Lecture 3: Variational Autoencoders and Generative Adversarial Networks

CS 6501-005: Constrained-Aware Generative AI

Tuesday, January 20, 2026

Abstract

In Lectures 1 and 2, we framed generative modeling as approximate inference under a target distribution that may include hard or soft constraints. We emphasized likelihood-based modeling and latent-variable formulations as the first systematic tools for controlling generation through probabilistic structure. This lecture focuses on two canonical model families that depart from fully explicit likelihoods in different ways: variational autoencoders (VAEs) and generative adversarial networks (GANs). Both families introduce mechanisms that can be interpreted as *weak control*: they shape the support, geometry, or statistics of generated samples, but do not provide explicit feasibility objectives with respect to external constraints.

1 Latent-variable models

We recall from Lecture 2 that a latent-variable model specifies a joint distribution over observations $x \in \mathcal{X}$ and latent variables $z \in \mathcal{Z}$ and induces the marginal

$$p_\theta(x) \;=\; \int p_\theta(x \mid z)\, p(z)\, dz, \tag{1}$$

where $p(z)$ is a prior and $p_\theta(x \mid z)$ is a conditional likelihood (decoder). This decomposition is attractive because it allows us to represent a complex $p_\theta(x)$ using simpler building blocks. In particular, if z captures the dominant factors of variation, then $p_\theta(x \mid z)$ can be chosen to have a comparatively simple form (for instance a factorized Gaussian or Bernoulli likelihood), while the marginal $p_\theta(x)$ can still be highly expressive due to the mixing induced by integrating out z [2]. As an example, if x is an image of a person and z is low-dimensional, such as describing the eye color, hair style, and pose, then $p_\theta(x \mid z)$ can be interpreted as a local model around the manifold defined by z, and integrating over $p(z)$ blends these local models into a global distribution over $\mathcal{X}$ [6, 8].

The central technical obstacle is inference. When $z \in \mathbb{R}^d$, the marginal likelihood involves a high-dimensional integral,

$$p_\theta(x) = \int p_\theta(x \mid z)\, p(z)\, dz, \tag{2}$$

and the posterior normalization constant is given by the same integral. Even if both $p_\theta(x \mid z)$ and $p(z)$ have simple closed forms, the integral typically does not admit a closed-form expression once $p_\theta(x \mid z)$ is parameterized by a neural network. Moreover, gradients of $\log p_\theta(x)$ require differentiating through this integral, which entails computing expectations with respect to the true posterior $p_\theta(z \mid x)$, itself defined only implicitly through the intractable normalization constant:

$$p_\theta(z \mid x) \;=\; \frac{p_\theta(x \mid z)\, p(z)}{\int p_\theta(x \mid z')\, p(z')\, dz'}. \tag{3}$$

Even when $p_\theta(x \mid z)$ and $p(z)$ are tractable, the denominator in (3) couples all latent configurations and is typically intractable in high dimensions [2].

For discrete latent variables, the difficulty is combinatorial. Suppose $z \in \{0,1\}^d$ is a vector of d binary latent features. Evaluating the marginal likelihood or the posterior normalization constant requires summing over all 2^d possible configurations,

$$p_\theta(x) = \sum_{z \in \{0,1\}^d} p_\theta(x, z), \tag{4}$$

which becomes computationally infeasible even for moderate d. This phenomenon already appears in classical mixture models: if z indexes K mixture components, then posterior inference requires evaluating K likelihood terms for each datapoint. In deep latent-variable models, where z may represent dozens or hundreds of latent factors, this exponential scaling makes exact summation impossible.

In both cases, the core issue is the same: posterior inference requires aggregating contributions from all latent configurations that are consistent with a given observation. For discrete latents this aggregation takes the form of an exponentially large sum, while for continuous latents it takes the form of a high-dimensional integral with no analytic solution. This is the reason why approximate inference techniques are required. The variational framework introduced next can be understood as a principled way to replace this intractable posterior with a tractable surrogate that can be optimized efficiently [2, 6, 8, 3].

2 Variational inference and the ELBO

Given data $x \sim p_{\text{data}}$, maximum likelihood learning seeks parameters θ that maximize $\mathbb{E}_{p_{\text{data}}}[\log p_\theta(x)]$. For a latent-variable model, using (1) we have $\log p_\theta(x) = \log \int p_\theta(x \mid z)\, p(z)\, dz$. As mentioned above, the obstacle is the *marginal likelihood integral* inside the logarithm.

How the ELBO addresses intractable marginal likelihoods. The key idea of variational inference is to replace the intractable quantity $\log p_\theta(x)$ with a *tractable lower bound* that can be optimized instead. We do this by introducing a variational distribution $q_\phi(z \mid x)$ that is easy to sample from and evaluate; the resulting objective is called the *evidence lower bound* (ELBO). The ELBO is constructed so that

$$\mathcal{L}(x; \theta, \phi) \;\le\; \log p_\theta(x) \qquad \text{and} \qquad \mathcal{L}(x; \theta, \phi) \text{ is tractable,} \tag{5}$$

meaning that maximizing $\mathcal{L}$ increases a guaranteed lower bound on the true log-marginal likelihood. Importantly, the ELBO depends only on expectations under $q_\phi(z \mid x)$ and on terms $\log p_\theta(x \mid z)$, $\log p(z)$, and $\log q_\phi(z \mid x)$, all of which are chosen to be computationally manageable. In this way, the ELBO *sidesteps* the need to evaluate the intractable integral in (1) directly.

Jensen's inequality and the ELBO derivation. To derive the bound, we begin by multiplying and dividing the integrand by $q_\phi(z \mid x)$:

$$p_\theta(x) \;=\; \int p_\theta(x, z)\, dz \;=\; \int q_\phi(z \mid x)\, \frac{p_\theta(x, z)}{q_\phi(z \mid x)}\, dz \;=\; \mathbb{E}_{z \sim q_\phi(\cdot \mid x)}\left[\frac{p_\theta(x, z)}{q_\phi(z \mid x)}\right]. \tag{6}$$

Taking logs gives

$$\log p_\theta(x) \;=\; \log \mathbb{E}_{z \sim q_\phi(\cdot \mid x)}\left[\frac{p_\theta(x, z)}{q_\phi(z \mid x)}\right]. \tag{7}$$

At this point, the intractability is precisely that we have a *log of an expectation*. Jensen's inequality for a concave function f states that

$$f(\mathbb{E}[Y]) \;\ge\; \mathbb{E}[f(Y)]. \tag{8}$$

Since $\log(\cdot)$ is concave, applying Jensen to (7) yields

$$\log \mathbb{E}_{q_\phi}\left[\frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right] \;\geq\; \mathbb{E}_{q_\phi}\left[\log \frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right]. \tag{9}$$

We define the right-hand side to be the ELBO [2]:

$$\mathcal{L}(x;\theta,\phi) \;\triangleq\; \mathbb{E}_{z\sim q_\phi(\cdot|x)}\left[\log \frac{p_\theta(x,z)}{q_\phi(z \mid x)}\right]. \tag{10}$$

Combining (7) and (9) gives the fundamental guarantee

$$\log p_\theta(x) \;\geq\; \mathcal{L}(x;\theta,\phi). \tag{11}$$

Expanding the joint as $p_\theta(x,z) = p_\theta(x \mid z)\, p(z)$, the ELBO becomes [6, 8]

$$\mathcal{L}(x;\theta,\phi) \;=\; \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z))\,. \tag{12}$$

This form makes the tractability explicit: the expectation is taken under $q_\phi(z \mid x)$, and the KL term involves q_ϕ and p only. No evaluation of $\int p_\theta(x,z)\,dz$ is required.

Tightness and the variational gap. A complementary identity clarifies when the bound is tight:

$$\log p_\theta(x) \;=\; \mathcal{L}(x;\theta,\phi) + \mathrm{KL}(q_\phi(z \mid x) \,\|\, p_\theta(z \mid x))\,. \tag{13}$$

Because the KL divergence is nonnegative, the ELBO is always a lower bound, and it becomes exact if and only if $q_\phi(z \mid x) = p_\theta(z \mid x)$ almost everywhere. Thus, optimizing the ELBO simultaneously (i) increases a guaranteed surrogate for the intractable $\log p_\theta(x)$ and (ii) learns an approximate inference model that reduces the gap to the true posterior.

Dataset objective. For a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^M$, VAE training maximizes the empirical ELBO

$$\max_{\theta,\phi} \sum_{i=1}^{M} \mathcal{L}(x^{(i)};\theta,\phi), \tag{14}$$

which yields a tractable learning criterion that avoids evaluating intractable marginal likelihoods while still providing a principled connection to maximum likelihood through the lower-bound guarantee [6, 8, 3, 2].

3 Conditional VAEs and structured priors

Conditional VAEs extend the latent-variable framework by incorporating side information c that specifies context, desired attributes, or partial observations [6, 2]. The conditional marginal is

$$p_\theta(x \mid c) \;=\; \int p_\theta(x \mid z, c)\, p(z \mid c)\, dz, \tag{15}$$

and learning proceeds by maximizing a conditional ELBO that replaces the intractable posterior $p_\theta(z \mid x, c)$ with an amortized approximation $q_\phi(z \mid x, c)$ [6, 8]:

$$\log p_\theta(x \mid c) \;\geq\; \mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x \mid z, c)] - \mathrm{KL}(q_\phi(z \mid x, c) \,\|\, p(z \mid c))\,. \tag{16}$$

Algorithmically, conditioning is implemented by feeding c into both encoder and decoder, so that inference produces a context-dependent latent code and generation maps (z, c) into an output x.

Conditioning provides a first, practically important mechanism for control. In vision, c might specify a class label; in inverse problems, c might be a partial observation or a measurement operator; in control and design, c might encode a goal specification, environment parameters, or task descriptors. This viewpoint aligns directly with the constrained-aware target distribution from Lecture 1: conditioning modifies the base model $p_\theta(x \mid c)$ so that "preference" information enters generation through the likelihood term rather than only through post-processing.

The prior also becomes a design lever. The simplest choice is $p(z \mid c) = \mathcal{N}(0, I)$, independent of c, which encourages a single shared latent geometry across all contexts. More expressive *structured priors* bias the latent space toward multimodality and interpretability. A common example is a mixture prior,

$$p(z \mid c) \;=\; \sum_{k=1}^{K} \pi_k(c)\, \mathcal{N}(z; \mu_k(c), \Sigma_k(c)), \tag{17}$$

which can represent distinct "modes" of feasible solutions under the same context c. This is particularly relevant in constrained settings where multiple qualitatively different outputs satisfy the same specification, such as multiple grasps for a target object, multiple classes of paths in motion planning, or multiple designs that meet engineering requirements.

3.1 Posterior regularization as weak control

One can strengthen control in VAEs by modifying the ELBO with additional regularization terms:

$$\mathcal{L}(x) = \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)] - \mathrm{KL}(q_\phi(z \mid x) \,\|\, p(z)) - \lambda\, \mathbb{E}_{q_\phi(z|x)}[\psi(z, x)], \tag{18}$$

where ψ penalizes violations of desired properties.

This posterior regularization viewpoint makes explicit the connection between VAEs and constrained optimization. Nevertheless, the enforcement remains soft and approximate, and the resulting optimization problem is sensitive to weighting and approximation error.

4 Likelihood-free learning and the motivation for GANs

VAEs remain firmly within the likelihood-based paradigm: even though the marginal likelihood $p_\theta(x)$ is intractable, training is still justified as approximate maximum likelihood through the ELBO lower bound. This however inherits a central limitation: *likelihood is not always aligned with perceptual sample quality*, especially in high-dimensional spaces [9]. In particular, a model can achieve strong test log-likelihood while allocating substantial probability mass to visually implausible or low-quality regions (for example, by mixing a small amount of data distribution with a broad "noise" component), and the gap between likelihood metrics and sample quality can become more pronounced as dimensionality increases.

At the same time, improving sample realism in VAEs is often achieved by modifying likelihood assumptions or decoder capacity in ways that can adversely affect likelihood-based metrics. For example, powerful decoders can reduce the need to encode information in z, leading to weak latent representations and posterior collapse, while restrictive likelihood models can produce overly smooth reconstructions. These tensions foreshadow a broader point: optimizing a tractable surrogate for $\mathrm{KL}(p_{\mathrm{data}} \| p_\theta)$ does not necessarily optimize the notion of similarity that matters for downstream use, especially when "quality" is determined by complex, task-dependent criteria.

This motivates a shift in perspective. Instead of training by maximizing (approximate) likelihood, one can compare p_θ to p_{data} using objectives that depend only on samples. If we can draw samples $x \sim p_{\mathrm{data}}$ and $x \sim p_\theta$, we may replace likelihood evaluation with a *new metric test*: learn a statistic that distinguishes the two sample sets, and then train the generator so that the samples become indistinguishable under this statistic.

5 Generative Adversarial Networks

Generative Adversarial Networks instantiates this likelihood-free perspective by learning the test statistic as a discriminator network and coupling it to the generator through an adversarial game, yielding a training objective that directly targets sample-level indistinguishability rather than explicit density estimation.

A generative adversarial network (GAN) [4] consists of two components trained simultaneously. The generator G_θ is a deterministic mapping that transforms a latent variable $z \sim p(z)$, drawn from a simple prior such as a standard Gaussian, into a sample $x = G_\theta(z)$. This defines an implicit model distribution $p_\theta(x)$ as the pushforward of $p(z)$ through G_θ. In contrast to VAEs, this distribution is not associated with a tractable density or an explicit probabilistic decoder.

The discriminator $D_\phi(x)$ is a binary classifier trained to distinguish samples drawn from the data distribution p_{data} from samples produced by the generator. Formally, GAN training is posed as the minimax problem

$$\min_\theta \max_\phi \; \mathbb{E}_{x\sim p_{\text{data}}}[\log D_\phi(x)] + \mathbb{E}_{z\sim p(z)}[\log(1 - D_\phi(G_\theta(z)))]. \tag{19}$$

For fixed generator parameters θ, optimizing (19) with respect to ϕ corresponds to standard binary classification with cross-entropy loss. For fixed discriminator parameters ϕ, the generator is trained to produce samples that the discriminator cannot reliably distinguish from real data.

The connection to two-sample testing becomes explicit by considering the optimal discriminator for a fixed generator. Under mild assumptions on model capacity, the optimal discriminator satisfies

$$D^*_\theta(x) \;=\; \frac{p_{\text{data}}(x)}{p_{\text{data}}(x) + p_\theta(x)}. \tag{20}$$

Substituting D^*_θ back into (19) shows that, at the population level, GAN training minimizes the Jensen–Shannon divergence between p_{data} and p_θ [4]. This derivation also clarifies why the vanilla GAN objective can lead to unstable gradients when the supports of p_{data} and p_θ are nearly disjoint, which motivates alternative divergences and integral probability metrics such as Wasserstein GANs [1, 5] and the broader f-divergence viewpoint [7].

From a modeling perspective, this marks a sharp departure from VAEs. The generator is not required to explain individual datapoints under a conditional likelihood, nor to produce a posterior over latent variables. Instead, it is only required to produce samples whose aggregate statistics, as detected by the discriminator, match those of the data [4]. This design choice explains both the empirical strength and the practical fragility of GANs: the learned notion of similarity can align closely with perceptual quality, but it is also implicit, task-dependent, and difficult to constrain or interpret [9]. These properties will be central when we later discuss why GANs are awkward vehicles for enforcing explicit feasibility constraints and why iterative, optimization-based generative procedures offer a more natural interface for constraint injection.

GANs as implicit control mechanisms From the perspective of constrained-aware generation, GANs can be viewed as learning an implicit feasibility region defined by the discriminator. Samples outside this region are penalized, even if the penalty is not explicitly interpretable. This implicit control explains the empirical success of GANs in producing visually realistic samples. However, it also explains their limitations: feasibility is defined only relative to the discriminator, not with respect to external constraints or symbolic rules.

Optimization pathologies and mode collapse. GAN training is notoriously unstable [4]. The minimax structure in (19) defines a two-player game rather than a single convex (or even stationary) optimization problem, so standard descent arguments do not apply. In practice, alternating updates of (θ, ϕ) often produce oscillatory dynamics in which the discriminator and

generator continually "chase" each other, and there is typically no reliable, likelihood-based stopping criterion or monotone objective to certify progress [4]. These instabilities are amplified by function approximation and finite-sample training: when the discriminator becomes too strong relative to the generator, gradients can become uninformative; when it is too weak, it fails to provide a meaningful signal. This mismatch partly explains the empirical dependence of GANs on architectural choices and regularization "tricks," and motivates alternative formulations such as Wasserstein GANs that replace the Jensen–Shannon divergence with a smoother integral probability metric [1, 5].

A particularly important failure mode is *mode collapse* [4], where the generator concentrates on producing samples from only a few modes of p_{data} (sometimes degenerating to near-duplicates), despite the data distribution being highly multimodal. One way to view this behavior is through the geometry of the game: the generator can reduce its loss by moving probability mass toward regions that the current discriminator scores highly, even if doing so sacrifices coverage elsewhere. Since the discriminator is trained on finite minibatches and has limited capacity, it may not penalize missing modes strongly enough to force recovery.

From a control standpoint, mode collapse represents a failure to enforce *coverage* or *diversity* constraints. The generator can satisfy the discriminator locally, in the sense of producing samples that appear realistic under the discriminator's current decision boundary, while violating global requirements such as representing all feasible solution families or maintaining diversity across valid outputs. The key point is that this failure mode is structural: because the training signal is mediated by an adaptive discriminator and does not directly penalize missing support, mode collapse cannot be ruled out by the vanilla GAN objective alone, and addressing it typically requires additional mechanisms (regularization, alternative divergences, or explicit diversity-promoting constraints) [7, 1].

6 Where constraints enter: VAEs vs. GANs vs. iterative methods

We now explicitly connect VAEs and GANs to the constrained target distribution introduced in Lecture 1,

$$\pi(x \mid c) \;\propto\; p_\theta(x \mid c)\, \exp(-\lambda \phi(x, c))\, \mathbf{1}\{x \in \mathcal{C}(c)\}, \tag{21}$$

and ask where, if at all, each modeling paradigm provides a mechanism to approximate sampling from π.

This perspective makes precise why VAEs and GANs constitute *early* and *weak* forms of control, and why neither is well suited for enforcing hard feasibility constraints.

6.1 VAEs as distribution-level regularization

In a VAE, control enters implicitly through the choice of prior $p(z)$ and the KL regularization term in the ELBO. The learned model approximates

$$p_\theta(x \mid c) = \int p_\theta(x \mid z, c)\, p(z)\, dz, \tag{22}$$

with the constraint signal acting only indirectly through training data statistics or posterior regularization terms.

When viewed through (21), VAEs primarily affect the *base distribution* $p_\theta(x \mid c)$. Any constraint $\phi(x, c)$ or feasibility set $\mathcal{C}(c)$ must be either: (i) absorbed into the training distribution, (ii) approximated by a soft penalty inside the ELBO, or (iii) encoded indirectly through the latent prior.

Crucially, once training is complete, sampling from a VAE is a one-shot procedure. There is no mechanism to iteratively correct violations of $\mathcal{C}(c)$ at inference time.

6.2 GANs as implicit feasibility shaping

GANs remove the likelihood entirely and replace it with an adversarially learned discriminator. In effect, the discriminator induces an *implicit constraint* on generated samples by penalizing deviations from the data distribution.

From the standpoint of (21), GANs do not expose either $p_\theta(x \mid c)$ or an explicit energy $\phi(x, c)$. Instead, feasibility is defined only relative to the discriminator's decision boundary.

This has two important consequences. First, constraints that are not well represented in the data cannot be enforced reliably. Second, there is no principled way to project or repair infeasible samples, because feasibility is not explicitly represented.

Note that both VAEs and GANs generate samples in a single forward pass. As a result, constraint satisfaction must be *pre-compiled* into the model parameters. This design fundamentally limits the types of constraints that can be enforced. In contrast, iterative generative procedures expose intermediate states that can be modified, corrected, or projected. This structural difference, rather than any specific loss function, is what enables strong constraint enforcement.

7 Summary and preview

VAEs and GANs represent two influential but ultimately limited approaches to control in generative modeling. VAEs offer explicit probabilistic structure and weak regularization through priors and posteriors, while GANs provide implicit control through adversarial discrimination.

Neither framework provides a principled mechanism for enforcing hard constraints at inference time. This observation motivates the central theme of the course: generation as an iterative process in which optimization, projection, and correction can be interleaved with probabilistic modeling.

In the next lectures, we will see how autoregressive decoding, diffusion, and flow-based methods expose explicit iteration structures that make constraint-aware generation far more tractable.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

[2] David M. Blei, Alp Kucukelbir, and Jon D. McAuliffe. Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877, 2017.

[3] Yuri Burda, Roger Grosse, and Ruslan Salakhutdinov. Importance weighted autoencoders. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[5] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[6] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[7] Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2016.

[8] Danilo J. Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, 2014.

[9] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv preprint arXiv:1511.01844*, 2016.