigms:

- **Maximum Likelihood (flow-based models, autoregressive models)**: These directly (or tractably) optimize $\mathbb{E}_{p_{\text{data}}} \log p_\theta(x)$, i.e., minimize $\text{KL}(p_{\text{data}}\|p_\theta)$. They are mass-covering by construction, and true mode collapse is atypical. Pathologies tend to be overly diffuse samples in ambiguous regions.
- **GANs (JS-divergencestyle adversarial training)**: Generator gradients are expectations under p_g. They excel at sharpness and realism but are prone to mode dropping due to sample myopia and support mismatch.
- **Wasserstein GANs (WGAN, WGAN-GP)**: Replace JS with an approximation to the Earth Movers (Wasserstein-1) distance. The discriminator (critic) estimates differences in expectations $\mathbb{E}_{p_{\text{data}}} f - \mathbb{E}_{p_g} f$ over 1-Lipschitz functions f. The signal is often more stable and informative when supports are disjoint, alleviating vanishing gradients. However, generator updates are still expectations under p_g; WGANs mitigate but do not eliminate mode dropping.
- **Diffusion and Score-Based Models**: Train by denoising score matching or related objectives that match the score $\nabla_x \log p_\theta(x_t \mid t)$ across noise levels. Training reduces to supervised regression under the data noising process and does not involve a discriminator. These models are empirically coverage-friendly and less prone to mode dropping, though they can be computationally heavy at sampling time.

The central theme: Which distributions expectation appears in the gradient? If data expectation dominates (forward KL, diffusion training), coverage is favored. If model expectation dominates (typical adversarial setups), mode seeking is favored without extra safeguards.

9 A Precise Statement of Coverage vs. Mode Dropping in Equations

Consider two statements, phrased exactly:

1. **Maximum Likelihood Enforces Coverage**: Since

$$\text{KL}(p_{\text{data}}\|p_\theta) = \text{constant} - \mathbb{E}_{x\sim p_{\text{data}}}[\log p_\theta(x)],$$

minimizing forward KL requires maximizing expected log-likelihood over actual data. Missing any data mode (assigning $p_\theta(x) \approx 0$ on a set of nonzero data measure) makes the loss extremely large. Therefore a learned p_θ is incentivized to assign mass across all modes: coverage.

2. **Standard GAN Training Tolerates Modeling Only a Subset**: For the non-saturating loss under optimal discrimination,
$$L_G^{\mathrm{NS}}(\theta) = \mathbb{E}_{x\sim p_g}\left[\log\left(1+\frac{p_g(x)}{p_{\mathrm{data}}(x)}\right)\right].$$
The expectation is computed under p_g. Any region unvisited by p_g contributes zero to the loss and zero to its gradient, so the generator can be happy modeling a subset of the support extremely well, leaving other valid modes under-modeled or ignoredmode dropping.

These two lines capture the intuitive difference completely, without any external reference.

10 Practical Diagnostics for Mode Collapse

If you plan to publish this discussion, readers will appreciate explicit metrics and what they capture:

- **Inception Score (IS)**: Measures how confidently a classifier recognizes generated images (sharpness) and how diverse those recognitions are. High IS can still occur with diversity limited to a subset of real classes (partial collapse).
- **Fréchet Inception Distance (FID)**: Compares means/covariances of features for real vs. generated images. Sensitive to both fidelity and diversity, but it reduces distributions to Gaussian approximations in feature space; it does not explicitly report how many modes are covered.
- **PrecisionRecall (PRD Curves)**: Decompose performance into precision (fidelity) and recall (coverage). Low recall at high precision is a classic signature of mode dropping.
- **Class-Conditional Coverage Tests**: When labels exist (e.g., CIFAR-10), compare class histograms of generated samples against real data. A skewed histogram is often the first red flag.
- **Latent Traversal Tests**: Interpolate in latent space and observe output variety. If many latent directions map to indistinguishable outputs, the generator has collapsed its mapping.

No single metric is definitive; a combination gives a more reliable picture.

11 Engineering Techniques that Reduce Mode Collapse

To make the write-up useful for practitioners, its worth grouping mitigations by the failure mechanism they target.

11.1 Increase Support Overlap / Keep Discriminator from Saturating

- Instance noise (add small Gaussian noise to real and fake inputs during training).
- Data augmentation with consistency constraints for the discriminator (augment both real and fake).
- Label smoothing and noisy labels for D, to prevent overconfident discrimination early.

11.2 Stabilize the Adversarial Game

- Spectral normalization on the discriminator to enforce Lipschitz-like constraints.
- Gradient penalties (e.g., WGAN-GP) or consistency regularization to stabilize the critic.
- Two Time-Scale Update Rule (TTUR): different learning rates for D and G to balance progress.
- Balanced update counts (e.g., $n_D : n_G$) with monitoringtoo strong D invites collapse, too weak D yields low fidelity.

11.3 Encourage Diversity Explicitly

- Minibatch discrimination / minibatch standard deviation layers. Give D features that detect lack of variation, forcing G to diversify.
- Feature matching: have G match intermediate feature statistics of D over a minibatch, not just fool D on individual samples.
- Entropy or dispersion regularizers on p_g (e.g., encourage spread in generator outputs for nearby z).
- Top-k training and path-length regularization (used in StyleGAN variants) that smooth generator mapping and discourage brittle collapses.

11.4 Architectural and Training Heuristics

- Ensure sufficient generator capacity and avoid excessively tight bottlenecks.
- Use progressive growing (for very high-res images) so G learns coarse diversity before fine details.
- Early stopping based on recall-oriented metrics to catch collapse early.
- Latent mixing or noise injection at multiple layers (e.g., in StyleGAN) to encourage diverse stochasticity.

None of these is a magic bullet; in practice, two or three combined (e.g., spectral norm + gradient penalty + augmentation) often produce large gains.

12 A Short Derivation: Why Forward KL Punishes Missing Mass and Reverse KL Doesnt

Let $A \subset \mathscr{X}$ with $p(A) > 0$. Consider what happens if the model assigns zero mass on A, i.e., $q(x) = 0$ for $x \in A$.

- **Forward KL**:

$$\mathrm{KL}(p\|q) = \int_{\mathscr{X}} p(x) \log \frac{p(x)}{q(x)}\, dx = \int_A p(x) \log \frac{p(x)}{0}\, dx + \int_{\mathscr{X}\setminus A} p(x) \log \frac{p(x)}{q(x)}\, dx.$$

  The first integral diverges to $+\infty$. Therefore, any optimizer minimizing forward KL must assign nonzero $q(x)$ on A.

- **Reverse KL**:

$$\mathrm{KL}(q\|p) = \int_{\mathscr{X}} q(x) \log \frac{q(x)}{p(x)}\, dx = \int_A 0 \cdot \log \frac{0}{p(x)}\, dx + \int_{\mathscr{X}\setminus A} q(x) \log \frac{q(x)}{p(x)}\, dx.$$

  The contribution from A is zero. The optimizer can happily set q to zero on A without penalty. This is the mathematical statement of mode seeking: the model can concentrate elsewhere if that reduces cost, with no incentive to cover A.

13 A Worked Toy Example: Two-Mode Target, One-Mode Model

Let p be the symmetric two-Gaussian mixture from earlier and $q = \mathcal{N}(m, s^2)$.

- For forward KL, in the limit of large separation ($\mu \gg \sigma$), we can reason qualitatively: maximizing $\mathbb{E}_p \log q(x)$ pushes q to place probability across both lobes. The optimal m is near 0 (the mixture mean) and the optimal s must be large enough that q allocates non-negligible density at $x \approx \pm\mu$. The cost of underweighting either lobe is extreme (log-likelihood plummets), so the solution inflates s. You cover both modesmass covering.
- For reverse KL, $\mathbb{E}_q \log p(x)$ places emphasis on regions where q already has mass. Since p itself is a mixture, $\log p(x)$ near $+\mu$ (or $-\mu$) is dominated by the closer component; the other component contributes exponentially small density. The optimizer therefore centers m near $+\mu$ (or $-\mu$) and keeps s small to avoid low-density tails where p is tiny. One lobe is chosen, the other is droppedmode seeking.

This example explains why a reverse-KL-flavored learning signal prefers peaky, single-mode fits, while forward KL prefers broader, covering fits.

14 How This Connects to Variational Inference Intuitions

Many readers have seen the same dichotomy in variational inference:

- The ELBO for latent-variable models involves a $\mathrm{KL}(q(z \mid x) \| p(z \mid x))$a reverse KL between approximate and true posteriors. The result is often an approximation that underestimates posterior variance and locks onto one high-density explanation rather than covering all plausible explanations.
- In contrast, fitting a generative model by maximizing the marginal likelihood over data corresponds to minimizing forward KL in data space, which is mass covering.

The GAN generators gradient, being expectation under p_g, parallels the reverse-KL bias (mode seeking). Maximum likelihood parallels forward KL (coverage). Keeping these two mental models aligned helps reason about failure modes and fixes.

15 Putting It All Together in a Crisp Set of Takeaways

- **Why GANs Collapse**: (i) Generator updates use expectations under p_g, so unseen modes produce no gradient; (ii) Discriminator saturation under support mismatch weakens gradients to the generator; (iii) The adversarial games dynamics (capacity, update balance) easily reinforce safe pocketsleading to mode dropping.
- **Why Forward KL Enforces Coverage**: The forward KL decomposition $\mathrm{KL}(p_{\text{data}} \| p_\theta) = \text{constant} - \mathbb{E}_{p_{\text{data}}} \log p_\theta$ places the learning signal on real data. Missing a data mode is catastrophically costly, so maximum likelihood pushes probability mass everywhere data live.
- **Mode-Seeking vs. Mass-Covering**: Reverse KL (and reverse-KL-like learning signals) encourages mode seekingconcentrate where the target is high, ignore where its moderate. Forward KL encourages mass coveringallocate non-negligible density across all data support, even if samples look blurry in ambiguous regions.
- **How to Mitigate Collapse**: Increase support overlap (instance noise, augmentation), stabilize D (spectral norm, gradient penalties), balance updates (TTUR), and directly teach D to detect lack of diversity (minibatch discrimination). Monitor with recall-sensitive metrics (PR curves), not just IS or FID.

These points, expressed in the equations included above, let you explain the coverage claim and the subset or mode dropping behavior of standard GAN training without invoking outside references.

A Technical Details on Mode Collapse and Divergence Objectives

A.1 Generator Gradient Forms: Minimax vs. Non-Saturating

Let $D(x) = \sigma(a(x))$ with $\sigma(t) = 1/(1+e^{-t})$ and $a(x) \in \mathbb{R}$ the discriminator logit. For a generated sample $x = G_\theta(z)$, the scalar loss $\ell(a)$ induces the generator gradient by the chain rule:

$$\nabla_\theta \ell(a(G_\theta(z))) = \left.\frac{\partial \ell}{\partial a}\right|_{a=a(G_\theta(z))} \cdot \nabla_\theta a(G_\theta(z)).$$

Minimax (saturating) generator loss.

$$\ell_{\min}(a) = \log(1-\sigma(a)), \quad \frac{\partial \ell_{\min}}{\partial a} = -\sigma(a).$$

If $a \ll 0$ (i.e., $D(x) \approx 0$ on generated samples), then $\sigma(a) \approx 0$ and $\frac{\partial \ell_{\min}}{\partial a} \approx 0$: generator gradients vanish.
Non-saturating generator loss.

$$\ell_{\text{ns}}(a) = -\log(\sigma(a)), \quad \frac{\partial \ell_{\text{ns}}}{\partial a} = \sigma(a) - 1.$$

If $a \ll 0$, then $\sigma(a) \approx 0$ and $\frac{\partial \ell_{\text{ns}}}{\partial a} \approx -1$: the generator receives a strong, non-vanishing signal. This explains the empirical stability of the non-saturating variant when the discriminator is confident. Even with the non-saturating loss, expectations remain taken over $x \sim p_g$, so regions never sampled by the generator contribute neither loss nor gradient. This structural property underlies mode dropping.

A.2 Instance Noise and Support Overlap

Let p and q denote the data and generator distributions. Adding isotropic Gaussian noise $\varepsilon \sim \mathcal{N}(0, \sigma^2 I)$ to inputs yields convolved density

$$\tilde{p} = p * \varphi_\sigma, \quad \tilde{q} = q * \varphi_\sigma, \quad \varphi_\sigma(x) = \frac{1}{(2\pi\sigma^2)^{d/2}} e^{-\|x\|^2/(2\sigma^2)}.$$

For any $\sigma > 0$, $\tilde{p}$ and $\tilde{q}$ are strictly positive and share full support on $\mathbb{R}^d$. The optimal discriminator cannot saturate immediately, and generator gradients are non-degenerate at early iterations. An annealing schedule $\sigma_t \downarrow 0$ recovers the original problem as training proceeds, while maintaining informative gradients in the transient regime. Formally, convolution commutes with weak limits and preserves differentiability properties relevant to gradient signals; moreover, divergences such as $\mathrm{JS}(\tilde{p}\|\tilde{q})$ are continuous in σ, ensuring a smooth homotopy from the noise-smoothed objective to the target objective.

A.3 Wasserstein GANs and Lipschitz Critics

The Wasserstein-1 distance between P and Q is

$$W(P,Q) = \sup_{\|f\|_{\text{Lip}} \le 1} \big(\mathbb{E}_{x\sim P}[f(x)] - \mathbb{E}_{x\sim Q}[f(x)]\big),$$

(KantorovichRubinstein duality). In WGAN, a critic f_ψ approximates the supremum over 1-Lipschitz functions, and the generator minimizes $\mathbb{E}_{x\sim p_g} f_\psi(x) - \mathbb{E}_{x\sim p_{\text{data}}} f_\psi(x)$. Lipschitz enforcement. Two practical mechanisms:

1. **Gradient Penalty (GP)**: With $\hat{x} = \varepsilon x_{\text{real}} + (1-\varepsilon)x_{\text{fake}}$, $\varepsilon \sim \mathscr{U}(0,1)$,

$$\mathscr{L}_{\text{critic}} = \mathbb{E}_{x\sim p_g}[f_\psi(x)] - \mathbb{E}_{x\sim p_{\text{data}}}[f_\psi(x)] + \lambda \mathbb{E}_{\hat{x}}\big(\|\nabla_{\hat{x}} f_\psi(\hat{x})\|_2 - 1\big)^2.$$

   This encourages $\|\nabla f_\psi\| \approx 1$ along the datamodel interpolation manifold.

2. **Spectral Normalization**: Each linear operator W in the critic is normalized by its spectral norm $\sigma_{\max}(W)$ to control the global Lipschitz constant.

Compared with JS-based training, $W(P,Q)$ provides informative gradients even when supports are disjoint. Nonetheless, generator updates still average over p_g; thus, while vanishing-gradient pathologies are mitigated, the lack of direct penalties for unvisited modes remains.

A.4 PrecisionRecall Evaluation for Generative Models

Let $\phi : \mathscr{X} \to \mathbb{R}^m$ denote a feature embedding (e.g., an intermediate layer of a pretrained classifier). Define empirical real and generated sets $R = \{\phi(x_i^{\text{real}})\}_{i=1}^n$ and $G = \{\phi(x_j^{\text{gen}})\}_{j=1}^m$. A common operationalization introduces scale parameters $\tau > 0$ (or adaptive k-NN radii) to define support estimators:

$$\mathscr{S}_R(\tau) = \bigcup_{u\in R} B(u,\tau), \quad \mathscr{S}_G(\tau) = \bigcup_{v\in G} B(v,\tau),$$

with $B(u,\tau)$ the ℓ_2 ball of radius τ centered at u. Then:

$$\text{Precision}(\tau) = \frac{1}{m}\sum_{v\in G} \mathbf{1}\{v \in \mathscr{S}_R(\tau)\}, \quad \text{Recall}(\tau) = \frac{1}{n}\sum_{u\in R} \mathbf{1}\{u \in \mathscr{S}_G(\tau)\}.$$

Precision measures fidelity (generated samples lying in the neighborhood of the real support), while recall measures coverage (real support captured by the generated set). Varying τ (or k) yields PR curves; low recall at fixed precision indicates mode dropping. For completeness, the Fréchet Inception Distance (FID) models $\phi(x)$ under Gaussian approximations with means μ_r, μ_g and covariances Σ_r, Σ_g:

$$\text{FID} = \|\mu_r - \mu_g\|_2^2 + \text{Tr}\left(\Sigma_r + \Sigma_g - 2(\Sigma_r^{1/2}\Sigma_g\Sigma_r^{1/2})^{1/2}\right).$$

FID summarizes both fidelity and diversity but does not directly decompose them; PR metrics complement FID by exposing recall deficits.

A.5 Minibatch-Level Diversity Signals for the Discriminator

Two mechanisms are commonly employed to make the discriminator sensitive to lack of diversity across a minibatch.

A.5.1 Minibatch Discrimination

Given feature vectors $h_i = \phi(x_i) \in \mathbb{R}^d$ for samples $\{x_i\}_{i=1}^B$, learn a tensor $T \in \mathbb{R}^{d\times k\times c}$ and form

$$M_i = \left[\sum_{j=1}^{B} \exp(-\|T_{:,r,:}^\top h_i - T_{:,r,:}^\top h_j\|_1)\right]_{r=1}^{k} \in \mathbb{R}^k.$$

Concatenate M_i to h_i before classification. When generator outputs across the batch are highly similar, M_i changes characteristically, enabling the discriminator to penalize collapses.

A.5.2 Minibatch Standard-Deviation Channel

Augment the discriminators penultimate layer with a scalar or small-channel feature representing the per-feature standard deviation across the minibatch. The discriminator then prefers batches with non-trivial variation. Both methods provide the discriminator with direct signals tied to sample-to-sample diversity, thereby discouraging generator mappings that collapse large regions of latent space to near-identical outputs.

A.6 Feature Matching Loss for the Generator

Let $f(\cdot)$ denote an intermediate feature map of the discriminator. The feature matching objective adds

$$\mathscr{L}_{\mathrm{FM}} = \left\| \mathbb{E}_{x\sim p_{\mathrm{data}}}[f(x)] - \mathbb{E}_{z\sim p(z)}[f(G_\theta(z))] \right\|_2^2,$$

to the adversarial loss. This encourages the generator to match batch-level statistics rather than solely maximizing the discriminators misclassification of individual samples. The effect is a smoother objective with implicit encouragement of diversity (multiple modes are typically needed to match feature moments).

A.7 Two-Gaussian Toy Problem: Forward vs. Reverse KL Optima

Consider the target distribution

$$p(x) = \frac{1}{2}\mathscr{N}(x \mid -\mu, \sigma^2) + \frac{1}{2}\mathscr{N}(x \mid +\mu, \sigma^2), \quad \mu > 0,$$

and the unimodal model family $q(x) = \mathscr{N}(x \mid m, s^2)$.

A.7.1 Forward KL Optimum (Mass Covering)

Maximizing $\mathbb{E}_p \log q(x)$ gives

$$\mathbb{E}_p\left[(x-m)^2\right] = \mathrm{Var}_p(x) + (m - \mathbb{E}_p[x])^2 = \mu^2 + \sigma^2 + m^2,$$

since $\mathbb{E}_p[x] = 0$ and $\mathrm{Var}_p(x) = \mu^2 + \sigma^2$. The expected log-likelihood (up to constants) is

$$-\frac{1}{2s^2}\left(\mu^2 + \sigma^2 + m^2\right) - \frac{1}{2}\log s^2.$$

Optimization yields $m^* = 0$ and $s^{*2} = \mu^2 + \sigma^2$. Thus the forward-KL solution centers between the modes and inflates variance to cover both.

A.7.2 Reverse KL Optimum (Mode Seeking)

For large separation $\mu/\sigma \to \infty$, $\log p(x) \approx \log \frac{1}{2} + \log \mathscr{N}(x \mid \pm\mu, \sigma^2)$ in the vicinity of one chosen mode and is much smaller elsewhere. The reverse-KL objective

$$\mathrm{KL}(q\|p) = \mathbb{E}_q \log q(x) - \mathbb{E}_q \log p(x)$$

is then minimized by choosing $m^* \approx +\mu$ (or $-\mu$) and $s^{*2} \approx \sigma^2$, i.e., by locking onto a single mode and ignoring the other. This establishes the canonical mass-covering vs. mode-seeking contrast.

A.8 Two-Time-Scale Updates and Optimization Balance

Let η_D and η_G denote learning rates for the discriminator and generator. Empirically, stable training benefits from a two-time-scale regime with η_D modestly larger than η_G and possibly multiple discriminator steps per generator step ($n_D > n_G$). Excessively strong discriminators (large η_D and/or n_D) induce near-perfect classification early, re-introducing saturation and exacerbating collapse; excessively weak discriminators reduce fidelity. Convergence analyses for two-time-scale stochastic approximation justify such asymmetry under suitable conditions, although the full GAN game is non-convex/non-concave and outside classical guarantees.

A.9 Recall-Sensitive Early-Stopping and Curriculum

When labeled subpopulations are available, class-conditional coverage can be tracked by histogram divergence (e.g., total variation or χ^2 distance) between generated and real class allocations. Unsupervised settings may employ recall proxies (Section D). An effective curriculum emphasizes broad support at low resolutions or under strong augmentations before increasing resolution or reducing stochasticity; this aligns the generators early learning signal with coverage rather than fine-grained fidelity.

A.10 Summary of Structural Causes and Remedies

1. **Structural Cause**: Generator gradients are expectations over p_g; unvisited modes yield no learning signal.
   Remedy: Incorporate diversity-sensitive signals (minibatch discrimination, feature matching) and recall-oriented evaluation.

2. **Structural Cause**: Discriminator saturation under support mismatch.
   Remedy: Instance noise, augmentation, Wasserstein objectives, spectral normalization, and gradient penalties.

3. **Structural Cause**: Fragile game dynamics and capacity bottlenecks.
   Remedy: Two-time-scale updates, balanced step ratios, adequate generator capacity, and curriculum strategies.