# Background

Introduce VAE, $\beta$-VAE, and mutual information preference as introduced by lossy VAE.

Rearraing we have the special primal forms for VAE and $\beta$-VAE (introduce consistency constraints), which leads us to the general primal form.

# General Primal Form with Consistency Constraints

List a range of existing methods that fall into this category. InfoGAN, BiGAN, InfoVAE, CycleGAN.

# Tractable Solutions to the Lagrangian Dual Problem

Problem: We use gradient descent methods in practice; consistency is never achieved, so $\lambda$ will always increase to $\infty$. We need tractable methods to learn lagrangian parameters!

## Relaxation of Constraints

Consider the following optimization problem, which is a general primal form with soft consistency constraints:

$$\begin{aligned} & \min_{\theta} -\alpha I_q(x; z) \\ \text{s.t.} \quad & \mathbb{E}_{x \sim q(x)}[\mathrm{KL}(q(x|z) \| p(x|z)))] \leq \epsilon_1 \\ & \mathbb{E}_{x \sim q(x), z \sim q(z|x)}[\mathrm{KL}(q(z) \| p(z)))] \leq \epsilon_2 \end{aligned}$$

This has the following dual form:

$$\min_{\theta} \max_{\lambda > 0} -\alpha I_q(x; z) + \lambda_1 (\mathbb{E}_{x \sim q(x)}[\mathrm{KL}(q(x|z) \| p(x|z)))] - \epsilon_1) + \lambda_2 (\mathbb{E}_{x \sim q(x), z \sim q(z|x)}[\mathrm{KL}(q(z) \| p(z)))] - \epsilon_2)$$

Intuitively, this objective will enforce "soft" consistency constraints as determined by $\epsilon_1$ and $\epsilon_2$. Moreover, if we use constant $\lambda_1$ and $\lambda_2$ then this is equivalent to the case without soft consistencies (since $\epsilon_1 \lambda_1 + \epsilon_2 \lambda_2$ is constant). Our goal (and **major contribution**) is to learn $\lambda_1$ and $\lambda_2$ **automatically**, as opposed to tuning for the best objective.

## Tractable Dual Form

However, $q(x|z) \propto q(x)q(z|x)$ is intractable, so it is not immediately obvious how the dual form can be tractable. By rearraging we have the following equivalent form:

$$\begin{aligned} \min_{\theta} \max_{\lambda > 0} & -\lambda_1 \mathbb{E}_{x \sim q(x), z \sim q(z|x)}[\log p(x|z)] \\ & + (\lambda_1 - \alpha) \mathbb{E}_{x \sim q(x)}[\mathrm{KL}(q(z|x) \| p(z)))] \\ & + (\lambda_2 - \lambda_1 + \alpha) \mathbb{E}_{x \sim q(x), z \sim q(z|x)}[\mathrm{KL}(q(z) \| p(z)))] \\ & - \lambda_1 \epsilon_1 - \lambda_2 \epsilon_2 \end{aligned}$$

Special cases:

- VAE: $\alpha = 0, \lambda_2 = \alpha - \lambda_1, \lambda_1 > 0$.
- $\beta$-VAE: $\lambda_1 > 0, \beta = (\lambda_1 - \alpha)/\lambda_1, \lambda_2 = \alpha - \lambda_1$.

Moreover, one could scale $\lambda_1, \lambda_2$ by $1/\alpha$, so we end up with three canonical forms, with $\alpha = 1, 0, -1$. These measures our objectives for mutual information between $x$ and $z$, corresponding to "maximizing", "no preference" and "minimizing" mutual information.

Unfortunately, $\nabla \mathrm{KL}(q(z) \| p(z))$ is intractable, so one could use Stein to obtain the optimal gradient direction:

$$\nabla_f \mathrm{KL}(q_{[T]} \| p)|_{f=0} = -\phi^\star_{q,p}(x)$$

where $\phi_{q,p}(\cdot) = \mathbb{E}_{x \sim q}[\nabla_x \log p(x) k(x, \cdot) + \nabla_x k(x, \cdot)]$. Empirically, we can also use MMD between the two distributions, or other two-sample tests.

## Practical Algorithm and Hyperparameter Selection

In practice, we still need to select two hyperparameters $\epsilon_1$ and $\epsilon_2$ — how is this better than tuning $\lambda_1$ and $\lambda_2$? There are two reasons:

1. The effect of $\epsilon_1$ and $\epsilon_2$ on the objective is much more interpretable — it imposes "softer" consistency constraints + "geometric interpretation"
2. We can find reasonable $\epsilon_1$ and $\epsilon_2$ values through one initial experiment. This allow us to automatically select the best hyperparameters for different network structures, since the ability to satisfy constraints depend on the capacity of the network.

# Experiments

MNIST - two architectures, compare VAE, $\beta$-VAE, InfoVAE and LagVAE.