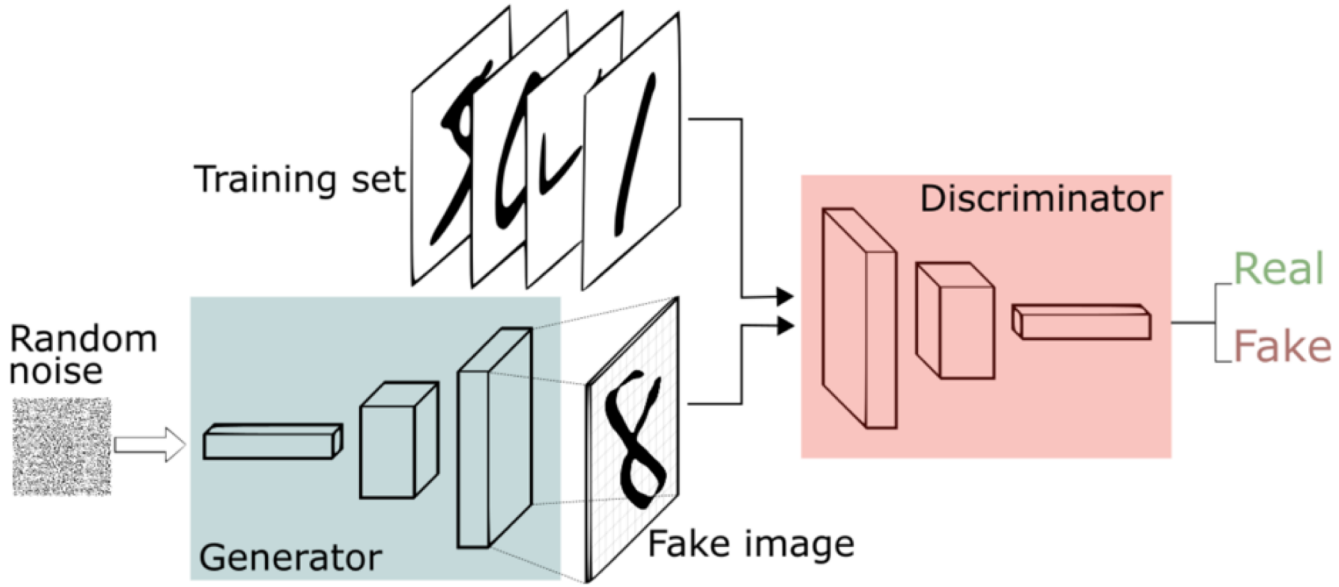
VAE+GAN

Monday, Oct 8, 2018

Jianguo Zhang

Generative Adversarial Network (GAN)

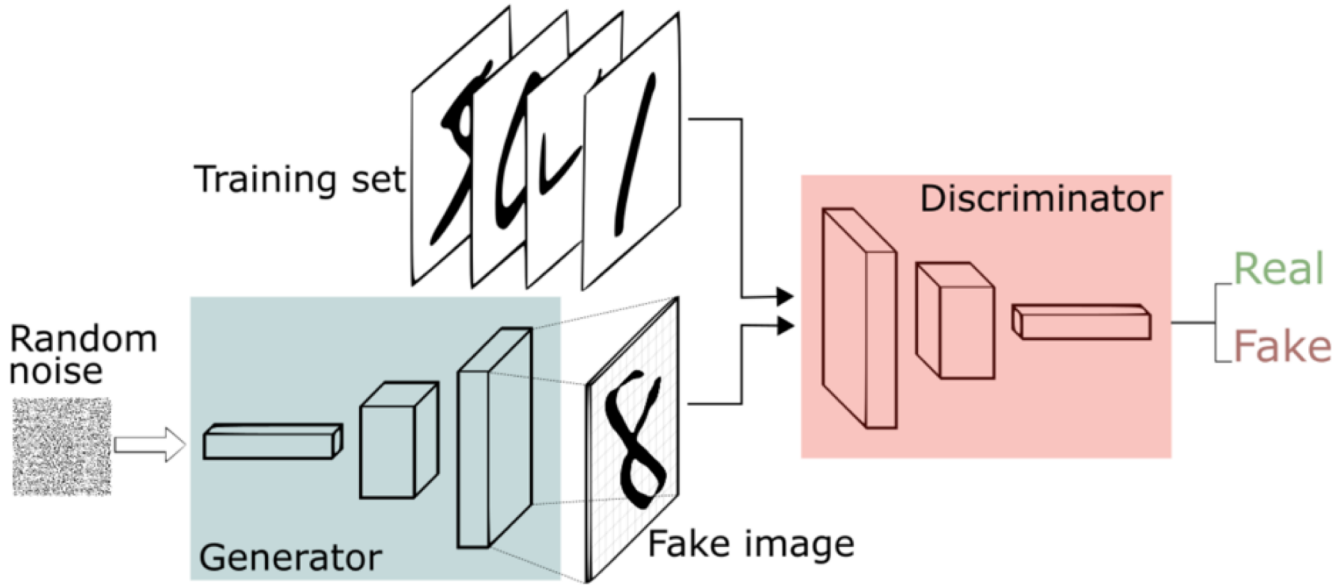


Generative Adversarial Network framework.

- **Discriminator tries to correctly distinguish the true data and the fake data**
- **Generator tries to generate high-quality data to fool discriminator**
- **Ideally, when D cannot distinguish the true and generated data, G can keeps generate realistic images**

Goodfellow et. al , “Generative Adversarial Nets”, NIPS, 2014

Generative Adversarial Network (GAN)



Generative Adversarial Network framework.

Probability that a real sample is classified as Real

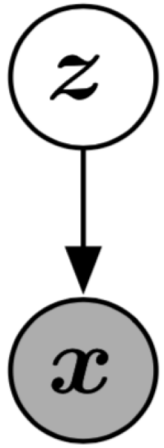
$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[log\ D(x)] + E_{z\sim p_z(z)}[log(1 - D(G(z)))]$$

Probability that a fake sample is classified as Fake

Goodfellow et. al , “Generative Adversarial Nets”, NIPS, 2014

Generative Adversarial Network (GAN)

Attractiveness



- Generator Networks $x = G(z; \theta^{(G)})$
- It is only required that, G is differentiable.
- So, having training data $x \sim p_{data}(x)$
  what we want is a model that can draw samples
  $x \sim p_{model}(x)$, where $p_{model} \approx p_{data}$
- Don't write a formula for $p_{data}(x)$, just learn to draw sample directly.

Goodfellow et. al , "Generative Adversarial Nets", NIPS, 2014

Generative Adversarial Network (GAN)

Drawbacks-1： Gradient Vanishing

Drawbacks-2： Training Instability

Drawbacks-3： Mode Collapse

(Generate Undesirable Modes)
(Generate a Single Mode, lack of variety)

Generative Adversarial Network (GAN)

Drawbacks-1： Gradient Vanishing

using the original form of the objective of ***G***

$$\mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]$$

will result in gradient vanishing issue of ***D*** for ***G*** because *intuitively*, at the very early phase of training, ***D*** is very easy to be confident in detecting ***G***, so ***D*** will output almost always 0

Generative Adversarial Network (GAN)

Drawbacks-1： **Gradient Vanishing**

using the original form of the objective of ***G***

$$\mathbb{E}_{z\sim p(z)}[\log(1 - D(g_\theta(z)))]$$

will result in gradient vanishing issue of ***D*** for ***G*** because *theoretically*, when ***D*** is *optimal*, minimizing the loss is equal to minimizing the *JS divergence* (Arjovsky & Bottou, 2017)

Generative Adversarial Network (GAN)

Drawbacks-1： Gradient Vanishing

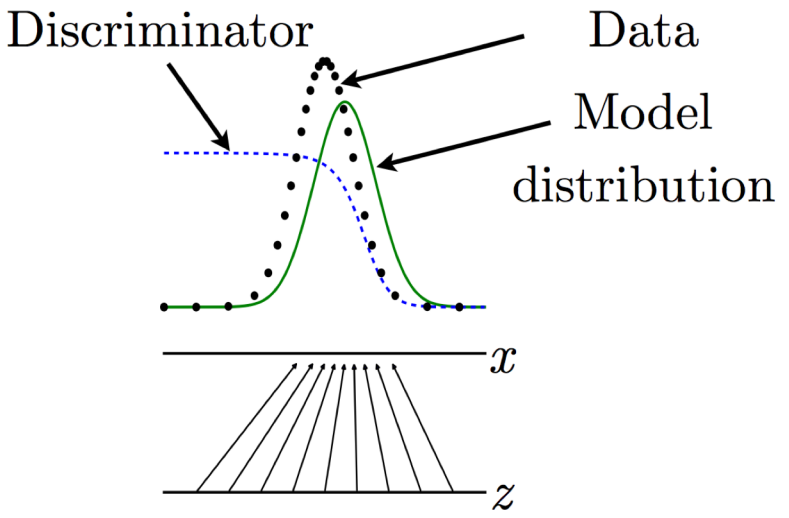
$$L(D, g_\theta) = \mathbb{E}_{x\sim\mathbb{P}_r}[\log D(x)] + \mathbb{E}_{x\sim\mathbb{P}_g}[\log(1 - D(x))]$$

Derivative

$$\frac{\delta}{\delta D(\boldsymbol{x})} J^{(D)} = 0.$$

$$D^*(\boldsymbol{x}) = \frac{p_{\text{data}}(\boldsymbol{x})}{p_{\text{data}}(\boldsymbol{x}) + p_{\text{model}}(\boldsymbol{x})}.$$

$$L(D^*, g_\theta) = 2JSD(\mathbb{P}_r \| \mathbb{P}_g) - 2\log 2$$



Generative Adversarial Network (GAN)

Drawbacks-1： Gradient Vanishing

$$L(D^*, g_\theta) = 2JSD(\mathbb{P}_r \| \mathbb{P}_g) - 2\log 2$$

JSD

$$D_{JS}(p|q) = D_{JS}(q|p) = \frac{1}{2}D_{KL}(p|r) + \frac{1}{2}D_{KL}(q|r)$$

$$r = \tfrac{1}{2}(p+q)$$

The JS divergence for the two distributions P_r and P_g is (almost) always log2 because P_r and P_g hardly can overlap (Arjovsky & Bottou, 2017, Theorem 2.1~2.3)

Generative Adversarial Network (GAN)

Drawbacks-2： Training Instability

It is heuristically motivated that generator can still learn even when discriminator successfully rejects all generator samples, but not theoretically guaranteed

using the alternative form of the objective of ***G***

$$\mathbb{E}_{z\sim p(z)}\left[-\log D(g_\theta(z))\right]$$

will result in gradient unstable issue and mode missing problem because *theoretically*, when ***D*** is *optimal*, minimizing the loss is equal to minimizing the *KL divergence* meanwhile maximizing the *JS divergence* (Arjovsky & Bottou, 2017, Theorem 2.5):

$$KL(\mathbb{P}_{g_\theta}\|\mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta}\|\mathbb{P}_r)]$$

Generative Adversarial Network (GAN)

Drawbacks-3： Mode Collapse

minimizing the *KL divergence* only is biased:

$$KL(\mathbb{P}_{g_\theta} \| \mathbb{P}_r) - 2JSD(\mathbb{P}_{g_\theta} \| \mathbb{P}_r)]$$

because *KL divergence* is asymmetric, and thus it is not equally treated when ***G*** generates an unreal sample and when ***G*** fails to generate real sample

Therefore, ***G*** will generate too many few-mode (less diverse) but real samples , a safer strategy

Generative Adversarial Network (GAN)

Wasserstein GANs (Arjovsky et al., 2017)

Wasserstein-1 Distance (Earth-Mover Distance):

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\| \right]$$

- *Why is it superior to KL and JS divergence?*

Generative Adversarial Network (GAN)

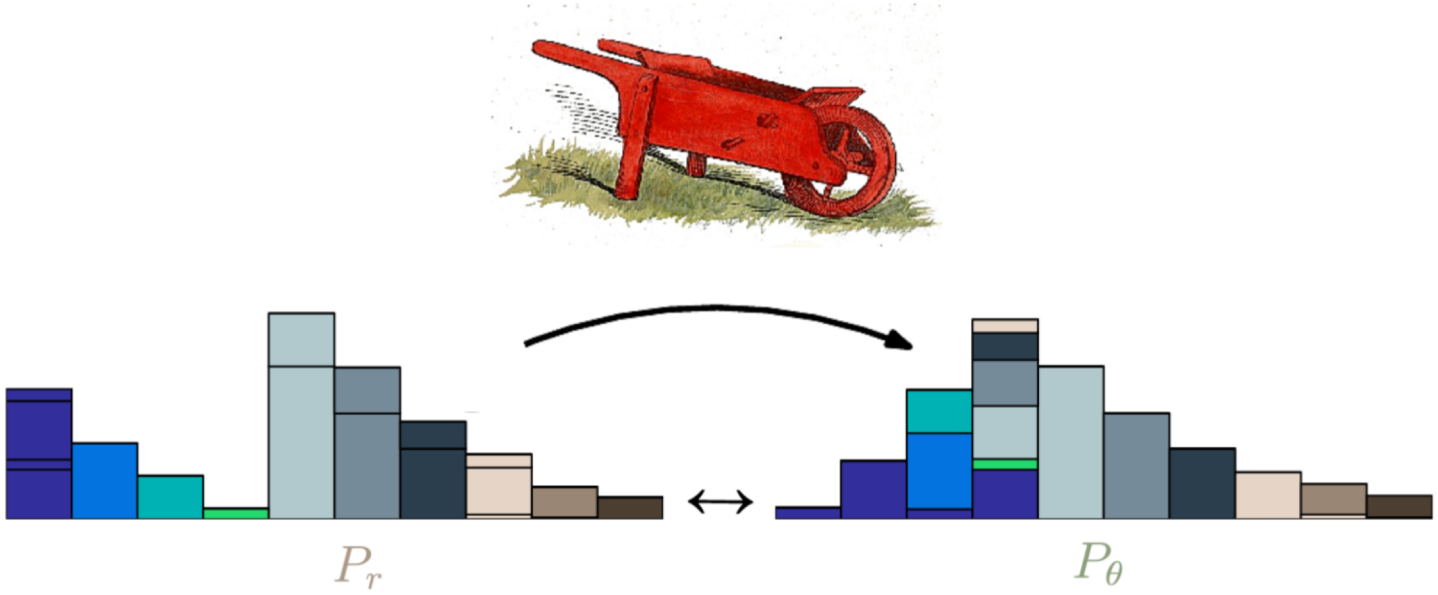
- Wasserstein-1 Distance (Earth-Mover Distance):

$$W(\mathbb{P}_r, \mathbb{P}_g) = \inf_{\gamma \in \Pi(\mathbb{P}_r, \mathbb{P}_g)} \mathbb{E}_{(x,y)\sim\gamma}\left[\|x - y\| \right]$$

where $\Pi(\mathbb{P}_r, \mathbb{P}_g)$ denotes the set of all joint distributions $\gamma(x, y)$ whose marginals are respectively $\mathbb{P}_r$ and $\mathbb{P}_g$. Intuitively, $\gamma(x, y)$ indicates how much "mass" must be transported from x to y in order to transform the distributions $\mathbb{P}_r$ into the distribution $\mathbb{P}_g$. The EM distance then is the "cost" of the optimal transport plan.

Generative Adversarial Network (GAN)

- Wasserstein-1 Distance (Earth-Mover Distance):



(Vincent Herrmann's blog)

- It is continuous everywhere and
- differentiable almost everywhere.
- *And most importantly, it can reflect the distance of two distributions even if they do not overlap, and thus can provide meaningful gradients*

Wasserstein GANs

- This new value function of WGAN gives rise to the additional requirement that the discriminator must lie within in the space of 1-Lipschitz functions:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right]$$

- In other words, ***D*** is the set of 1-Lipschitz functions

  - Lipschitz continuity

Wasserstein GANs

Lipschitz Continuity

- real-value function: $f : R \rightarrow R$
- positive constant: K

$$|f(x_1) - f(x_2)| \leq K|x_1 - x_2|$$

- In other words, a Lipschitz continuous function has bounded first derivative. Intuitively, the slope of a KK-Lipschitz function never exceeds KK, for a more general definition of slope.

$$d_Y(f(x_1), f(x_2)) \leq K d_X(x_1, x_2)$$

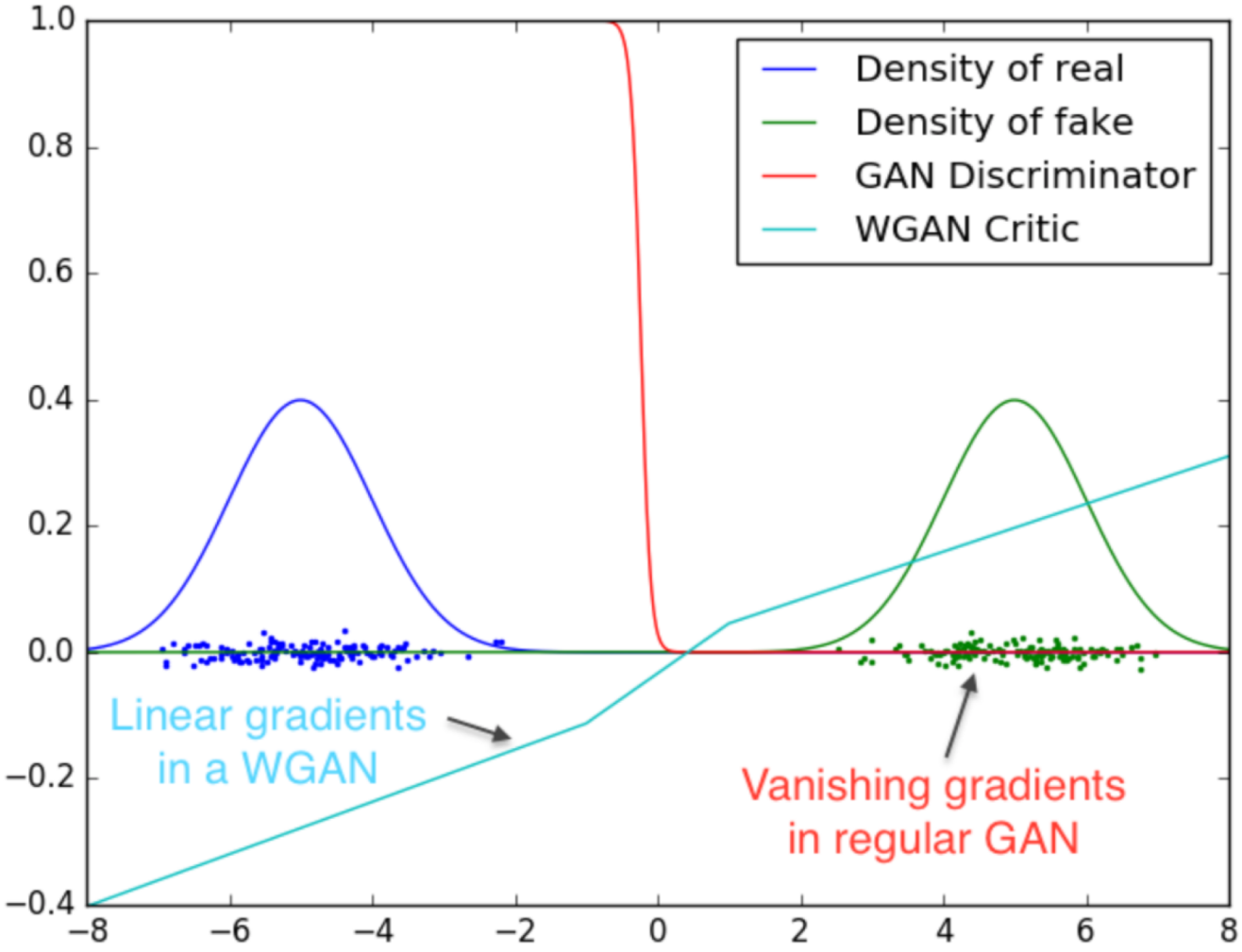
Wasserstein GANs

- This new value function of WGAN gives rise to the additional requirement that the discriminator must lie within in the space of 1-Lipschitz functions:

$$\min_{G} \max_{D \in \mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{x} \sim \mathbb{P}_r} \left[D(\boldsymbol{x}) \right] - \mathop{\mathbb{E}}_{\tilde{\boldsymbol{x}} \sim \mathbb{P}_g} \left[D(\tilde{\boldsymbol{x}}) \right]$$

- To satisfy this requirement, WGAN enforces the weights of ***D*** lie within a compact space *[-c, c]* by applying weight clipping

Wasserstein GANs



Improving Variational Encoder-Decoders in Dialogue Generation

VAE

$$
\begin{aligned}
&-\log p_\theta(x) \leq -\log p_\theta(x) + \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) \\
&= -\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x|z)] + \mathrm{KL}(q_\phi(z|x||p(z))
\end{aligned}
$$

CVAE

$$
-\mathbb{E}_{q_\phi(z|x,c)}[\log p_\theta(x|c,z)] + \mathrm{KL}(q_\phi(z|x,c)\|p_\theta(z|c))
$$

CVAE can be rewrote as

$$
-\log \int_z p_\theta(z|c)p_\theta(x|z,c)dz + \mathrm{KL}(q_\phi(z|x,c)||p_\theta(z|x,c))
$$

Let's first take a look at the first item, $\log \int_z p_\theta(z|c)p_\theta(x|z,c)dz = \log p_\theta(x|c)$. When the family of $p_\theta(x|z,c)$ is complex enough and includes the real distribution of x, the optimal value of this item is $p(x|c)$ and the reliance on z is not necessary. However, reliance on z provides the model with a chance of taking advantage of $z's$ distribution and reduces the complexity requirement for the distribution family $p_\theta(x|z,c)$. For

The weaker the decoder family is, the more it will be biased to utilizing latent variables.

A more flexible prior distribution pθ(z) will also increase the chance as it provides more possibilities for the utilisation 。

CVAE can be rewrote as

$$-\log \int_z p_\theta(z|c)p_\theta(x|z,c)dz + \mathrm{KL}(q_\phi(z|x,c)||p_\theta(z|x,c))$$

KL divergence

$$p_\theta(z|x,c) = \frac{p_\theta(x|z,c)p_\theta(z|c)}{p_\theta(x|c)}$$

modelled by $q_\phi(z|x)$ (Hinton et al. 1995). However, when x represents sentences with variable length, the value of $p_\theta(x|z)$ vanishes greatly when the length grows, which makes the adjusting task much more difficult. This implies the second item will always prefer ignoring the latent variables, so long as the approximated posterior is not powerful enough to perfectly match the real posterior. The weaker the approximating

The weaker the decoder family is, the more it will be biased to utilizing latent variables.

A more flexible prior distribution pθ(z) will also increase the chance as it provides more possibilities for the utilisation 。

The weaker the approximating posterior distribution family is, the more it will be biased to ignoring latent variables

Proposed Method

Original: **Adversarial Encoder-Decoder (AED)**

(real posterior)

$$-\mathbb{E}_{q_\phi(z|c,x)} p_\theta(x|c,z) + JS(q_\phi(z|c)||p_\theta(z|c))$$

(AE phase) (GAN phase)

$p_\theta(z|c)$ are implicitly defined by passing context-dependent Gaussian random variables ε through multi-layer perceptions

New: **Replacing GAN with VAE**

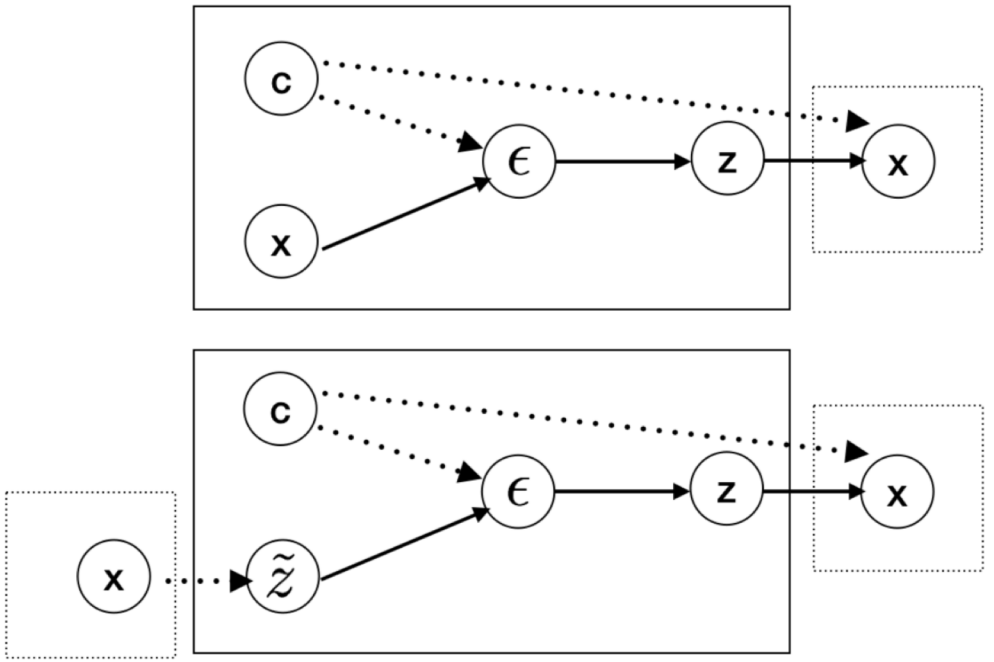
$$-\mathbb{E}_{q_\phi(\epsilon|c,\tilde{z})} p_\theta(z|c,\epsilon) + KL(q_\phi(\epsilon|c,\tilde{z})||p_\theta(\epsilon|c))$$

(approximated posterior)

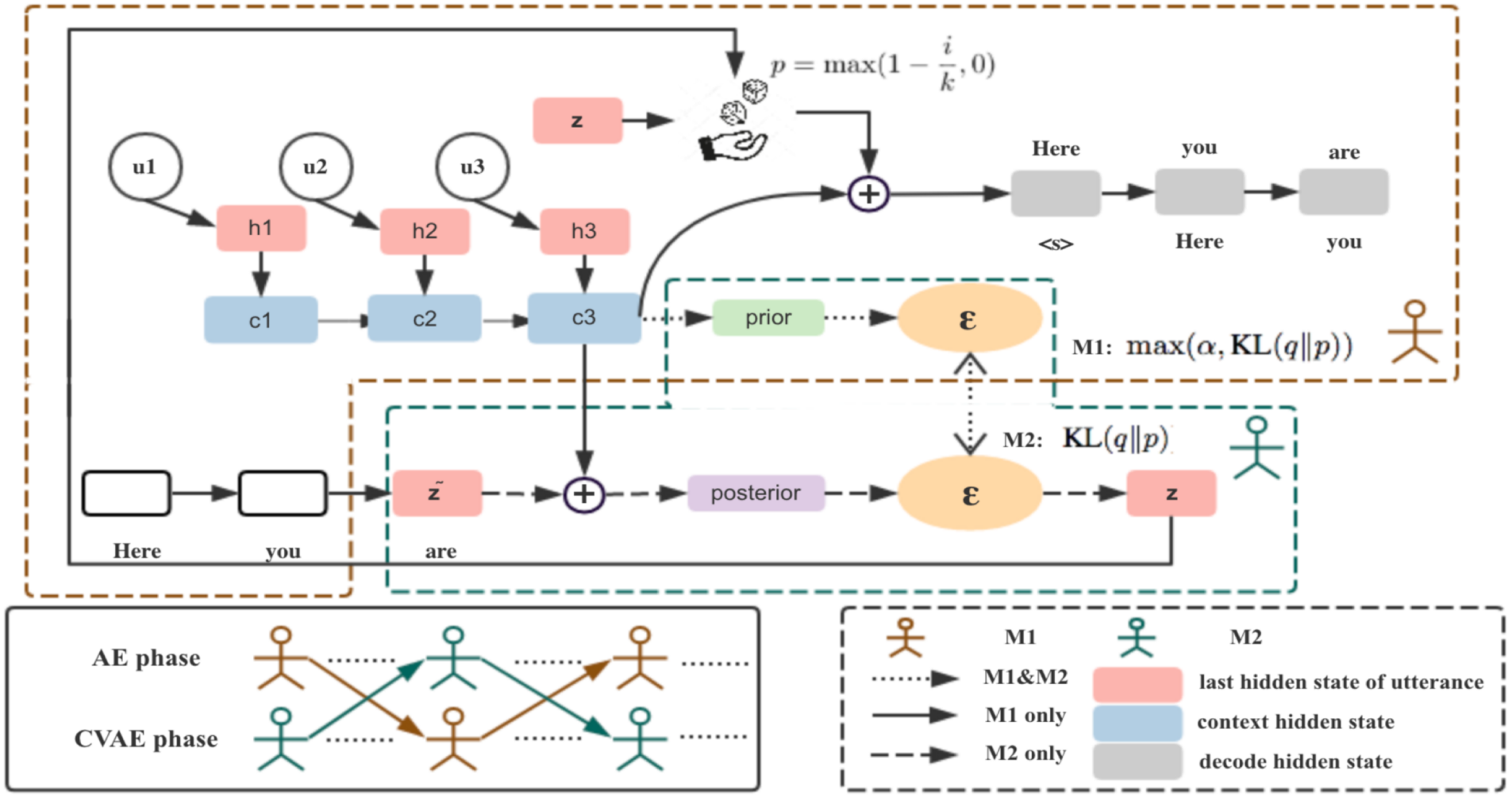
real hidden vector $\tilde{z}$ or the noisy z

The accuracy of the CVAE objective relies on the matching degree of $q_\phi(\epsilon|c,\tilde{z})$ and $p_\theta(\epsilon|c,\tilde{z})$

Original and New have the same global optimum



Proposed Method



$$-\mathbb{E}_{q_\phi(\epsilon|c,\tilde{z})} p_\theta(z|c,\epsilon) + KL(q_\phi(\epsilon|c,\tilde{z})||p_\theta(\epsilon|c))$$

Training Process

CVAE phase

$$\min_{\phi} \mathrm{KL}(q_\phi(\epsilon|\tilde{z}, c)\|p_\phi(\epsilon|c)) + \frac{1}{2}\mathbb{E}_{q_\phi(\epsilon|\tilde{z},c)}||g_\phi(\epsilon) - \tilde{z}||_2^2;$$

$$\tilde{z} = f_\theta(x)$$

A sample $\tilde{z}$ is obtained from the AE

The corresponding latent variable z is sampled from the posterior distribution $q_\phi(z|\tilde{z}, c)$ provided by the CVAE part

AE phase

$$\min_{\theta} \max(\alpha, \mathrm{KL}(q_\phi(\epsilon|\tilde{z}, c)\|p_\phi(\epsilon|c)))$$

$$- \mathbb{E}_{q_\phi(z|\tilde{z},c)}[log(p_\theta(x|z, c))];$$

$$\tilde{z} = f_\theta(x), z = (1-p)g_\phi(\epsilon) + p\tilde{z}$$

The output of the CVAE part are latent variables, which can represent a much broader distribution family than mean-field Gaussian

DialogWAE: Multimodal Response Generation with Conditional Wasserstein Auto-Encoder

Data-driven Conversation

Safe Response Problem

Fail to generate meaningful, diverse on-topic responses

the Posterior Collapse Problem

The decoder learns to ignore the latent variable and degrades to a vanilla RNN

GAN-based Methods

However, training with REINFORCE has been observed to be unstable due to the high variance of the gradient estimate

A new approach injects diversity in the word level rather than the level of the whole responses

Our main contributions are two-fold: (1) A novel GAN-based model for neural dialogue modeling, which employs GAN to generate samples of latent variables. (2) A Gaussian mixture prior network to sample random noise from a multimodal prior distribution. To the best of our knowledge, the proposed DialogWAE is the first GAN conversation model that exploits multimodal latent structures.

of the whole responses. DialogWAE differs from exiting GAN conversation models in that it shapes the distribution of responses in a high level latent space rather than direct tokens and does not rely on RL where the gradient variances are large.

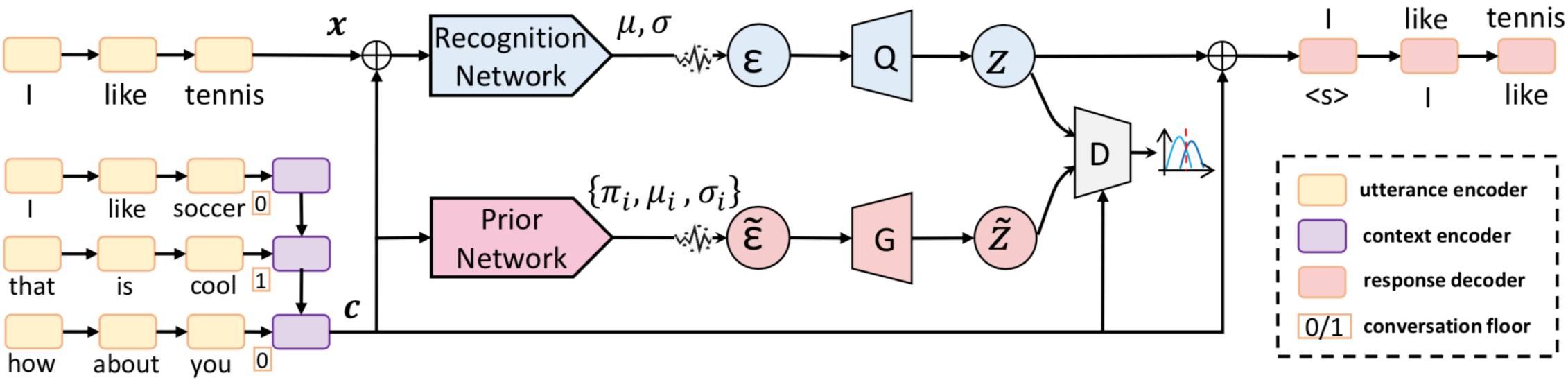


Figure 1: Architecture of DialogWAE

Let d=[u_1, ..., u_k] denote a dialogue of k utterances where u_i=[w_1, ..., $w_{|u_i|}$] represents an utterance and w_n denotes the n-th word in u_i. Let c=[u_1, ..., u_{k-1}] denote a dialogue context, the k-1 historical utterances, and x=u_k be a response which means the next utterance. Our goal is to estimate the conditional distribution $p_\theta(x|c)$.

Log-likelihood of a response

$$p_\theta(x|c) = \int_z p(x|c,z)p(z|c)d_z.$$

Goal

$$\min_{\theta,\phi,\psi} -E_{q_\phi(z|x,c)} \log p_\psi(x|z,c) + W(q_\phi(z|x,c)||p_\theta(z|c)),$$

Evidence Lower Bound (ELBO)

$$\begin{aligned}\log p_\theta(x|c) &= \log \int_z p(x|c,z)p(z|c)d_z \\ &\geq \ell(x,c) = \mathbf{E}_{z\sim q_\phi(z|x,c)}[\log p_\psi(x|c,z)] - \mathrm{KL}(q_\phi(z|x,c)||p(z|c)),\end{aligned}$$

Reconstruction loss

$$\mathcal{L}_{rec} = -E_{z=Q(\epsilon),\epsilon\sim \mathrm{RecNet}(x,c)} \log p_\psi(x|c,z)$$

approximate posterior sample $z\sim q_\phi(z|c,x)$

Discriminator loss

prior network and *recognition network*

$$\tilde{z} = G_\theta(\tilde{\epsilon}), \quad \tilde{\epsilon} \sim \mathcal{N}(\epsilon; \tilde{\mu}, \tilde{\sigma}^2 I), \quad \begin{bmatrix}\tilde{\mu} \\ \log \tilde{\sigma}^2\end{bmatrix} = \tilde{W} f_\theta(c) + \tilde{b}$$

$$z = Q_\phi(\epsilon), \quad \epsilon \sim \mathcal{N}(\epsilon; \mu, \sigma^2 I), \quad \begin{bmatrix}\mu \\ \log \sigma^2\end{bmatrix} = W g_\phi(\begin{bmatrix}x \\ c\end{bmatrix}) + b,$$

$$\mathcal{L}_{disc} = E_{\epsilon\sim \mathrm{RecNet}(x,c)}[D(Q(\epsilon),c)] - E_{\tilde{\epsilon}\sim \mathrm{PriNet}(c)}[D(G(\tilde{\epsilon}),c)]$$

situations [Sato *et al.*, 2017], topics and sentiments. A random noise with normal distribution could restrict the generator to output a latent space with a single dominant mode due to the single-modal nature of Gaussian distribution. Consequently, the generated responses could follow simple prototypes.

Mixture of Gaussian distributions

$$\mathrm{GMM}(\{\pi_k, \mu_k, \sigma_k^2 I\}_{k=1}^K)$$

$$p(\epsilon|c) = \sum_{k=1}^{K} v_k \mathcal{N}(\epsilon; \mu_k, \sigma_k^2 I),$$

$$\boldsymbol{\pi}_k = \frac{\exp(e_k)}{\sum_{i=1}^K \exp(e_i)}, \quad \text{where} \quad \begin{bmatrix} e_k \\ \mu_k \\ \log \sigma_k^2 \end{bmatrix} = W f_\theta(c) + b$$

Gumbel-Softmax re-parametrization

$$v_k = \frac{\exp((e_k + g_k)/\tau)}{\sum_{i=1}^K \exp((e_i + g_i)/\tau)},$$

$$g_i = -\log(-\log(u_i)), u_i \sim U(0, 1)$$

Algorithm 1: DialogWAE Training (UEnc: utterance encoder; CEnc: context encoder; RecNet: recognition network; PriNet: prior network; Dec: decoder) K=3, n_{disc}=5 in all experiments

In: a dialog corpus $\mathcal{D}$=$\{(c_i, x_i)\}_{i=1}^{|\mathcal{D}|}$, the number of prior modes K, discriminator iterations n_{critic}

Initialize $\{\theta_{\text{UEnc}}, \theta_{\text{CEnc}}, \theta_{\text{PriNet}}, \theta_{\text{RecNet}}, \theta_Q, \theta_G, \theta_D, \theta_{\text{Dec}}\}$
while *not convergence* **do**
  Initialize $\mathcal{D}$
  while $\mathcal{D}$ has unsampled batches **do**
    Sample a mini-batch of N instances $\{(x_n, c_n)\}_{n=1}^{N}$ from $\mathcal{D}$
    Get the representations of context and response $\boldsymbol{x}_n$=UEnc(x_n), $\boldsymbol{c}_n$=CEnc(c_n)
    Sample $\boldsymbol{\epsilon}_n$ from RecNet($\boldsymbol{x}_n$, $\boldsymbol{c}_n$) according to Equation 4
    Sample $\hat{\boldsymbol{\epsilon}}_n$ from PriNet($\boldsymbol{c}_n$, K) according to Equation 8–10
    Generate z_n = Q($\boldsymbol{\epsilon}_n$), $\tilde{z}_n$ = G($\hat{\boldsymbol{\epsilon}}_n$)
    Update $\{\theta_Q, \theta_G, \theta_{\text{PriNet}}, \theta_{\text{RecNet}}\}$ by gradient ascent on discriminator loss

$$\mathcal{L}_{disc} = \frac{1}{N}\sum_{n=1}^{N} D(z_n, c_n) - \frac{1}{N}\sum_{n=1}^{N} D(\tilde{z}_n, c_n)$$

    for $i \in \{1, \cdots, n_{\text{critic}}\}$ **do**
      Repeat 5–9
      Update θ_D by gradient descent on the discriminator loss $\mathcal{L}_{disc}$ with gradient penalty
    end
    Update $\{\theta_{\text{UEnc}}, \theta_{\text{CEnc}}, \theta_{\text{RecNet}}, \theta_Q, \theta_{\text{Dec}}\}$ by gradient descent on the reconstruction loss

$$\mathcal{L}_{rec} = -\frac{1}{N}\sum_{n=1}^{N} \log p(x_n | z_n, c_n)$$

  end
end