# LLM-VAE

Omri Drori Topaz freizeit

# VAEs at a Glance: Purpose and Use Cases

**What is a Variational Autoencoder (VAE)?**

- A powerful type of **generative model**.
- It learns to **compress** high-dimensional data (like images) into a lower-dimensional, continuous **latent space**.
- From this latent space, it can then **generate** new, similar data.

# VAEs at a Glance: Purpose and Use Cases

**What is a Variational Autoencoder (VAE)?**

- A powerful type of **generative model**.
- It learns to **compress** high-dimensional data (like images) into a lower-dimensional, continuous **latent space**.
- From this latent space, it can then **generate** new, similar data.

**What are they used for?**

- **Data Generation:** Creating novel, realistic data samples (e.g., new faces, music, or text).

# VAEs at a Glance: Purpose and Use Cases

**What is a Variational Autoencoder (VAE)?**

- A powerful type of **generative model**.
- It learns to **compress** high-dimensional data (like images) into a lower-dimensional, continuous **latent space**.
- From this latent space, it can then **generate** new, similar data.

**What are they used for?**

- **Data Generation:** Creating novel, realistic data samples (e.g., new faces, music, or text).
- **Representation Learning:** The compressed latent space provides a meaningful, compact representation of the data, useful for tasks like clustering or classification.

# VAEs at a Glance: Purpose and Use Cases

**What is a Variational Autoencoder (VAE)?**

- A powerful type of **generative model**.
- It learns to **compress** high-dimensional data (like images) into a lower-dimensional, continuous **latent space**.
- From this latent space, it can then **generate** new, similar data.

**What are they used for?**

- **Data Generation:** Creating novel, realistic data samples (e.g., new faces, music, or text).
- **Representation Learning:** The compressed latent space provides a meaningful, compact representation of the data, useful for tasks like clustering or classification.
- **Data Interpolation:** Smoothly morphing between two data points by traversing the path between them in the latent space.

# VAEs at a Glance: Purpose and Use Cases

**What is a Variational Autoencoder (VAE)?**

- A powerful type of **generative model**.
- It learns to **compress** high-dimensional data (like images) into a lower-dimensional, continuous **latent space**.
- From this latent space, it can then **generate** new, similar data.

**What are they used for?**

- **Data Generation:** Creating novel, realistic data samples (e.g., new faces, music, or text).
- **Representation Learning:** The compressed latent space provides a meaningful, compact representation of the data, useful for tasks like clustering or classification.
- **Data Interpolation:** Smoothly morphing between two data points by traversing the path between them in the latent space.
- **Anomaly Detection:** Identifying unusual data points that the model struggles to reconstruct.

# The Goal: Modeling Data

- Goal: Find model parameters $\theta^*$ that make observed data $X = \{x_1, ..., x_N\}$ most probable.
- Model: $p_\theta(x)$.

# The Goal: Modeling Data

- Goal: Find model parameters $\theta^*$ that make observed data $X = \{x_1, ..., x_N\}$ most probable.
- Model: $p_\theta(x)$.
- Maximize Likelihood: $L(\theta|X) = \prod_{i=1}^{N} p_\theta(x_i)$.
- Or Log-Likelihood (common practice): $\mathcal{L}(\theta|X) = \sum_{i=1}^{N} \log p_\theta(x_i)$.

# The Goal: Modeling Data

- Goal: Find model parameters $\theta^*$ that make observed data $X = \{x_1, ..., x_N\}$ most probable.
- Model: $p_\theta(x)$.
- Maximize Likelihood: $L(\theta|X) = \prod_{i=1}^N p_\theta(x_i)$.
- Or Log-Likelihood (common practice): $\mathcal{L}(\theta|X) = \sum_{i=1}^N \log p_\theta(x_i)$.

**The Problem: Limited Expressive Power**

- Simple models (e.g., single Gaussian) for $p_\theta(x)$ are often too restrictive for complex data (images, text).

# The Goal: Modeling Data

- Goal: Find model parameters $\theta^*$ that make observed data $X = \{x_1, ..., x_N\}$ most probable.
- Model: $p_\theta(x)$.
- Maximize Likelihood: $L(\theta|X) = \prod_{i=1}^{N} p_\theta(x_i)$.
- Or Log-Likelihood (common practice): $\mathcal{L}(\theta|X) = \sum_{i=1}^{N} \log p_\theta(x_i)$.

**The Problem: Limited Expressive Power**

- Simple models (e.g., single Gaussian) for $p_\theta(x)$ are often too restrictive for complex data (images, text).

**Key Takeaway:**

- Need models flexible enough for complex data, yet tractable to learn.

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.
- Data probability by marginalizing $h$: $p_\theta(x) = \sum_h p_\theta(x, h)$ (for discrete $h$).

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.
- Data probability by marginalizing $h$: $p_\theta(x) = \sum_h p_\theta(x, h)$ (for discrete $h$).

**Example: Gaussian Mixture Models (GMMs)**

- Latent $h$: discrete, indicates which of $K$ Gaussian components generated $x$.

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.
- Data probability by marginalizing $h$: $p_\theta(x) = \sum_h p_\theta(x, h)$ (for discrete $h$).

**Example: Gaussian Mixture Models (GMMs)**

- Latent $h$: discrete, indicates which of $K$ Gaussian components generated $x$.
- Component prior: $p(h = k)$

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.
- Data probability by marginalizing $h$: $p_\theta(x) = \sum_h p_\theta(x, h)$ (for discrete $h$).

**Example: Gaussian Mixture Models (GMMs)**

- Latent $h$: discrete, indicates which of $K$ Gaussian components generated $x$.
- Component prior: $p(h = k)$
- Component likelihood: $p_\theta(x|h = k) = \mathcal{N}(x|\mu_k, \Sigma_k)$.

# Enhancing Models with Discrete Latent Variables

**Latent Variable Models (LVMs)**

- Introduce unobserved (latent) variables $h$ to explain structure in observed data $x$.
- Define a joint distribution: $p_\theta(x, h)$.
- Data probability by marginalizing $h$: $p_\theta(x) = \sum_h p_\theta(x, h)$ (for discrete $h$).

**Example: Gaussian Mixture Models (GMMs)**

- Latent $h$: discrete, indicates which of $K$ Gaussian components generated $x$.
- Component prior: $p(h = k)$
- Component likelihood: $p_\theta(x|h = k) = \mathcal{N}(x|\mu_k, \Sigma_k)$.
- Marginal likelihood: $p_\theta(x) = \sum_h p_\theta(x, h) = \sum_{k=1}^K p_\theta(x|h = k)p(h = k) = \sum_{k=1}^K p(h = k)\mathcal{N}(x|\mu_k, \Sigma_k)$.

# GMMs: Advantages & Lingering Limitations

**Advantage over simple MLE (with GMMs):**

- More powerful: Combine simple distributions (e.g., Gaussians in GMM) to represent complex, multi-modal data.

# GMMs: Advantages & Lingering Limitations

**Advantage over simple MLE (with GMMs):**

- More powerful: Combine simple distributions (e.g., Gaussians in GMM) to represent complex, multi-modal data.

**Remaining Limitations of Discrete LVMs:**

- May still struggle with very high-dimensional data or intricate continuous variations.
- Astronomical $K$ (number of components) might be needed for subtle nuances → impractical.

# Seeking More Expressive Power: Continuous Latent Variables

**A Potential Path Forward: Continuous Latent Space**

- What if the latent variable $h$ could take on continuous values?

# Seeking More Expressive Power: Continuous Latent Variables

**A Potential Path Forward: Continuous Latent Space**

- What if the latent variable $h$ could take on continuous values?
- Instead of a discrete set of choices, $h$ could live in a continuous space (e.g., $h \in \mathbb{R}^D$).
- This could potentially capture smoother, more nuanced variations in the data.

# Seeking More Expressive Power: Continuous Latent Variables

**A Potential Path Forward: Continuous Latent Space**

- What if the latent variable $h$ could take on continuous values?
- Instead of a discrete set of choices, $h$ could live in a continuous space (e.g., $h \in \mathbb{R}^D$).
- This could potentially capture smoother, more nuanced variations in the data.

**The New Formulation:**

- Model: $p_\theta(x, h)$, where $h$ is now continuous.

# Seeking More Expressive Power: Continuous Latent Variables

**A Potential Path Forward: Continuous Latent Space**

- What if the latent variable $h$ could take on continuous values?
- Instead of a discrete set of choices, $h$ could live in a continuous space (e.g., $h \in \mathbb{R}^D$).
- This could potentially capture smoother, more nuanced variations in the data.

**The New Formulation:**

- Model: $p_\theta(x, h)$, where $h$ is now continuous.
- Data probability: $p_\theta(x) = \int p_\theta(x, h) dh$.

# Continuous Latents: Defining $p_\theta(x|h)$

# Continuous Latents: Defining $p_\theta(x|h)$

**Recall: Discrete Latent Variables**

- For each discrete state $h = k$, $p_\theta(x|h = k)$ had its own learned parameters (e.g., $\mu_k, \Sigma_k$ for a Gaussian component).

# Continuous Latents: Defining $p_\theta(x|h)$

**Recall: Discrete Latent Variables**

- For each discrete state $h = k$, $p_\theta(x|h = k)$ had its own learned parameters (e.g., $\mu_k, \Sigma_k$ for a Gaussian component).

**Challenge with Continuous $h \in \mathbb{R}^D$**

- Cannot store separate parameters for infinitely many $h$ values.
- The parameters of $p_\theta(x|h)$ must be *derived from $h$ itself*.

# Continuous Latents: Defining $p_\theta(x|h)$

**Recall: Discrete Latent Variables**

- For each discrete state $h = k$, $p_\theta(x|h = k)$ had its own learned parameters (e.g., $\mu_k, \Sigma_k$ for a Gaussian component).

**Challenge with Continuous $h \in \mathbb{R}^D$**

- Cannot store separate parameters for infinitely many $h$ values.
- The parameters of $p_\theta(x|h)$ must be *derived from $h$ itself*.

**Solution: Parameters are Outputs of a Function**

- We use a single function, often a neural network, that maps the latent variable $h$ to all the necessary parameters of the conditional distribution. Let's call this function $f_\theta$.
- If $p_\theta(x|h)$ is a Gaussian, this function produces both its mean $\mu$ and covariance $\Sigma$:

$$(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f_\theta(h)$$

- The parameters $\theta$ (e.g., the weights of the network $f$) are what we learn during training.

# Model Structure & The Target: $p_\theta(x)$

**Components of our Generative Model so far:**

- A continuous latent variable $h \in \mathbb{R}^D$.
- A prior distribution over these latent variables, $p(h)$ (e.g., $p(h) = \mathcal{N}(h|0, I)$).
- A conditional distribution $p_\theta(x|h)$, whose parameters are determined by the output of a function $f_\theta(h)$:
  - e.g., $p_\theta(x|h) = \mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f_\theta(h)$.

# Model Structure & The Target: $p_\theta(x)$

**Components of our Generative Model so far:**

- A continuous latent variable $h \in \mathbb{R}^D$.
- A prior distribution over these latent variables, $p(h)$ (e.g., $p(h) = \mathcal{N}(h|0, I)$).
- A conditional distribution $p_\theta(x|h)$, whose parameters are determined by the output of a function $f_\theta(h)$:
  - e.g., $p_\theta(x|h) = \mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f_\theta(h)$.

**The Joint Distribution**

- Combining these, the joint probability is $p_\theta(x, h) = p_\theta(x|h)p(h)$.

# Model Structure & The Target: $p_\theta(x)$

**Components of our Generative Model so far:**

- A continuous latent variable $h \in \mathbb{R}^D$.
- A prior distribution over these latent variables, $p(h)$ (e.g., $p(h) = \mathcal{N}(h|0, I)$).
- A conditional distribution $p_\theta(x|h)$, whose parameters are determined by the output of a function $f_\theta(h)$:
  - e.g., $p_\theta(x|h) = \mathcal{N}(x|\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = f_\theta(h)$.

**The Joint Distribution**

- Combining these, the joint probability is $p_\theta(x, h) = p_\theta(x|h)p(h)$.

**Obtaining the Likelihood of Data $p_\theta(x)$**

- Our ultimate goal for MLE is to calculate $p_\theta(x)$.
- With continuous $h$, this requires integrating out $h$:

$$p_\theta(x) = \int p_\theta(x, h)dh = \int p_\theta(x|h)p(h)dh$$

# Integral Hurdle 1: Latent Dimensionality

- our log-likelihood objective depends on

$$\int p_\theta(x_i|h)p(h)dh)$$

- Our latent variable $h$ can live in a high-dimensional space (e.g., $h \in \mathbb{R}^D$, $D \gg 1$).
- The mapping from $h$ to the parameters of $x$'s distribution is often a complex, non-linear function.
- (e.g., $f_\theta(h)$ is a neural network that outputs the parameters for the distribution of $x$).
- This makes the integrand $p_\theta(x|h)p(h)$ itself highly complex.
- Generally, no analytical (closed-form) solution exists for the integral.

# Result: An Intractable Likelihood

- Due to high latent dimensionality and a complex integrand:

$$p_\theta(x) = \int p_\theta(x|h)p_\theta(h)dh$$

- This integral is generally **intractable**.
- So, we cannot compute the exact log-likelihood $\mathcal{L}(\theta|X)$.
- Direct MLE training is therefore not feasible.

# A Path Forward? Approximation

- Exact computation of $p_\theta(x)$ is off the table.
- What if we could **approximate** the integral instead?
- A common method for this is Monte Carlo estimation.

# Naive Solution: Monte Carlo Estimation

- Our integral $p_\theta(x) = \int p_\theta(x|h)p_\theta(h)dh$ can be seen as an expectation:

$$p_\theta(x) = \mathbb{E}_{h \sim p_\theta(h)}[p_\theta(x|h)]$$

- Monte Carlo methods approximate expectations using samples.

# MC Approximation for $p_\theta(x)$

**The Method:**

1. To approximate $p_\theta(x) = \mathbb{E}_{h \sim p_\theta(h)}[p_\theta(x|h)]$:
2. Draw $S$ independent samples $h^{(1)}, h^{(2)}, \ldots, h^{(S)}$ from the prior distribution $p_\theta(h)$.
3. For each sample $h^{(s)}$, calculate the conditional probability $p_\theta(x|h^{(s)})$.
4. The Monte Carlo estimate is the average:

$$\hat{p}_\theta(x) = \frac{1}{S} \sum_{s=1}^{S} p_\theta(x|h^{(s)})$$

# Pitfalls of Naive Monte Carlo

**Using the Estimate in Log-Likelihood:**

- We would approximate the log-likelihood as:

$$\mathcal{L}(\theta|X) \approx \sum_{i=1}^{N} \log\left(\frac{1}{S}\sum_{s=1}^{S} p_\theta(x_i|h^{(s,i)})\right)$$

# Pitfalls of Naive Monte Carlo

**Using the Estimate in Log-Likelihood:**

- We would approximate the log-likelihood as:

$$\mathcal{L}(\theta|X) \approx \sum_{i=1}^{N} \log\left(\frac{1}{S}\sum_{s=1}^{S} p_\theta(x_i|h^{(s,i)})\right)$$

**The "Mismatch" Problem:**

- For a specific data point $x_i$, most samples $h$ drawn from the general prior $p_\theta(h)$ might result in a tiny $p_\theta(x_i|h)$.
- This means many $h$ samples are "wasted" as they don't meaningfully contribute to explaining $x_i$.

# Pitfalls of Naive Monte Carlo

**Using the Estimate in Log-Likelihood:**

- We would approximate the log-likelihood as:

$$\mathcal{L}(\theta|X) \approx \sum_{i=1}^{N} \log \left( \frac{1}{S} \sum_{s=1}^{S} p_\theta(x_i|h^{(s,i)}) \right)$$

**The "Mismatch" Problem:**

- For a specific data point $x_i$, most samples $h$ drawn from the general prior $p_\theta(h)$ might result in a tiny $p_\theta(x_i|h)$.
- This means many $h$ samples are "wasted" as they don't meaningfully contribute to explaining $x_i$.

**Consequence: High Variance & Inefficiency**

- The estimate $\hat{p}_\theta(x_i)$ can be very unreliable (high variance) if only a few $h$ samples "hit" the relevant region for $x_i$.
- To avoid "missing" $x_i$ and get a stable estimate, a very large $S$ is often needed, making this naive approach inefficient.

# The Quest for Better Latent Samples

- **Idea:** What if we could draw $h$ samples from a distribution that already "knows" about $x_i$?
- Such samples would be inherently more relevant.

# The Posterior $p_\theta(h|x)$: Relevant Samples

- Consider the **posterior distribution**: $p_\theta(h|x)$.
- This distribution answers: "Given that I've observed $x$, what are the most probable latent variables $h$ that could have generated it?"

# The Posterior $p_\theta(h|x)$: Relevant Samples

- Consider the **posterior distribution**: $p_\theta(h|x)$.
- This distribution answers: ”Given that I’ve observed $x$, what are the most probable latent variables $h$ that could have generated it?”
- Samples $h \sim p_\theta(h|x)$ would, by definition, be concentrated in regions of the latent space that are highly relevant to the specific data point $x$.
- This seems like the perfect source for ”good” $h$ samples!

# The Posterior Problem: A Catch-22

- A natural way to find the posterior $p_\theta(h|x)$ is by using **Bayes' Theorem**.

# The Posterior Problem: A Catch-22

- A natural way to find the posterior $p_\theta(h|x)$ is by using **Bayes' Theorem**.

## Bayes' Theorem

$$p_\theta(h|x) = \frac{p_\theta(x|h)p_\theta(h)}{p_\theta(x)}$$

# The Posterior Problem: A Catch-22

- A natural way to find the posterior $p_\theta(h|x)$ is by using **Bayes' Theorem**.

## Bayes' Theorem

$$p_\theta(h|x) = \frac{p_\theta(x|h)p_\theta(h)}{p_\theta(x)}$$

**The "Chicken and Egg" Problem**

- The denominator is the marginal likelihood of the data, $p_\theta(x)$.
- This is the exact quantity we found to be intractable:

$$p_\theta(x) = \int p_\theta(x|h)p_\theta(h)dh$$

- We need $p_\theta(x)$ to find the true posterior, but our original motivation to find the posterior was to help us deal with the intractability of $p_\theta(x)$. We are stuck in a loop.

# The Way Out: Approximating the Posterior

**Recap of our Impasse:**

- The true posterior for a specific data point, $p_\theta(h|x_i)$, is the ideal distribution to sample from, but it is intractable to compute.

# The Way Out: Approximating the Posterior

**Recap of our Impasse:**

- The true posterior for a specific data point, $p_\theta(h|x_i)$, is the ideal distribution to sample from, but it is intractable to compute.

**The Core Idea: Variational Inference (VI)**

- Instead of computing the true posterior, we find an **approximate distribution** for it.
- We define a family of simpler, tractable distributions, called the variational family $\mathcal{Q}$.

# The Way Out: Approximating the Posterior

**Recap of our Impasse:**

- The true posterior for a specific data point, $p_\theta(h|x_i)$, is the ideal distribution to sample from, but it is intractable to compute.

**The Core Idea: Variational Inference (VI)**

- Instead of computing the true posterior, we find an **approximate distribution** for it.
- We define a family of simpler, tractable distributions, called the variational family $\mathcal{Q}$.
- We then select a distribution $q_{\lambda_i}(h|x_i)$ from this family, controlled by its own parameters $\lambda_i$ for each data point $x_i$.

# The Way Out: Approximating the Posterior

**Recap of our Impasse:**

- The true posterior for a specific data point, $p_\theta(h|x_i)$, is the ideal distribution to sample from, but it is intractable to compute.

**The Core Idea: Variational Inference (VI)**

- Instead of computing the true posterior, we find an **approximate distribution** for it.
- We define a family of simpler, tractable distributions, called the variational family $\mathcal{Q}$.
- We then select a distribution $q_{\lambda_i}(h|x_i)$ from this family, controlled by its own parameters $\lambda_i$ for each data point $x_i$.
- The goal is to make our approximation $q_{\lambda_i}(h|x_i)$ as "close" as possible to the true posterior $p_\theta(h|x_i)$.

# The Variational Objective: Minimizing Divergence

**How do we measure ”closeness”?**

- We use a divergence measure to quantify the similarity between two distributions.
- A common choice is the reverse Kullback-Leibler (KL) divergence.

# The Variational Objective: Minimizing Divergence

**How do we measure "closeness"?**

- We use a divergence measure to quantify the similarity between two distributions.
- A common choice is the reverse Kullback-Leibler (KL) divergence.

**The Optimization Problem (for a single $x_i$):**

- We seek the variational parameters $\lambda_i^*$ that **minimize** the KL divergence between our approximation and the true posterior:

$$\lambda_i^* = \arg\min_{\lambda_i} D_{KL}\left(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i)\right)$$

# The Variational Objective: Minimizing Divergence

**How do we measure ”closeness”?**

- We use a divergence measure to quantify the similarity between two distributions.
- A common choice is the reverse Kullback-Leibler (KL) divergence.

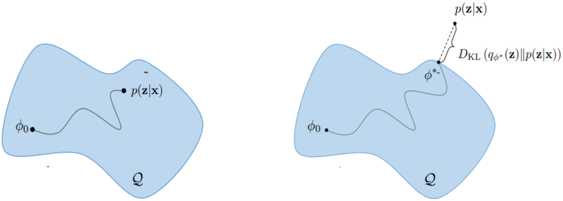
**The Optimization Problem (for a single $x_i$):**

- We seek the variational parameters $\lambda_i^*$ that **minimize** the KL divergence between our approximation and the true posterior:

$$\lambda_i^* = \arg\min_{\lambda_i} D_{KL}\left(q_{\lambda_i}(h|x_i)\|p_\theta(h|x_i)\right)$$

- **Important Note:** The approximation will be inherently biased unless the true posterior $p_\theta(h|x_i)$ happens to be a member of our simpler variational family $\mathcal{Q}$.

# Visualizing the Approximation


$\phi_0$
$p(\mathbf{z}|\mathbf{x})$
$\mathcal{Q}$
$p(\mathbf{z}|\mathbf{x})$
$D_{\mathrm{KL}}\left(q_{\phi^*}(\mathbf{z})\|p(\mathbf{z}|\mathbf{x})\right)$
$\phi^*$
$\phi_0$
$\mathcal{Q}$

(a) $p(\mathbf{z}|\mathbf{x}) \in \mathcal{Q}$

(b) $p(\mathbf{z}|\mathbf{x}) \notin \mathcal{Q}$

Figure: If the true posterior (black line) for a given $x_i$ is in the variational family $\mathcal{Q}$ (blue area), VI can find it (a). If not, it finds the closest approximation (b).

# From KL Divergence to a Tractable Objective

**The Challenge:**

- Our objective, $D_{KL}(q_{\lambda_i}(h|x_i)\|p_\theta(h|x_i))$, depends on the unknown posterior for $x_i$ and can't be optimized directly.

# From KL Divergence to a Tractable Objective

**The Challenge:**

- Our objective, $D_{KL}(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i))$, depends on the unknown posterior for $x_i$ and can't be optimized directly.

**The Solution:**

- We derive an alternative, *tractable* objective function called the **Evidence Lower Bound (ELBO)**.

# ELBO Derivation (1): The Setup

We start with the log-likelihood of our data point $x_i$ and introduce our approximate posterior $q_{\lambda_i}(h|x_i)$:

$$\log p_\theta(x_i) = \log \int p_\theta(x_i, h) dh$$

# ELBO Derivation (1): The Setup

We start with the log-likelihood of our data point $x_i$ and introduce our approximate posterior $q_{\lambda_i}(h|x_i)$:

$$\begin{aligned}\log p_\theta(x_i) &= \log \int p_\theta(x_i, h) dh \\ &= \log \int p_\theta(x_i, h) \frac{q_{\lambda_i}(h|x_i)}{q_{\lambda_i}(h|x_i)} dh\end{aligned}$$

# ELBO Derivation (1): The Setup

We start with the log-likelihood of our data point $x_i$ and introduce our approximate posterior $q_{\lambda_i}(h|x_i)$:

$$\begin{aligned}
\log p_\theta(x_i) &= \log \int p_\theta(x_i, h) dh \\
&= \log \int p_\theta(x_i, h) \frac{q_{\lambda_i}(h|x_i)}{q_{\lambda_i}(h|x_i)} dh \\
&= \log \left( \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] \right)
\end{aligned}$$

# ELBO Derivation (2): Jensen's Inequality

**From our previous expression for $x_i$:**

$$\log p_\theta(x_i) = \log \left( \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] \right)$$

# ELBO Derivation (2): Jensen's Inequality

**From our previous expression for $x_i$:**

$$\log p_\theta(x_i) = \log \left( \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] \right)$$

Since log is a concave function, we apply Jensen's Inequality ($\log(\mathbb{E}[Y]) \geq \mathbb{E}[\log Y]$) to get a lower bound:

$$\log p_\theta(x_i) \geq \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \log \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] := \mathcal{L}(\lambda_i, \theta)$$

# ELBO Derivation (2): Jensen's Inequality

**From our previous expression for $x_i$:**

$$\log p_\theta(x_i) = \log \left( \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] \right)$$

Since log is a concave function, we apply Jensen's Inequality ($\log(\mathbb{E}[Y]) \geq \mathbb{E}[\log Y]$) to get a lower bound:

$$\log p_\theta(x_i) \geq \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \log \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right] := \mathcal{L}(\lambda_i, \theta)$$

This lower bound, $\mathcal{L}(\lambda_i, \theta)$, is our Evidence Lower Bound (ELBO).

# Deriving the ELBO-KL Relationship (1/3)

Recall the definition of the Evidence Lower Bound (ELBO) for a data point $x_i$:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}(h|x_i)}\left[\log \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)}\right]$$

# Deriving the ELBO-KL Relationship (1/3)

Recall the definition of the Evidence Lower Bound (ELBO) for a data point $x_i$:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \log \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)} \right]$$

First, using the property of logarithms, we can split the fraction:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}(h|x_i)} \left[ \log p_\theta(x_i, h) - \log q_{\lambda_i}(h|x_i) \right]$$

# Deriving the ELBO-KL Relationship (1/3)

Recall the definition of the Evidence Lower Bound (ELBO) for a data point $x_i$:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}(h|x_i)}\left[\log \frac{p_\theta(x_i, h)}{q_{\lambda_i}(h|x_i)}\right]$$

First, using the property of logarithms, we can split the fraction:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}(h|x_i)}\left[\log p_\theta(x_i, h) - \log q_{\lambda_i}(h|x_i)\right]$$

Next, we can distribute the expectation across the two terms:

$$\mathcal{L}(\lambda_i, \theta) = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(x_i, h)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (2/3)

From our last step, we had:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(x_i, h)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (2/3)

From our last step, we had:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(x_i, h)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

Now, let's focus on the first term. We can rewrite the joint probability $p_\theta(x_i, h)$ using the chain rule of probability (i.e., Bayes' rule):

$$p_\theta(x_i, h) = p_\theta(h|x_i) p_\theta(x_i)$$

# Deriving the ELBO-KL Relationship (2/3)

From our last step, we had:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(x_i, h)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

Now, let's focus on the first term. We can rewrite the joint probability $p_\theta(x_i, h)$ using the chain rule of probability (i.e., Bayes' rule):

$$p_\theta(x_i, h) = p_\theta(h|x_i)p_\theta(x_i)$$

Substituting this back into our equation for the ELBO:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log(p_\theta(h|x_i)p_\theta(x_i))] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (2/3)

From our last step, we had:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(x_i, h)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

Now, let's focus on the first term. We can rewrite the joint probability $p_\theta(x_i, h)$ using the chain rule of probability (i.e., Bayes' rule):

$$p_\theta(x_i, h) = p_\theta(h|x_i) p_\theta(x_i)$$

Substituting this back into our equation for the ELBO:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log(p_\theta(h|x_i) p_\theta(x_i))] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

And applying the logarithm property one more time:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i) + \log p_\theta(x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (3/3)

Our expression for the ELBO is now:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i) + \log p_\theta(x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (3/3)

Our expression for the ELBO is now:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i) + \log p_\theta(x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

The term $\log p_\theta(x_i)$ is a constant with respect to the expectation over $h$, so we can pull it out:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i)] + \log p_\theta(x_i) - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

# Deriving the ELBO-KL Relationship (3/3)

Our expression for the ELBO is now:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i) + \log p_\theta(x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

The term $\log p_\theta(x_i)$ is a constant with respect to the expectation over $h$, so we can pull it out:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i)] + \log p_\theta(x_i) - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

Now, let's rearrange the terms to isolate $\log p_\theta(x_i)$:

$$\mathcal{L} = \log p_\theta(x_i) - \left(\mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i)]\right)$$

# Deriving the ELBO-KL Relationship (3/3)

Our expression for the ELBO is now:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i) + \log p_\theta(x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

The term $\log p_\theta(x_i)$ is a constant with respect to the expectation over $h$, so we can pull it out:

$$\mathcal{L} = \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i)] + \log p_\theta(x_i) - \mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)]$$

Now, let's rearrange the terms to isolate $\log p_\theta(x_i)$:

$$\mathcal{L} = \log p_\theta(x_i) - \left(\mathbb{E}_{q_{\lambda_i}}[\log q_{\lambda_i}(h|x_i)] - \mathbb{E}_{q_{\lambda_i}}[\log p_\theta(h|x_i)]\right)$$

We can recognize the expression in the parentheses as the definition of the KL divergence, $D_{KL}(q_{\lambda_i} \| p_\theta)$:

$$\mathcal{L}(\lambda_i, \theta) = \log p_\theta(x_i) - D_{KL}\left(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i)\right)$$

# Why Maximizing the ELBO Works

Our new derivation gives us the fundamental relationship:

$$\log p_\theta(x_i) = \mathcal{L}(\lambda_i, \theta) + D_{KL}\left(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i)\right)$$

# Why Maximizing the ELBO Works

Our new derivation gives us the fundamental relationship:

$$\log p_\theta(x_i) = \mathcal{L}(\lambda_i, \theta) + D_{KL}\left(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i)\right)$$

This tells us that the (intractable) true log-likelihood of our data can be decomposed into two parts:

- The (tractable) **Evidence Lower Bound (ELBO)**, $\mathcal{L}$.
- The (intractable) **KL Divergence**, $D_{KL}$, between our approximation $q_{\lambda_i}$ and the true posterior $p_\theta$.

# Why Maximizing the ELBO Works

Our new derivation gives us the fundamental relationship:

$$\log p_\theta(x_i) = \mathcal{L}(\lambda_i, \theta) + D_{KL}\left(q_{\lambda_i}(h|x_i) \| p_\theta(h|x_i)\right)$$

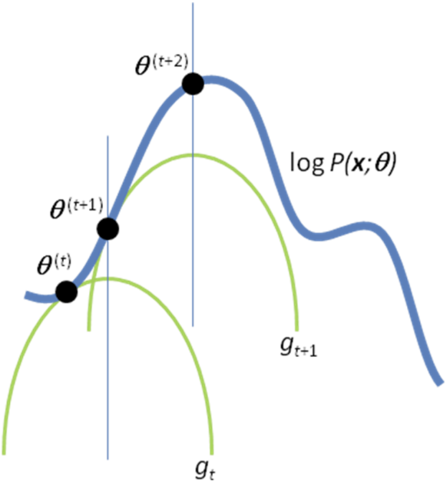
This tells us that the (intractable) true log-likelihood of our data can be decomposed into two parts:

- The (tractable) **Evidence Lower Bound (ELBO)**, $\mathcal{L}$.
- The (intractable) **KL Divergence**, $D_{KL}$, between our approximation $q_{\lambda_i}$ and the true posterior $p_\theta$.

**The Key Insight:**

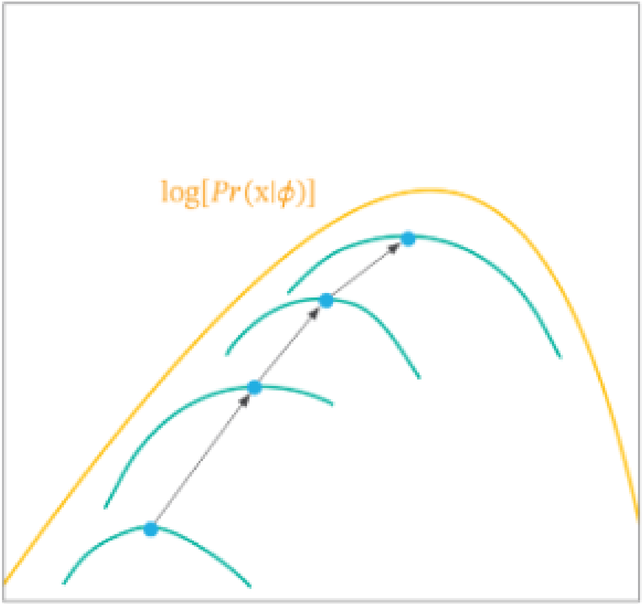
- For a given data point $x_i$, the term $\log p_\theta(x_i)$ is a fixed value.
- The KL Divergence is always non-negative ($D_{KL} \geq 0$).
- Therefore, increasing the ELBO **must** decrease the KL divergence. Maximizing the ELBO is equivalent to minimizing the gap between our approximation and the true posterior.

# EM VS VI



**Supplementary Figure 1** Convergence of the EM algorithm. Starting from initial parameters $\theta^{(t)}$, the E-step of the EM algorithm constructs a function $g_t$ that lower-bounds the objective function $\log P(x;\theta)$. In the M-step, $\theta^{(t+1)}$ is computed as the maximum of $g_t$. In the next E-step, a new lower-bound $g_{t+1}$ is constructed; maximization of $g_{t+1}$ in the next M-step gives $\theta^{(t+2)}$, etc.

# EM VS VI



# The Challenge of Scaling Variational Inference

In classical variational inference, this per-datapoint process is inefficient for large datasets.

- Each data point $x_i$ requires its own unique set of variational parameters, $\lambda_i$.

# The Challenge of Scaling Variational Inference

In classical variational inference, this per-datapoint process is inefficient for large datasets.

- Each data point $x_i$ requires its own unique set of variational parameters, $\lambda_i$.
- This means we must run a separate, iterative optimization loop for **every single data point** in our dataset to find its optimal $\lambda_i^*$.

# The Challenge of Scaling Variational Inference

In classical variational inference, this per-datapoint process is inefficient for large datasets.

- Each data point $x_i$ requires its own unique set of variational parameters, $\lambda_i$.
- This means we must run a separate, iterative optimization loop for **every single data point** in our dataset to find its optimal $\lambda_i^*$.
- This becomes computationally expensive and completely impractical to scale to datasets with millions of examples.

# The Solution: Amortized Inference

Amortized inference bypasses this bottleneck by introducing a single function that does the work for all data points.

- Instead of learning individual parameters $\lambda_i$, we learn a single, parameterized function, let's call it $f_\phi(x)$.
- This function takes a data point $x$ as input and predicts its optimal variational parameters.
- The parameters of this function, $\phi$, are shared across all data points. The cost of inference is thus "amortized".

# The Inference Network (Encoder)

This new function defines our approximate posterior.

- In a VAE, this function is a deep neural network called the **inference network** or **encoder**.
- We use the notation $q_\phi(h|x)$ to show that the approximate posterior for the latent variable $h$ is conditioned on the input data $x$ and depends on the shared parameters $\phi$.

# The Inference Network (Encoder)

This new function defines our approximate posterior.

- In a VAE, this function is a deep neural network called the **inference network** or **encoder**.
- We use the notation $q_\phi(h|x)$ to show that the approximate posterior for the latent variable $h$ is conditioned on the input data $x$ and depends on the shared parameters $\phi$.
- **Example:** For a Gaussian posterior, the encoder takes $x$ and outputs a mean $\mu(x)$ and a variance $\sigma^2(x)$.

$$q_\phi(h_i|x_i) = \mathcal{N}(\mu(x_i), \mathrm{diag}(\sigma^2(x_i)))$$

# The Other Half: The Generative Network (Decoder)

A VAE also employs a second neural network to model the generative process.

- This is the **generative network** or **decoder**, which defines the conditional probability $p_\theta(x|h)$.
- It is controlled by a separate set of parameters, $\theta$.

# The Full Objective Function

With both an encoder and a decoder, the ELBO now depends on both sets of parameters, $\phi$ and $\theta$:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(h|x)} \left[ \log \frac{p_\theta(x, h)}{q_\phi(h|x)} \right]$$

# The Full Objective Function

With both an encoder and a decoder, the ELBO now depends on both sets of parameters, $\phi$ and $\theta$:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(h|x)} \left[ \log \frac{p_\theta(x, h)}{q_\phi(h|x)} \right]$$

Instead of optimizing for each data point, we now simultaneously optimize the global parameters $\phi$ and $\theta$ for the entire dataset using gradient-based methods.

# A More Intuitive Form of the ELBO

Before seeing the final loss, let's rearrange the ELBO into its most common and intuitive form. This will clarify the two competing goals we are optimizing.

We start with the ELBO definition, now including the encoder/decoder parameters:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(h|x)}\left[\log \frac{p_\theta(x, h)}{q_\phi(h|x)}\right]$$

# A More Intuitive Form of the ELBO

Before seeing the final loss, let's rearrange the ELBO into its most common and intuitive form. This will clarify the two competing goals we are optimizing.
We start with the ELBO definition, now including the encoder/decoder parameters:

$$\mathcal{L}(\phi, \theta) = \mathbb{E}_{q_\phi(h|x)}\left[\log \frac{p_\theta(x, h)}{q_\phi(h|x)}\right]$$

We expand the joint probability $p_\theta(x, h)$ into $p_\theta(x|h)p(h)$:

$$= \mathbb{E}_{q_\phi(h|x)}\left[\log \frac{p_\theta(x|h)p(h)}{q_\phi(h|x)}\right]$$

# Deriving the Practical Objective

Using the properties of logarithms, we can split the expression for a single data point $x_i$:

$$\mathcal{L}(\phi, \theta; x_i) = \mathbb{E}_{q_\phi(h|x_i)} \left[ \log p_\theta(x_i|h) + \log \frac{p(h)}{q_\phi(h|x_i)} \right]$$

# Deriving the Practical Objective

Using the properties of logarithms, we can split the expression for a single data point $x_i$:

$$\begin{aligned}\mathcal{L}(\phi, \theta; x_i) &= \mathbb{E}_{q_\phi(h|x_i)} \left[\log p_\theta(x_i|h) + \log \frac{p(h)}{q_\phi(h|x_i)}\right] \\ &= \mathbb{E}_{q_\phi(h|x_i)}[\log p_\theta(x_i|h)] + \mathbb{E}_{q_\phi(h|x_i)} \left[\log \frac{p(h)}{q_\phi(h|x_i)}\right]\end{aligned}$$

# Deriving the Practical Objective

Using the properties of logarithms, we can split the expression for a single data point $x_i$:

$$
\begin{aligned}
\mathcal{L}(\phi, \theta; x_i) &= \mathbb{E}_{q_\phi(h|x_i)} \left[ \log p_\theta(x_i|h) + \log \frac{p(h)}{q_\phi(h|x_i)} \right] \\
&= \mathbb{E}_{q_\phi(h|x_i)}[\log p_\theta(x_i|h)] + \mathbb{E}_{q_\phi(h|x_i)} \left[ \log \frac{p(h)}{q_\phi(h|x_i)} \right] \\
&= \mathbb{E}_{q_\phi(h|x_i)}[\log p_\theta(x_i|h)] - D_{KL} \left( q_\phi(h|x_i) \| p(h) \right)
\end{aligned}
$$

# Interpreting the Practical Objective

This new form reveals a fundamental trade-off:

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]}_{\text{Reconstruction Fidelity}} - \underbrace{D_{KL}\left(q_\phi(h|x) \| p(h)\right)}_{\text{Regularization}}$$

# Interpreting the Practical Objective

This new form reveals a fundamental trade-off:

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]}_{\text{Reconstruction Fidelity}} - \underbrace{D_{KL}\left(q_\phi(h|x)\|p(h)\right)}_{\text{Regularization}}$$

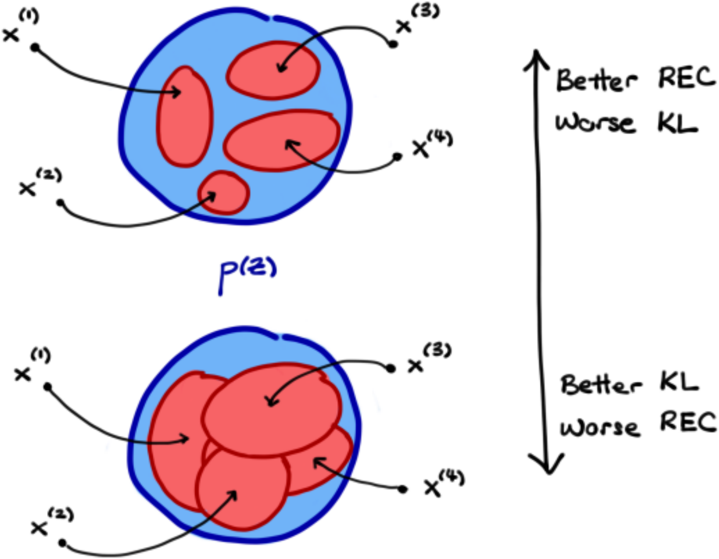
- **Reconstruction Fidelity:** Pushes the model to learn latent codes $h$ that accurately reconstruct the original data $x$. This acts like a reconstruction loss.

# Interpreting the Practical Objective

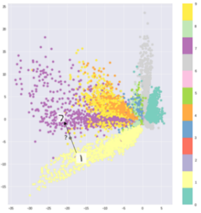
This new form reveals a fundamental trade-off:

$$\mathcal{L}_{\text{ELBO}} = \underbrace{\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]}_{\text{Reconstruction Fidelity}} - \underbrace{D_{KL}\left(q_\phi(h|x) \| p(h)\right)}_{\text{Regularization}}$$

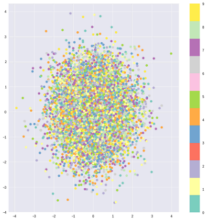
- **Reconstruction Fidelity:** Pushes the model to learn latent codes $h$ that accurately reconstruct the original data $x$. This acts like a reconstruction loss.
- **Regularization:** Pushes the approximate posterior $q_\phi(h|x)$ to be close to the simple prior $p(h)$. This organizes the latent space and is crucial for generation.


$x^{(1)}$
$x^{(3)}$
$x^{(4)}$
$x^{(2)}$
$p(z)$
Better REC
Worse KL
Better KL
Worse REC

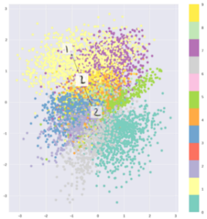
Only reconstruction loss



Only KL divergence



Combination



# VISUALIZATION

https://xnought.github.io/vae-explainer/

# A Deeper Question: What is the VAE Objective Really Optimizing?

**Going Beyond the Intuition**

- While "Reconstruction vs. Regularization" is a helpful intuition, it raises deeper questions:
  - How much "information" is the latent code $h$ truly capturing about the input $x$?
  - Is there a more rigorous way to understand this trade-off?

# A Deeper Question: What is the VAE Objective Really Optimizing?

**Going Beyond the Intuition**

- While "Reconstruction vs. Regularization" is a helpful intuition, it raises deeper questions:
  - How much "information" is the latent code $h$ truly capturing about the input $x$?
  - Is there a more rigorous way to understand this trade-off?

**A New Perspective: VAEs as Information Compressors**

- To answer these questions, we can turn to the tools of **Information Theory**.

# A Deeper Question: What is the VAE Objective Really Optimizing?

**Going Beyond the Intuition**

- While "Reconstruction vs. Regularization" is a helpful intuition, it raises deeper questions:
  - How much "information" is the latent code $h$ truly capturing about the input $x$?
  - Is there a more rigorous way to understand this trade-off?

**A New Perspective: VAEs as Information Compressors**

- To answer these questions, we can turn to the tools of **Information Theory**.
- This allows us to re-frame the VAE as a system that performs **lossy compression** on data, providing a precise mathematical language to describe the trade-off.

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**The Foundational Idea**

- Rate-Distortion (R-D) theory is the mathematical framework for **lossy compression**.

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**The Foundational Idea**

- Rate-Distortion (R-D) theory is the mathematical framework for **lossy compression**.

**The Inescapable Trade-off**

- To compress the signal (i.e., to use fewer bits), we must discard some information. This inevitably introduces errors or **Distortion (D)**.

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

### The Foundational Idea

- Rate-Distortion (R-D) theory is the mathematical framework for **lossy compression**.

### The Inescapable Trade-off

- To compress the signal (i.e., to use fewer bits), we must discard some information. This inevitably introduces errors or **Distortion (D)**.
- The number of bits needed to store the compressed code is the **Rate (R)**.

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**The Foundational Idea**

- Rate-Distortion (R-D) theory is the mathematical framework for **lossy compression**.

**The Inescapable Trade-off**

- To compress the signal (i.e., to use fewer bits), we must discard some information. This inevitably introduces errors or **Distortion (D)**.
- The number of bits needed to store the compressed code is the **Rate (R)**.
- **Core Principle:** There is a fundamental, inverse relationship between Rate and Distortion. To achieve a lower Rate (more compression), one must tolerate higher Distortion.

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**Formalizing the Terms**

- **Distortion (D):** The expected ”cost” or error of representing the original signal $X$ with its compressed version $Z$.

$$D = \mathbb{E}[d(X, Z)]$$

(where $d(\cdot, \cdot)$ is a metric like squared error)

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**Formalizing the Terms**

- **Distortion (D):** The expected "cost" or error of representing the original signal $X$ with its compressed version $Z$.

  $$D = \mathbb{E}[d(X, Z)]$$

  (where $d(\cdot, \cdot)$ is a metric like squared error)

- **Rate (R):** The mutual information between $X$ and $Z$. This measures how much information the code $Z$ contains about the original signal $X$.

  $$R = I(X; Z)$$

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**Navigating the Trade-off**

- To find an optimal balance, we can minimize a single Lagrangian objective, controlled by a hyperparameter $\beta$:

$$\min \left( D + \beta \cdot R \right)$$

# A Quick Refresher: The Core Idea of Rate-Distortion (R-D) Theory

**Navigating the Trade-off**

- To find an optimal balance, we can minimize a single Lagrangian objective, controlled by a hyperparameter $\beta$:

$$\min(D + \beta \cdot R)$$

- The parameter $\beta$ explicitly sets our priority:
  - **Low** $\beta$: Prioritizes minimizing Distortion (fidelity is cheap).
  - **High** $\beta$: Prioritizes minimizing Rate (compression is critical).

# The VAE Objective Through the Lens of R-D Theory : The Distortion Term

**Step 1: Recall R-D Distortion**

- In R-D theory, Distortion $D$ measures the average error in reconstructing the source $X$ from the compressed code $Z$.

# The VAE Objective Through the Lens of R-D Theory : The Distortion Term

**Step 1: Recall R-D Distortion**

- In R-D theory, Distortion $D$ measures the average error in reconstructing the source $X$ from the compressed code $Z$.

**Step 2: Examine the VAE's Reconstruction Term**

- The first term in our ELBO objective is the expected log-likelihood of the data given the latent code:

$$\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]$$

# The VAE Objective Through the Lens of R-D Theory : The Distortion Term

**Step 1: Recall R-D Distortion**

- In R-D theory, Distortion $D$ measures the average error in reconstructing the source $X$ from the compressed code $Z$.

**Step 2: Examine the VAE's Reconstruction Term**

- The first term in our ELBO objective is the expected log-likelihood of the data given the latent code:

$$\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]$$

- Maximizing this term forces the decoder to generate a faithful reconstruction of the input $x$.
- Therefore, the **negative** of this term can be interpreted as a measure of reconstruction error.

# The VAE Objective Through the Lens of R-D Theory : The Distortion Term

**Step 1: Recall R-D Distortion**

- In R-D theory, Distortion $D$ measures the average error in reconstructing the source $X$ from the compressed code $Z$.

**Step 2: Examine the VAE's Reconstruction Term**

- The first term in our ELBO objective is the expected log-likelihood of the data given the latent code:

$$\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)]$$

- Maximizing this term forces the decoder to generate a faithful reconstruction of the input $x$.
- Therefore, the **negative** of this term can be interpreted as a measure of reconstruction error.

## The VAE Distortion Term

The Distortion $D$ in a VAE is its reconstruction error, represented by the negative log-likelihood

# The VAE Objective Through the Lens of R-D Theory : The Rate Term

**Step 1: Recall the R-D Rate**

- In R-D theory, the Rate $R$ is the mutual information $I(X;Z)$, measuring the information capacity of the compressed code.

# The VAE Objective Through the Lens of R-D Theory : The Rate Term

**Step 1: Recall the R-D Rate**

- In R-D theory, the Rate $R$ is the mutual information $I(X;Z)$, measuring the information capacity of the compressed code.

**Step 2: The VAE's Regularization Term**

- The second term in our ELBO is the KL divergence:

$$D_{KL}\left(q_\phi(h|x)\|p(h)\right)$$

# The VAE Objective Through the Lens of R-D Theory : The Rate Term

**Step 1: Recall the R-D Rate**

- In R-D theory, the Rate $R$ is the mutual information $I(X; Z)$, measuring the information capacity of the compressed code.

**Step 2: The VAE's Regularization Term**

- The second term in our ELBO is the KL divergence:

$$D_{KL}\left(q_\phi(h|x) \| p(h)\right)$$

- A key result from information theory is that this KL term forms a **variational upper bound** on the mutual information between the input $x$ and the latent code $h$:

$$I(x; h) \leq D_{KL}\left(q_\phi(h|x) \| p(h)\right)$$

# The VAE Objective Through the Lens of R-D Theory: The Rate Term

### Making the Final Connection

- Since the KL divergence is a tractable upper bound on the mutual information, we use it as our practical proxy for the Rate.

# The VAE Objective Through the Lens of R-D Theory: The Rate Term

### Making the Final Connection

- Since the KL divergence is a tractable upper bound on the mutual information, we use it as our practical proxy for the Rate.

### The VAE Rate Term

The Rate $R$ in a VAE is effectively defined by the KL divergence. Minimizing this term corresponds to minimizing the information Rate (i.e., compressing the data).

$$R_{\text{VAE}} = D_{KL}\left(q_\phi(h|x) \| p(h)\right)$$

# Revisiting the -VAE: A Principled Lagrangian Formulation

**Let's Re-examine the VAE Objective**

- Our goal is to **maximize** the ELBO:

$$\max_{\phi,\theta} \left( \mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)] - D_{KL}\left(q_\phi(h|x) \| p(h)\right) \right)$$

# Revisiting the -VAE: A Principled Lagrangian Formulation

## Let's Re-examine the VAE Objective

- Our goal is to **maximize** the ELBO:

$$\max_{\phi,\theta} \left( \mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)] - D_{KL}\left(q_\phi(h|x) \| p(h)\right) \right)$$

- Maximizing a value is identical to **minimizing** its negative. Let's flip the signs:

$$\min_{\phi,\theta} \left( -\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)] + D_{KL}\left(q_\phi(h|x) \| p(h)\right) \right)$$

# Revisiting the -VAE: A Principled Lagrangian Formulation

**Let's Re-examine the VAE Objective**

- Our goal is to **maximize** the ELBO:

$$\max_{\phi,\theta}\left(\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)] - D_{KL}\left(q_\phi(h|x)\|p(h)\right)\right)$$

- Maximizing a value is identical to **minimizing** its negative. Let's flip the signs:

$$\min_{\phi,\theta}\left(-\mathbb{E}_{q_\phi(h|x)}[\log p_\theta(x|h)] + D_{KL}\left(q_\phi(h|x)\|p(h)\right)\right)$$

- Now, let's substitute the R-D terms we just defined ($D_{\text{VAE}}$ and $R_{\text{VAE}}$):

A standard VAE is equivalent to solving:

$$\min_{\phi,\theta}\left(D_{\text{VAE}} + R_{\text{VAE}}\right)$$

# Revisiting the -VAE: A Principled Lagrangian Formulation

## The Insight

- The standard VAE objective is an R-D problem with a **fixed trade-off** where $\beta = 1$.
- What if we want to explicitly control this trade-off, just like in formal R-D theory?

# Revisiting the -VAE: A Principled Lagrangian Formulation

**The Insight**

- The standard VAE objective is an R-D problem with a **fixed trade-off** where $\beta = 1$.
- What if we want to explicitly control this trade-off, just like in formal R-D theory?

**The $\beta$-VAE Objective**

- We introduce the hyperparameter $\beta$ as the Lagrange multiplier. This gives us the $\beta$-VAE objective function:

$$\mathcal{L}_{\beta-VAE} = \min_{\phi,\theta} \left( D_{\text{VAE}} + \beta \cdot R_{\text{VAE}} \right)$$

# Revisiting the -VAE: A Principled Lagrangian Formulation

### The Insight

- The standard VAE objective is an R-D problem with a **fixed trade-off** where $\beta = 1$.
- What if we want to explicitly control this trade-off, just like in formal R-D theory?

### The $\beta$-VAE Objective

- We introduce the hyperparameter $\beta$ as the Lagrange multiplier. This gives us the $\beta$-VAE objective function:

$$\mathcal{L}_{\beta-VAE} = \min_{\phi,\theta} \left( D_{\text{VAE}} + \beta \cdot R_{\text{VAE}} \right)$$

- This is no longer a heuristic modification. The $\beta$-VAE is a principled method for exploring the Rate-Distortion trade-off. By changing $\beta$, we choose how much we care about compression versus reconstruction fidelity.

# Visualizing the Trade-off: The Rate-Distortion (R-D) Plane

We can visualize the performance of a VAE on a 2D plot.

- **Y-axis: Distortion (D)**
  Reconstruction error. Lower is better.

- **X-axis: Rate (R)**
  Latent capacity / complexity. Lower means more compression.

# Visualizing the Trade-off: The Rate-Distortion (R-D) Plane

We can visualize the performance of a VAE on a 2D plot.

- **Y-axis: Distortion (D)**
  Reconstruction error. Lower is better.

- **X-axis: Rate (R)**
  Latent capacity / complexity. Lower means more compression.

The curve shows the **optimal frontier**. No model can perform better (lower D for a given R).

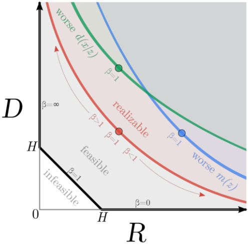
# Visualizing the Trade-off: The Rate-Distortion (R-D) Plane

We can visualize the performance of a VAE on a 2D plot.

- **Y-axis: Distortion (D)**
  Reconstruction error. Lower is better.

- **X-axis: Rate (R)**
  Latent capacity / complexity. Lower means more compression.

The curve shows the **optimal frontier**. No model can perform better (lower D for a given R). By tuning $\beta$ in the objective $\min(D + \beta R)$, we select different optimal points on this curve.



# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- $I(x; h)$ is the **Mutual Information**: How much information the latent code $h$ contains about the input $x$.

# Bounding the Mutual Information

It can be shown that the mutual information $I(x;h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x;h) \leq R$$

- $I(x;h)$ is the **Mutual Information**: How much information the latent code $h$ contains about the input $x$.
- $R$ is the **Rate**: Our familiar KL-divergence term, $D_{KL}(q_\phi(h|x) \| p(h))$.

# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- $I(x; h)$ is the **Mutual Information**: How much information the latent code $h$ contains about the input $x$.
- $R$ is the **Rate**: Our familiar KL-divergence term, $D_{KL}(q_\phi(h|x)\|p(h))$.
- $D$ is the **Distortion**: Our familiar reconstruction error, $-\mathbb{E}[\log p_\theta(x|h)]$.

# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- $I(x; h)$ is the **Mutual Information**: How much information the latent code $h$ contains about the input $x$.
- $R$ is the **Rate**: Our familiar KL-divergence term, $D_{KL}(q_\phi(h|x) \| p(h))$.
- $D$ is the **Distortion**: Our familiar reconstruction error, $-\mathbb{E}[\log p_\theta(x|h)]$.
- $\mathcal{H}(X)$ is the **Data Entropy**. This is a constant value representing the inherent complexity of the dataset itself.

# Bounding the Mutual Information

It can be shown that the mutual information $I(x; h)$ is formally "sandwiched" between two quantities related to our VAE objective:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- $I(x; h)$ is the **Mutual Information**: How much information the latent code $h$ contains about the input $x$.
- $R$ is the **Rate**: Our familiar KL-divergence term, $D_{KL}(q_\phi(h|x) \| p(h))$.
- $D$ is the **Distortion**: Our familiar reconstruction error, $-\mathbb{E}[\log p_\theta(x|h)]$.
- $\mathcal{H}(X)$ is the **Data Entropy**. This is a constant value representing the inherent complexity of the dataset itself.

# Exploring the R-D Frontier (1): The Auto-Encoding Limit

**Case: Perfect Reconstruction ($D \to 0$)**

- Let's analyze our inequality when we demand zero distortion:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

# Exploring the R-D Frontier (1): The Auto-Encoding Limit

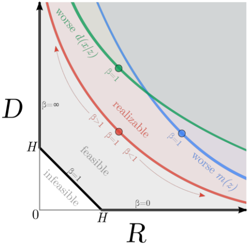
**Case: Perfect Reconstruction ($D \to 0$)**

- Let's analyze our inequality when we demand zero distortion:

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- If we set $D = 0$, the inequality becomes:

$$\mathcal{H}(X) \leq I(x; h) \leq R$$



# Exploring the R-D Frontier (1): The Auto-Encoding Limit

**Implication of Perfect Reconstruction**

From the relationship $\mathcal{H}(X) \leq I(x; h) \leq R$, we conclude:

- To achieve lossless reconstruction $(D = 0)$, the Rate $R$ must be at least as large as the data's entropy $\mathcal{H}(X)$.

# Exploring the R-D Frontier (1): The Auto-Encoding Limit

**Implication of Perfect Reconstruction**

From the relationship $\mathcal{H}(X) \leq I(x; h) \leq R$, we conclude:

- To achieve lossless reconstruction $(D = 0)$, the Rate $R$ must be at least as large as the data's entropy $\mathcal{H}(X)$.
- The latent channel must have enough capacity to carry all the information present in the original data.

# Exploring the R-D Frontier (1): The Auto-Encoding Limit

**Implication of Perfect Reconstruction**

From the relationship $\mathcal{H}(X) \leq I(x; h) \leq R$, we conclude:

- To achieve lossless reconstruction $(D = 0)$, the Rate $R$ must be at least as large as the data's entropy $\mathcal{H}(X)$.
- The latent channel must have enough capacity to carry all the information present in the original data.
- This defines the optimal point $(R = \mathcal{H}(X), D = 0)$ on the R-D plane.

# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Case: Maximum Compression ($R \to 0$)**

- Now let's analyze the inequality when we force the Rate to be zero (i.e., very high $\beta$):

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Case: Maximum Compression ($R \to 0$)**

- Now let's analyze the inequality when we force the Rate to be zero (i.e., very high $\beta$):

$$\mathcal{H}(X) - D \leq I(x;h) \leq R$$

- If we set $R = 0$, the inequality becomes:

$$\mathcal{H}(X) - D \leq I(x;h) \leq 0$$

# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Case: Maximum Compression ($R \to 0$)**

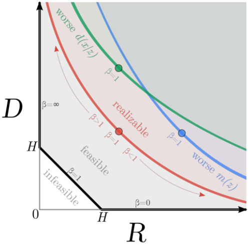
- Now let's analyze the inequality when we force the Rate to be zero (i.e., very high $\beta$):

$$\mathcal{H}(X) - D \leq I(x; h) \leq R$$

- If we set $R = 0$, the inequality becomes:

$$\mathcal{H}(X) - D \leq I(x; h) \leq 0$$

- Since mutual information cannot be negative, this forces $I(x; h) = 0$.



# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Implication of Maximum Compression**

We found that when $R = 0$, the mutual information $I(x; h)$ must be 0.

- Plugging $I(x; h) = 0$ into the left side of our main inequality, $\mathcal{H}(X) - D \leq I(x; h)$, gives us:

$$\mathcal{H}(X) - D \leq 0 \quad \Longrightarrow \quad D \geq \mathcal{H}(X)$$

# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Implication of Maximum Compression**
We found that when $R = 0$, the mutual information $I(x; h)$ must be 0.

- Plugging $I(x; h) = 0$ into the left side of our main inequality, $\mathcal{H}(X) - D \leq I(x; h)$, gives us:

$$\mathcal{H}(X) - D \leq 0 \quad \Longrightarrow \quad D \geq \mathcal{H}(X)$$

- The minimum possible distortion is the data entropy itself. This is achieved by a powerful decoder that ignores $h$ and just models the average data distribution.

# Exploring the R-D Frontier (2): The Auto-Decoding Limit

**Implication of Maximum Compression**

We found that when $R = 0$, the mutual information $I(x; h)$ must be 0.

- Plugging $I(x; h) = 0$ into the left side of our main inequality, $\mathcal{H}(X) - D \leq I(x; h)$, gives us:

$$\mathcal{H}(X) - D \leq 0 \quad \Longrightarrow \quad D \geq \mathcal{H}(X)$$

- The minimum possible distortion is the data entropy itself. This is achieved by a powerful decoder that ignores $h$ and just models the average data distribution.
- This defines the point $(R = 0, D = \mathcal{H}(X))$ on the R-D plane.

# The ”Posterior Collapse” Phenomenon

**What is Posterior Collapse?**

- When we force the Rate to zero ($R \to 0$) with a very high $\beta$, the objective is best satisfied by forcing the approximate posterior to match the prior:

$$q_\phi(h|x) \to p(h)$$

# The "Posterior Collapse" Phenomenon

**What is Posterior Collapse?**

- When we force the Rate to zero ($R \to 0$) with a very high $\beta$, the objective is best satisfied by forcing the approximate posterior to match the prior:

$$q_\phi(h|x) \to p(h)$$

- This phenomenon, where the posterior becomes **independent of the input** $x$, is known as **Posterior Collapse**.

# The "Posterior Collapse" Phenomenon

**What is Posterior Collapse?**

- When we force the Rate to zero ($R \to 0$) with a very high $\beta$, the objective is best satisfied by forcing the approximate posterior to match the prior:

$$q_\phi(h|x) \to p(h)$$

- This phenomenon, where the posterior becomes **independent of the input** $x$, is known as **Posterior Collapse**.
- The result is that the encoder is effectively ignored, and the latent code $h$ becomes uninformative.

# Deriving Distortion Under Collapse

**How does $D$ become $\mathcal{H}(X)$?**

- If the decoder $p_\theta(x|h)$ cannot use the uninformative $h$, its best strategy is to learn the marginal data distribution directly:

$$p_\theta(x|h) \approx p_{data}(x)$$

# Deriving Distortion Under Collapse

**How does $D$ become $\mathcal{H}(X)$?**

- If the decoder $p_\theta(x|h)$ cannot use the uninformative $h$, its best strategy is to learn the marginal data distribution directly:

$$p_\theta(x|h) \approx p_{data}(x)$$

- The formula for Distortion is:

$$D = \mathbb{E}_{x \sim p_{data}} \left[ \mathbb{E}_{h \sim q_\phi(h|x)} [-\log p_\theta(x|h)] \right]$$

# Deriving Distortion Under Collapse

**How does $D$ become $\mathcal{H}(X)$?**

- If the decoder $p_\theta(x|h)$ cannot use the uninformative $h$, its best strategy is to learn the marginal data distribution directly:
$$p_\theta(x|h) \approx p_{data}(x)$$
- The formula for Distortion is:
$$D = \mathbb{E}_{x\sim p_{data}}\left[\mathbb{E}_{h\sim q_\phi(h|x)}[-\log p_\theta(x|h)]\right]$$
- Since $p_\theta(x|h)$ no longer depends on $h$ and approximates $p_{data}(x)$, the expression simplifies to:
$$D \approx \mathbb{E}_{x\sim p_{data}}[-\log p_{data}(x)]$$

# Deriving Distortion Under Collapse

**How does $D$ become $\mathcal{H}(X)$?**

- If the decoder $p_\theta(x|h)$ cannot use the uninformative $h$, its best strategy is to learn the marginal data distribution directly:

$$p_\theta(x|h) \approx p_{data}(x)$$

- The formula for Distortion is:

$$D = \mathbb{E}_{x \sim p_{data}} \left[ \mathbb{E}_{h \sim q_\phi(h|x)} [-\log p_\theta(x|h)] \right]$$

- Since $p_\theta(x|h)$ no longer depends on $h$ and approximates $p_{data}(x)$, the expression simplifies to:

$$D \approx \mathbb{E}_{x \sim p_{data}} [-\log p_{data}(x)]$$

- By definition, this is the **entropy of the data**, $\mathcal{H}(X)$.

# The Crucial Link: Why Low Rate Encourages Disentanglement

**The Ultimate Goal: A Disentangled Representation**

- Our true goal is often not just to compress data, but to learn a **disentangled** representation.
- This means each latent dimension $h_j$ should correspond to a single, real-world factor of variation (e.g., object position, rotation, color).

# The Crucial Link: Why Low Rate Encourages Disentanglement

## The Ultimate Goal: A Disentangled Representation

- Our true goal is often not just to compress data, but to learn a **disentangled** representation.
- This means each latent dimension $h_j$ should correspond to a single, real-world factor of variation (e.g., object position, rotation, color).

## The Disentanglement Hypothesis

- For data generated from independent factors, the most **information-theoretically efficient** (i.e., most compressed) representation is one that is itself factorized or disentangled.

# The Crucial Link: Why Low Rate Encourages Disentanglement

**The Ultimate Goal: A Disentangled Representation**

- Our true goal is often not just to compress data, but to learn a **disentangled** representation.
- This means each latent dimension $h_j$ should correspond to a single, real-world factor of variation (e.g., object position, rotation, color).

**The Disentanglement Hypothesis**

- For data generated from independent factors, the most **information-theoretically efficient** (i.e., most compressed) representation is one that is itself factorized or disentangled.
- By forcing a low Rate $R$ with a high $\beta$, we create an "information bottleneck" that pressures the model to discover and encode these independent factors in the most compact way possible.

# The Limitation of -VAE: What Are We *Actually* Penalizing?

**Recap of the Problem**

- We've established that increasing $\beta$ encourages disentanglement at the cost of worse reconstruction.

# The Limitation of -VAE: What Are We *Actually* Penalizing?

### Recap of the Problem

- We've established that increasing $\beta$ encourages disentanglement at the cost of worse reconstruction.

### A Deeper Look at the Rate Term

- The Rate, $R = D_{KL}(q_\phi(h|x) \| p(h))$, appears to be a single, monolithic term.

# The Limitation of -VAE: What Are We *Actually* Penalizing?

### Recap of the Problem

- We've established that increasing $\beta$ encourages disentanglement at the cost of worse reconstruction.

### A Deeper Look at the Rate Term

- The Rate, $R = D_{KL}(q_\phi(h|x)\|p(h))$, appears to be a single, monolithic term.
- However, it's actually a composite of several distinct, meaningful information-theoretic quantities.

# The Limitation of -VAE: What Are We *Actually* Penalizing?

### Recap of the Problem

- We've established that increasing $\beta$ encourages disentanglement at the cost of worse reconstruction.

### A Deeper Look at the Rate Term

- The Rate, $R = D_{KL}(q_\phi(h|x) \| p(h))$, appears to be a single, monolithic term.
- However, it's actually a composite of several distinct, meaningful information-theoretic quantities.
- To understand the limitation of $\beta$-VAE, we need to **decompose** this KL divergence.

# Decomposing the Rate Term

It can be shown that the expected Rate term can be decomposed as follows:

$$\mathbb{E}_{p_{data}(x)}[R] = \mathbb{E}_{p_{data}(x)}\left[D_{KL}(q_\phi(h|x) \| p(h))\right]$$

# Decomposing the Rate Term

It can be shown that the expected Rate term can be decomposed as follows:

$$\begin{aligned}\mathbb{E}_{p_{data}(x)}[R] &= \mathbb{E}_{p_{data}(x)}\left[D_{KL}(q_\phi(h|x)\|p(h))\right] \\ &= \underbrace{I(x;h)}_{\text{Mutual Info}} + \underbrace{D_{KL}(q(h)\|p(h))}_{\text{Posterior Matching}}\end{aligned}$$

- **Mutual Information** $I(x;h)$: Measures how much $h$ tells us about $x$. Penalizing this directly hurts reconstruction.

# Decomposing the Rate Term

It can be shown that the expected Rate term can be decomposed as follows:

$$\begin{aligned}\mathbb{E}_{p_{data}(x)}[R] &= \mathbb{E}_{p_{data}(x)}\left[D_{KL}(q_\phi(h|x)\|p(h))\right] \\ &= \underbrace{I(x;h)}_{\text{Mutual Info}} + \underbrace{D_{KL}(q(h)\|p(h))}_{\text{Posterior Matching}}\end{aligned}$$

- **Mutual Information** $I(x;h)$: Measures how much $h$ tells us about $x$. Penalizing this directly hurts reconstruction.
- **Posterior Matching** $D_{KL}(q(h)\|p(h))$: Pushes the distribution of all our latent codes, known as the **aggregated posterior** $q(h) = \mathbb{E}_{p_{data}(x)}[q_\phi(h|x)]$, to match the simple prior $p(h)$.

# Decomposing the Rate Term

It can be shown that the expected Rate term can be decomposed as follows:

$$\begin{aligned}\mathbb{E}_{p_{data}(x)}[R] &= \mathbb{E}_{p_{data}(x)}\left[D_{KL}(q_\phi(h|x)\|p(h))\right] \\ &= \underbrace{I(x;h)}_{\text{Mutual Info}} + \underbrace{D_{KL}(q(h)\|p(h))}_{\text{Posterior Matching}}\end{aligned}$$

- **Mutual Information** $I(x;h)$: Measures how much $h$ tells us about $x$. Penalizing this directly hurts reconstruction.
- **Posterior Matching** $D_{KL}(q(h)\|p(h))$: Pushes the distribution of all our latent codes, known as the **aggregated posterior** $q(h) = \mathbb{E}_{p_{data}(x)}[q_\phi(h|x)]$, to match the simple prior $p(h)$.

## The Flaw

$\beta$-VAE penalizes both terms equally. It doesn't distinguish between reducing reconstruction-relevant information ($I(x;h)$) and structuring the overall latent space.

# Decomposing the Rate: The Full Picture

The "Posterior Matching" term can be further broken down. This reveals that the total expected Rate is composed of three distinct terms:

$$\mathbb{E}[R] = \underbrace{I(x;h)}_{\text{(1) Mutual Info}} + \underbrace{D_{KL}\left(q(h)\middle\|\prod_j q(h_j)\right)}_{\text{(2) Total Correlation}} + \underbrace{\sum_j D_{KL}\left(q(h_j)\|p(h_j)\right)}_{\text{(3) Dimension-wise KL}}$$

# Decomposing the Rate: Interpreting the Components

Here is the meaning of each term from the previous slide's equation:

**(1)** **Mutual Information:** $I(x; h)$

- Measures how much information the latent code $h$ retains about the input $x$. Penalizing it harms reconstruction.

# Decomposing the Rate: Interpreting the Components

Here is the meaning of each term from the previous slide's equation:

(1) **Mutual Information:** $I(x; h)$

- Measures how much information the latent code $h$ retains about the input $x$. Penalizing it harms reconstruction.

(2) **Total Correlation (TC):**

- Measures the statistical dependence between the latent dimensions. If the dimensions are independent, the TC is zero.

# Decomposing the Rate: Interpreting the Components

Here is the meaning of each term from the previous slide's equation:

(1) **Mutual Information:** $I(x; h)$

- Measures how much information the latent code $h$ retains about the input $x$. Penalizing it harms reconstruction.

(2) **Total Correlation (TC):**

- Measures the statistical dependence between the latent dimensions. If the dimensions are independent, the TC is zero.

(3) **Dimension-wise KL**:

- Encourages the distribution of each individual latent dimension to match the prior's marginal distribution.

# Decomposing the Rate: Interpreting the Components

Here is the meaning of each term from the previous slide's equation:

(1) **Mutual Information:** $I(x; h)$
- Measures how much information the latent code $h$ retains about the input $x$. Penalizing it harms reconstruction.

(2) **Total Correlation (TC):**
- Measures the statistical dependence between the latent dimensions. If the dimensions are independent, the TC is zero.

(3) **Dimension-wise KL**:
- Encourages the distribution of each individual latent dimension to match the prior's marginal distribution.

Minimizing the **Total Correlation (TC)** is the most direct way to enforce statistical independence. **This is the true objective for disentanglement.**

# The Path Forward: Towards Targeted Disentanglement

**The Flaw of $\beta$-VAE Revisited**

- We want to minimize the Total Correlation (TC) to get disentangled latents.

# The Path Forward: Towards Targeted Disentanglement

**The Flaw of $\beta$-VAE Revisited**

- We want to minimize the Total Correlation (TC) to get disentangled latents.
- $\beta$-VAE penalizes the entire Rate $R = I(x; h) + \text{TC} + \dots$.

# The Path Forward: Towards Targeted Disentanglement

**The Flaw of $\beta$-VAE Revisited**

- We want to minimize the Total Correlation (TC) to get disentangled latents.
- $\beta$-VAE penalizes the entire Rate $R = I(x; h) + \mathrm{TC} + \dots$.
- To apply pressure on TC, we must also apply the same pressure on the Mutual Information $I(x; h)$, which inevitably harms reconstruction quality. It's a blunt instrument.

# The Path Forward: Towards Targeted Disentanglement

### The Flaw of $\beta$-VAE Revisited

- We want to minimize the Total Correlation (TC) to get disentangled latents.
- $\beta$-VAE penalizes the entire Rate $R = I(x; h) + \mathrm{TC} + \dots$.
- To apply pressure on TC, we must also apply the same pressure on the Mutual Information $I(x; h)$, which inevitably harms reconstruction quality. It's a blunt instrument.

### The Solution: Isolate the Total Correlation

- The key insight is to create objective functions that can **isolate and directly regularize the Total Correlation term**, without excessively penalizing mutual information.

# The Path Forward: Towards Targeted Disentanglement

## The Flaw of $\beta$-VAE Revisited

- We want to minimize the Total Correlation (TC) to get disentangled latents.
- $\beta$-VAE penalizes the entire Rate $R = I(x; h) + \text{TC} + \dots$.
- To apply pressure on TC, we must also apply the same pressure on the Mutual Information $I(x; h)$, which inevitably harms reconstruction quality. It's a blunt instrument.

## The Solution: Isolate the Total Correlation

- The key insight is to create objective functions that can **isolate and directly regularize the Total Correlation term**, without excessively penalizing mutual information.
- This is the core idea behind more advanced models like **FactorVAE** and $\beta$-**TCVAE**, which introduce clever ways to approximate the TC term and penalize it specifically.