KL collapse: A survey

Problem

- KL will vanish when using strong auto-aggressive decoder
- Model collapse to a standard auto-regressive model
- Cannot learn good representation using z (the latent values)

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$$

From a regularization perspective

- The very first paper Bowman et al., 2016
- Use KL annealing (weighting the KL from a small number to 1)
- Word dropout to limit the power of decoder
- Unable to deal with collapse on complex text datasets with very large LSTM decoders

$$\mathcal{L}_{\text{VAE}}(\theta, \phi) + \lambda_1 \mathbb{E}_{\hat{p}(x)}[R_1(q_\phi(z|x))] + \lambda_2 R_2(q_\phi(z))$$

WORK	$\mathcal{L}$.	R_1	R_2	Y
β-VAE [2]	VAE	$D_{\mathrm{KL}}(q_\phi(z\|x)\|p(z))$		
VIB [9]	VAE	$D_{\mathrm{KL}}(q_\phi(z\|x)\|p(z))$		O
PixelGAN-AE[18]	VAE	$-I_{q_\phi}(x;z)$		O
InfoVAE [5]	VAE	$D_{\mathrm{KL}}(q_\phi(z\|x)\|p(z))$	$D_{\mathrm{KL}}(q_\phi(z)\|p(z))$	
Info. dropout [10]	VAE	$D_{\mathrm{KL}}(q_\phi(z\|x)\|p(z))$	$\mathrm{TC}(q_\phi(z))$	O
HFVAE [8]	VAE	$-I_{q_\phi}(x;z)$	$R_{\mathcal{G}}(q_\phi(z))+\lambda_2'\sum_{G\in\mathcal{G}} R_{\mathcal{G}}(q_\phi(z))$	
FactorVAE [3, 4]	VAE		$\mathrm{TC}(q_\phi(z))$	
DIP-VAE [6]	VAE		$\|\mathrm{Cov}_{q_\phi(z)}[z]-I\|_{\mathrm{F}}^2$	
HSIC-VAE [7]	VAE		$\mathrm{HSIC}(q_\phi(z_{G_1}), q_\phi(z_{G_2}))$	O
VFAE [13]	VAE		$\mathrm{MMD}(q_\phi(z\|s=0), q_\phi(z\|s=1))$	✓
DC-IGN [11]	VAE			✓
FaderNet. [12]; [37][2]	AE	$-\mathbb{E}_{\hat{p}(x,y)}[\log P_\psi(1-y\|E_\phi(x))]$		✓
AAE/WAE [22, 36]	AE		$D_{\mathrm{JS}}(E_\phi(z)\|p(z))$	O

Treat the weight as a hyperparameter

- Beta-VAE

$$\max_{\phi,\theta} \mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z})]$$

$$\text{subject to } D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z})) < \delta$$

- Use the information bottleneck to control the distance between real and estimated data
- explicitly encourages the use of latent variable, they may implicitly sacrifice density estimation performance at the same time

Replace KL with others (InfoVAE)

- Kl may be not strong enough to regularize the reconstruction loss
- Replace KL with Maximum-Mean Discrepancy (match only in expection)
- MMD is a framework to quantify the distance between two distributions by comparing all of their moments.

$$D_{\mathrm{MMD}}(q\|p) = \mathbb{E}_{p(z),p(z')}[k(z,z')] - 2\mathbb{E}_{q(z),p(z')}[k(z,z')] + \mathbb{E}_{q(z),q(z')}[k(z,z')]$$

$$D_{\mathrm{MMD}} = 0 \text{ if and only if } p = q.$$

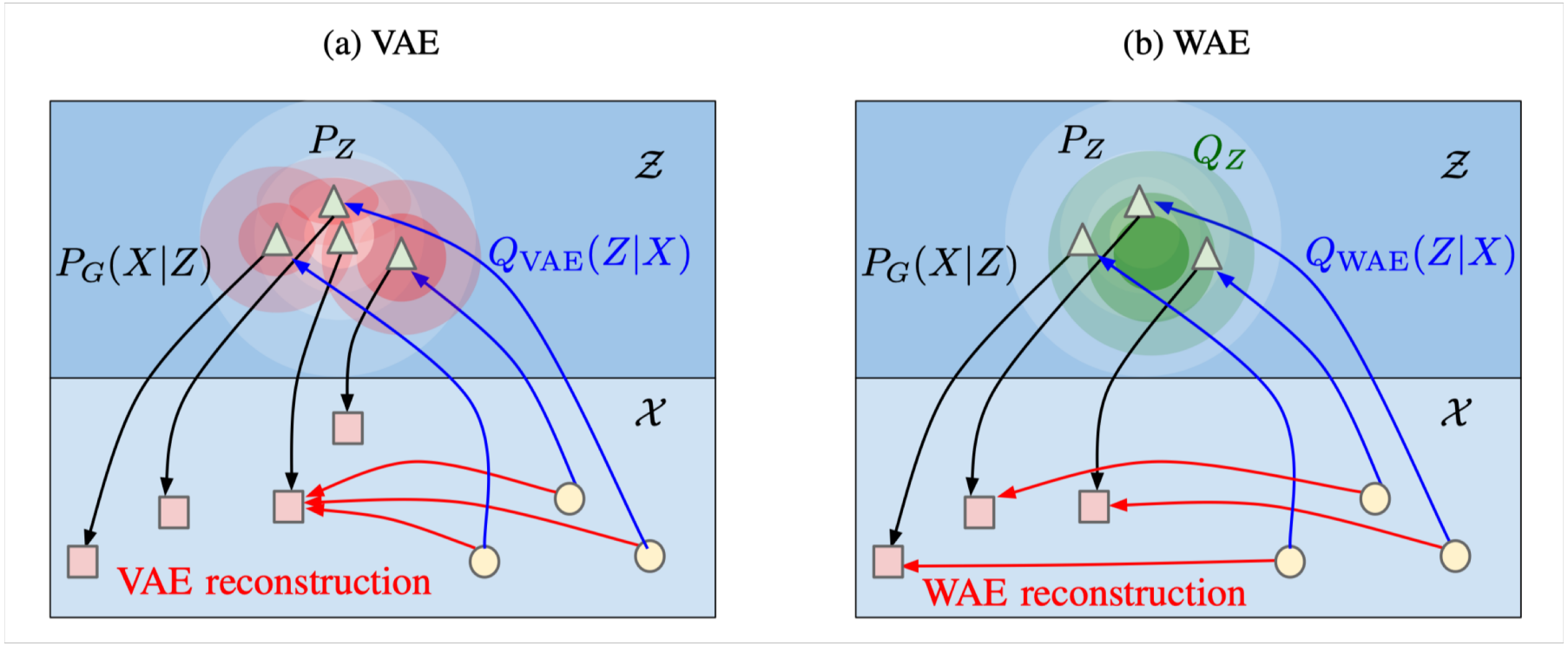
Replace Gaussian with other Prior

- KL vanished to 0 when decoding, how about choose a prior with KL is a constant?
- von Mises-Fisher prior is provided (K can be set as a hyperparameter, however, if k is learned collapses happens again)

$$\mathbf{KL}(\mathbf{vMF}(\mu,\kappa)||\mathbf{vMF}(\cdot,0)) = \kappa\frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}$$
$$+\left(\frac{d}{2}-1\right)\log\kappa - \frac{d}{2}\log(2\pi) - \log I_{d/2-1}(\kappa)$$
$$+\frac{d}{2}\log\pi + \log 2 - \log\Gamma\left(\frac{d}{2}\right)$$

WASSERSTEIN AUTO-ENCODERS

- VAE forces Q to match P for every data points then collapse to the prior
- WAE forces forces the mixture to match Pz



Trained by W-GAN

- The VDB is a followed up work

ALGORITHM 1 Wasserstein Auto-Encoder with GAN-based penalty (WAE-GAN).

Require: Regularization coefficient $\lambda > 0$.

Initialize the parameters of the encoder Q_ϕ, decoder G_θ, and latent discriminator D_γ.

while (ϕ, θ) not converged **do**

Sample $\{x_1, \dots, x_n\}$ from the training set

Sample $\{z_1, \dots, z_n\}$ from the prior P_Z

Sample $\tilde{z}_i$ from $Q_\phi(Z|x_i)$ for $i = 1, \dots, n$

Update D_γ by ascending:

$$\frac{\lambda}{n} \sum_{i=1}^{n} \log D_\gamma(z_i) + \log\big(1 - D_\gamma(\tilde{z}_i)\big)$$

Update Q_ϕ and G_θ by descending:

$$\frac{1}{n} \sum_{i=1}^{n} c\big(x_i, G_\theta(\tilde{z}_i)\big) - \lambda \cdot \log D_\gamma(\tilde{z}_i)$$

end while

Increase disentanglement

- Similar to the infoVAE, MAE (Ma etal 2019) adds a data-dependent regularization

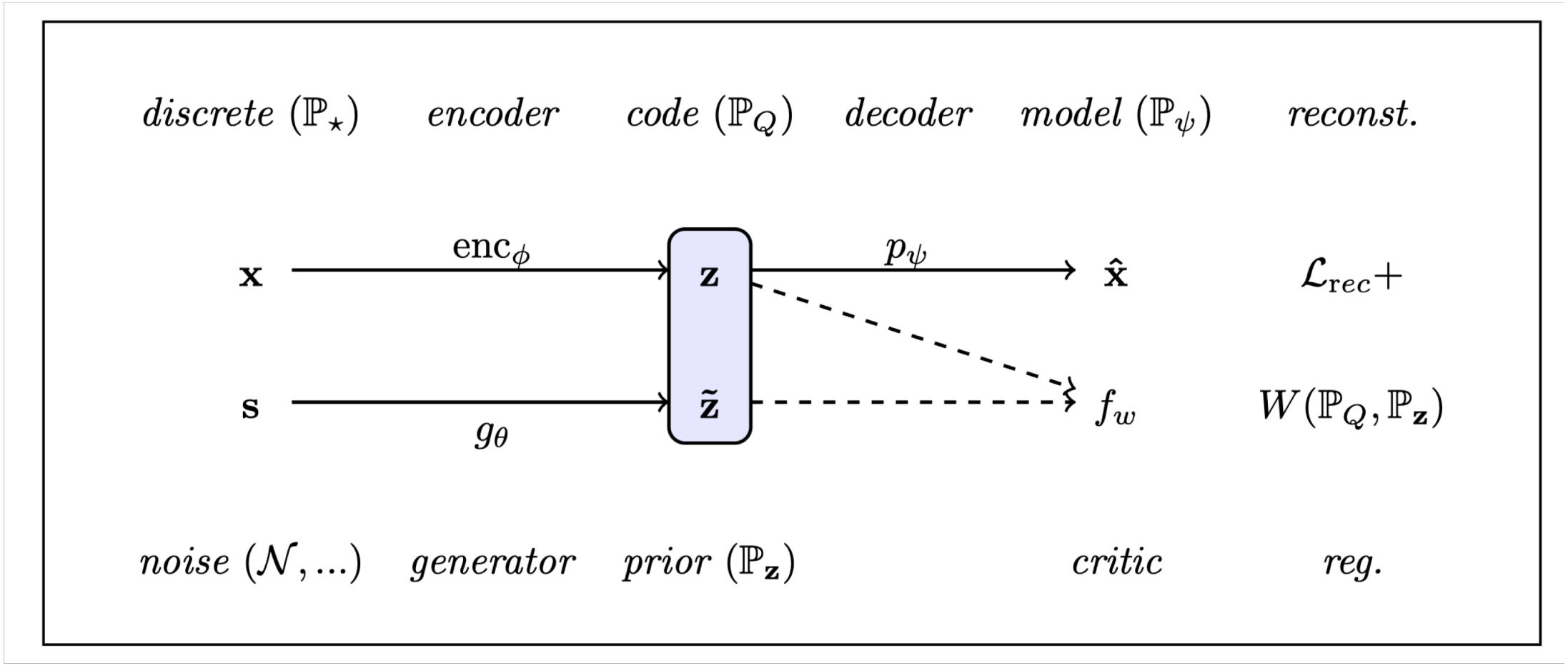
$$\mathcal{L}_{MAE} = \mathcal{L}_{elbo} + \eta \mathcal{L}_{diverse} + \gamma \mathcal{L}_{smooth}$$

$$\mathcal{L}_{diverse} = \mathrm{E}_{X_1, X_2 \sim P(X)} \left[\sum_{k=1}^{K} \log(1 + \exp(-\mathrm{KL}(q_\phi(Z_k|X_1)||q_\phi(Z_k|X_2)))) \right]$$

$$\mathcal{L}_{smooth} = \mathrm{STD}_{X_1, X_2 \sim P(X)} [\mathrm{KL}(q_\phi(Z|X_1)||q_\phi(Z|X_2))]$$

Adversarially Regularized Autoencoders

- Apply WAE to text



Algorithm 1 ARAE Training

for each training iteration **do**

(1) Train the encoder/decoder for reconstruction (ϕ, ψ)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$

Backprop loss, $\mathcal{L}_{\text{rec}} = -\frac{1}{m}\sum_{i=1}^m \log p_\psi(\mathbf{x}^{(i)} \mid \mathbf{z}^{(i)})$

(2) Train the critic (w)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and $\{\mathbf{s}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$

Compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{z}}^{(i)} = g_\theta(\mathbf{z}^{(i)})$

Backprop loss $-\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{z}^{(i)}) + \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{z}}^{(i)})$

Clip critic w to $[-\epsilon, \epsilon]^d$.

(3) Train the encoder/generator adversarially (ϕ, θ)

Sample $\{\mathbf{x}^{(i)}\}_{i=1}^m \sim \mathbb{P}_\star$ and $\{\mathbf{s}^{(i)}\}_{i=1}^m \sim \mathcal{N}(0, \mathbf{I})$

Compute $\mathbf{z}^{(i)} = \text{enc}_\phi(\mathbf{x}^{(i)})$ and $\tilde{\mathbf{z}}^{(i)} = g_\theta(\mathbf{s}^{(i)})$.

Backprop loss $\frac{1}{m}\sum_{i=1}^m f_w(\mathbf{z}^{(i)}) - \frac{1}{m}\sum_{i=1}^m f_w(\tilde{\mathbf{z}}^{(i)})$

end for

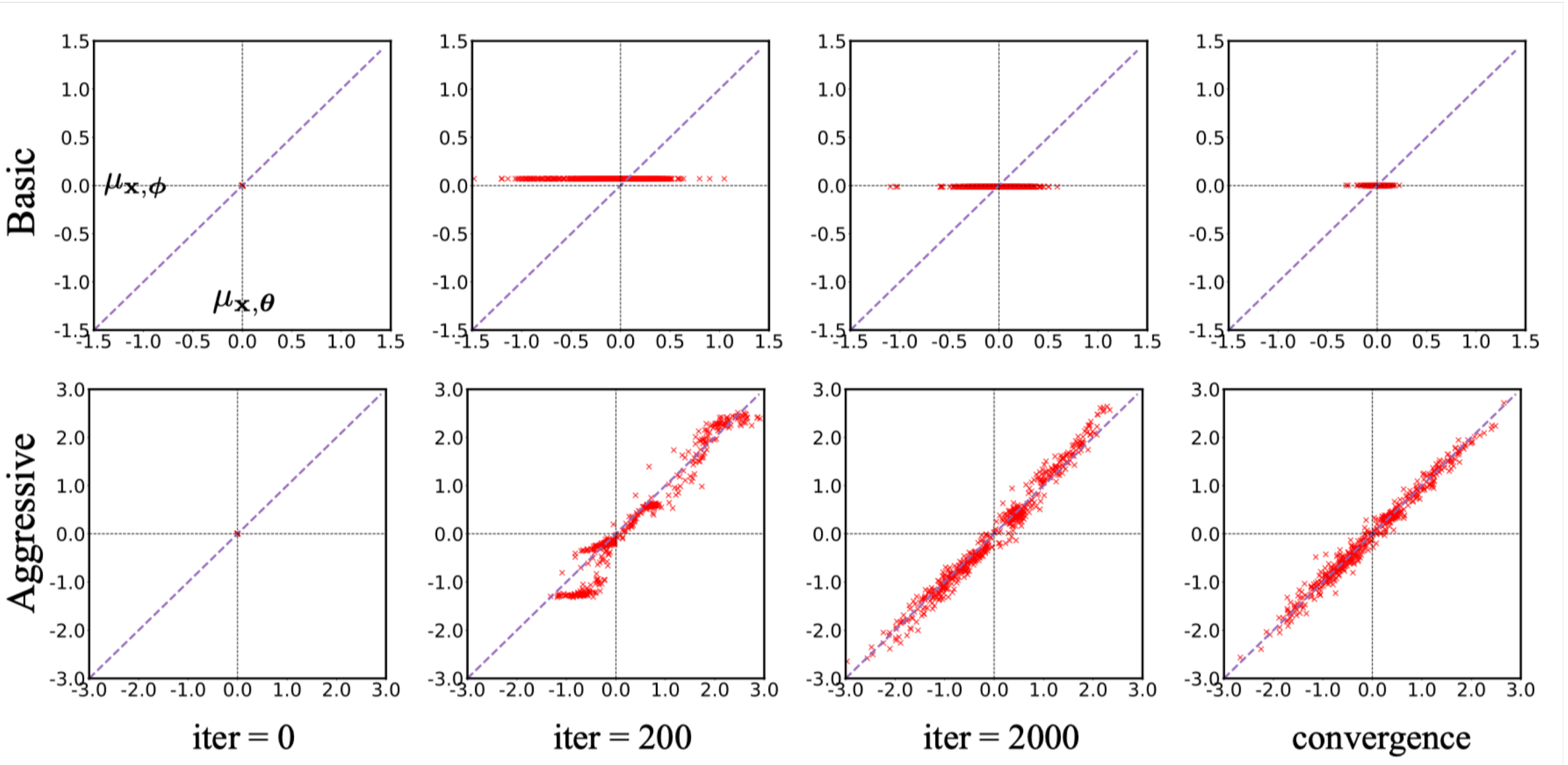
Analysis the training progress

- L AGGING I NFERENCE N ETWORKS AND P OSTERIOR C OLLAPSE IN VARIATIONAL AUTOENCODERS (ICLR 2019)
- There are two kind of collapse
  - Model collapse where

$$p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$$

  - Inference collapse where

$$q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x}) = p(\mathbf{z})$$


Basic
Aggressive
$\mu_{\mathbf{x},\phi}$
$\mu_{\mathbf{x},\theta}$
iter = 0
iter = 200
iter = 2000
convergence

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

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)	≤332.1	10.0	–	–	≤359.1	7.6	–	–
SA-VAE + anneal (Kim et al., 2018)	≤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)

Weaken the decoder

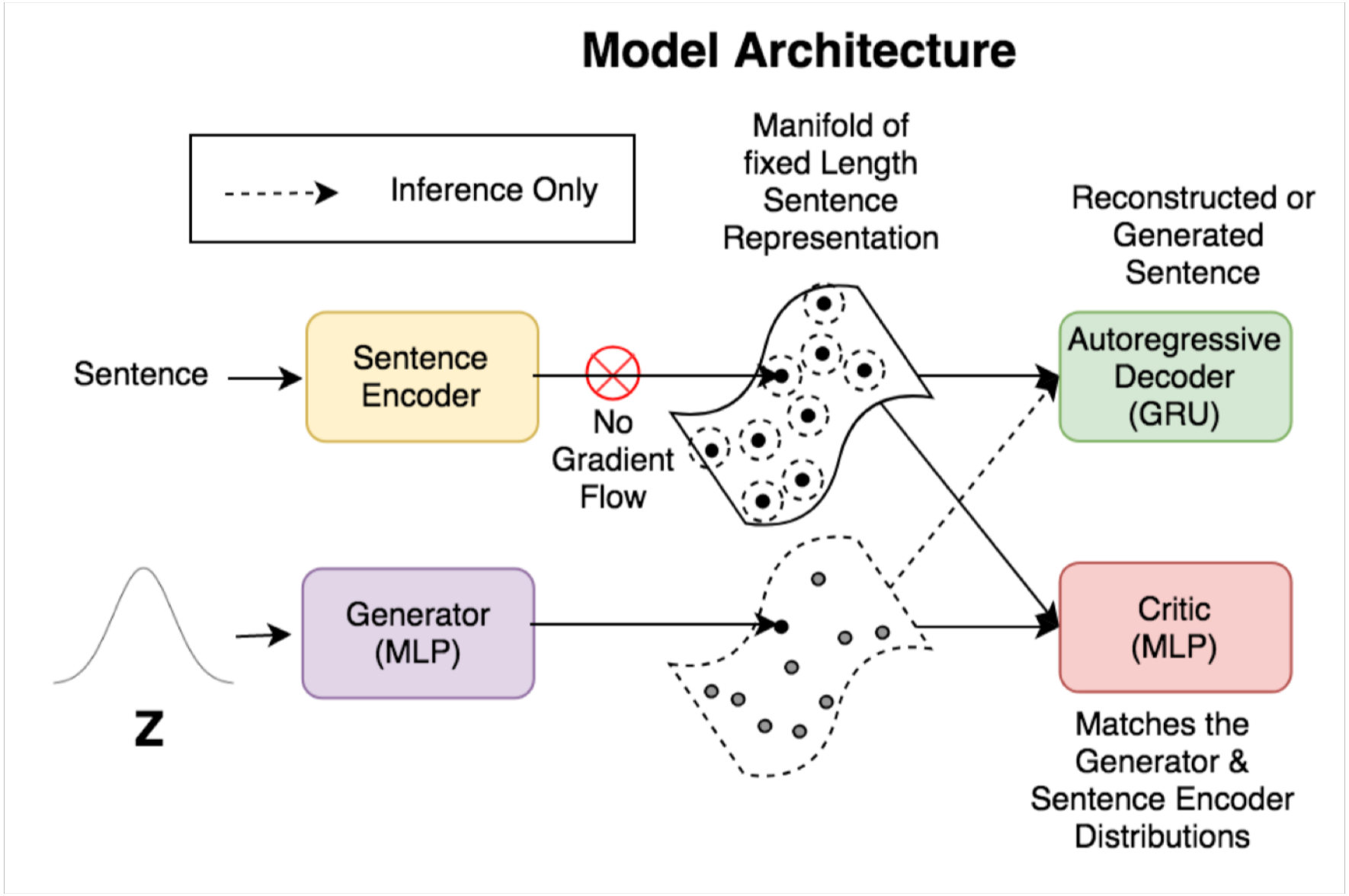
- Replace the RNN decoder with
  - CNN based decoder
  - Gaines are limited
  - Need to specific design for tasks

Summaries

- In short, optimize the KL part is easier than the likelihood part (annealing, beta-vae, and the aggressive training schema)
- Replace the prior with a prior whose KL can be controlled into a constant
- Using a weakened decoder
- Add regularization to push individual data point away from each other (W-AE, Ma et al 2018, MMD-VAE)
- Or to force the z to have some relation between input, for example, reconstruct input from z

Towards Text Generation with Adversarially Learned Neural Outlines (NIPS 2018)

- Unsupervised text generation with GAN without RL
- Use pre-trained context representation (ELMO)
- Results show that using ELMO improve NLL in a big margin


Model Architecture
Inference Only
Manifold of fixed Length Sentence Representation
Reconstructed or Generated Sentence
Sentence
Sentence Encoder
No Gradient Flow
Autoregressive Decoder (GRU)
Generator (MLP)
Critic (MLP)
z
Matches the Generator & Sentence Encoder Distributions

Dataset	ARAE						WDLSTM				Ours					
	FPPL			RPPL			FPPL		RPPL		FPPL			RPPL		
	0.5	1.0	B=1	0.5	1.0	B=1	0.5	1.0	0.5	1.0	0.5	1.0	B=5	0.5	1.0	B=5
BookCorpus	389.6	555.6	364.2	209.2	206.2	213.3	9.4	185.2	280.7	137.2	25.5	66.6	10.5	220.4	152.8	250.9
WMT15	448.7	965.1	385.8	476.2	378.7	626.3	21.4	369.0	528.9	250.5	105.5	212.9	19.9	350.5	254.1	373.2
SNLI	67.5	109.1	62.0	54.8	54.0	59.9	5.9	57.0	86.8	34.5	18.6	35.6	15.3	90.8	49.5	59.8

Table 6: Quantitative evaluation of sample quality from ARAE, WD-LSTM and our model. We report the FPPL and RPPL from a KN smoothed 5-gram language modeled trained on a distinct but large subset of the data. We also report FPPLs and RPPLs on samples generated with multinomial sampling with temperatures of 0.5 and 1.0, as well as deterministic decoding with beam search (B). Note that deterministic decoding is not suitable for the WDLSTM since there needs to be some stochasticity to produce diverse samples.