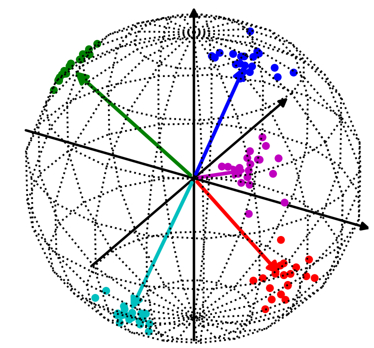
SEPTEMBER 2018

TEXAS
The University of Texas at Austin

SPHERICAL LATENT SPACES FOR STABLE VARIATIONAL AUTOENCODERS

EMNLP 2018

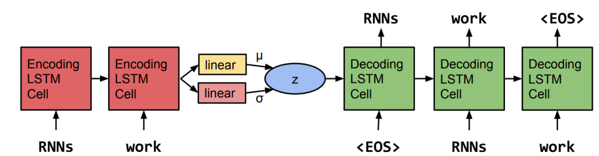


JIACHENG XU & GREG DURRETT
TAUR Lab, Department of Computer Science, The University of Texas at Austin

I will first introduce the background of the VAE and the problem we want to solve. And then I will introduce our model, von mises fisher VAE and its comparison and improvement over the original Gaussian VAE. Then followed by the experimental findings.

VAE x NLP

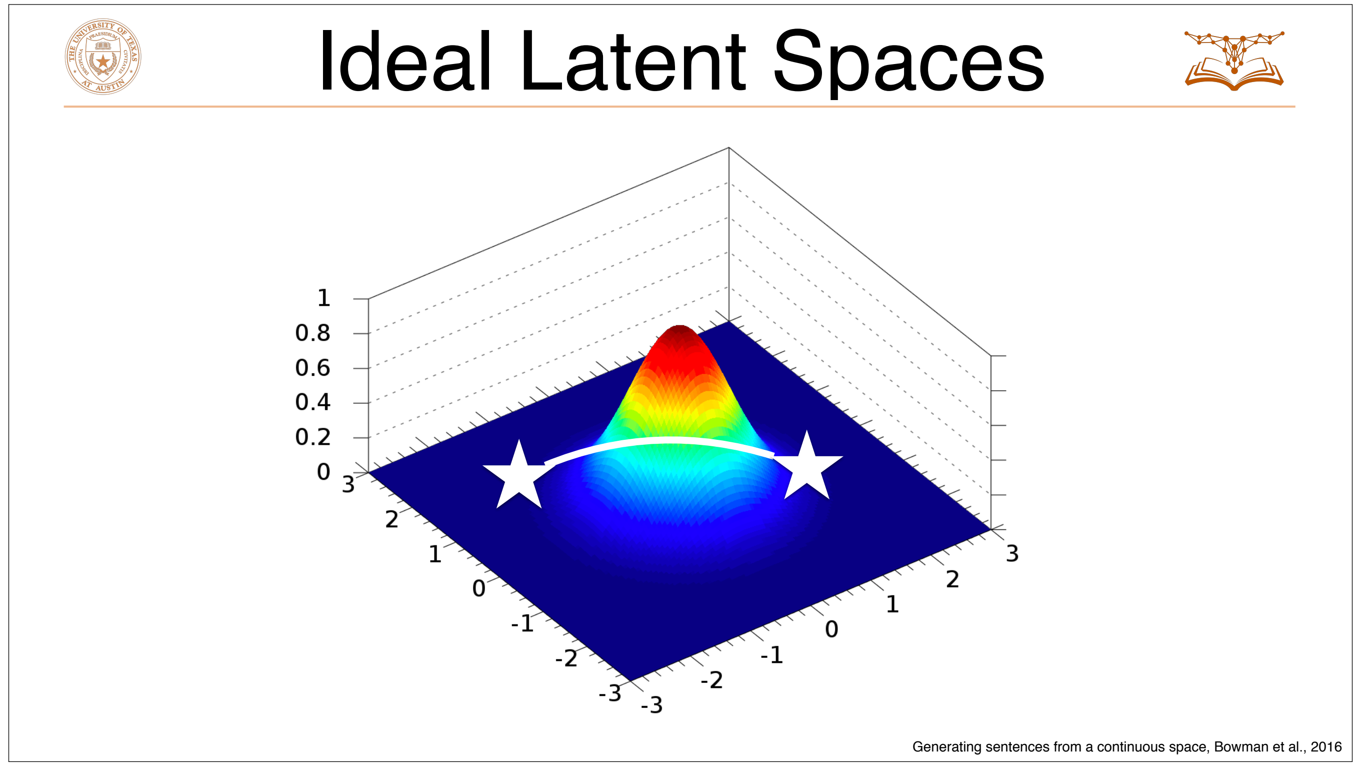
- Unsupervised Latent Variable Model
  - Factorization & Decoupling
  - Style Transfer
- Representations in Latent Spaces
  - Soft Ellipsoidal Regions
  - Better Generalization
- Cases: RNN Language Model, Document Model, Dialogue System



Generating sentences from a continuous space, Bowman et al., 2016

Recent advances in variational inference and learning enable the incorporation of distributed latent representations of the whole sentence or the document.

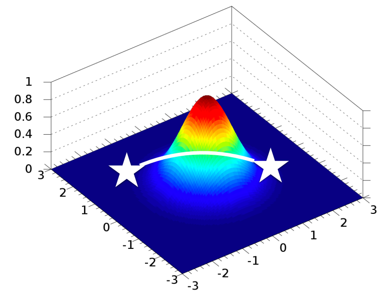
The intuition of this unsupervised latent variable model is that to model the holistic properties of the whole sequences such as style, topic, and high-level syntactic features. Besides, this approach of factorization allows us to decouple the semantics, syntax, sentiment, topic and many other aspects of the texts to be encoded, which can be used in style transfer or domain transfer for text generation.
VAE learns codes not as single points, but as soft ellipsoidal regions in latent space, forcing the codes to fill the space rather than memorizing the training data as isolated codes. Hence, the learned representations are more diverse and well-formed.



decoding from points between two sentence encodings

Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English

Ideal Latent Spaces

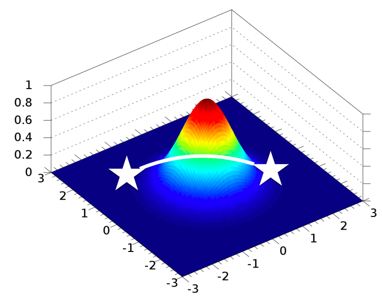


Generating sentences from a continuous space, Bowman et al., 2016

decoding from points between two sentence encodings

Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English

Ideal Latent Spaces



Comparison: decoding from points between two sentence encodings

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

“ i want to talk to you . ”
“i want to be with you . ”
“i do n’t want to be with you . ”
i do n’t want to be with you .
she did n’t want to be with him .

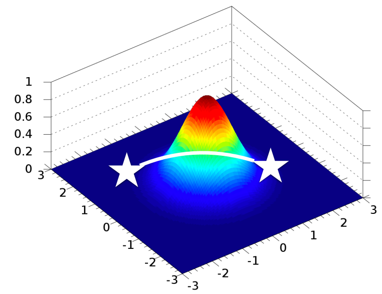
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

Generating sentences from a continuous space, Bowman et al., 2016

decoding from points between two sentence encodings

Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English

Ideal Latent Spaces



Comparison: decoding from points between two sentence encodings

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

Traditional AE

" i want to talk to you . "
"i want to be with you . "
"i do n't want to be with you . "
i do n't want to be with you .
she did n't want to be with him .

he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

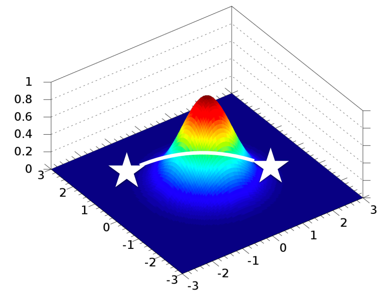
Variational AE

Generating sentences from a continuous space, Bowman et al., 2016

decoding from points between two sentence encodings

Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English

Ideal Latent Spaces



Comparison: decoding from points between two sentence encodings

i went to the store to buy some groceries .
i store to buy some groceries .
i were to buy any groceries .
horses are to buy any groceries .
horses are to buy any animal .
horses the favorite any animal .
horses the favorite favorite animal .
horses are my favorite animal .

Traditional AE ✘

" i want to talk to you . "
"i want to be with you . "
"i do n't want to be with you . "
i do n't want to be with you .
she did n't want to be with him .
he was silent for a long moment .
he was silent for a moment .
it was quiet for a moment .
it was dark and cold .
there was a pause .
it was my turn .

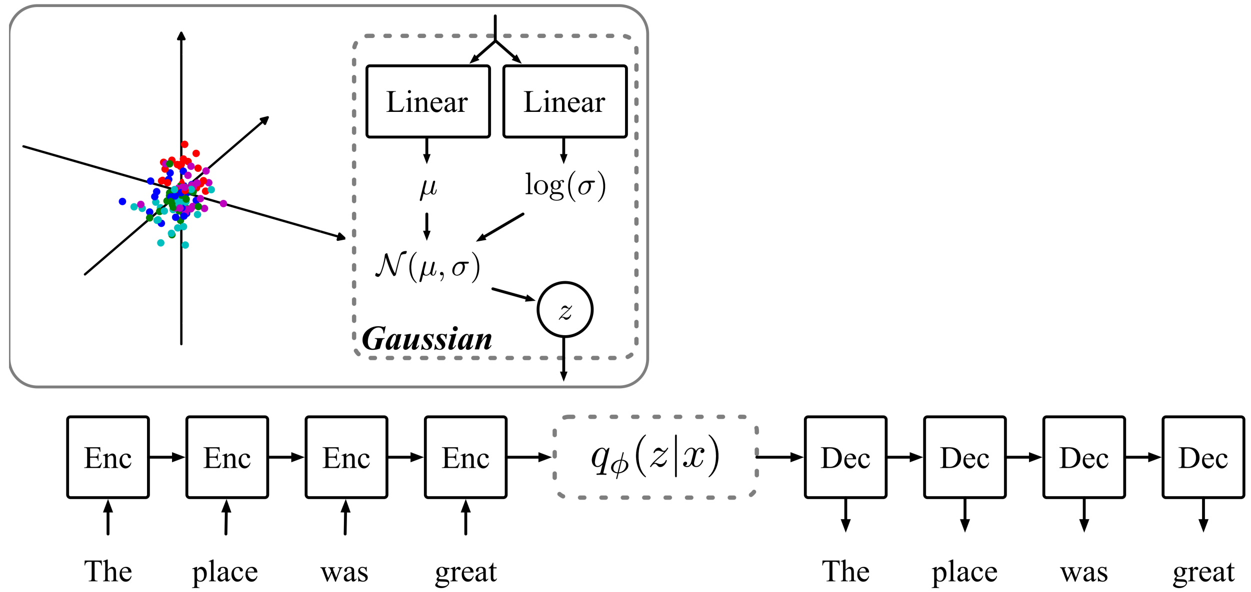
Variational AE ✓

Generating sentences from a continuous space, Bowman et al., 2016

decoding from points between two sentence encodings

Sentences produced by greedily decoding from points between two sentence encodings with a conventional autoencoder. The intermediate sentences are not plausible English

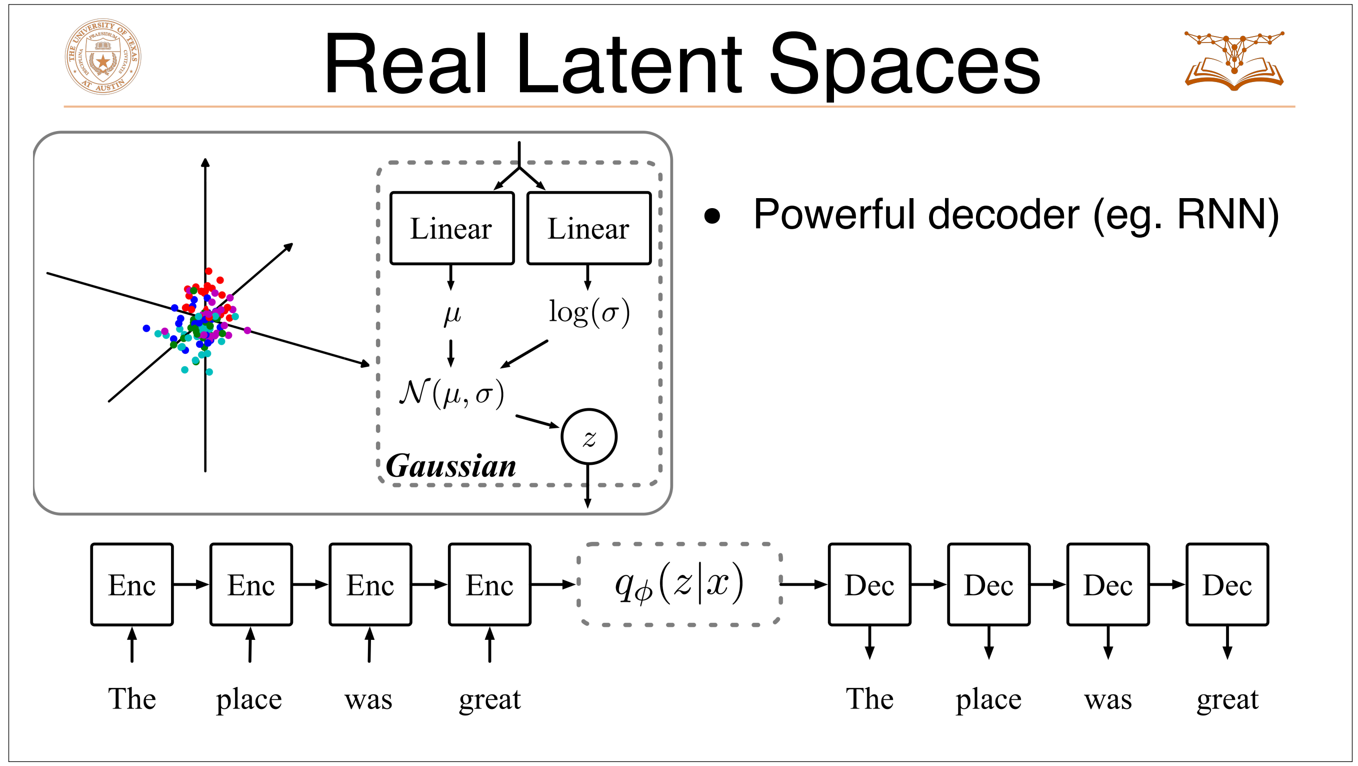
Real Latent Spaces



As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

And we name the issue with KL collapse

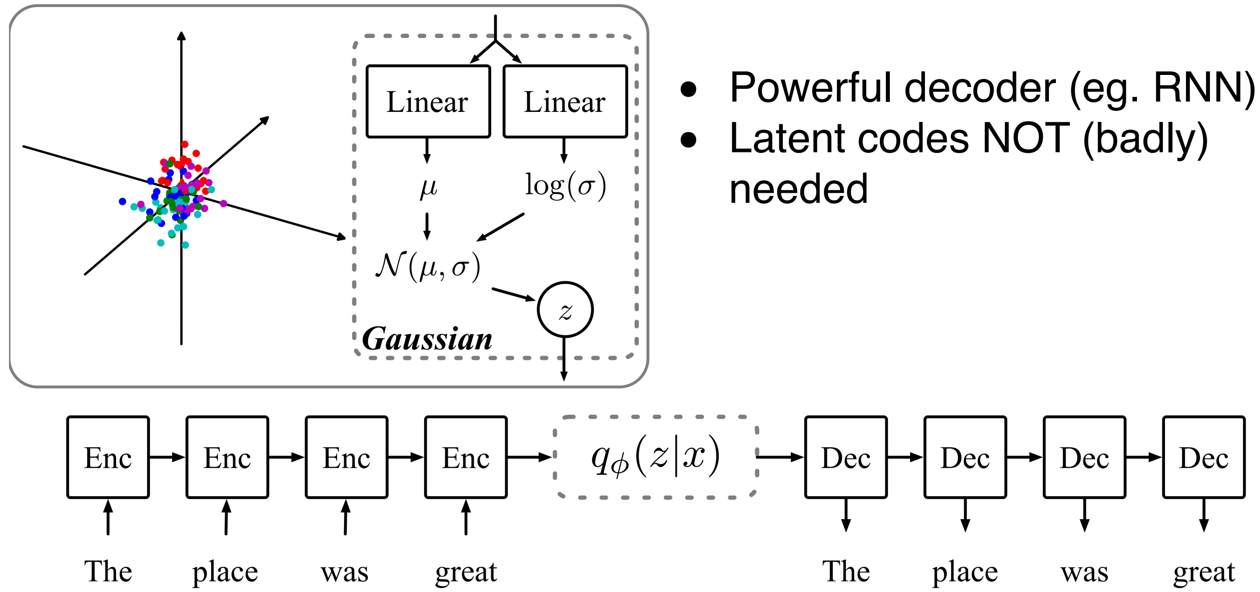


As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

And we name the issue with KL collapse

Real Latent Spaces

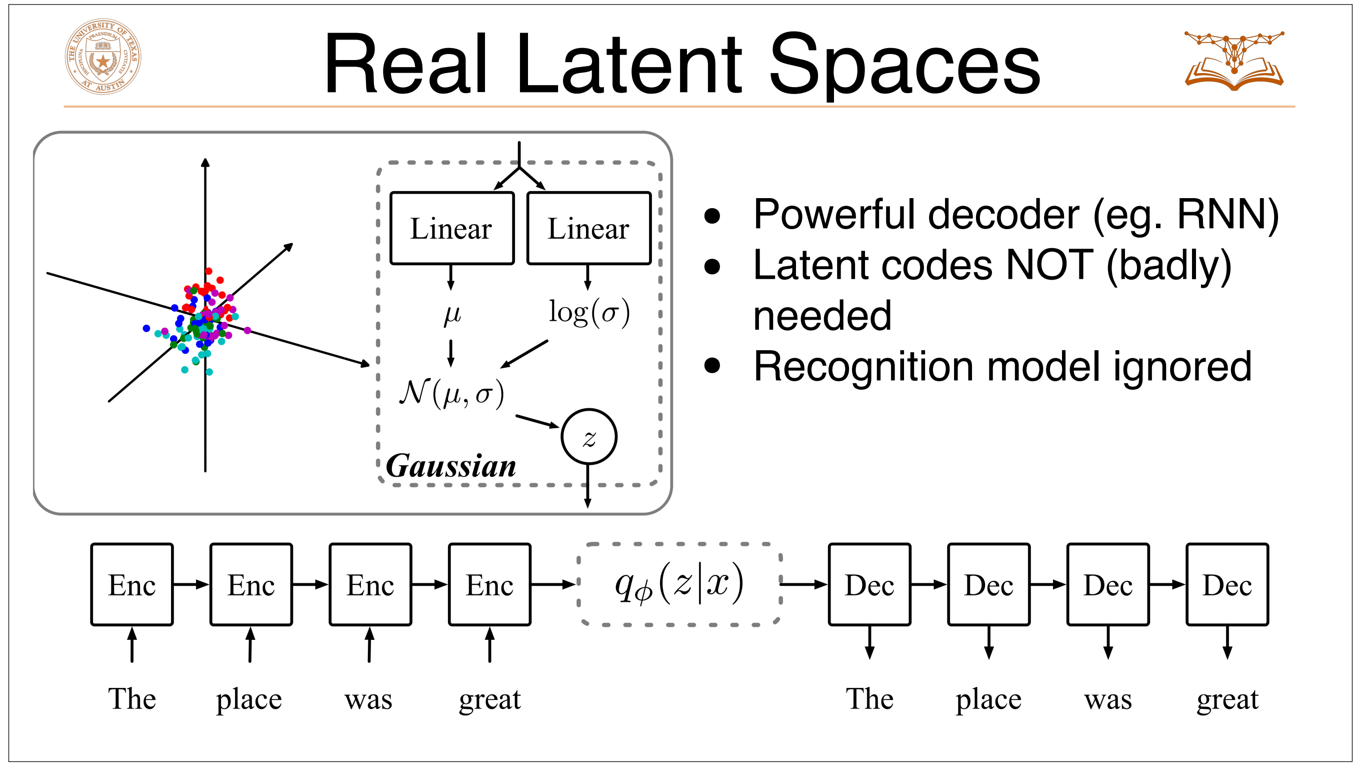


- Powerful decoder (eg. RNN)
- Latent codes NOT (badly) needed

As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

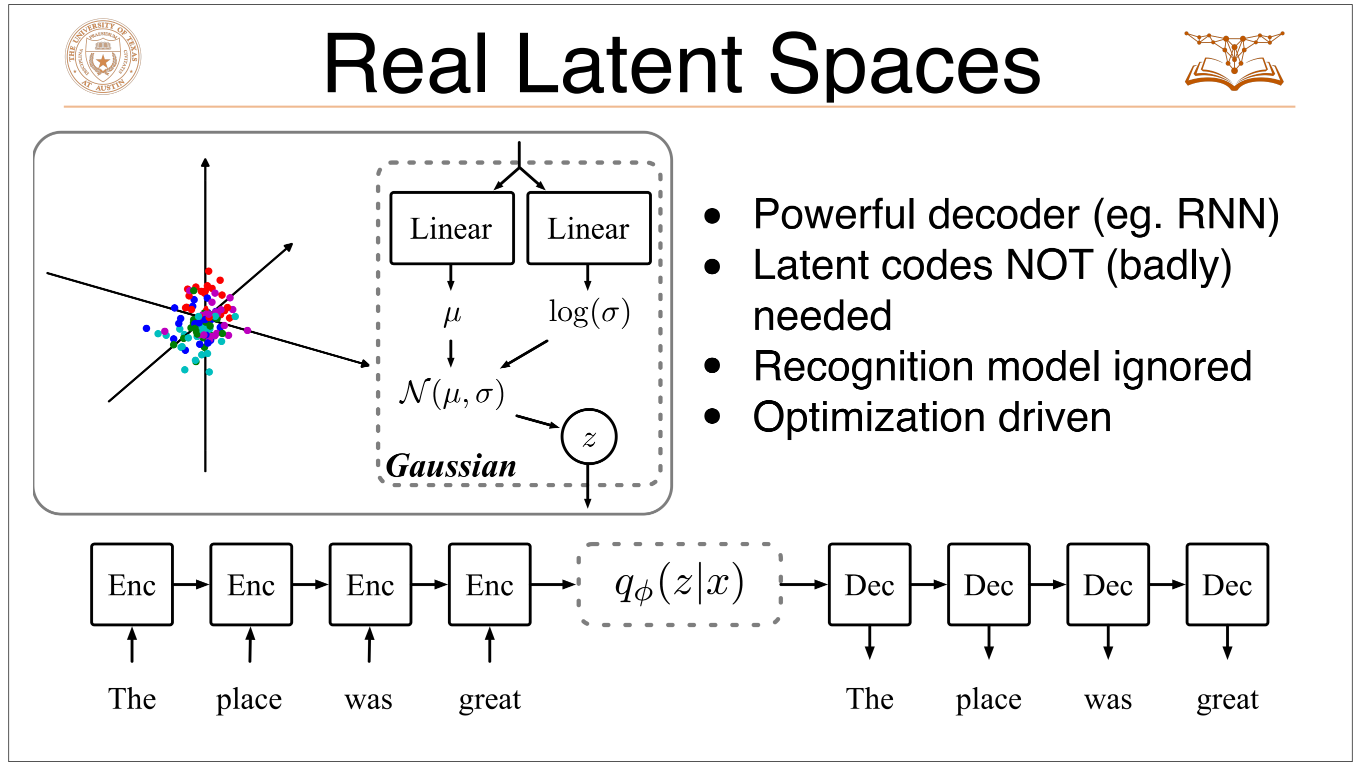
And we name the issue with KL collapse



As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

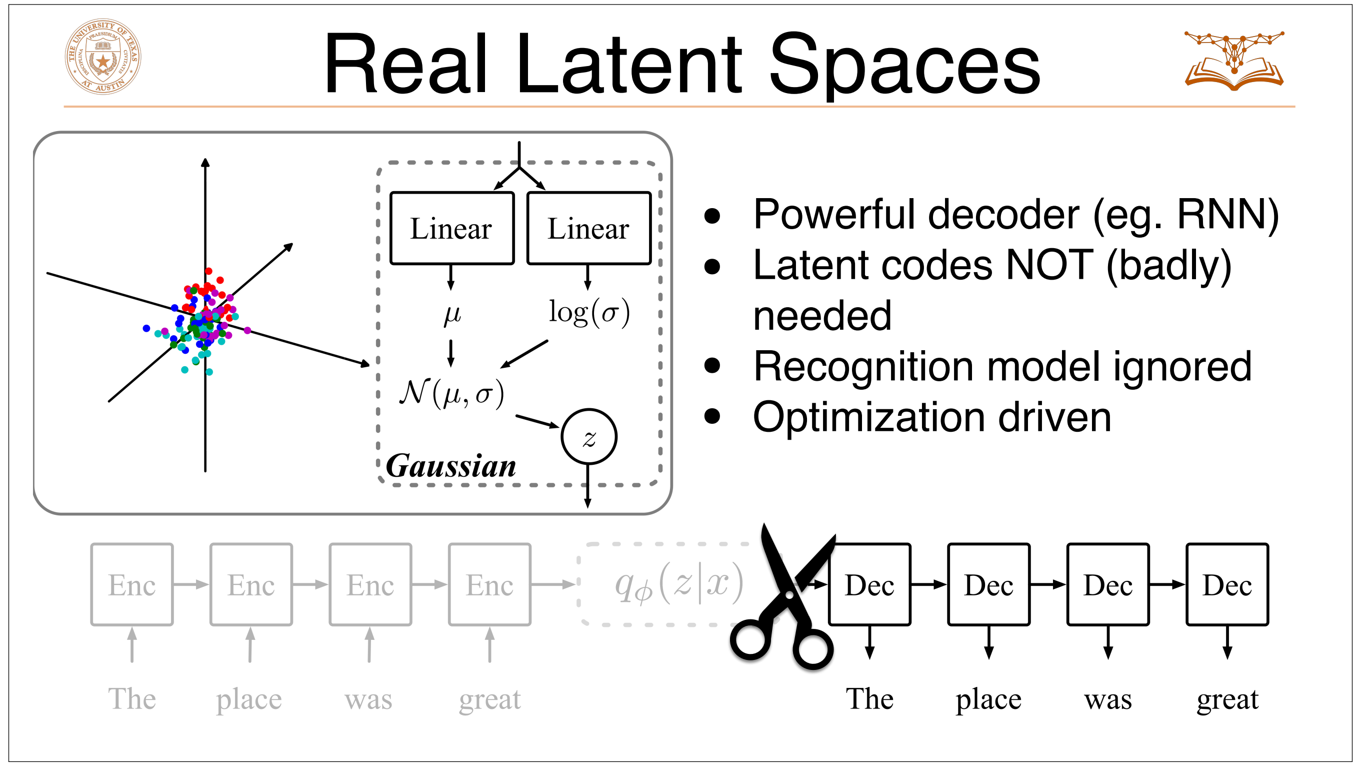
And we name the issue with KL collapse



As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

And we name the issue with KL collapse

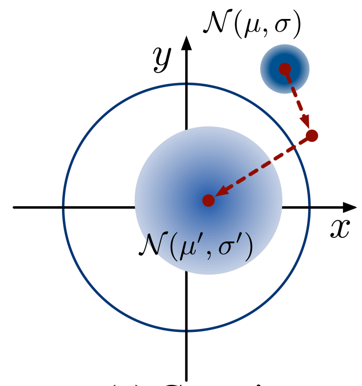


As you see in the figure, the representations learned in Gaussian VAE is more or less random noise and doesn't help the downstream task at all.

The Neural Variational RNN (NVRNN) language model based on a Gaussian prior (left) and a vMF prior (right). The encoder model first computes the parameters for the variational approximation qϕ(zlx) (see dotted box); we then sample z and generate the word sequence x given z. We show samples from N (0, I) and vMF(·,κ = 100); the latter samples lie on the surface of the unit sphere. While κ can be predicted from the encoder network, we find experimentally that fixing it leads to more stable optimization and better performance.

And we name the issue with KL collapse

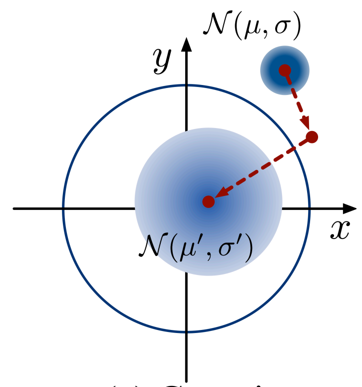
KL Collapse



(a) Gaussian

Visualization of optimization of how q varies over time for a single example during learning. In the Gaussian case, the KL term tends to pull the model towards the prior (moving from μ, σ to μ 0 , σ0), whereas in the vMF case there is no such pressure towards a single distribution

KL Collapse

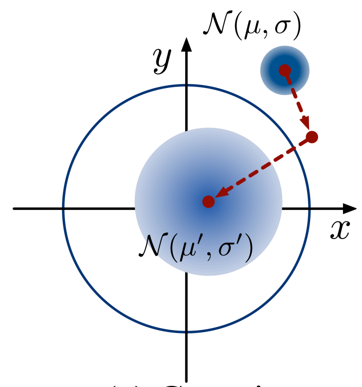


(a) Gaussian

- Original Gaussian VAE ``learns" to escape from using the latent spaces

Visualization of optimization of how q varies over time for a single example during learning. In the Gaussian case, the KL term tends to pull the model towards the prior (moving from μ, σ to μ 0 , σ0), whereas in the vMF case there is no such pressure towards a single distribution

KL Collapse

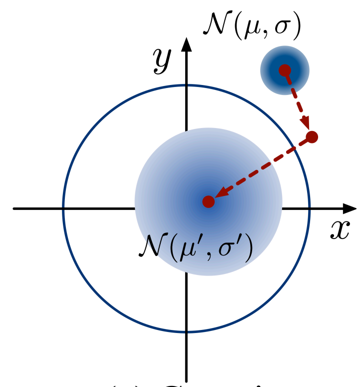


(a) Gaussian

- Original Gaussian VAE ``learns" to escape from using the latent spaces
- Let's change the Assumption of the Prior and Posterior Distribution in VAE models

Visualization of optimization of how q varies over time for a single example during learning. In the Gaussian case, the KL term tends to pull the model towards the prior (moving from μ, σ to μ 0 , σ0), whereas in the vMF case there is no such pressure towards a single distribution

KL Collapse

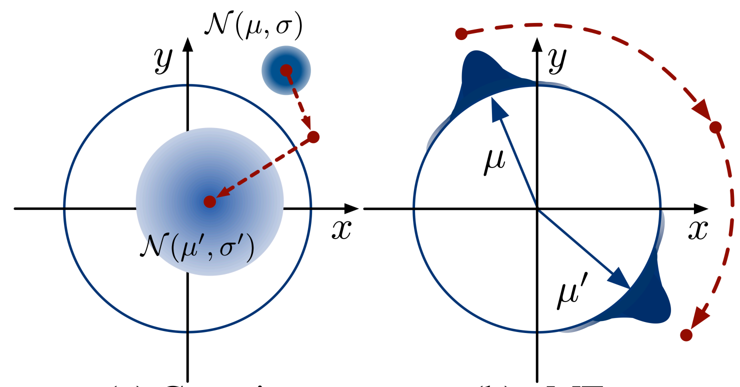


(a) Gaussian

- Original Gaussian VAE ``learns'' to escape from using the latent spaces
- Let's change the Assumption of the Prior and Posterior Distribution in VAE models
- Directional Distribution: von Mises-Fisher (vMF)

Visualization of optimization of how q varies over time for a single example during learning. In the Gaussian case, the KL term tends to pull the model towards the prior (moving from μ, σ to μ 0 , σ0), whereas in the vMF case there is no such pressure towards a single distribution

KL Collapse



(a) Gaussian (b) vMF

- Original Gaussian VAE ``learns" to escape from using the latent spaces
- Let's change the Assumption of the Prior and Posterior Distribution in VAE models
- Directional Distribution: von Mises-Fisher (vMF)

Visualization of optimization of how q varies over time for a single example during learning. In the Gaussian case, the KL term tends to pull the model towards the prior (moving from μ, σ to μ 0 , σ0), whereas in the vMF case there is no such pressure towards a single distribution

Objective Function Decomposition

$$\boxed{\log p_\theta(x)} = \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) + \boxed{\mathcal{L}(\theta,\phi;x)}$$

$$\boxed{\mathcal{L}(\theta,\phi;x)} = -\mathrm{KL}(q_\phi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}\log p_\theta(x|z)$$

ELBO

It's time to dive into the math behind the VAEs.

The variational autoencoder is a generative model that is based on a regularized version of the standard autoencoder. the vae uses an objective which encourages the model to keep its posterior distributions close to a prior p(z). this objective is a valid lower bound on the true log likelihood of the data.

The first term of ELBO is the KL divergence of the approximate posterior from prior and the second term is an expected reconstruction error.

Objective Function Decomposition

$$\boxed{\log p_\theta(x)} = \mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z\,|\,x)) + \boxed{\mathscr{L}(\theta, \phi; x)}$$

$$\boxed{\mathscr{L}(\theta, \phi; x)} = \boxed{-\,\mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z))} + \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x\,|\,z)$$

ELBO

Regularization Term

It’s time to dive into the math behind the VAEs.

The variational autoencoder is a generative model that is based on a regularized version of the standard autoencoder. the vae uses an objective which encourages the model to keep its posterior distributions
close to a prior p(z). this objective is a valid lower bound on the true log likelihood of the data.

The first term of ELBO is the KL divergence of the approximate posterior from prior and the second term is an expected reconstruction error.

Objective Function Decomposition

$$\log p_\theta(x) = \mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z\,|\,x)) + \mathscr{L}(\theta, \phi; x)$$

$$\mathscr{L}(\theta, \phi; x) = -\,\mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)} \log p_\theta(x\,|\,z)$$

ELBO | Regularization Term | Reconstruction Term

It’s time to dive into the math behind the VAEs.

The variational autoencoder is a generative model that is based on a regularized version of the standard autoencoder. the vae uses an objective which encourages the model to keep its posterior distributions close to a prior p(z). this objective is a valid lower bound on the true log likelihood of the data.

The first term of ELBO is the KL divergence of the approximate posterior from prior and the second term is an expected reconstruction error.

Objective Function Decomposition

$$\boxed{\log p_\theta(x)} = \mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z\,|\,x)) + \boxed{\mathscr{L}(\theta,\phi;x)}$$

$$\boxed{\mathscr{L}(\theta,\phi;x)} = \boxed{-\,\mathrm{KL}(q_\phi(z\,|\,x)\,||\,p_\theta(z))} + \boxed{\mathbb{E}_{q_\phi(z|x)}\log p_\theta(x\,|\,z)}$$

ELBO | Regularization Term | Reconstruction Term

$$q_\phi(z\,|\,x) \sim \mathcal{N}(z;\mu,\sigma)$$

$$p_\theta(z) = \mathcal{N}(z;0,I)$$

It's time to dive into the math behind the VAEs.

The variational autoencoder is a generative model that is based on a regularized version of the standard autoencoder. the vae uses an objective which encourages the model to keep its posterior distributions
close to a prior p(z). this objective is a valid lower bound on the true log likelihood of the data.

The first term of ELBO is the KL divergence of the approximate posterior from prior and the second term is an expected reconstruction error.

Objective Function Decomposition

$$\log p_\theta(x) = \mathrm{KL}(q_\phi(z|x)||p_\theta(z|x)) + \mathscr{L}(\theta,\phi;x)$$

$$\mathscr{L}(\theta,\phi;x) = -\mathrm{KL}(q_\phi(z|x)||p_\theta(z)) + \mathbb{E}_{q_\phi(z|x)}\log p_\theta(x|z)$$

ELBO

Regularization Term

Reconstruction Term

$$q_\phi(z|x) \sim \mathcal{N}(z;\mu,\sigma)$$
$$p_\theta(z) = \mathcal{N}(z;0,I)$$

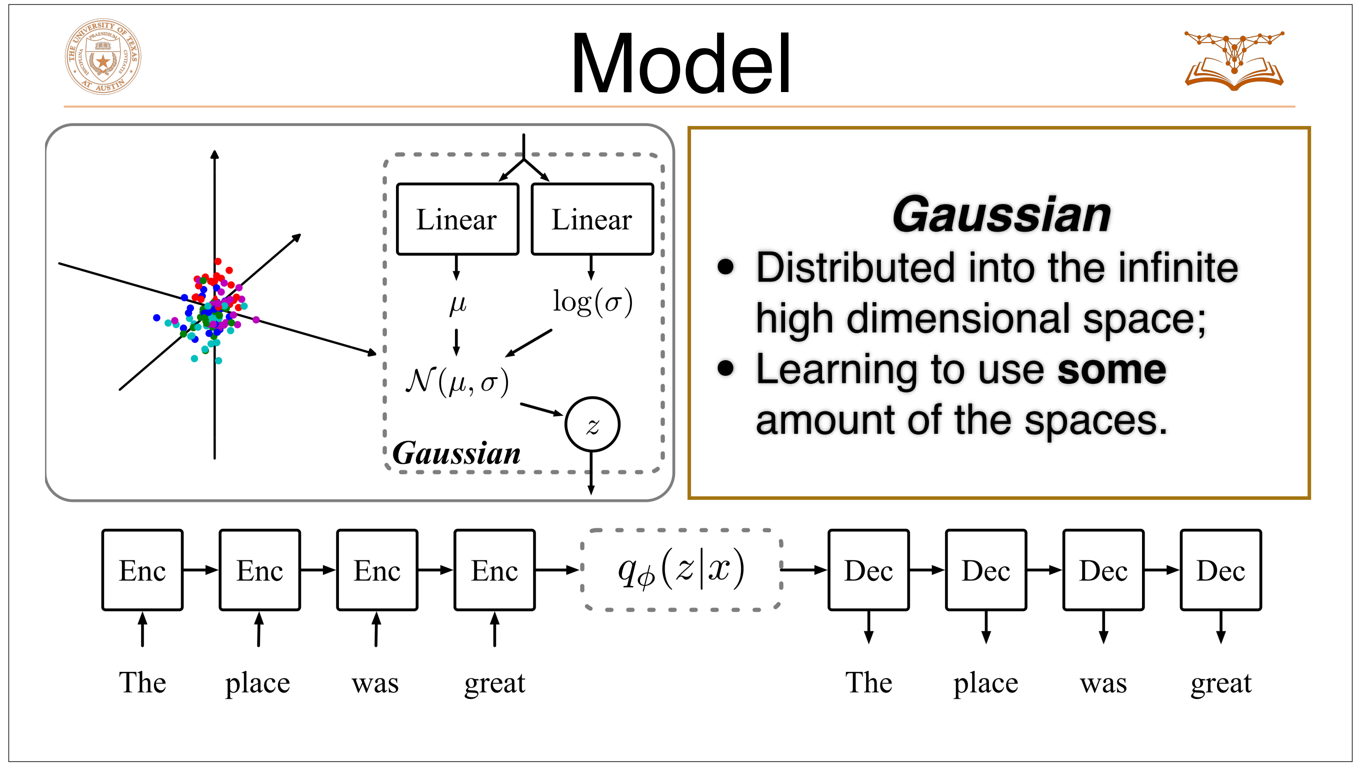
Assumption

$$q_\phi(z|x) \sim \mathrm{vMF}(z;\mu,\kappa)$$
$$p_\theta(z) = \mathrm{vMF}(\cdot,0)$$

It’s time to dive into the math behind the VAEs.

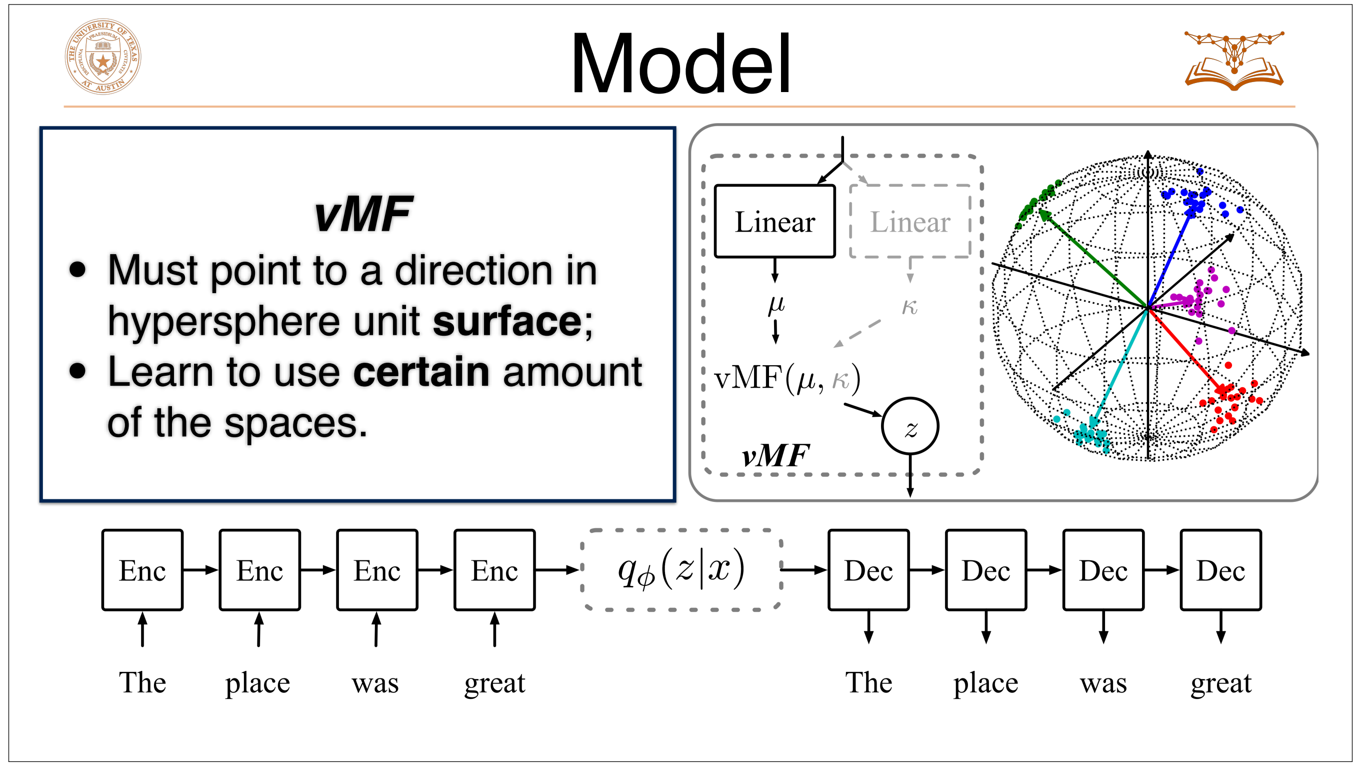
The variational autoencoder is a generative model that is based on a regularized version of the standard autoencoder. the vae uses an objective which encourages the model to keep its posterior distributions
close to a prior p(z). this objective is a valid lower bound on the true log likelihood of the data.

The first term of ELBO is the KL divergence of the approximate posterior from prior and the second term is an expected reconstruction error.



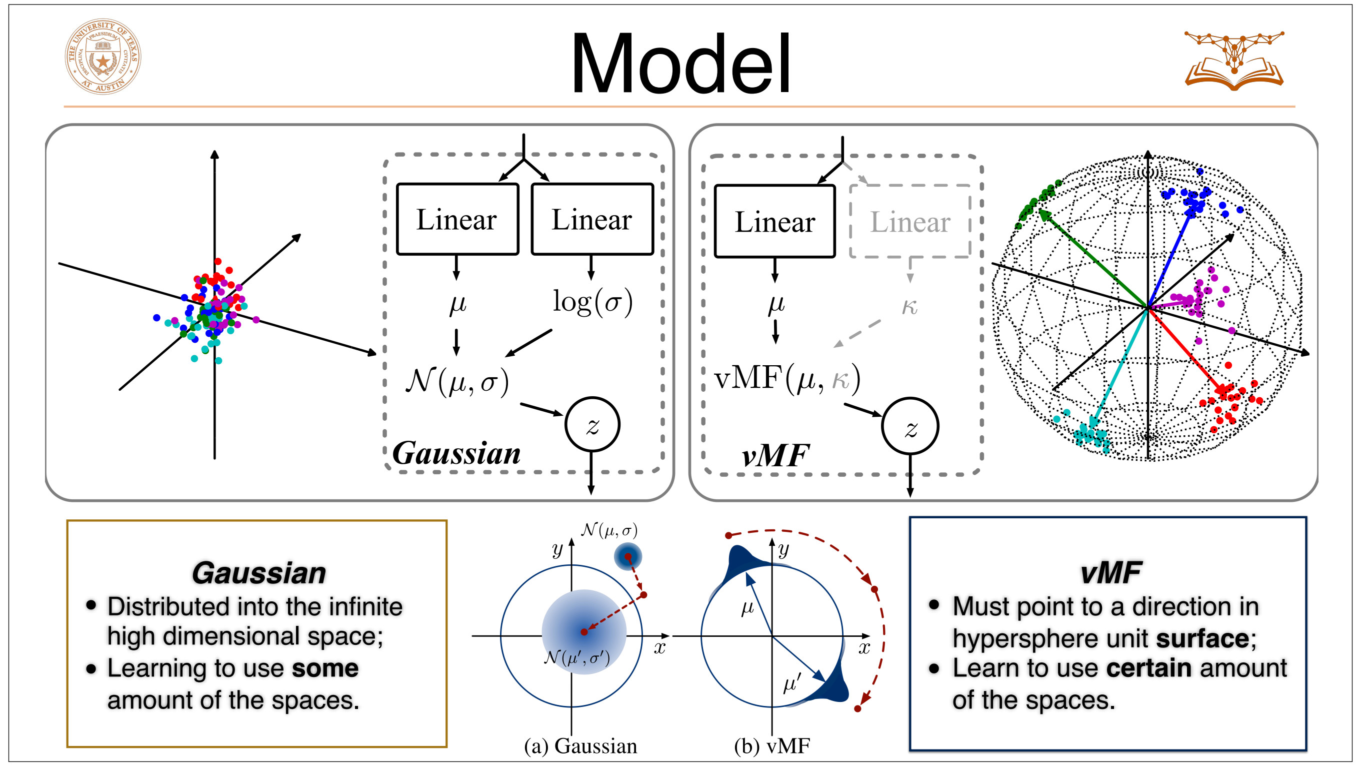
We fix and mannually choose the kappa in the vMF model because it's hard to learn the kappa in an end-to-end fashion and it's actually pretty easy to find a reasonable kappa value.

Gaussian



We fix and mannually choose the kappa in the vMF model because it’s hard to learn the kappa in an end-to-end fashion and it’s actually pretty easy to find a reasonable kappa value.

Gaussian



We fix and mannually choose the kappa in the vMF model because it’s hard to learn the kappa in an end-to-end fashion and it’s actually pretty easy to find a reasonable kappa value.

Gaussian

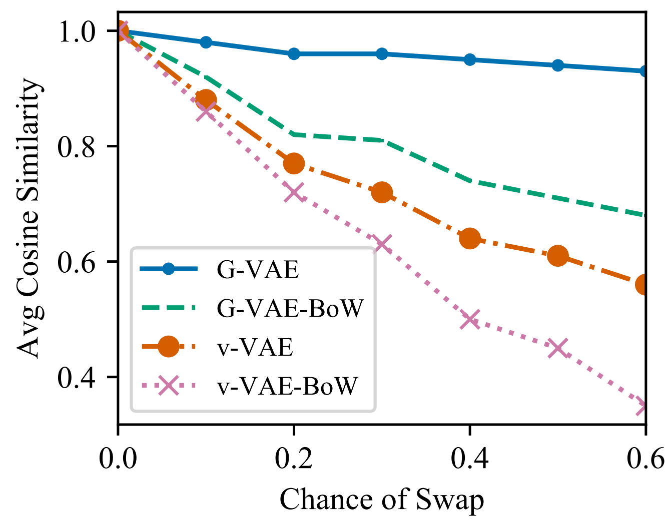
Experiments

Model	PTB				Yelp			
	Standard		Inputless		Standard		Inputless	
	NLL	PPL	NLL	PPL	NLL	PPL	NLL	PPL
RNNLM (2016)	100 (–)	116	135 (–)	>600	–	–	–	–
G-VAE (2016)	101 (2)	119	125 (15)	380	–	–	–	–
RNNLM (Ours)	100 (–)	114	134 (–)	596	199 (–)	55	300 (–)	432
G-VAE (Ours)	99 (4.4)	109	125 (6.3)	379	199 (0.5)	55	274 (13.4)	256
vMF-VAE (Ours)	**96 (5.7)**	**98**	**117 (18.6)**	**262**	**198 (6.4)**	**54**	**242 (48.5)**	**134**

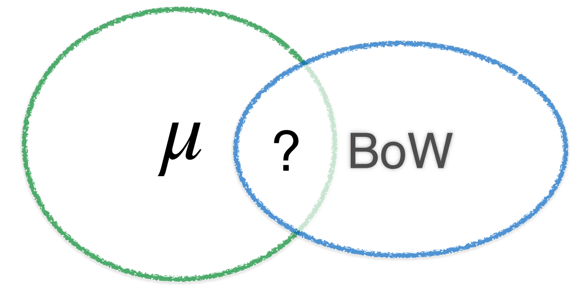
- RNN Language Model
  - PTB
  - Yelp
- Document Model
  - Reuters Corpus
  - 20 News Group

Model	Dim	20NG	RCV1
fDARN (2014)	50	917	724
	200	-	598
G-NVDM (2016)	50	836	563
	200	852	550
v-NVDM (Ours)	25	**793**	558
	50	830	**529**
	200	851	609

What Encoded?



Sensitive to Word Order

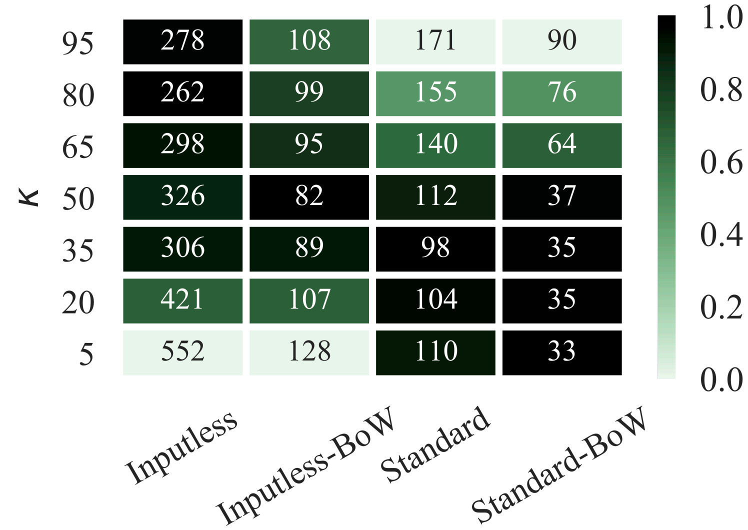


Model Setting	NVRNN $\mu \rightarrow$ BoW	NVRNN BoW $\rightarrow \mu$
G-VAE	0.74	0.74
v-VAE	0.77	0.57

Transferability & Overlap

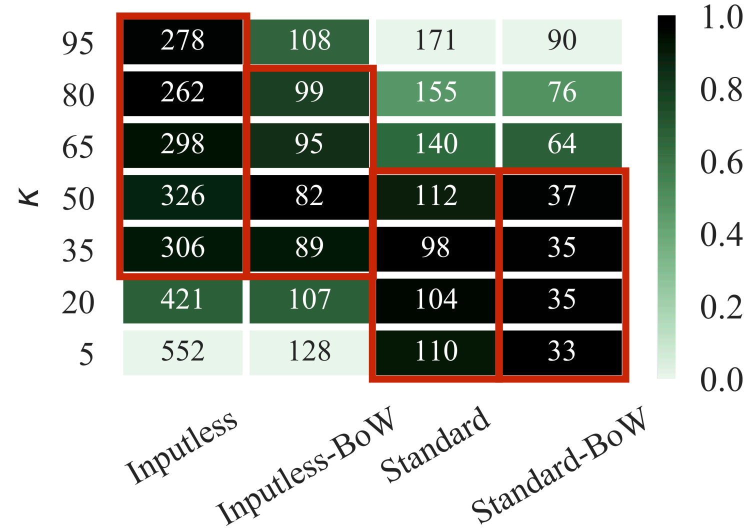
Average cosine similarity when trying to reconstruct the latent code μ from the bag of words and vice versa. In vMF, the latent code contains more information
beyond the bag of words, as shown by the lower cosine similarity when predicting BoW → μ (0.57). When the latent code is learned in a model conditioned on the bag of words (right column), it predicts the bag of words much less well, indicating that the model successfully learns orthogonal information.

Easy to Tune



Original Gaussian VAE is very tricky to tune.
What about the kappa in our model?
Perplexity of v-VAE in different settings with different κ values when the latent dimension is 50.
Darker colors correspond to perplexity values closer to the best observed for that setting. For each task, we see that there is a range of κ values that work well, and these transfer between comparable tasks

Easy to Tune



Original Gaussian VAE is very tricky to tune.
What about the kappa in our model?
Perplexity of v-VAE in different settings with different κ values when the latent dimension is 50.
Darker colors correspond to perplexity values closer to the best observed for that setting. For each task, we see that there is a range of κ values that work well, and these transfer between comparable tasks

Takehome

Takehome

- Original Gaussian VAE is tricky to wake it up
  - vMF is a ready-to-go and elegant solution
  - Directional distribution to be discovered

Takehome

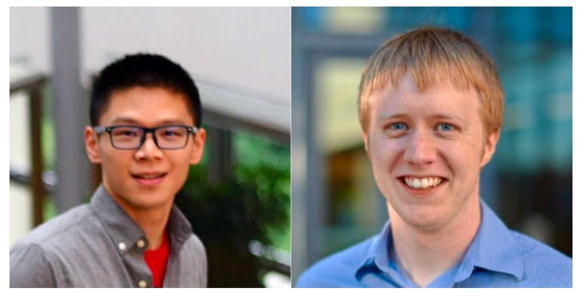
- Original Gaussian VAE is tricky to wake it up
  - vMF is a ready-to-go and elegant solution
  - Directional distribution to be discovered
- VAE (probably) helps in low resource / small data / …
  - Less data-hungry & more interoperability

Takehome

- Original Gaussian VAE is tricky to wake it up
  - vMF is a ready-to-go and elegant solution
  - Directional distribution to be discovered
- VAE (probably) helps in low resource / small data / …
  - Less data-hungry & more interoperability
- vMF VAE induces meaningful representations
  - Nature of VAE models or vMF specific effects?

Release

- arXiv: https://arxiv.org/abs/1808.10805
- Code & Data: **https://github.com/jiacheng-xu/vmf_vae_nlp**
- Contact: Jiacheng Xu (jcxu@utexas.edu)



gregd_nlp

NSF
TACC
Bloomberg
NVIDIA