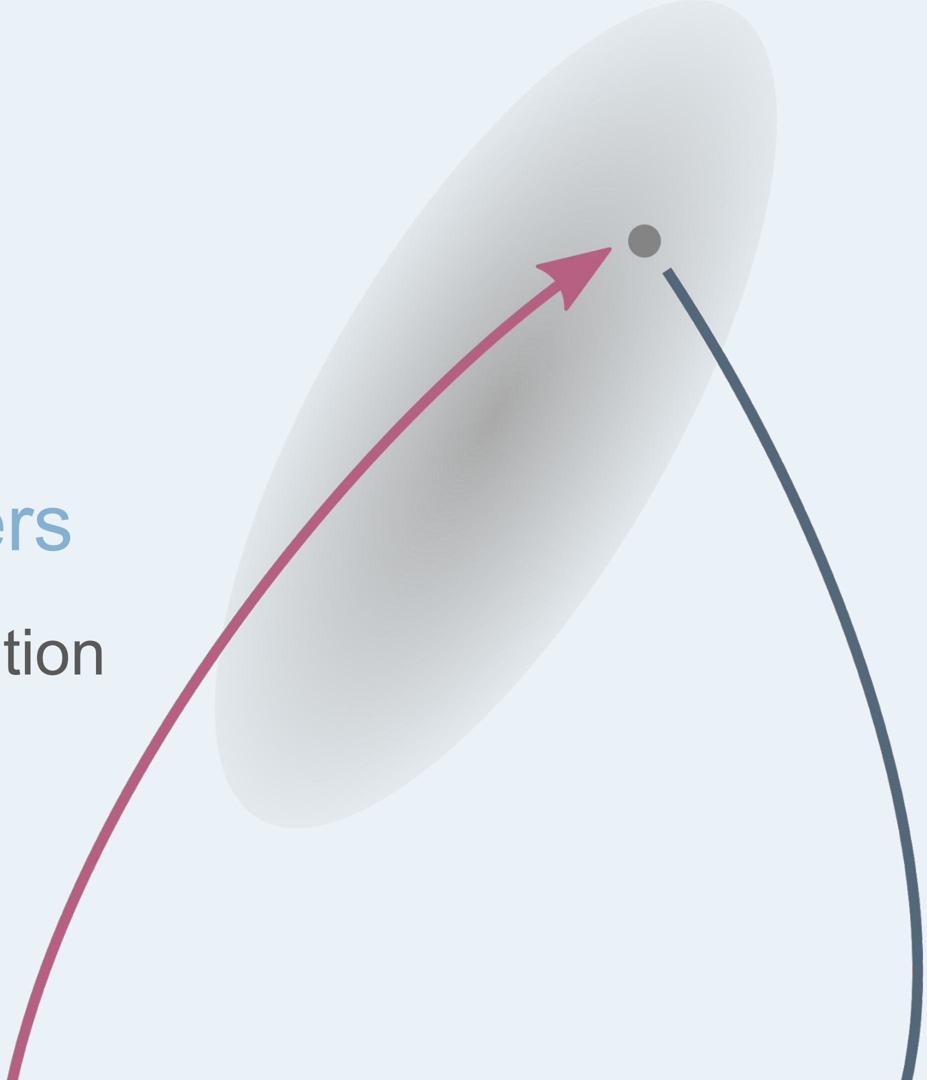
Variational Autoencoders

Discussion + an NLP Application



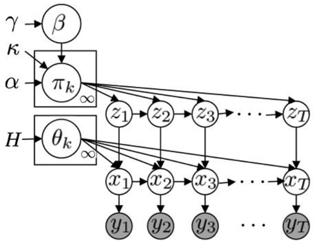
Kris Sankaran

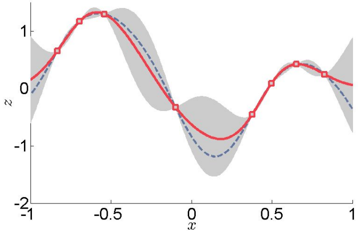
Probabilistic Inference ↔ Deep Learning

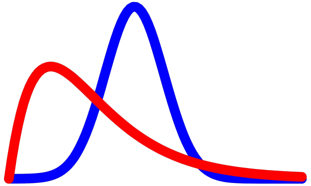
How can we blend,

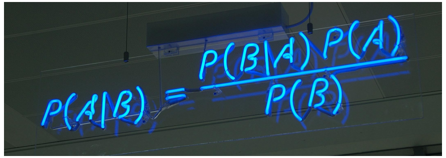
Rich probabilistic models

- Describe generative process
- Interpretable components
- Quantify uncertainty









Probabilistic Inference ↔ Deep Learning

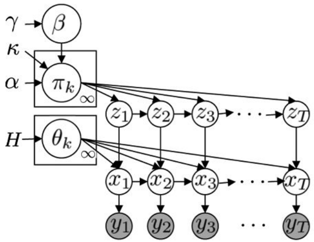
How can we blend,

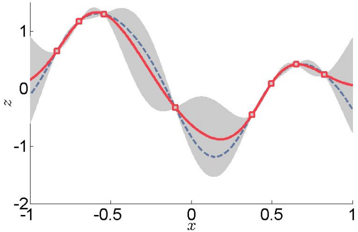
Rich probabilistic models

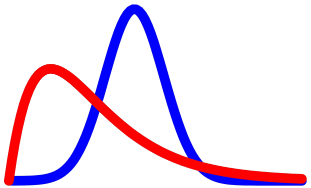
- Describe generative process
- Interpretable components
- Quantify uncertainty

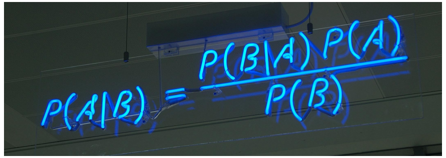
Powerful deep learning

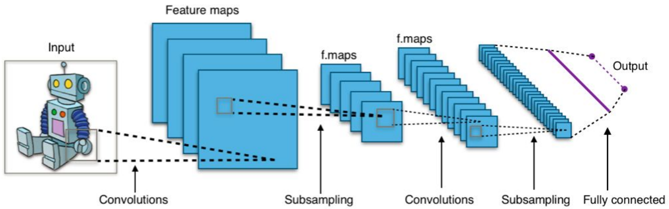
- State-of-the art performance
- Adaptable across problem types
- Scales to large datasets

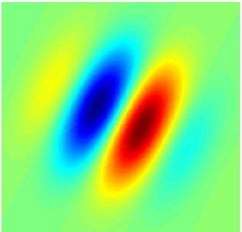


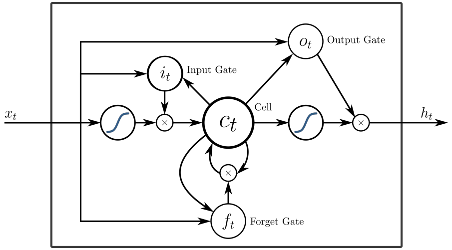


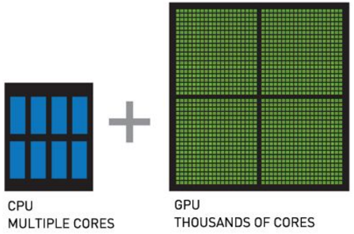












The Problem of Inference

- Generative models have interesting and useful properties

$$z \xrightarrow{\theta} x$$

The Problem of Inference

- Generative models have interesting and useful properties

$$z \xrightarrow{\theta} x$$

$$p(z) \qquad p_\theta(x|z)$$

prior likelihood

The Problem of Inference

- Generative models have interesting and useful properties

$$z \xrightarrow{\theta} x$$

$$p(z) \qquad p_\theta(x|z)$$

prior likelihood

'a crossed seven' →

The Problem of Inference

- Generative models have interesting and useful properties

$$z \xrightarrow{\theta} x$$

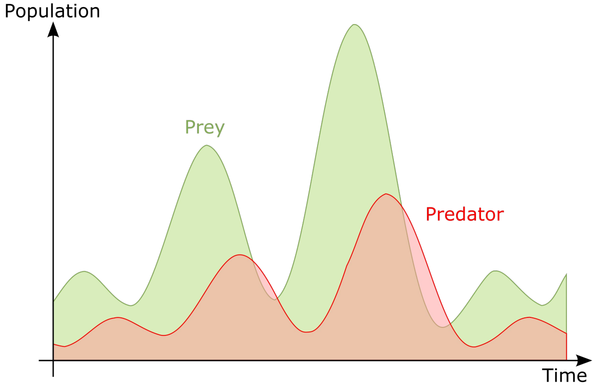
$$p(z) \qquad p_\theta(x|z)$$

prior   likelihood

'a crossed seven' →

$$\frac{dx}{dt} = \alpha x - \beta xy$$
$$\frac{dy}{dt} = \delta xy - \gamma y$$

→



The Problem of Inference

- The difficulty of using them lies in inference

$$x \xrightarrow{?} z$$

$p(z)$ $p_\theta(x|z)$

prior likelihood

$$p(z|x) = \frac{p_\theta(x|z)p(z)}{\int p_\theta(x|z)p(z)dz}$$

posterior

The Problem of Inference

- The difficulty of using them lies in inference

$$x \xrightarrow{?} z$$

$$p(z) \qquad p_\theta(x|z)$$

prior likelihood

$$p(z|x) = \frac{p_\theta(x|z)p(z)}{\int p_\theta(x|z)p(z)dz}$$

posterior

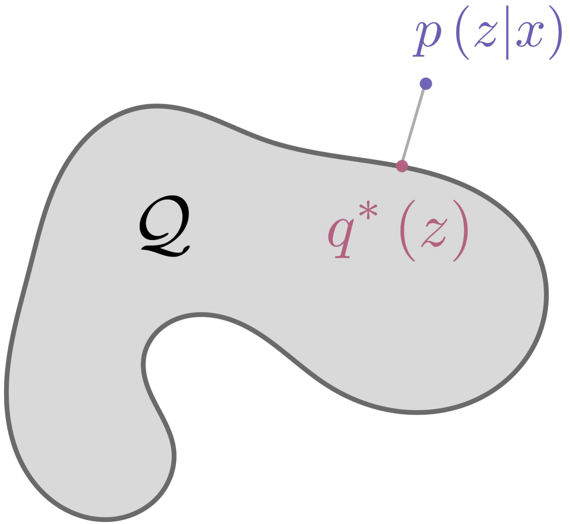
The Variational Idea

Integration → Optimization

[Wainwright and Jordan 2008]

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} D_{KL}\left(q(z), p(z|x)\right)$$

- Some families $\mathcal{Q}$ are easier to optimize over
- There is a trade-off between tractability and solution quality



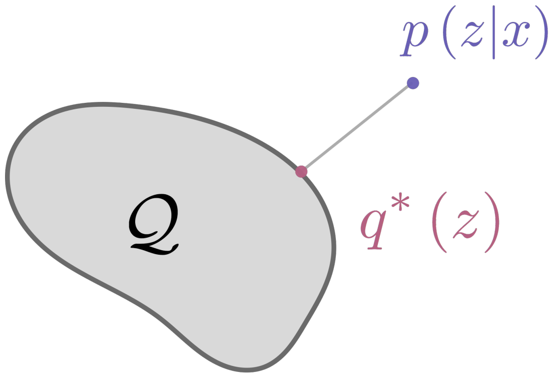
The Variational Idea

Integration $\rightarrow$ Optimization

[Wainwright and Jordan 2008]

$$q^*(z) = \arg\min_{q \in \mathcal{Q}} D_{KL}\left(q(z), p(z|x)\right)$$

- Some families $\mathcal{Q}$ are easier to optimize over
- There is a trade-off between tractability and solution quality



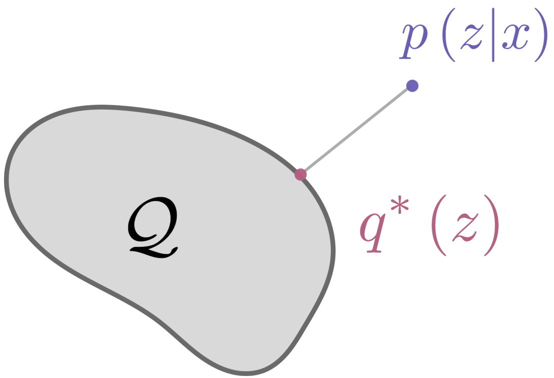
The Variational Idea

Integration → Optimization

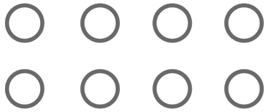
[Wainwright and Jordan 2008]

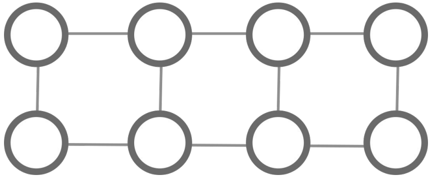
Typical choices of $\mathcal{Q}$

- Mean Field



$$\prod_{i=1}^{n} q_i(z_i)$$





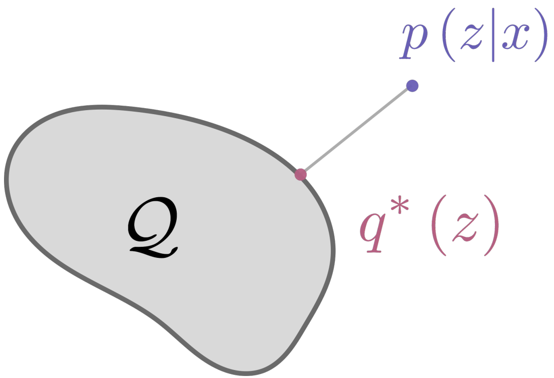
The Variational Idea

Integration → Optimization

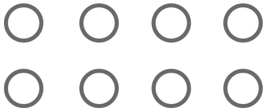
[Wainwright and Jordan 2008]

Typical choices of $\mathcal{Q}$

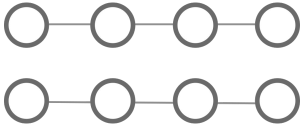
- Mean Field
- Structured Mean Field

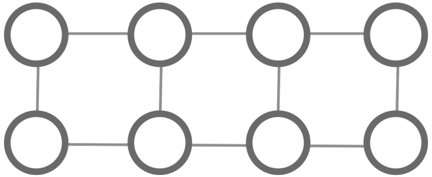


$$\prod_{i=1}^{n} q_i(z_i)$$



$$\prod_{i=1}^{n_G} q_g\left(z_{G(i)}\right)$$

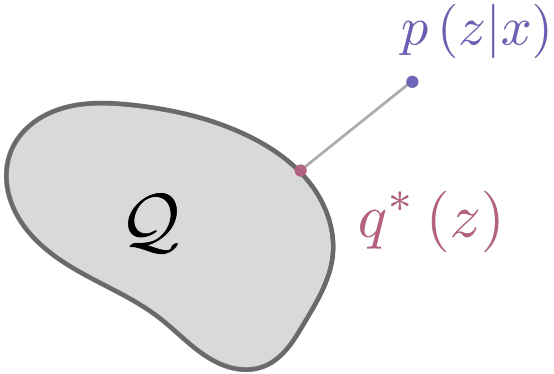




The Variational Idea

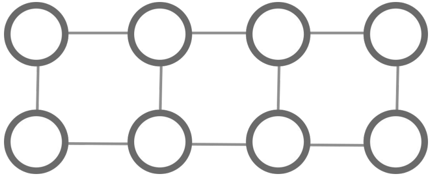
Integration → Optimization

[Wainwright and Jordan 2008]

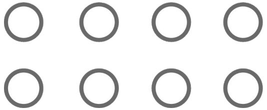


Typical choices of $\mathcal{Q}$

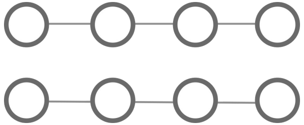
- Mean Field
- Structured Mean Field
- Global / Local factorizations



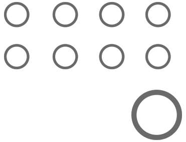
$$\prod_{i=1}^{n} q_i(z_i)$$



$$\prod_{i=1}^{n_G} q_g\left(z_{G(i)}\right)$$



$$q(z_{\text{global}}) \prod_{i=1}^{n} q_i(z_i)$$



Optimization

Typical strategies,

- Coordinate updates

$$q_i^*\left(z_i\right) \propto \exp\left(\mathbb{E}_{q_{-i}\left(z_{-i}\right)}\left[\log p\left(z_i | x, z_{-i}\right)\right]\right)$$

- For large data, only update minibatches (Stochastic Variational Inference [Hoffman+ 2013])
- For difficult expectations, can appeal to surrogate bounds [Jaakola and Jordan 1996]

Optimization

In Kingma and Welling [2014], the likelihood is a single layer MLP.

Typical strategies,

- Coordinate updates

$$q_i^*\left(z_i\right) \propto \exp\left(\mathbb{E}_{q_{-i}(z_{-i})}\left[\log p\left(z_i|x, z_{-i}\right)\right]\right)$$

- For large data, only update minibatches (Stochastic Variational Inference [Hoffman+ 2013])
- For difficult expectations, can appeal to surrogate bounds [Jaakola and Jordan 1996]

Optimization

In Kingma and Welling [2014], the likelihood is a single layer MLP.

Typical strategies,

This is not reasonable...

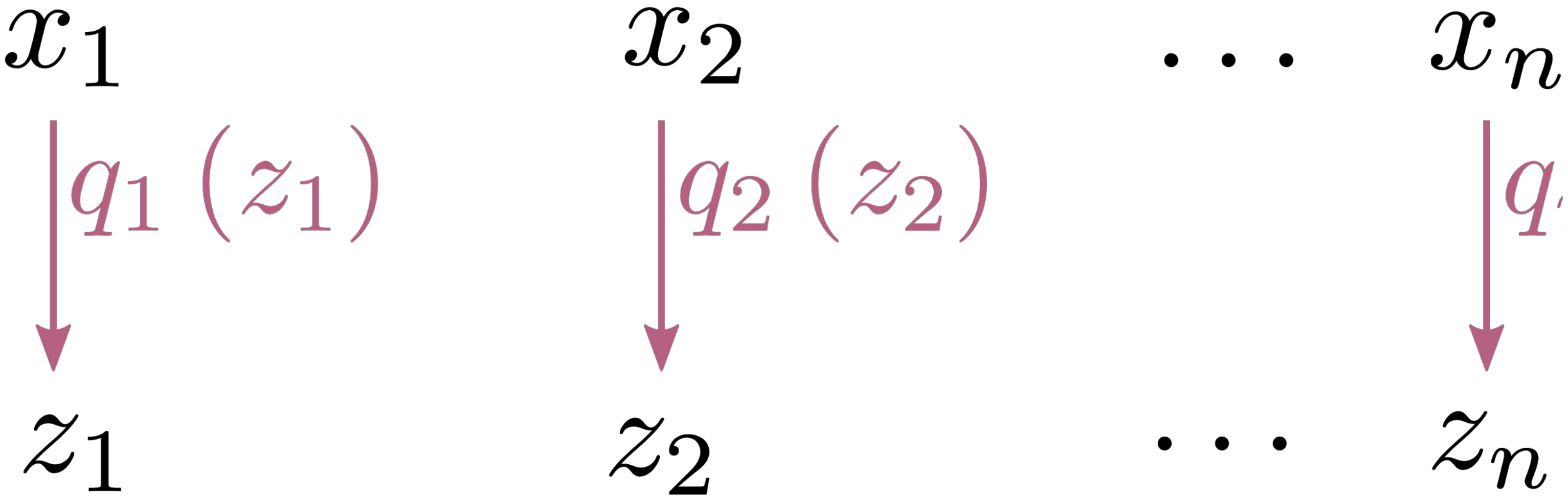
- Coordinate updates

$$q_i^*(z_i) \propto \exp\left(\mathbb{E}_{q_{-i}(z_{-i})}\left[\log p\left(z_i | x, z_{-i}\right)\right]\right)$$

- For large data, only update minibatches (Stochastic Variational Inference [Hoffman+ 2013])
- For difficult expectations, can appeal to surrogate bounds [Jaakola and Jordan 1996]

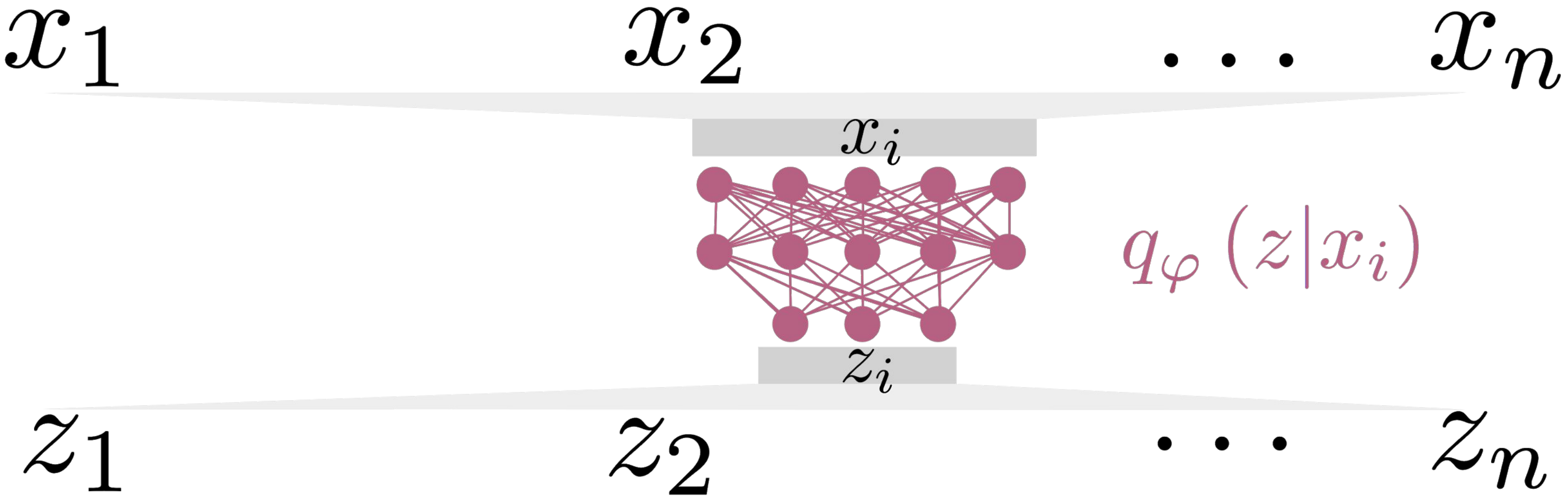
What to do? (1) Amortization

- **Typically**: Coordinate ascent on $\mathcal{Q}$, updating one q_i at a time
  - Nonparametric → Number of parameters grows with the data



What to do? (1) Amortization

- **Typically**: Coordinate ascent on $\mathcal{Q}$, updating one q_i at a time
  - Nonparametric → Number of parameters grows with the data
- **Instead**: Learn a mapping from data to latent variables
  - Parametric, but very flexible



What to do? (2) Reparameterization

- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

Minimizing the KL-divergence objective is equivalent to maximizing the Evidence Lower Bound (ELBO),

$$\mathcal{L}(q) = \mathbb{E}_{q(z)}\left[\log p(x, z)\right] + H(q)$$
$$= \mathbb{E}_{q(z)}\left[\log p(x|z)\right] - D_{KL}\left(q(z) || p(z)\right)$$

What to do? (2) Reparameterization

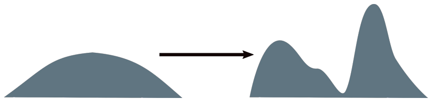
- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

Minimizing the KL-divergence objective is equivalent to maximizing the Evidence Lower Bound (ELBO),

$$\mathcal{L}(q) = \mathbb{E}_{q(z)}\left[\log p(x, z)\right] + H(q)$$
$$= \mathbb{E}_{q(z)}\left[\log p(x|z)\right] - D_{KL}\left(q(z) || p(z)\right)$$

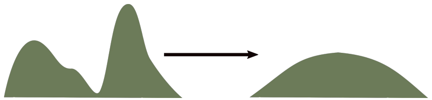
Reconstruction Measures

- Expected complete data log-likelihood
- Expected log-likelihood



Complexity Penalties

- Entropy
- Distance from prior

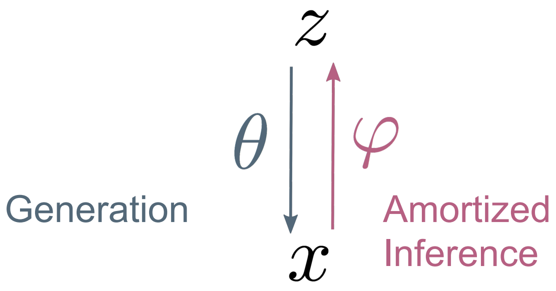


What to do? (2) Reparameterization

- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

$$\mathcal{L}(q) = \mathbb{E}_{q(z)}\left[\log p(x|z)\right] - D_{KL}\left(q(z)||p(z)\right)$$

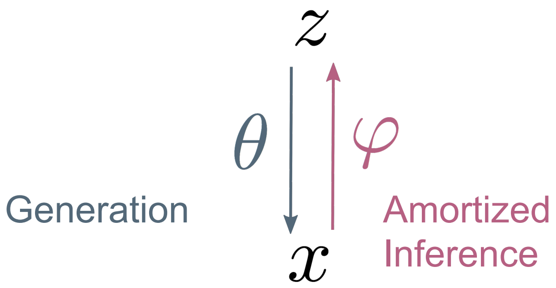
What to do? (2) Reparameterization



- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

$$\mathcal{L}(\varphi, \theta) = \mathbb{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\varphi(z|x)||p(z)\right)$$

What to do? (2) Reparameterization



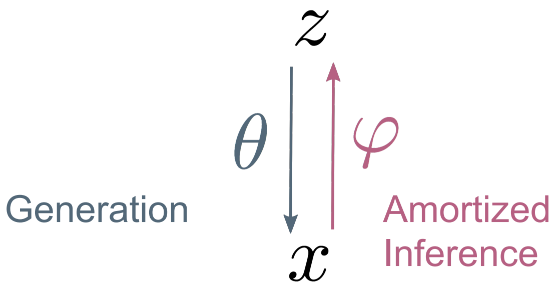
- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

$$\mathcal{L}(\varphi, \theta) = \mathbb{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\varphi(z|x) \| p(z)\right)$$

- Would like gradient updates of the form,

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \begin{pmatrix} \nabla_\varphi \mathcal{L}(\varphi, \theta) \\ \nabla_\theta \mathcal{L}(\varphi, \theta) \end{pmatrix} \Bigg|_{\varphi=\varphi_t, \theta=\theta_t}$$

What to do? (2) Reparameterization



- Mean-field updates are intractable
- **Idea**: Directly optimize using noisy gradients

$$\mathcal{L}(\varphi, \theta) = \mathbb{E}_{q_\varphi(z|x)}\left[\log p_\theta(x|z)\right] - D_{KL}\left(q_\varphi(z|x) \| p(z)\right)$$

- Would like gradient updates of the form,

Intractable

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \begin{pmatrix} \nabla_\varphi \mathcal{L}(\varphi, \theta) \\ \nabla_\theta \mathcal{L}(\varphi, \theta) \end{pmatrix}\Bigg|_{\varphi=\varphi_t, \theta=\theta_t}$$

Tractable

What to do? (2) Reparameterization

- **Reparameterization** allows *efficient* estimation of

$$\nabla_{\varphi} \mathbb{E}_{q_{\varphi}(z|x)} \left[f(z) \right]$$

Think $f(z) = \log p_{\theta}(x|z)$

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)]$$

- It's an alternative to the REINFORCE approach [Williams 1992],

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)] = \int f(z) \nabla_\varphi q_\varphi(z|x) dz$$

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right]$$

- It's an alternative to the REINFORCE approach [Williams 1992],

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right] = \int f(z) \nabla_\varphi \log q_\varphi(z|x)\, q_\varphi(z|x) dz$$

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right]$$

- It's an alternative to the REINFORCE approach [Williams 1992],

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right] \approx \frac{1}{N} \sum_{z \sim q_\varphi(z|x)} f(z) \nabla_\varphi \log q_\varphi(z|x)$$

but this unfortunately has very high variance…

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)]$$

- Instead, suppose we can reparameterize,

$$z \sim q_\varphi(z|x) \implies z \stackrel{D}{=} g_\varphi(\epsilon, x)$$

$$\epsilon \sim p(\epsilon)$$

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right]$$

- Instead, suppose we can reparameterize,

$$z \sim q_\varphi(z|x) \implies z \stackrel{D}{=} g_\varphi(\epsilon, x)$$

$$\epsilon \sim p(\epsilon)$$

φ modulates stochastic nodes

φ modulates deterministic nodes!

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right]$$

- Therefore,

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right] \stackrel{D}{=} \nabla_\varphi \mathbb{E}_{p(\epsilon)}\left[f(g_\varphi(\epsilon, x))\right]$$

φ modulates stochastic nodes

φ modulates deterministic nodes!

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right]$$

- Therefore,

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}\left[f(z)\right] \stackrel{D}{=} \mathbb{E}_{p(\epsilon)}\left[\nabla_\varphi f\left(g_\varphi(\epsilon, x)\right)\right]$$

φ modulates stochastic nodes

φ modulates deterministic nodes!

What to do? (2) Reparameterization

Think $f(z) = \log p_\theta(x|z)$

- **Reparameterization** allows *efficient* estimation of

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)]$$

- Therefore,

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)] \approx \frac{1}{N} \sum_{\epsilon \sim p(\epsilon)} \nabla_\varphi f(g_\varphi(\epsilon, x))$$

φ modulates stochastic nodes

φ modulates deterministic nodes!

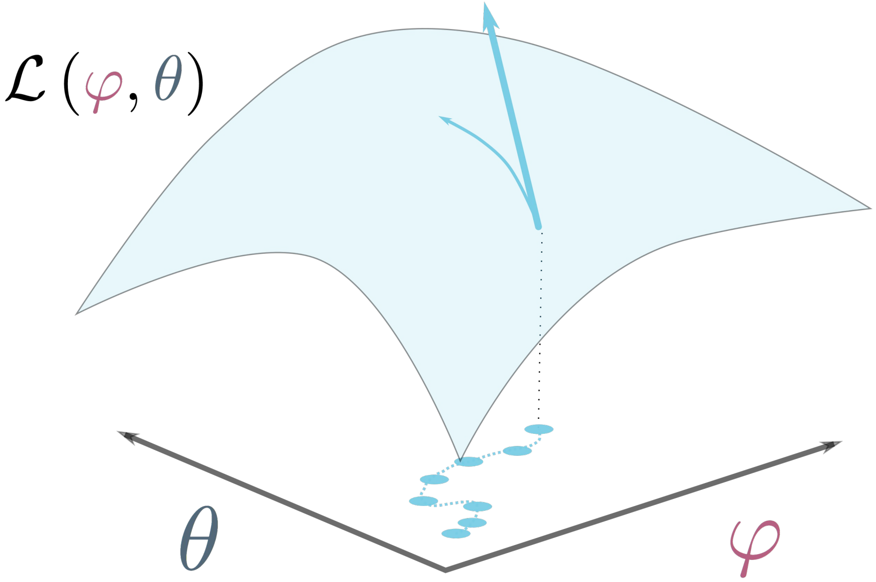
Algorithm Summary

- We now have everything in place to perform inference
- Our ideal algorithm has the form,

```
initialize
```
φ_0, θ_0

```
while not converged
    step along the gradient
```

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \begin{pmatrix} \nabla_\varphi \mathcal{L}(\varphi, \theta) \\ \nabla_\theta \mathcal{L}(\varphi, \theta) \end{pmatrix} \Bigg|_{\varphi=\varphi_t, \theta=\theta_t}$$



Algorithm Summary

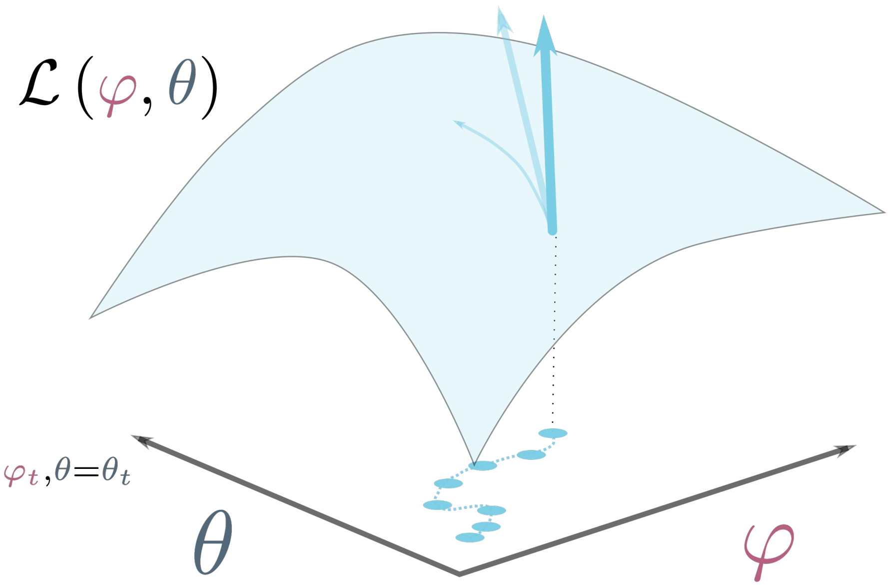
- We now have everything in place to perform inference
- It's better to use stochastic gradients

```
initialize
```
φ_0, θ_0

```
while not converged
    draw a minibatch
```
X^M

```
    step along the stochastic gradient
```

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \left. \begin{pmatrix} \nabla_\varphi \widehat{\mathcal{L}(\varphi, \theta)\left[X^M\right]} \\ \nabla_\theta \widehat{\mathcal{L}(\varphi, \theta)\left[X^M\right]} \end{pmatrix} \right|_{\varphi=\varphi_t, \theta=\theta_t}$$



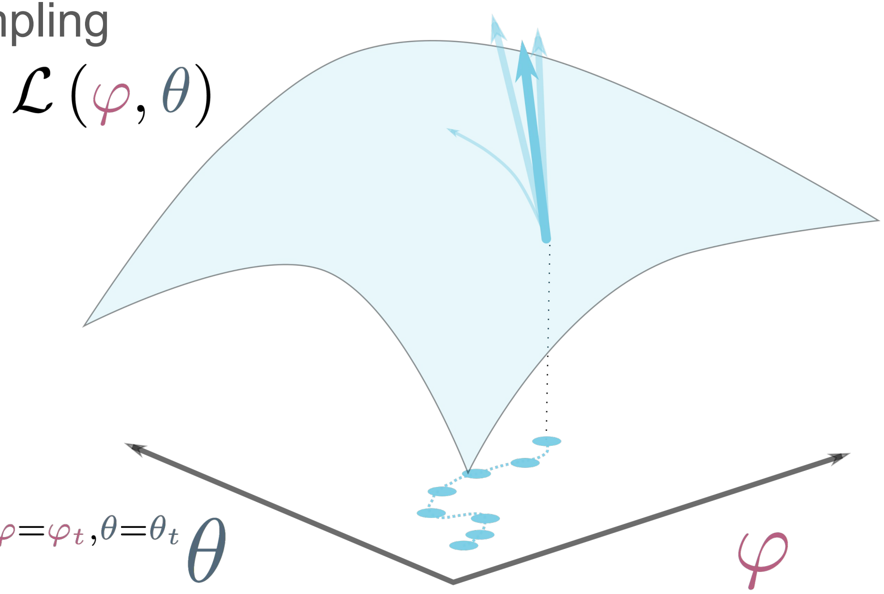
Algorithm Summary

- We now have everything in place to perform inference
- It's better to use stochastic gradients
- Reparameterization facilitates MC sampling

initialize φ_0, θ_0

while not converged
    draw a minibatch X^M
    Sample ϵ
    step along the **stochastic** gradient

$$\begin{pmatrix}\varphi_{t+1}\\ \theta_{t+1}\end{pmatrix} \leftarrow \begin{pmatrix}\varphi_t\\ \theta_t\end{pmatrix} + \eta \begin{pmatrix}\nabla_\varphi \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\\ \nabla_\theta \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\end{pmatrix}\Bigg|_{\varphi=\varphi_t,\theta=\theta_t}$$



Algorithm Cartoon

initialize φ_0, θ_0

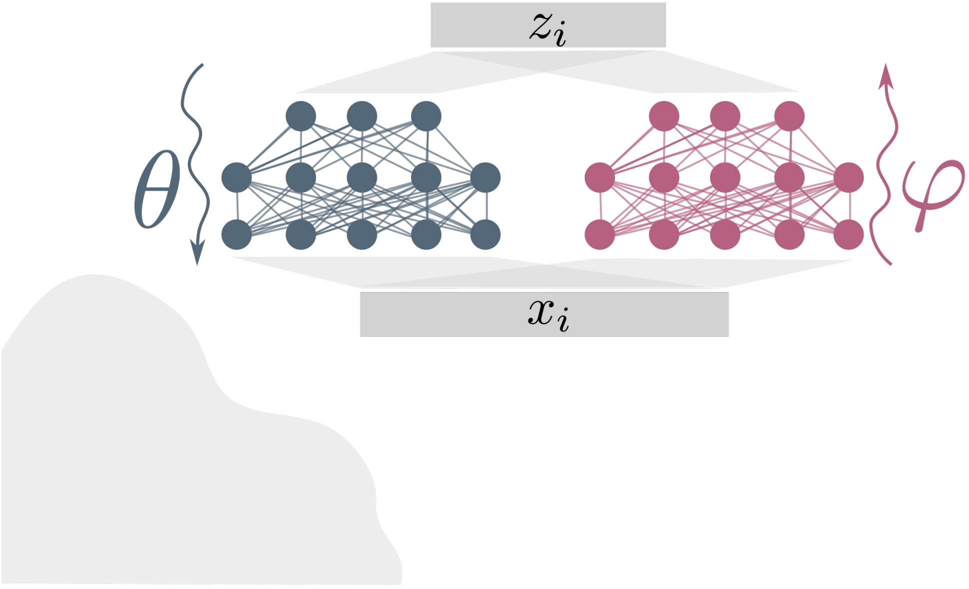
while not converged

    draw a minibatch X^M

    Sample ϵ

    step along the **stochastic** gradient

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \begin{pmatrix} \nabla_\varphi \widehat{\mathcal{L}(\varphi, \theta)} \left[X^M, \epsilon\right] \\ \nabla_\theta \widehat{\mathcal{L}(\varphi, \theta)} \left[X^M, \epsilon\right] \end{pmatrix} \Bigg|_{\varphi=\varphi_t, \theta=\theta_t}$$

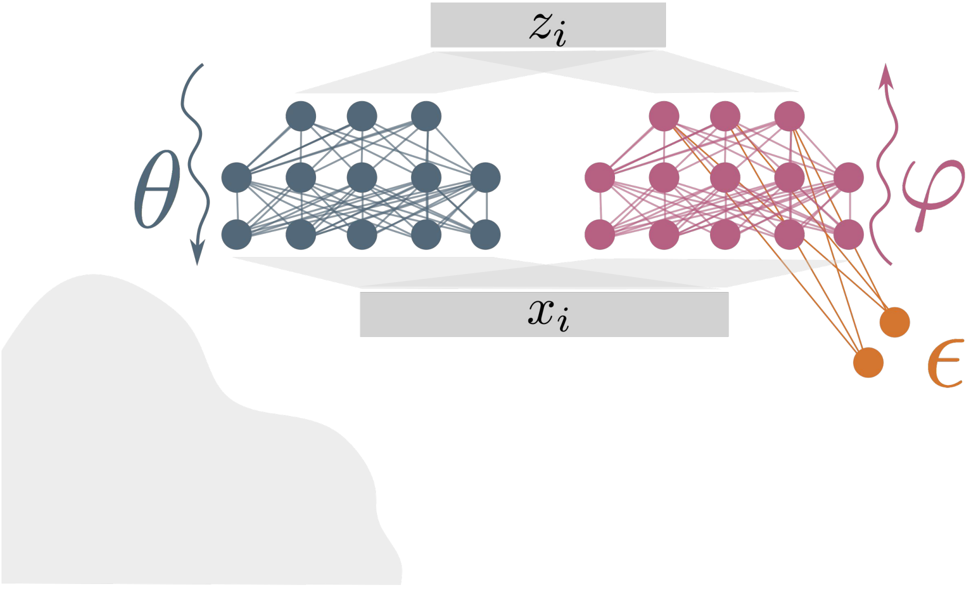


Algorithm Cartoon

```
initialize φ0, θ0

while not converged
    draw a minibatch X^M
     Sample ϵ
     step along the stochastic gradient
```

$$\begin{pmatrix}\varphi_{t+1}\\ \theta_{t+1}\end{pmatrix} \leftarrow \begin{pmatrix}\varphi_t\\ \theta_t\end{pmatrix} + \eta \left.\begin{pmatrix}\nabla_\varphi \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\\ \nabla_\theta \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\end{pmatrix}\right|_{\varphi=\varphi_t,\theta=\theta_t}$$

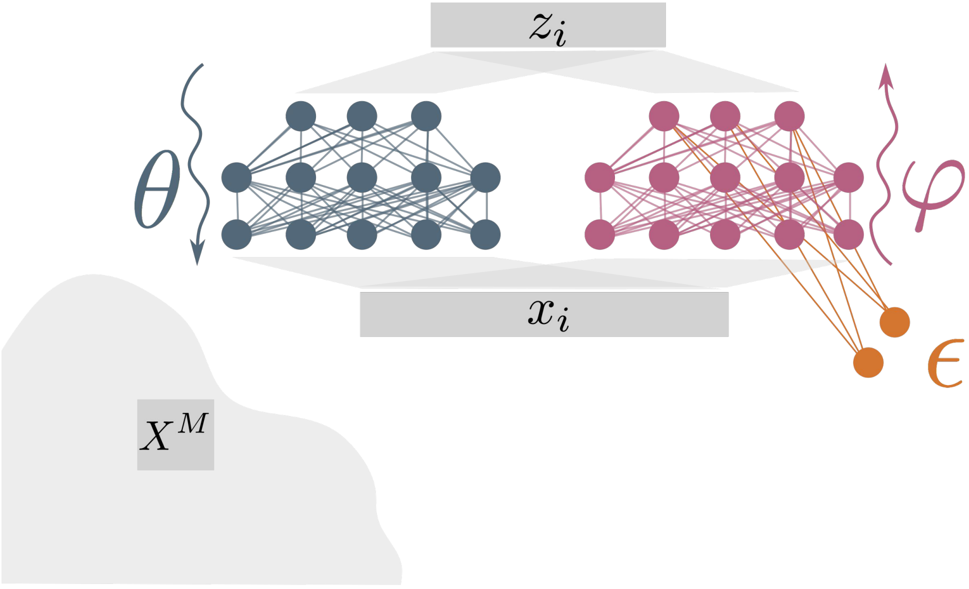


Algorithm Cartoon

```
initialize φ_0, θ_0

while not converged
    draw a minibatch X^M
    Sample ϵ
    step along the stochastic gradient
```

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_t \\ \theta_t \end{pmatrix} + \eta \left. \begin{pmatrix} \nabla_\varphi \widehat{\mathcal{L}(\varphi}, \theta) \left[X^M, \epsilon \right] \\ \nabla_\theta \widehat{\mathcal{L}(\varphi}, \theta) \left[X^M, \epsilon \right] \end{pmatrix} \right|_{\varphi=\varphi_t, \theta=\theta_t}$$

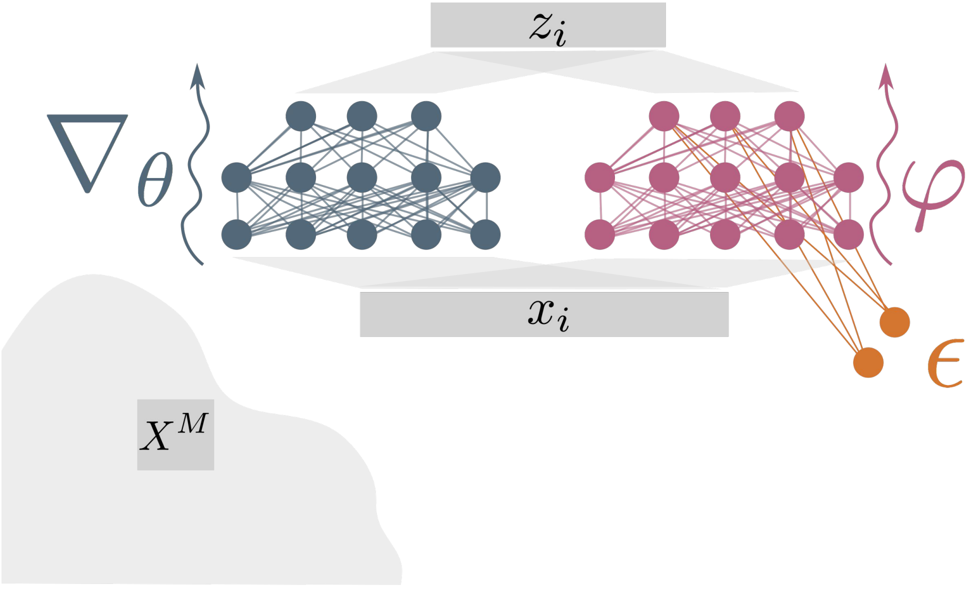


Algorithm Cartoon

```
initialize φ0, θ0

while not converged
    draw a minibatch X^M
    Sample ϵ
    step along the stochastic gradient
```

$$\begin{pmatrix} \varphi_{t+1} \\ \theta_{t+1} \end{pmatrix} \leftarrow \begin{pmatrix} \varphi_{t} \\ \theta_{t} \end{pmatrix} + \eta \begin{pmatrix} \nabla_\varphi \widehat{\mathcal{L}(\varphi,\theta)} \left[X^M, \epsilon\right] \\ \nabla_\theta \widehat{\mathcal{L}(\varphi,\theta)} \left[X^M, \epsilon\right] \end{pmatrix} \Bigg|_{\varphi=\varphi_t, \theta=\theta_t}$$

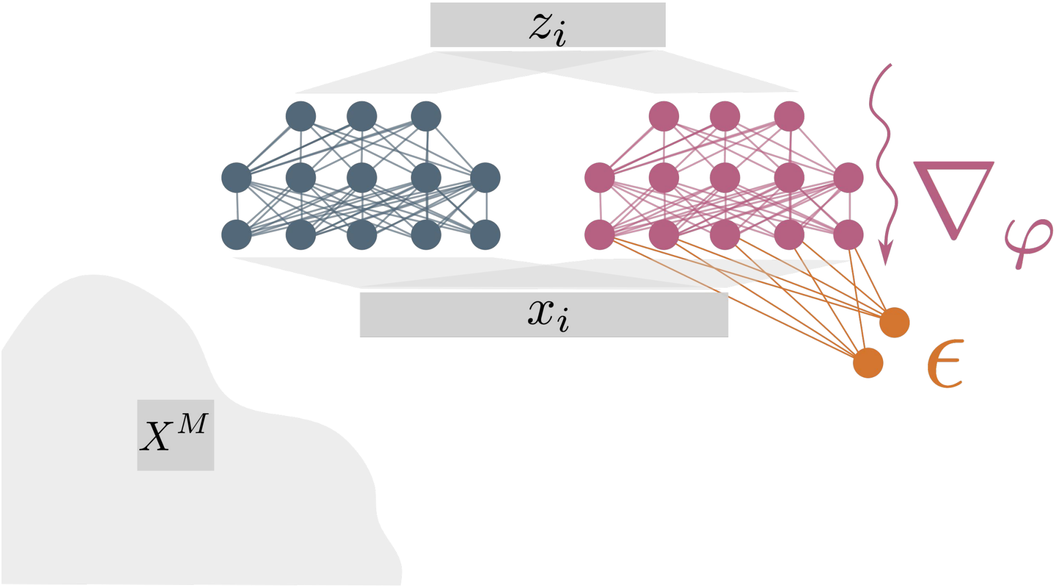


Algorithm Cartoon

initialize φ_0, θ_0

while not converged
    draw a minibatch X^M
    Sample ϵ
    step along the **stochastic** gradient

$$\begin{pmatrix}\varphi_{t+1}\\ \theta_{t+1}\end{pmatrix} \leftarrow \begin{pmatrix}\varphi_t\\ \theta_t\end{pmatrix} + \eta \left.\begin{pmatrix}\nabla_\varphi \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\\ \nabla_\theta \widehat{\mathcal{L}(\varphi,\theta)}\left[X^M,\epsilon\right]\end{pmatrix}\right|_{\varphi=\varphi_t,\theta=\theta_t}$$



Sequence-to-Sequence Modeling

Bowman+ [2015]: How can we combine the benefits of (1) **generative** and (2) **sequence** modeling?

- *Sampling / Uncertainty quantification*
- *Latent representations of full sequences*
- *Awareness of syntax and grammar*

Sequence-to-Sequence Modeling

Bowman+ [2015]: How can we combine the benefits of (1) **generative** and (2) **sequence** modeling?

- *Sampling / Uncertainty quantification*
- *Latent representations of full sequences*
- *Awareness of syntax and grammar*

Applications

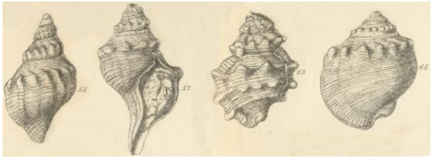
Text translation

Does this actually work?
これは実際には機能しますか？

Speech Recognition

“A B C”

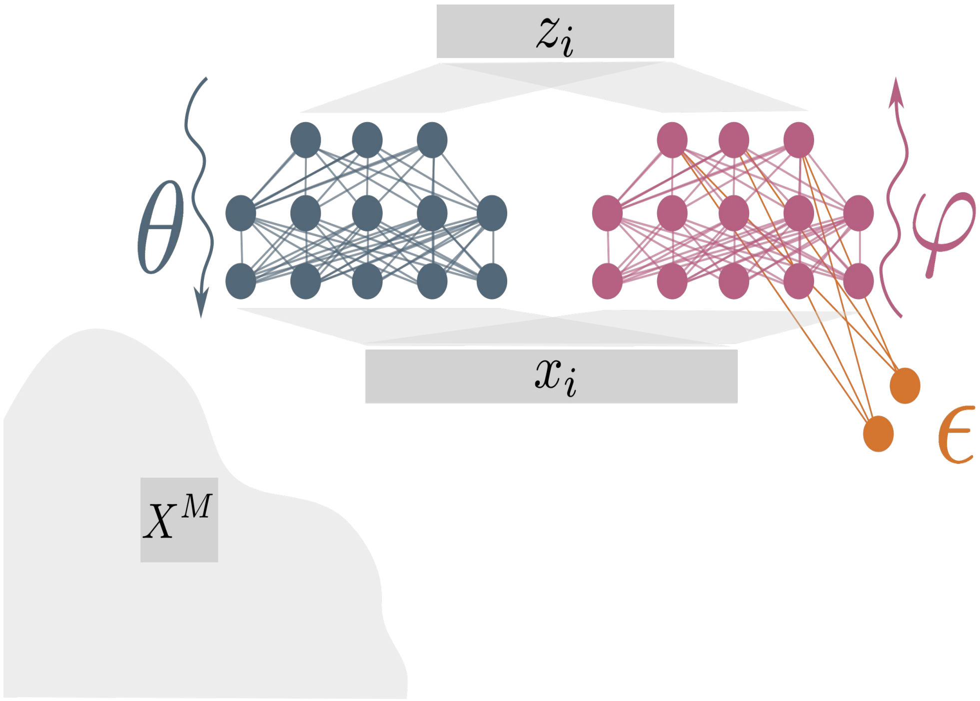
Image Captioning



Four sketches of seashells. [by Charles Darwin…]

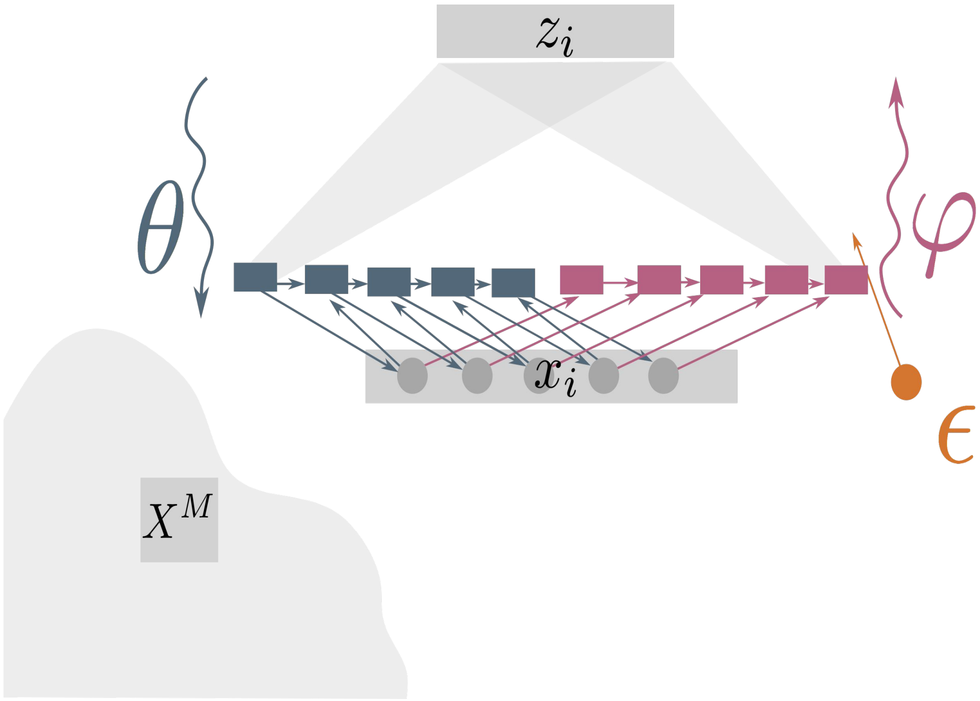
VAE Model

- Built from basic VAE approach



VAE Model

- Built from basic VAE approach
- The generator and inference networks are now RNNs with LSTM units



Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

KL Annealing

Word Dropout

Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

KL Annealing

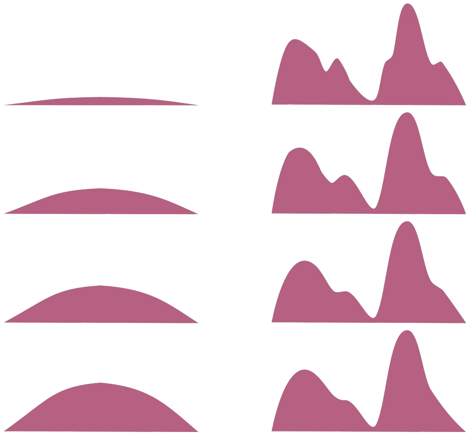
Using high KL from the very start prevents any learning in the encoder.

Word Dropout

Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

KL Annealing



Downweighting the KL early in training gives the encoder a chance to learn.

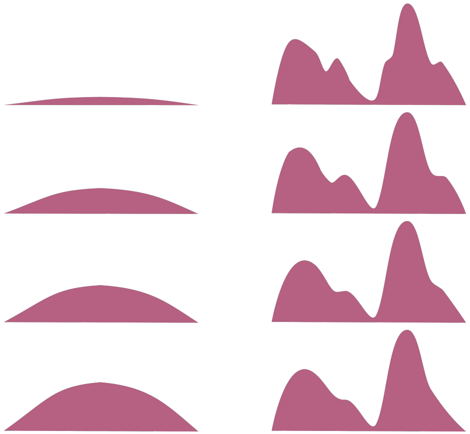
Analogy: Pruning in decision trees.

Word Dropout

Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

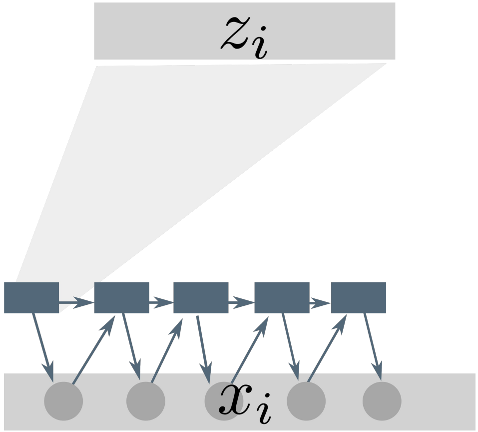
KL Annealing



Downweighting the KL early in training gives the encoder a chance to learn.

Analogy: Pruning in decision trees.

Word Dropout

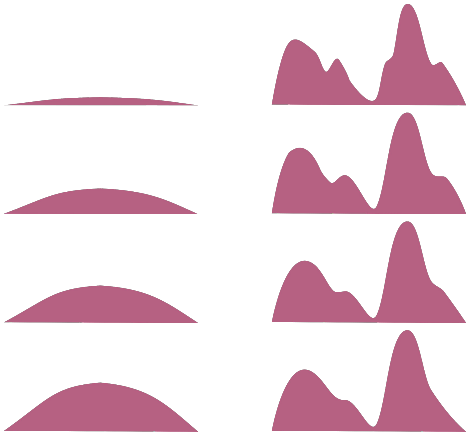


Access to previous words gives the decoder lots of power.

Optimization Hurdles

- The naive implementation fails!
- Decoder is too strong, encoder is too weak

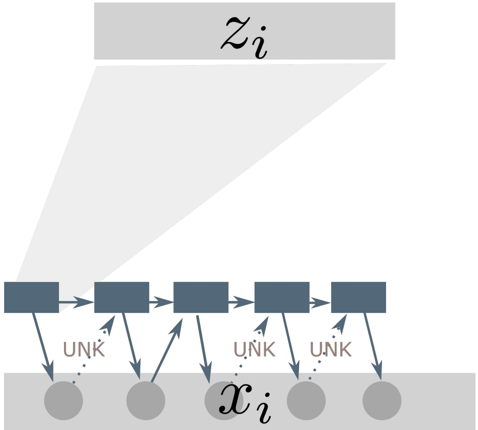
KL Annealing



Downweighting the KL early in training gives the encoder a chance to learn.

Analogy: Pruning in decision trees.

Word Dropout

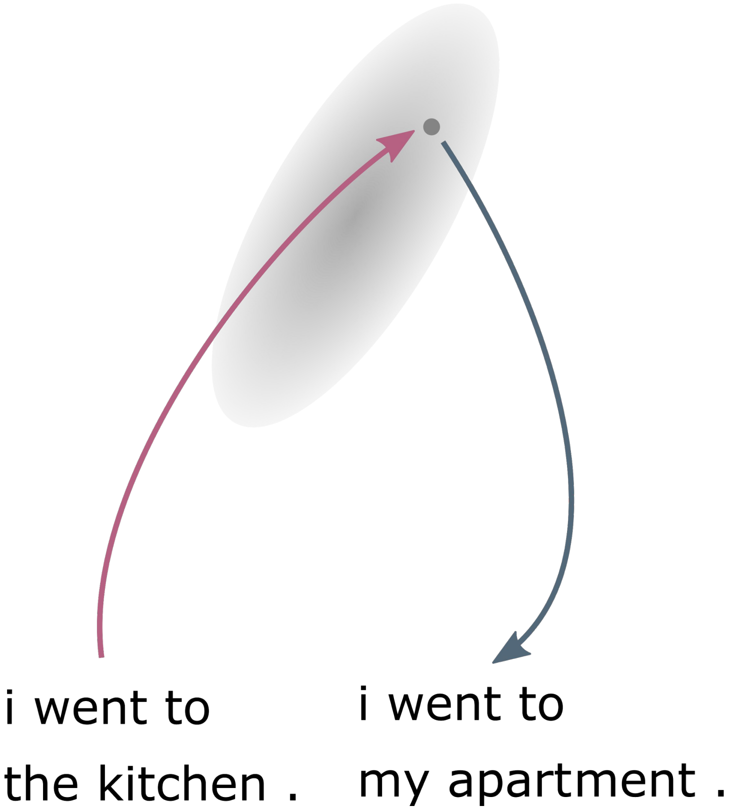


Randomly removing access weakens the decoder.

Qualitative Analysis

Sampling from the Posterior

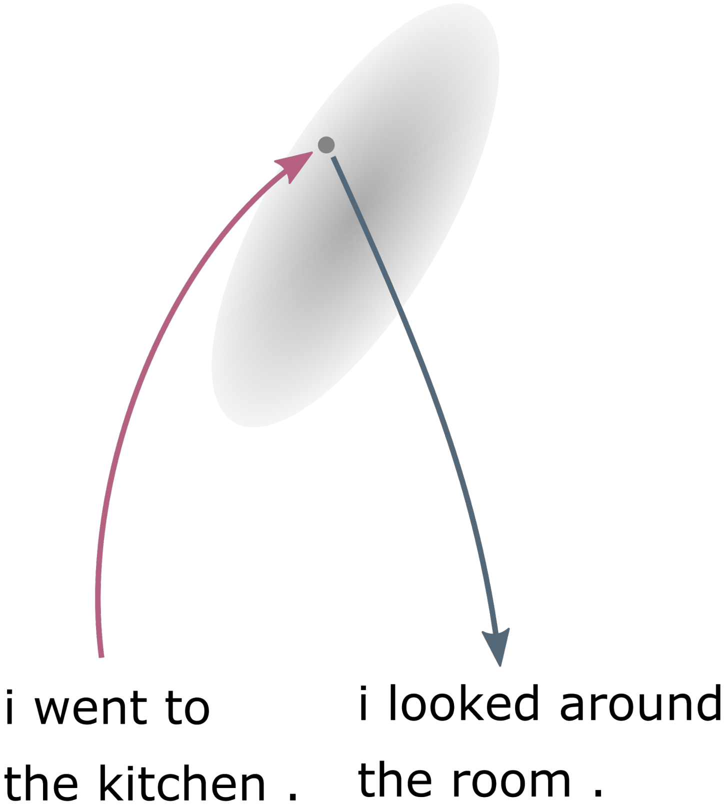
Since our encoder is probabilistic, we can view the *distribution* of sentences corresponding to an encoding.



Qualitative Analysis

Sampling from the Posterior

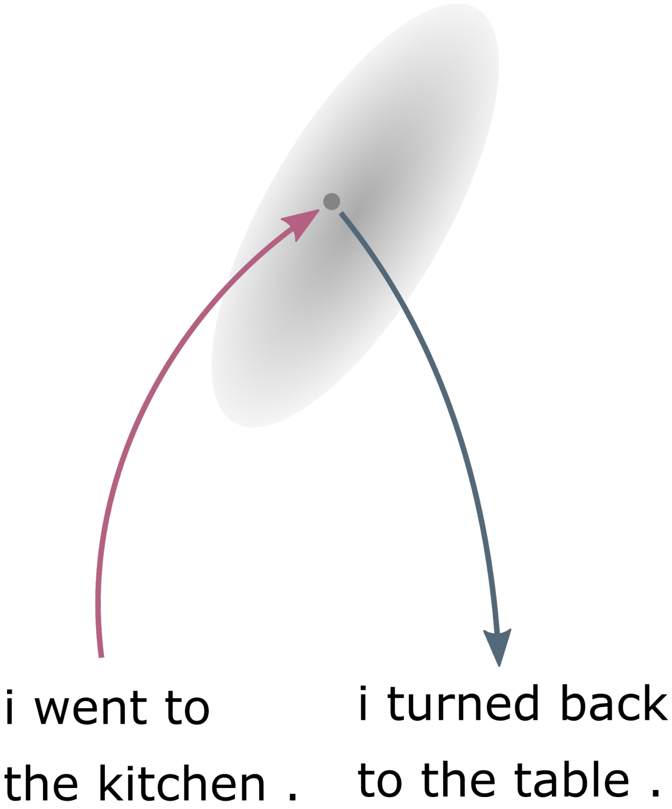
Since our encoder is probabilistic, we can view the *distribution* of sentences corresponding to an encoding.



Qualitative Analysis

Sampling from the Posterior

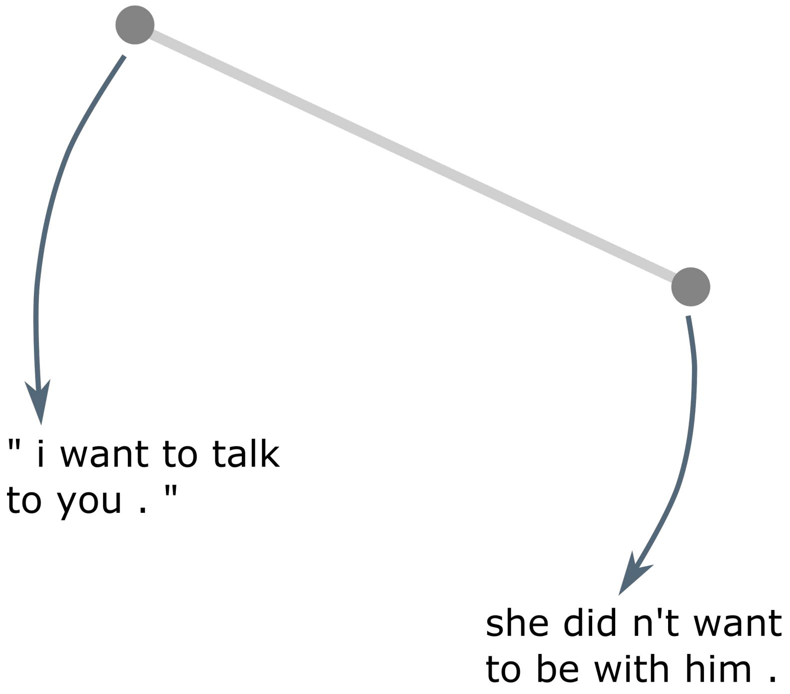
Since our encoder is probabilistic, we can view the *distribution* of sentences corresponding to an encoding.



Qualitative Analysis

Homotopies

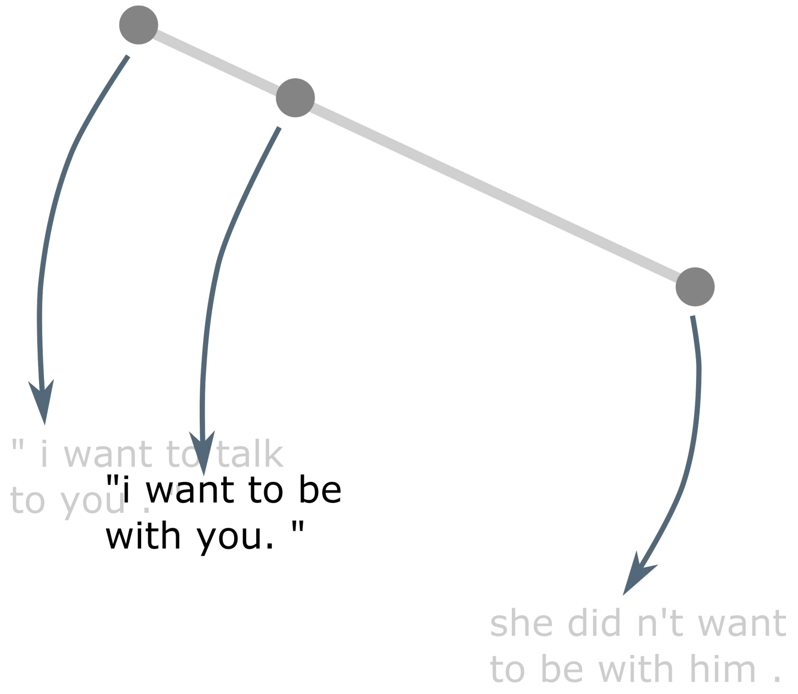
By tracing out the path of sentences in the encoded space, we can evaluate the degree to which it (1) captures topical information and (2) respects syntactic structure.



Qualitative Analysis

Homotopies

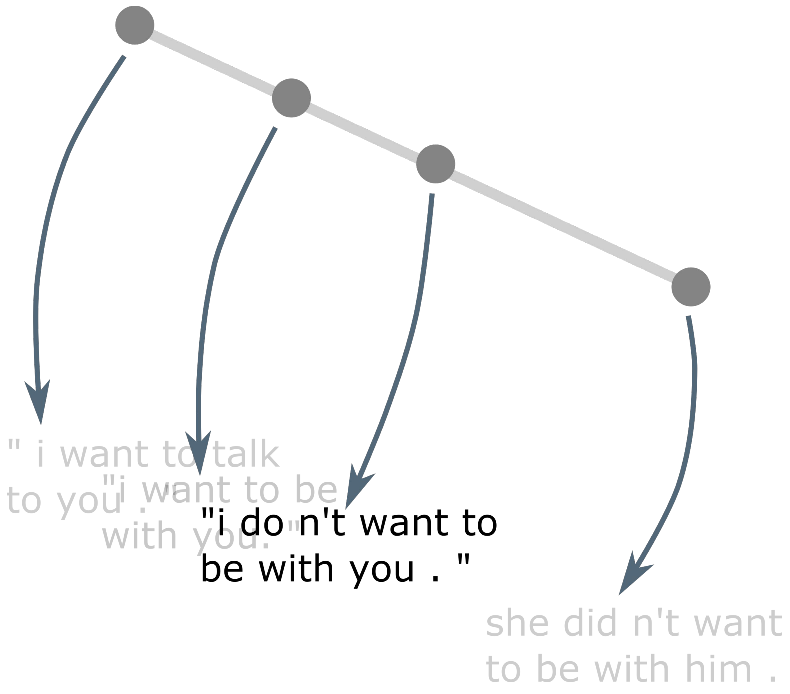
By tracing out the path of sentences in the encoded space, we can evaluate the degree to which it (1) captures topical information and (2) respects syntactic structure.



Qualitative Analysis

Homotopies

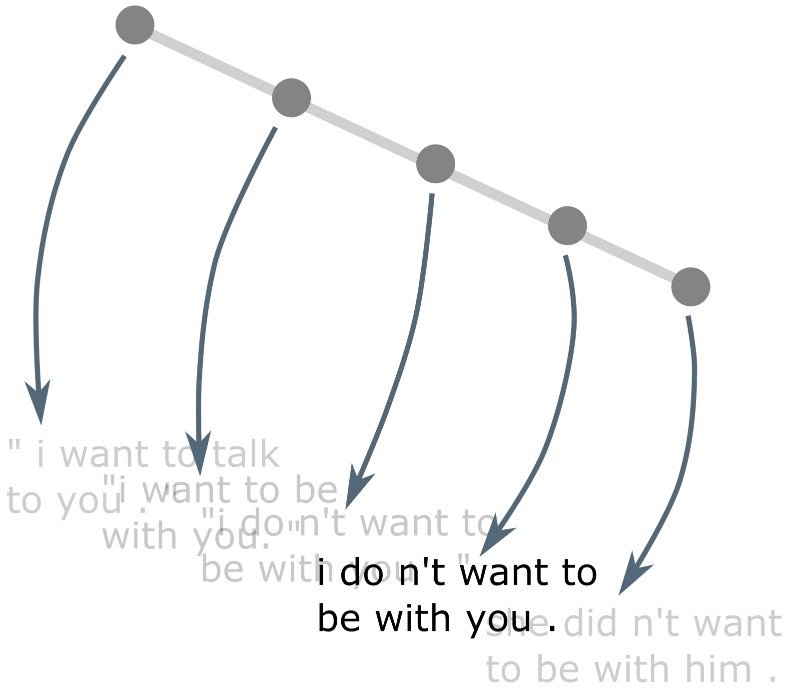
By tracing out the path of sentences in the encoded space, we can evaluate the degree to which it (1) captures topical information and (2) respects syntactic structure.



Qualitative Analysis

Homotopies

By tracing out the path of sentences in the encoded space, we can evaluate the degree to which it (1) captures topical information and (2) respects syntactic structure.



Follow-up Research

Powerful Reformulations

Are there reformulations that are easier to optimize, or which obtain tighter bounds?

- Makhzani+ [2015]
- Chen+ [2016]
- Kingma [2016]
- Sønderby+ [2016]

Incorporating Structure

What happens with more richly structured DAGS?

- Johnson+ [2015]
- Karl+ [2016]

Allowing Discreteness

The differentiability constraint is limiting, how can we get around it?

- Jang+ [2016]
- Maddison+ [2016]
- Naesseth+ [2017]

Probabilistic Inference ↔ Deep Learning

At the end of the day...

Develop methods for learning useful representations that are,

- **Powerful**: Reflect complex structure in real data
- **Automatic**: Don't require substantial human effort
- **Modular**: Easily assembled for new problems
- **Inferential**: Allow reasoning about uncertainty
- **Robust**, **Data Efficient**, **Fast**, ….

References

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.

Chen, X., Kingma, D. P., Salimans, T., Duan, Y., Dhariwal, P., Schulman, J., ... & Abbeel, P. (2016). Variational lossy autoencoder. *arXiv preprint arXiv:1611.02731*.

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, *14*(1), 1303-1347.

Jaakkola, T., & Jordan, M. (1997, January). A variational approach to Bayesian logistic regression models and their extensions. In *Sixth International Workshop on Artificial Intelligence and Statistics* (Vol. 82, p. 4).

Jang, E., Gu, S., & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*.

Johnson, M. J., Duvenaud, D., Wiltschko, A. B., Datta, S. R., & Adams, R. P. (2016). Structured VAEs: Composing probabilistic graphical models and variational autoencoders. *ArXiv e-prints*, *1603*, v1.

Karl, M., Soelch, M., Bayer, J., & van der Smagt, P. (2016). Deep variational bayes filters: Unsupervised learning of state space models from raw data. *arXiv preprint arXiv:1605.06432*.

Kingma, D. P., Salimans, T., Jozefowicz, R., Chen, X., Sutskever, I., & Welling, M. (2016). Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems* (pp. 4743-4751).

Kingma, D. P., & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Maddison, C. J., Mnih, A., & Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., & Frey, B. (2015). Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Naesseth, C., Ruiz, F., Linderman, S., & Blei, D. (2017, April). Reparameterization gradients through acceptance-rejection sampling algorithms. In *Artificial Intelligence and Statistics* (pp. 489-498).

Sønderby, C. K., Raiko, T., Maaløe, L., Sønderby, S. K., & Winther, O. (2016). Ladder variational autoencoders. In *Advances in neural information processing systems* (pp. 3738-3746)

Wainwright, Martin J., and Michael I. Jordan. "Graphical models, exponential families, and variational inference." *Foundations and Trends® in Machine Learning* 1, no. 1–2 (2008): 1-305..

Williams, Ronald J. "Simple statistical gradient-following algorithms for connectionist reinforcement learning." In *Reinforcement Learning*, pp. 5-32. Springer, Boston, MA, 1992.

Derivation of ELBO expressions

$$D_{KL}\left(q\left(z\right)||p\left(z|x\right)\right) \geq 0$$

$$\iff \mathbb{E}_{q(z)}\left[\log \frac{q\left(z\right)}{p\left(z|x\right)}\right] \geq 0$$

$$\iff \mathbb{E}_{q(z)}\left[\log \frac{q\left(z\right)}{p\left(x,z\right)}\right] + \log p\left(x\right) \geq 0$$

$$\iff \log p\left(x\right) \geq \mathbb{E}_{q}\left[\log p\left(x,z\right)\right] - H\left(q\right)$$

$$\iff \log p\left(x\right) \geq \mathbb{E}_{q}\left[\log p\left(x|z\right)\right] - D_{KL}\left(q\left(z\right)||p\left(z\right)\right)$$

High Variance of REINFORCE

Intuition 1: Consider “depth 0” generator and inference networks -- just univariate Gaussians. The REINFORCE estimate has form,

$$\frac{1}{\sigma_\theta^2(z)}(x-\mu_\theta(z))^2(z-\mu_\varphi(x))$$

which is generally a more complicated function of the gaussian noise than

$$\frac{1}{\sigma_\theta^2(z)}(\mu_\varphi(x)+\sigma_\varphi(x)\epsilon-\mu_\theta(z))$$

the pathwise gradient.

Intuition 2: If the variational parameters have additive, orthogonal influence on the log-likelihood, then the reparameterization estimate only depends on one term, since the rest are differentiated to zero.

Quantitative Evaluation Experiment

Task: Impute the ends of sentences in a Books Corpus

Inference: Beam search (breadth-first search of probable sequences), with or without Iterated Conditional Modes (deterministic Gibbs-sampling-like iteration)

Evaluation: Classify true vs. generated sentence completions