Posterior Collapse

“Posterior Collapse”问题起源

- **2016 Generating Sentences from a Continuous Space**

$$
\begin{aligned}
\log p(\mathbf{x}) &= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}\mid\mathbf{x})] + D_{KL}(q(\mathbf{z}\mid\mathbf{x})||p(\mathbf{z}\mid\mathbf{x})) \\
&\geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x},\mathbf{z}) - \log q(\mathbf{z}\mid\mathbf{x})] \\
&= \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}[\log p(\mathbf{x}\mid\mathbf{z})] - D_{KL}(q(\mathbf{z}\mid\mathbf{x})||p(\mathbf{z})) \qquad (:= ELBO)
\end{aligned}
$$

Reconstruction term **KL term**

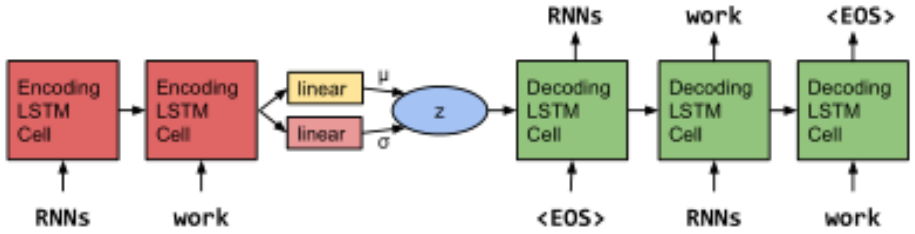


Figure 1: The core structure of our variational autoencoder language model. Words are represented using a learned dictionary of embedding vectors.

通常认为要将有用的信息编码到隐变量z中：

1、KL term $\neq$ 0

2、较小的cross entropy term

但是这个研究中发现，绝大部分情况下模型的KL项随着训练最终将变为零？？？

- **文章给出观点：**以前的VAE之所以能成是因为解码器太弱，**迫使模型接纳隐变量以获得更高的likelihood**。而**强大的解码器**导致模型学会忽略z，而去追求容易实现的目标，比如只用更容易优化的解码器来解释数据。
- **解决方案：**给KL项加上了一个权重，在训练最开始的时候将权重设为0，因而模型学着将尽可能多的信息编码进z中。接着随着训练的进行，逐渐的提升这个权重，迫使模型逐渐的平滑这个编码并将他们放进先验之中。这可以被看作是从AE到VAE的退火过程。

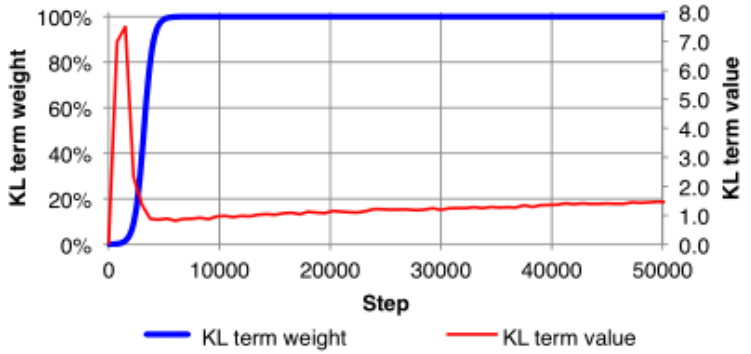


Figure 2: The weight of the KL divergence term of variational lower bound according to a typical sigmoid annealing schedule plotted alongside the (unweighted) value of the KL divergence term for our VAE on the Penn Treebank.

Lagging Inference Networks And Posterior Collapse In Variational Autoencoders

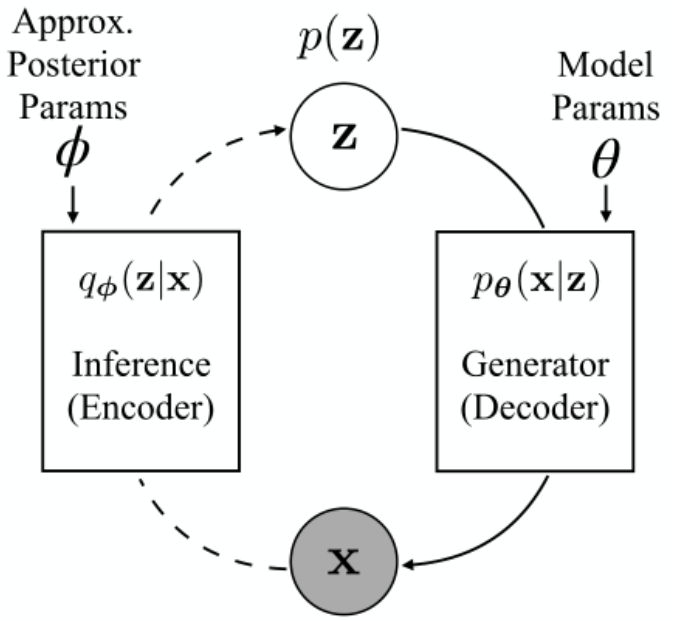
内容提要

前期工作

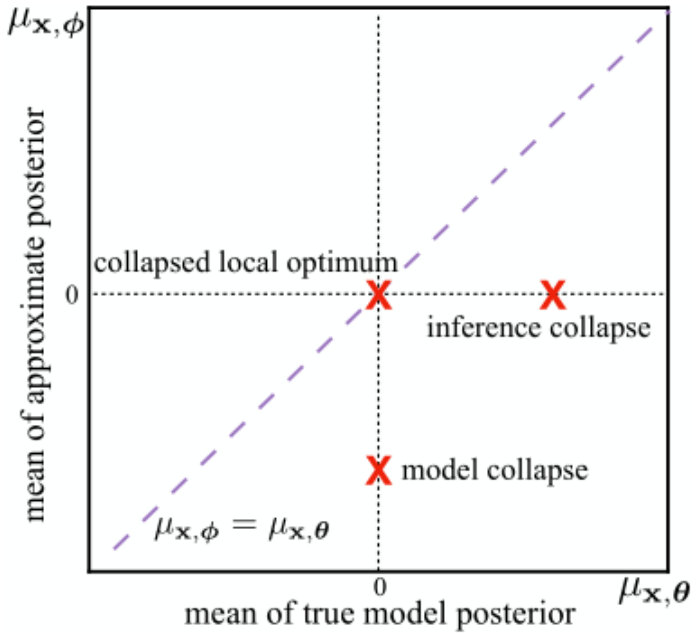
- 1、VAE尽管取得了很多成果，但是不少工作发现VAE训练的时候经常会遭遇一个称为“posterior collapse”的问题，它指的是模型忽略了隐变量z。**特别是当生成器$p_\theta(x|z)$采用的是较强的自回归神经网络时以及在对离散数据进行建模时这个问题更为常见。**
- 2、现有的工作大都是从一个**静态优化的角度**分析这个问题，注意到坍塌通常是ELBO的一个比较好的**局部最优**。因此许多解决方案关注于**削弱生成器或者修改训练目标函数**。

本文工作

- 1、本文的方案是从训练动力学的角度提出了一种新颖的训练方式来解决“posterior collapse”问题。相比于其它方案，我们的方案仍然优化的是标准的ELBO目标函数，不需要改变VAE模型或它的参数化。
- 2、在这篇文章中我们考虑了两点：**1、为什么VAE会掉入坍塌的局部最优；2、是否有一种简单的办法来改变训练轨迹以找到一个非平凡的局域最优。**
- 3、本文发现在训练的初始阶段**后验估计**通常是落后于**真实的模型后验。**接着证明了这样的滞后行为是如何驱使生成模型掉入局域坍塌的，并提出了一种新颖的训练方式，有针对性的更频繁的优化推理网络来减缓滞后。



(a) Variational autoencoders



(b) Posterior mean space

$$\mathcal{L}(\mathbf{x};\boldsymbol{\theta},\boldsymbol{\phi}) = \underbrace{\mathbb{E}_{\mathbf{z}\sim q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x}|\mathbf{z})]}_{\text{Reconstruction Loss}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})\|p(\mathbf{z}))}_{\text{KL Regularizer}},$$

Posterior Collapse:

$$q_\phi(\mathbf{z}|\mathbf{x}) = p_\theta(\mathbf{z}|\mathbf{x}) = p(\mathbf{z}) \text{ for all } \mathbf{x}.$$

问题拆分：

Model collapse: $p_\theta(z|x) = p(z)$

Inference collapse: $q_\varphi(z|x) = p(z)$

评估方式：

posterior mean space $\mathcal{U} = \{\mu : \mu = (\mu^1_{\mathbf{x},\boldsymbol{\theta}}, \mu^1_{\mathbf{x},\boldsymbol{\phi}})\}$,

$\mu_{\mathbf{x},\boldsymbol{\theta}}$ and $\mu_{\mathbf{x},\boldsymbol{\phi}}$ are the means of $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ and $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$

$\mu_{\mathbf{x},\boldsymbol{\theta}} = \mathbf{0}$ as model collapse

$\mu_{\mathbf{x},\boldsymbol{\phi}} = \mathbf{0}$ as inference collapse

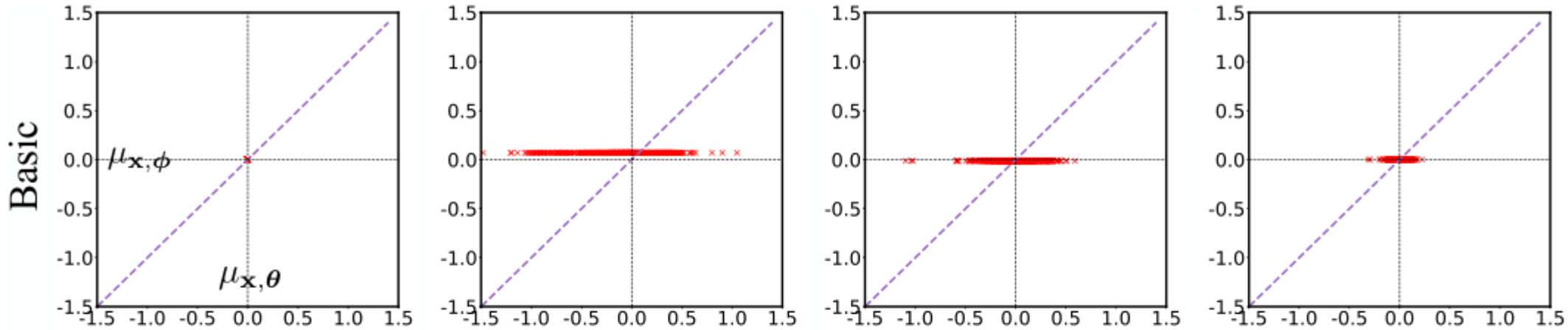


Figure 2: The projections of 500 data samples from a synthetic dataset on the posterior mean space over the course of training. "iter" denotes the number of updates of generators. The top row is from the basic VAE training, the bottom row is from our aggressive inference network training. The results show that while the approximate posterior is lagging far behind the true model posterior in basic VAE training, our aggressive training approach successfully moves the points onto the diagonal line and away from inference collapse.

Algorithm 1 VAE training with controlled aggressive inference network optimization.

$\boldsymbol{\theta}, \boldsymbol{\phi} \leftarrow$ Initialize parameters
$aggressive \leftarrow$ TRUE
repeat
  if $aggressive$ **then**
    repeat ▷ [aggressive updates]
      $\mathbf{X} \leftarrow$ Random data minibatch
      Compute gradients $\boldsymbol{g}_{\boldsymbol{\phi}} \leftarrow \nabla_{\boldsymbol{\phi}} \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi})$
      Update $\boldsymbol{\phi}$ using gradients $\boldsymbol{g}_{\boldsymbol{\phi}}$
    until convergence
    $\mathbf{X} \leftarrow$ Random data minibatch
    Compute gradients $\boldsymbol{g}_{\boldsymbol{\theta}} \leftarrow \nabla_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi})$
    Update $\boldsymbol{\theta}$ using gradients $\boldsymbol{g}_{\boldsymbol{\theta}}$
  else ▷ [basic VAE training]
    $\mathbf{X} \leftarrow$ Random data minibatch
    Compute gradients $\boldsymbol{g}_{\boldsymbol{\theta},\boldsymbol{\phi}} \leftarrow \nabla_{\boldsymbol{\phi},\boldsymbol{\theta}} \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi})$
    Update $\boldsymbol{\theta}, \boldsymbol{\phi}$ using $\boldsymbol{g}_{\boldsymbol{\theta},\boldsymbol{\phi}}$
  end if
  Update $aggressive$ as discussed in Section 4.2
until convergence

另一个视角：

$$\mathcal{L}(\mathbf{x}; \boldsymbol{\theta}, \boldsymbol{\phi}) = \underbrace{\log p_{\boldsymbol{\theta}}(\mathbf{x})}_{\text{marginal log data likelihood}} - \underbrace{D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}))}_{\text{agreement between approximate and model posteriors}},$$

更新策略：

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi}^*), \text{ where } \boldsymbol{\phi}^* = \arg\max_{\boldsymbol{\phi}} \mathcal{L}(\mathbf{X}; \boldsymbol{\theta}, \boldsymbol{\phi}),$$

Stopping criterion:

$$I_q = \mathbb{E}_{\mathbf{x} \sim p_d(\mathbf{x})}[D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) || p(\mathbf{z}))] - D_{\mathrm{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}) \| p(\mathbf{z})),$$

Mutual information between z and x under $q_\phi(z|x)$

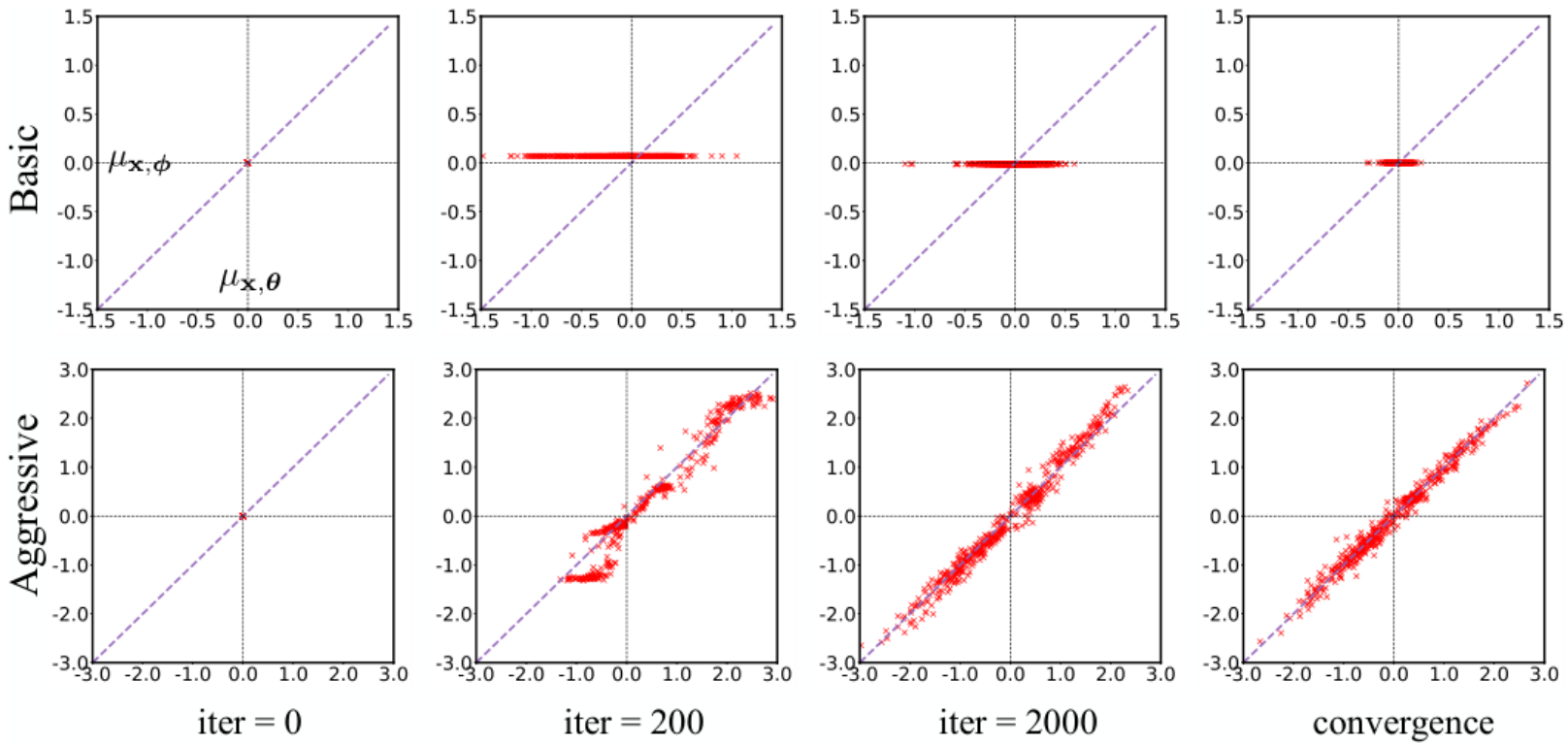


Figure 2: The projections of 500 data samples from a synthetic dataset on the posterior mean space over the course of training. "iter" denotes the number of updates of generators. The top row is from the basic VAE training, the bottom row is from our aggressive inference network training. The results show that while the approximate posterior is lagging far behind the true model posterior in basic VAE training, our aggressive training approach successfully moves the points onto the diagonal line and away from inference collapse.

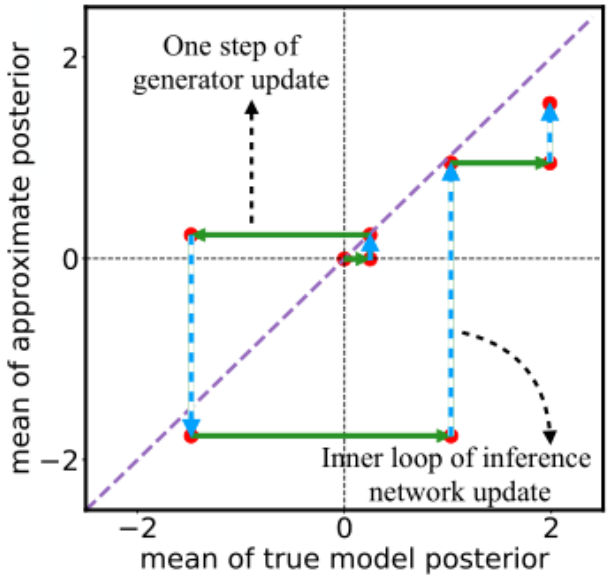


Figure 3: Trajectory of one data instance on the posterior mean space with our aggressive training procedure. Horizontal arrow denotes one step of generator update, and vertical arrow denotes the inner loop of inference network update. We note that the approximate posterior $q_{\phi}(\mathbf{z}|\mathbf{x})$ takes an aggressive step to catch up to the model posterior $p_{\theta}(\mathbf{z}|\mathbf{x})$.

True data experiment results

Table 1: Results on Yahoo and Yelp datasets. We report mean values across 5 different random restarts, and standard deviation is given in parentheses when available. For LSTM-LM* we report the exact negative log likelihood.

Model	Yahoo NLL	Yahoo KL	Yahoo MI	Yahoo AU	Yelp NLL	Yelp KL	Yelp MI	Yelp AU
Previous Reports								
CNN-VAE (Yang et al., 2017)	$\leq$332.1	10.0	–	–	$\leq$359.1	7.6	–	–
SA-VAE + anneal (Kim et al., 2018)	$\leq$327.5	7.19	–	–	–	–	–	–
Modified VAE Objective								
VAE + anneal	328.6 (0.0)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	357.9 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
β-VAE ($\beta = 0.2$)	332.2 (0.6)	19.1 (1.5)	3.3 (0.1)	20.4 (6.8)	360.7 (0.7)	11.7 (2.4)	3.0 (0.5)	10.0 (5.9)
β-VAE ($\beta = 0.4$)	328.7 (0.1)	6.3 (1.7)	2.8 (0.6)	8.0 (5.2)	358.2 (0.3)	4.2 (0.4)	2.0 (0.3)	4.2 (3.8)
β-VAE ($\beta = 0.6$)	328.5 (0.1)	0.3 (0.2)	0.2 (0.1)	1.0 (0.7)	357.9 (0.1)	0.2 (0.2)	0.1 (0.1)	3.8 (2.9)
β-VAE ($\beta = 0.8$)	328.8 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	358.1 (0.2)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
SA-VAE + anneal	327.2 (0.2)	5.2 (1.4)	2.7 (0.5)	9.8 (1.3)	355.9 (0.1)	2.8 (0.5)	1.7 (0.3)	8.4 (0.9)
Ours + anneal	**326.7 (0.1)**	5.7 (0.7)	2.9 (0.2)	15.0 (3.5)	**355.9 (0.1)**	3.8 (0.2)	2.4 (0.1)	11.3 (1.0)
Standard VAE Objective								
LSTM-LM*	**328.0 (0.3)**	–	–	–	358.1 (0.6)	–	–	–
VAE	329.0 (0.1)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)	358.3 (0.2)	0.0 (0.0)	0.0 (0.0)	0.0 (0.0)
SA-VAE	329.2 (0.2)	0.1 (0.0)	0.1 (0.0)	0.8 (0.4)	357.8 (0.2)	0.3 (0.1)	0.3 (0.0)	1.0 (0.0)
Ours	328.2 (0.2)	5.6 (0.2)	3.0 (0.0)	8.0 (0.0)	**356.9 (0.2)**	3.4 (0.3)	2.4 (0.1)	7.4 (1.3)

True data experiment results

Table 2: Results on OMNIGLOT dataset. We report mean values across 5 different random restarts, and standard deviation is given in parentheses when available. For PixelCNN* we report the exact negative log likelihood.

Model	NLL	KL	MI	AU
Previous Reports				
VLAE (Chen et al., 2017)	89.83	–	–	–
VampPrior (Tomczak & Welling, 2018)	89.76	–	–	–
Modified VAE Objective				
VAE + anneal	89.21 (0.04)	1.97 (0.12)	1.79 (0.11)	5.3 (1.0)
β-VAE ($\beta = 0.2$)	105.96 (0.38)	69.62 (2.16)	3.89 (0.03)	32.0 (0.0)
β-VAE ($\beta = 0.4$)	96.09 (0.36)	44.93 (12.17)	3.91 (0.03)	32.0 (0.0)
β-VAE ($\beta = 0.6$)	92.14 (0.12)	25.43 (9.12)	3.93 (0.03)	32.0 (0.0)
β-VAE ($\beta = 0.8$)	89.15 (0.04)	9.98 (0.20)	3.84 (0.03)	13.0 (0.7)
SA-VAE + anneal	**89.07 (0.06)**	3.32 (0.08)	2.63 (0.04)	8.6 (0.5)
Ours + anneal	89.11 (0.04)	2.36 (0.15)	2.02 (0.12)	7.2 (1.3)
Standard VAE Objective				
PixelCNN*	89.73 (0.04)	–	–	–
VAE	89.41 (0.04)	1.51 (0.05)	1.43 (0.07)	3.0 (0.0)
SA-VAE	89.29 (0.02)	2.55 (0.05)	2.20 (0.03)	4.0 (0.0)
Ours	**89.05 (0.05)**	2.51 (0.14)	2.19 (0.08)	5.6 (0.5)

对比本方法和VAE+annealing、β-vae

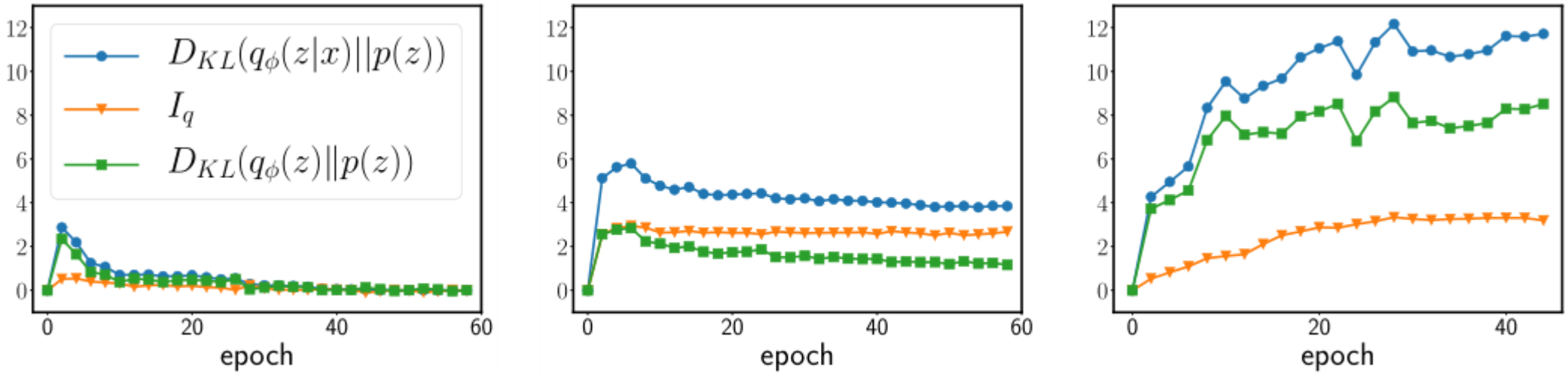


Figure 5: Training behavior on Yelp. **Left:** VAE + annealing. **Middle:** Our method. **Right:** β-VAE ($\beta = 0.2$).

对另外两个方法的评价：

1、之前通常的做法都是**减弱KL项**的正则化作用，因为ELBO中的KL项直观看起来**最像是罪魁祸首**

2、KL cost annealing和β-vae都是降低KL项的权重，等价于将$q_\varphi(z|x)$ 推离开p(z)

对三个评价量的作用：

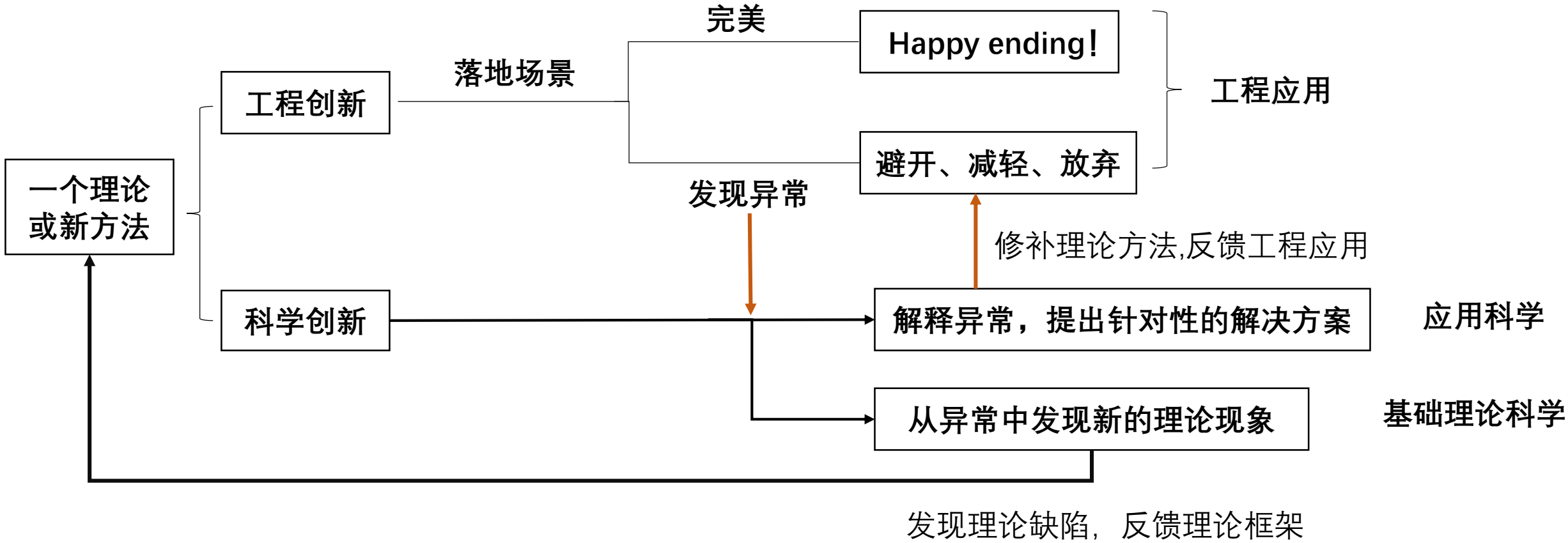
$$I_q = \mathbb{E}_{\mathbf{x}\sim p_d(\mathbf{x})}[D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z}))] - D_{\mathrm{KL}}(q_\phi(\mathbf{z})\|p(\mathbf{z})),$$

1、**互信息**代表了x和z之间的条件相关性，而**互信息**越大说明隐变量中被编码的信息越多。

2、**KL正则项**其实可以看成是互信息$\boldsymbol{I_q}$和$\boldsymbol{D_{KL}(q_\varphi(z)||p(z))}$的和

3、 $\mathbf{q_\varphi(z)}$是用来描述观察数据对应的隐变量分布，如果它与模型的先验分布**p(z)**相差较大，说明整个模型对于数据的描述较差。

吾之蜜糖、彼之砒霜


一个理论
或新方法
工程创新
科学创新
落地场景
完美
Happy ending!
避开、减轻、放弃
工程应用
发现异常
修补理论方法,反馈工程应用
解释异常，提出针对性的解决方案
应用科学
从异常中发现新的理论现象
基础理论科学
发现理论缺陷，反馈理论框架

Increasing the expressiveness of approximate posteriors

$$\log p_\theta(X) = \sum_{i=1}^{N} \log p_\theta(x_i) = \sum_{i=1}^{N} \log \int p_\theta(x_i, z_i) dz_i.$$

$$\log p(x) = \mathbb{E}_{q(z|x)} \left[\log \left(\frac{p(x,z)}{q(z|x)} \right) \right] + \mathrm{KL}\left(q(z|x)||p(z|x)\right) \tag{1}$$

$$\geq \mathbb{E}_{q(z|x)} \left[\log \left(\frac{p(x,z)}{q(z|x)} \right) \right] = \mathcal{L}_{\mathrm{VAE}}[q]. \tag{2}$$

- **1、Normalizing flows**

$$\mathbb{E}_{z_0 \sim q_0(z|x)} \left[\log \left(\frac{p(x, z_T)}{q_0(z_0|x) \prod_{t=1}^{T} \left| \det \frac{\partial z_t}{\partial z_{t-1}} \right|^{-1}} \right) \right]. \tag{5}$$

- 2、**Auxiliary variables:**

hierarchical variational models can induce dependencies between latent variables

$$\mathbb{E}_{z,v \sim q(z,v|x)} \left[\log \left(\frac{p(x,z) r(v|x,z)}{q(z,v|x)} \right) \right] \tag{6}$$

$$= \mathbb{E}_{q(z|x)} \left[\log \left(\frac{p(x,z)}{q(z|x)} \right) - \mathrm{KL}\Big(q(v|z,x) \| r(v|x,z) \Big) \right] \tag{7}$$

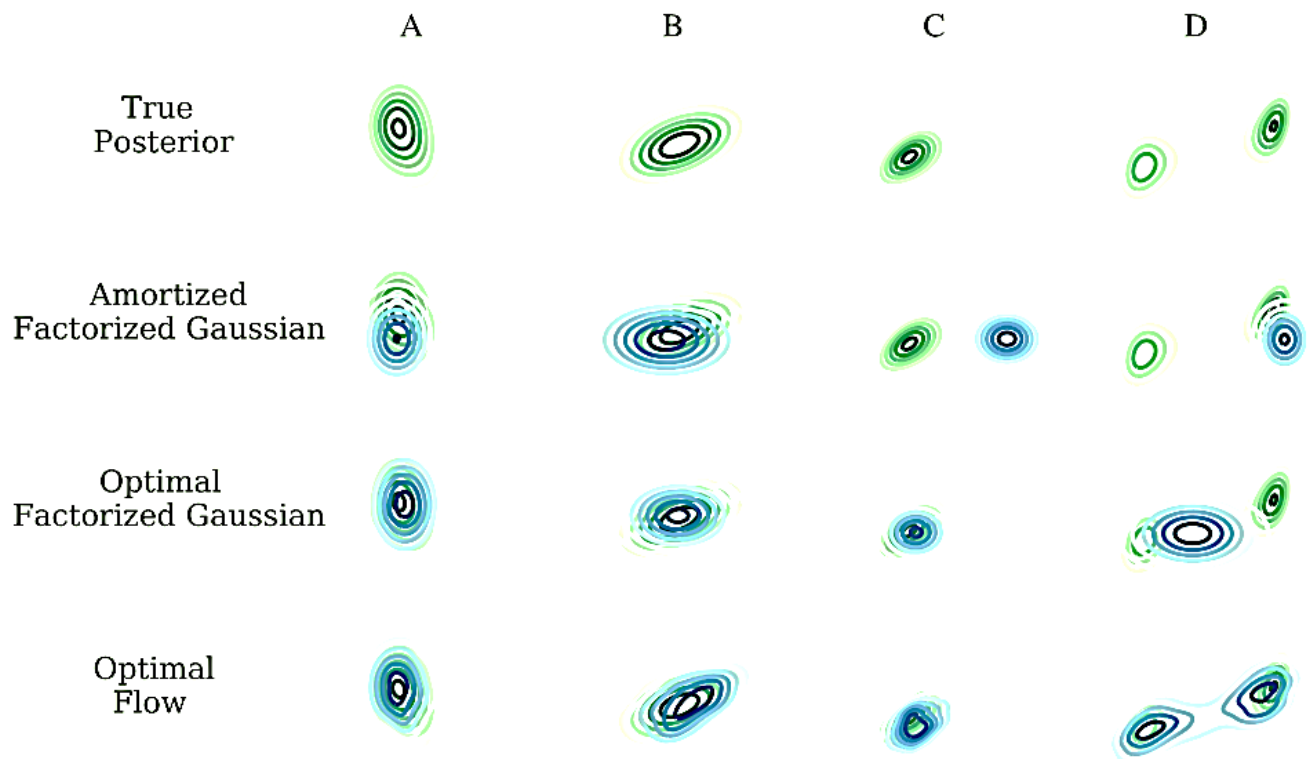


Figure 2. True Posterior and Approximate Distributions of a VAE with 2D latent space. The columns represent four different datapoints. The green distributions are the true posterior distributions, highlighting the mismatch with the blue approximations. Amortized: Variational parameters learned over the entire dataset. Optimal: Variational parameters optimized for each individual datapoint. Flow: Using a flexible approximate distribution.