How I Learned to Stop Worrying AND LOVE the Uncertainty

Representing Uncertainty in Language Models: Part 1

by Hande Celikkanat

01

The Magical Christmasland

and the Peace of Certainty

I'd rather be certain.

Thank you very much.

DEEP LEARNING TODAY (AND TOMORROW?)

PARAMETERS

fixed

REPRESENTATIONS

deterministic

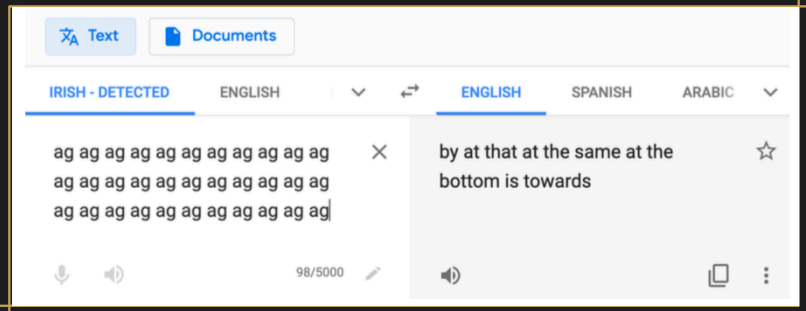
100%

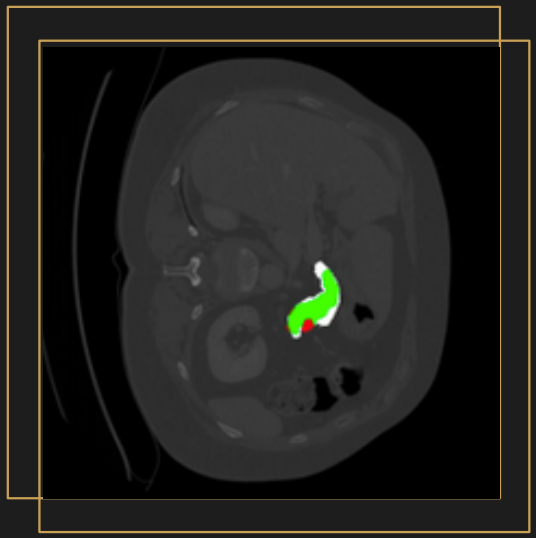
CERTAINTY

it's all there

EXPECTATIONS

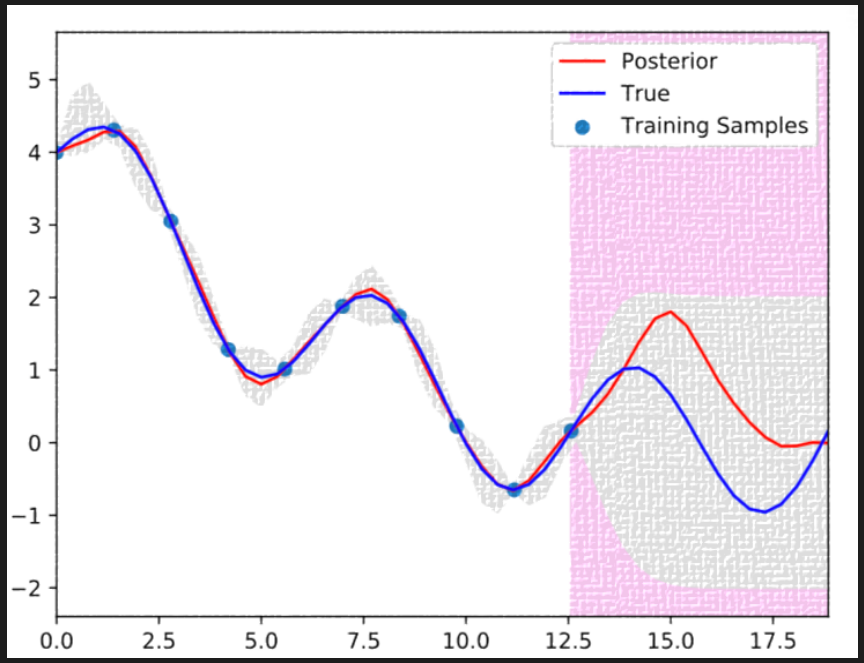
HOWEVER BUT...


Text
Documents
IRISH - DETECTED
ENGLISH
ENGLISH
SPANISH
ARABIC
ag ag ag ag ag ag ag ag ag ag ag
ag ag ag ag ag ag ag ag ag ag ag
ag ag ag ag ag ag ag ag ag ag ag
98/5000
by at that at the same at the
bottom is towards



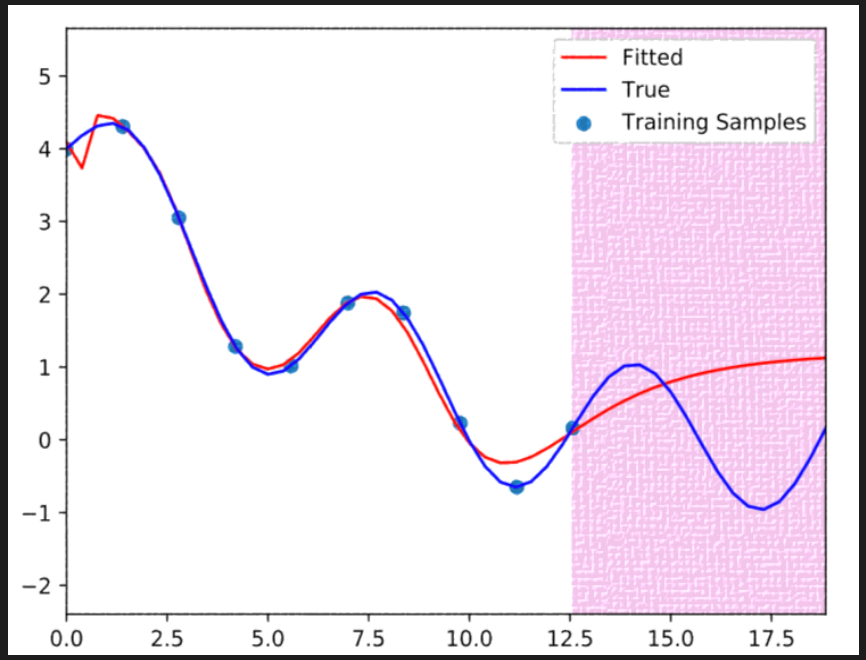

Self-Driving
48
kph

DREAMS vs. FACTS vs. CALIBRATION



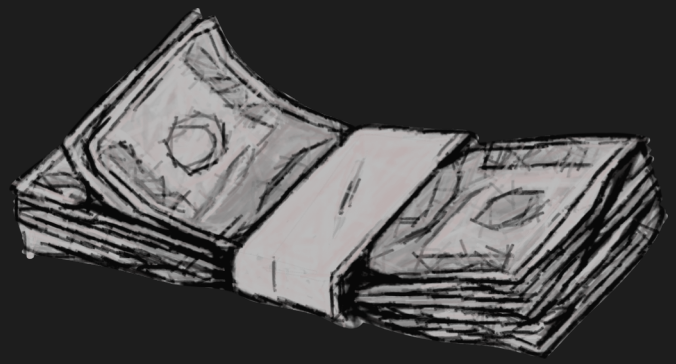
Gaussian Process

x



NN with 2 hidden layers

SOURCES OF UNCERTAINTY



EPISTEMIC

Model Uncertainty

Reducible

ALEATORIC

Data Uncertainty

Irreducible

Just speak clearly.


02
TWO
ROADS
diverged in a yellow
wood...

THE TWO ROADS OF ML

There exists ideal parameters.
I just need to find them.

Data gives all answers.

BEING FREQUENTIST

Model parameters also have a probability distribution.

We have an initial (prior) guess about these.

BEING BAYESIAN

THE TWO ROADS OF ML

Use Maximum Likelihood Estimation (MLE):

$$\theta_{ML} = \arg\max p(x|\theta)$$

model params

data

likelihood

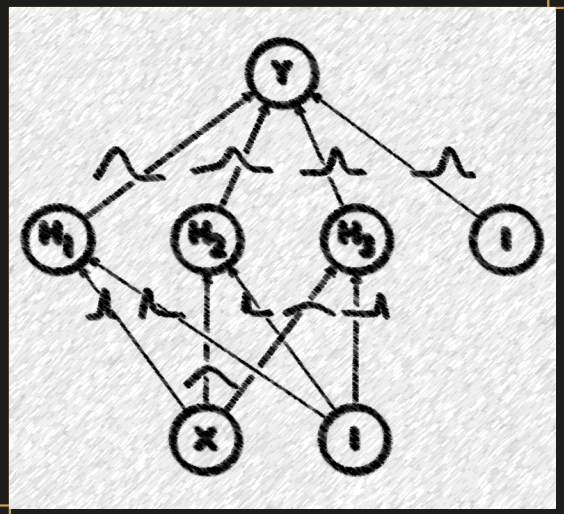
BEING FREQUENTIST

Assume a distribution over parameters:

$$\underbrace{p(\theta|x)}_{\text{posterior}} = \underbrace{\frac{\overbrace{p(x|\theta)}^{\text{likelihood}}\overbrace{p(\theta)}^{\text{prior}}}{\int p(x|\theta')p(\theta')d\theta'}}_{= p(x), \text{ the evidence}}$$

BEING BAYESIAN

(being) BAYESIAN (about) DEEP LEARNING



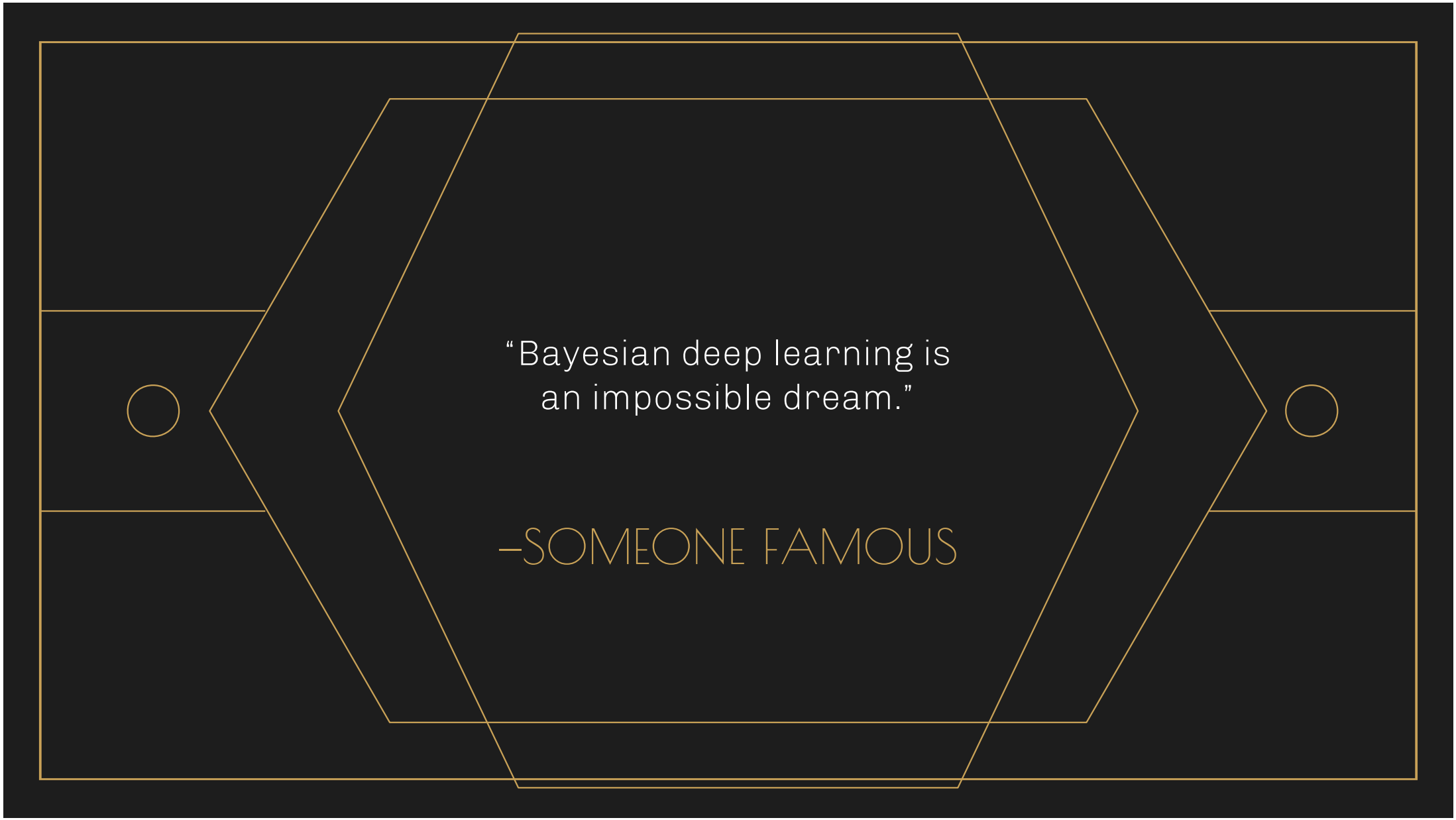
One way to be Bayesian about a model:

- Assuming that the model parameters come from a distribution also
  - + Putting a prior distribution on them

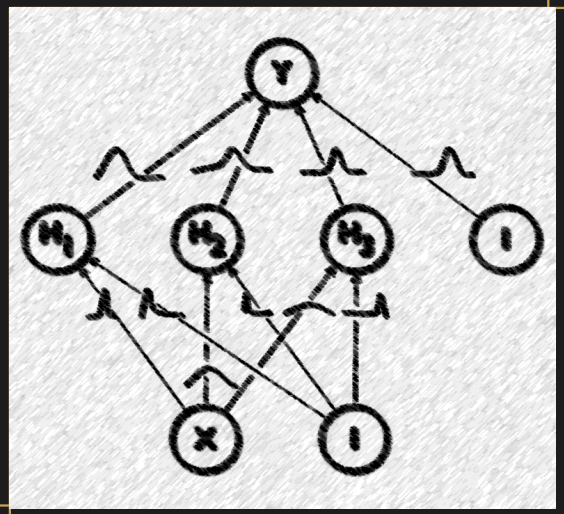
$$\theta \sim p(\theta)$$

$$p(\theta|x_{train})$$ } learning

$$p(x_{out}|x_{in}, x_{train}) = \int p(x_{out}|x_{in}, \theta')p(\theta'|x_{train})d\theta'$$ } prediction


“Bayesian deep learning is
an impossible dream.”
–SOMEONE FAMOUS

(being) BAYESIAN (about) DEEP LEARNING



One way to be Bayesian about a model:

- Assuming that the model parameters come from a distribution also
  - + Putting a prior distribution on them

$$\theta \sim p(\theta)$$

$$p(\theta|x_{train}) = \frac{p(x_{train}|\theta)\, p(\theta)}{\int p(x_{train}|\theta')p(\theta')d\theta'}$$

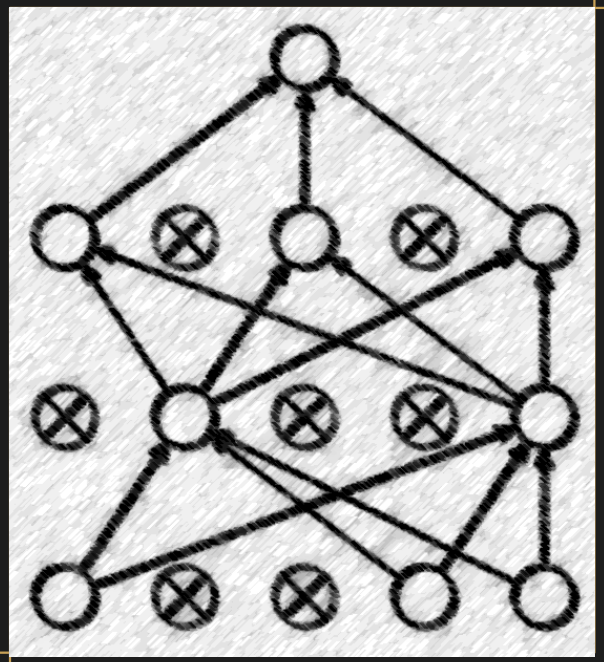
Intractable

03 APPROACHES

How can we even?

Model Selection and Uncertainty

MC-DROPOUT AS AN APPROXIMATION



IDEA

We want:

1. 1. Distributions over parameters
2. 2. The model to know when its uncertain

IMPLEMENTATION

Use Dropout
not only in training time
but also in *test time*

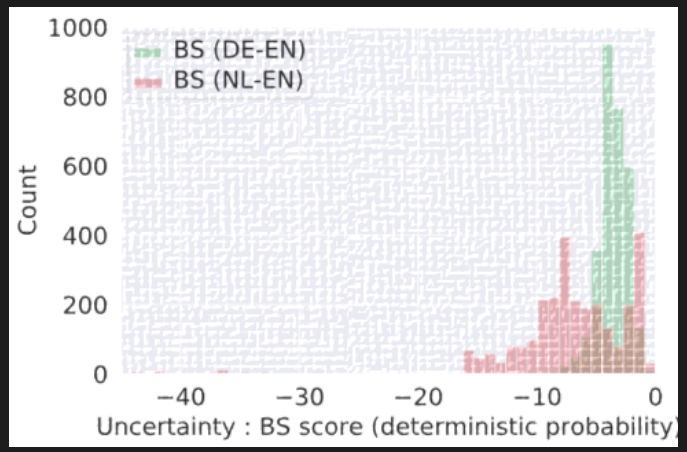
MATHEMATICAL VALIDITY

NN with MC-dropout is an
approximation to the Gaussian process

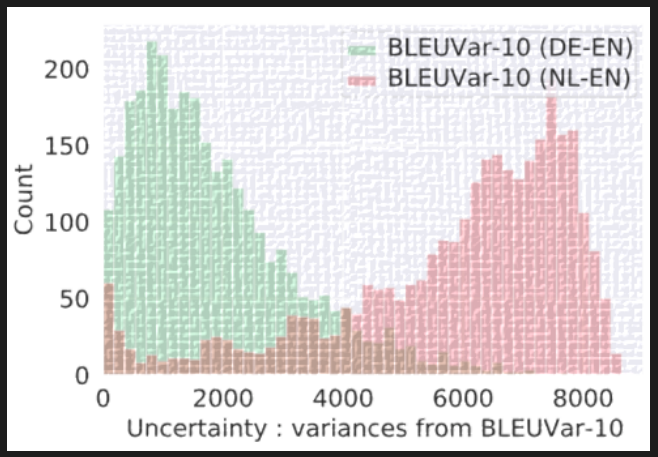
Gal (2016), Uncertainty in Deep Learning

MC-DROPOUT on TRANSFORMER

Xiao, Gomez and Gal (2020), *Wat zei je? Detecting Out-of-Distribution Translations with Variational Transformers*



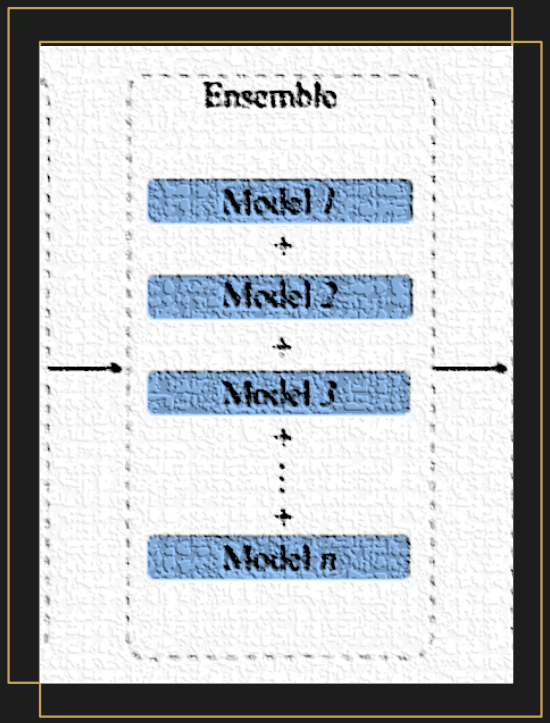
x



Lengths	1-10	11-20	21-30	31-40	41-50	51+
DE-EN (In-dist)	2579.93	1867.11	1613.95	1507.85	1502.78	2794.11
NL-EN (OOD)	4772.16	5388.25	5579.82	6039.02	6705.95	7042.69

Table 1: Average BLEUVar for output sentences of various lengths from Figure 2.

ENSEMBLE METHODS



IDEA

We want:

1. 1. Distributions over parameters
2. 2. The model to know when its uncertain

IMPLEMENTATION

Train multiple models and let them approximate the probability distribution.

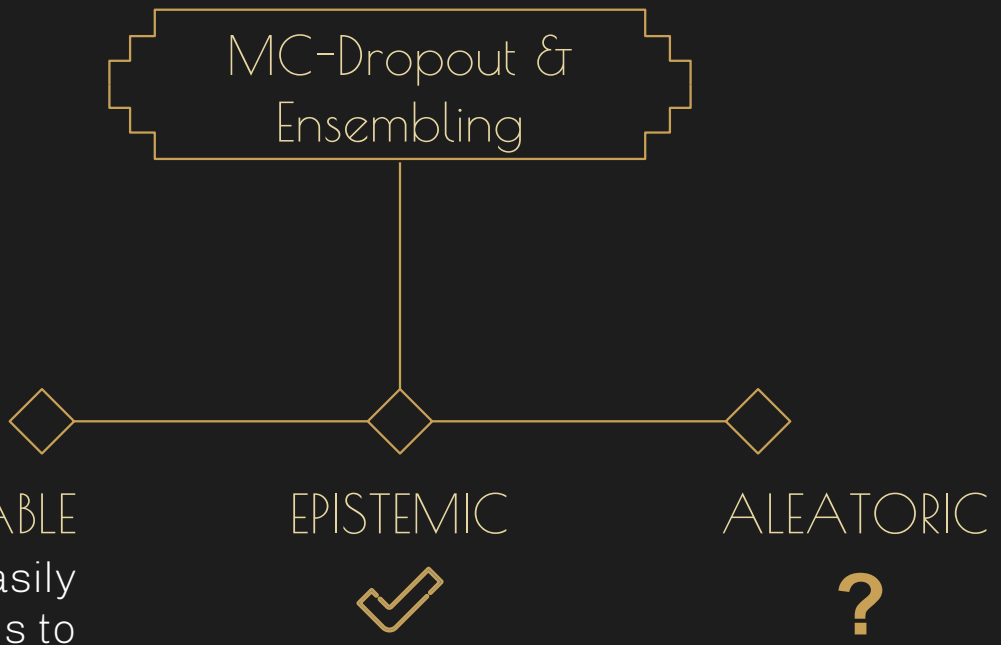
Lakshminarayanan, Pritzel and Blundell (2017), *Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles, NIPS.*

ENSEMBLING TRANSFORMERS?

Many for improving performance (as is a natural gain).

But none (that I could find) with an uncertainty focus.

I CAN USE THIS HOW?



THE OTHER APPROACH

Epistemic Uncertainty?

DISTRIBUTIONS ON PARAMETERS

Aleatoric Uncertainty?

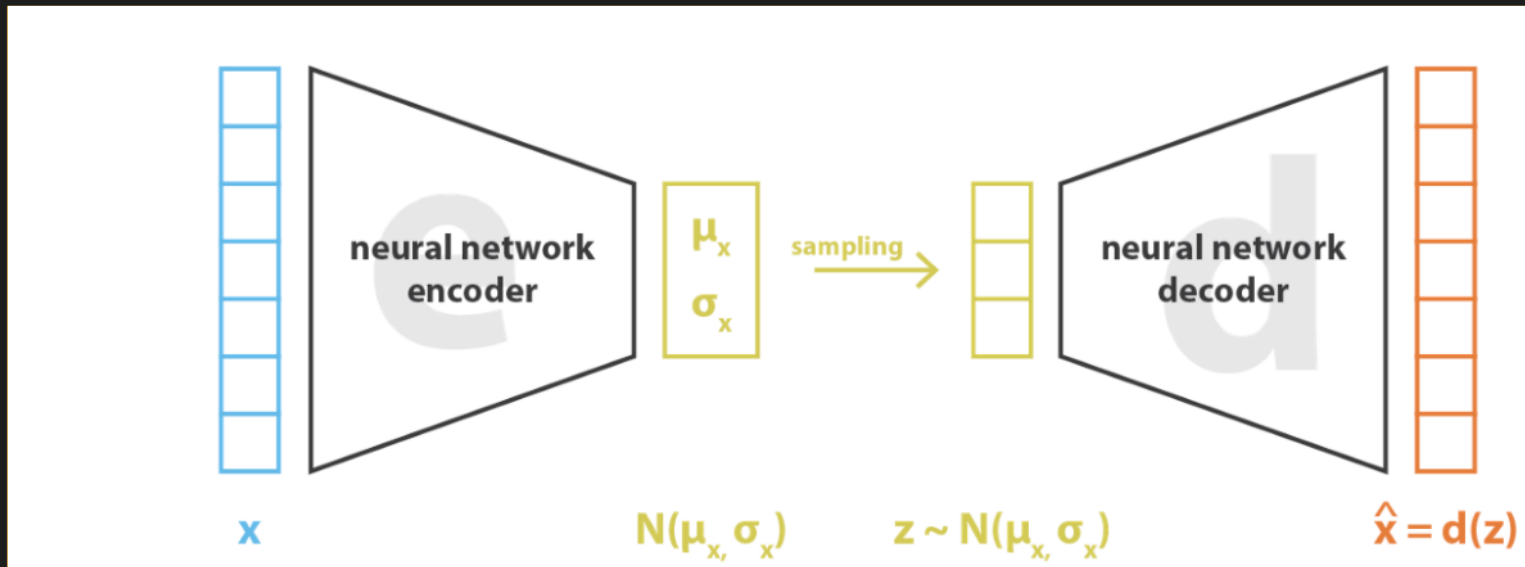
UNCERTAINTY IN REPRESENTATION LEARNING

* The second half

Representations and Uncertainty*

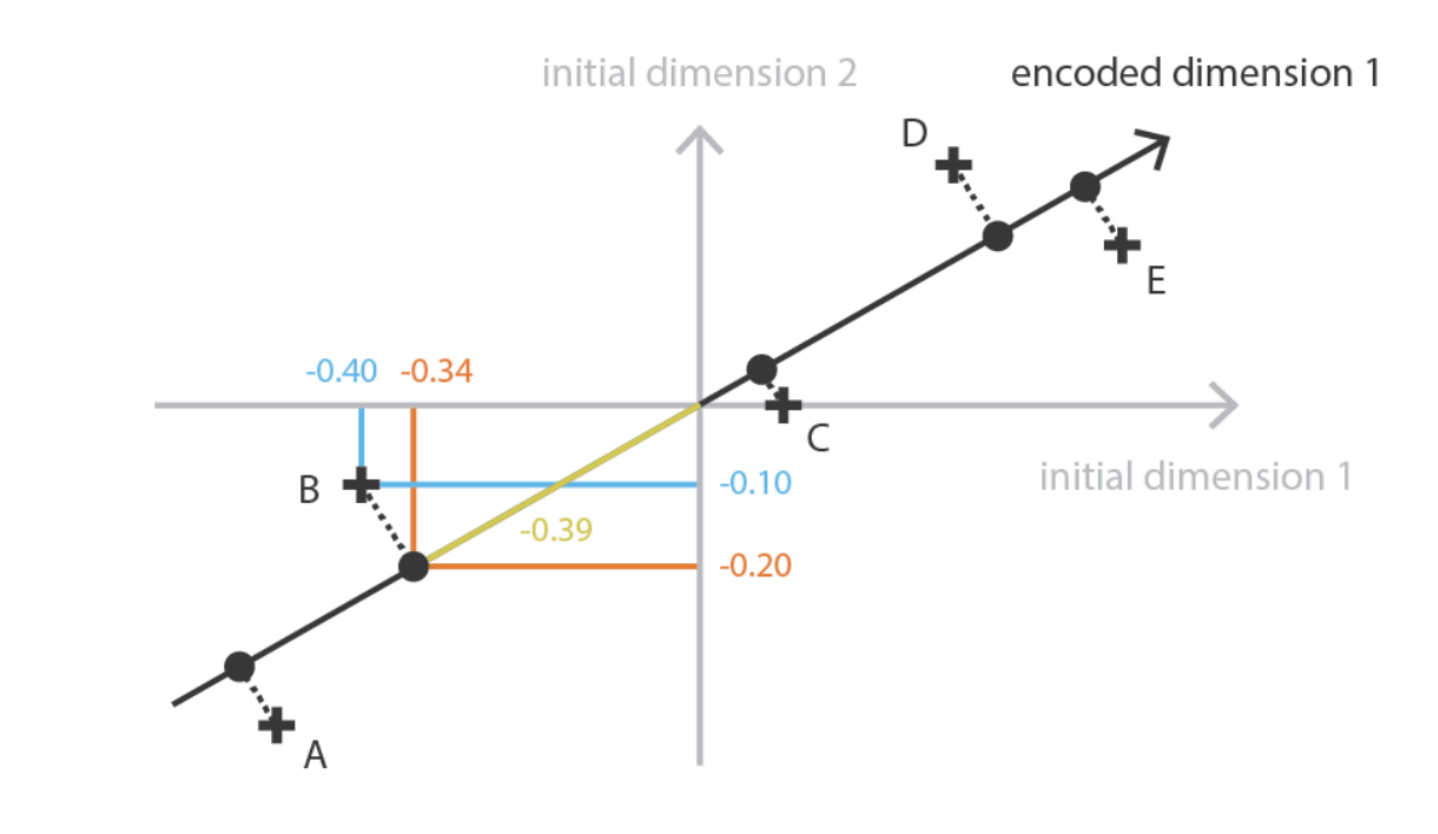
VARIATIONAL AUTO-ENCODERS

Kingma and Welling (2014), Auto-Encoding Variational Bayes

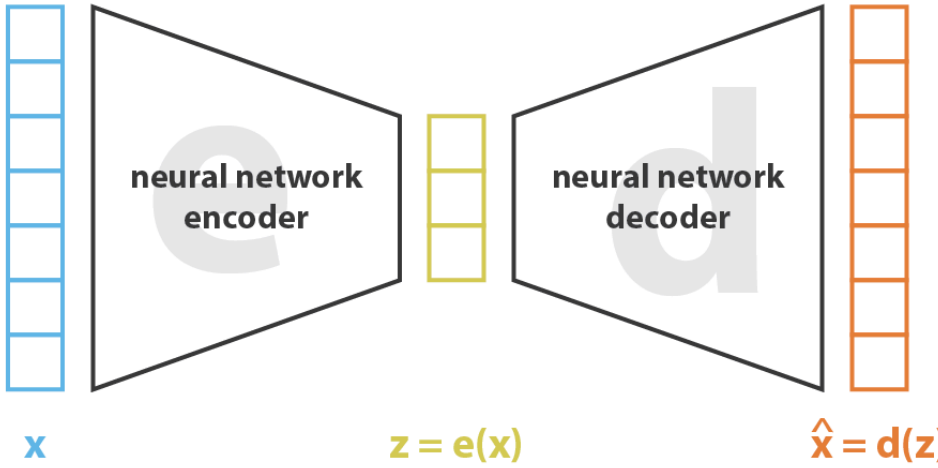


Illustrations: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

SIMPLEST EVER REPRESENTATION LEARNING



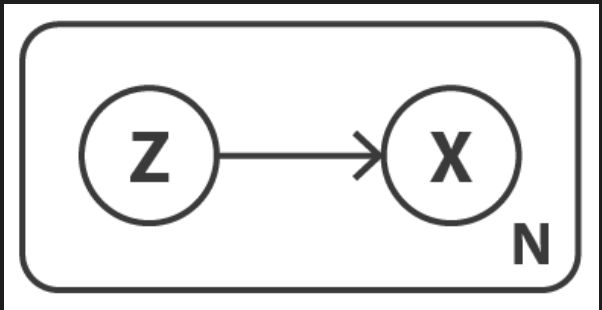
NONLINEARITY? AUTOENCODER



$$\text{loss} = \| x - \hat{x} \|^2 = \| x - d(z) \|^2 = \| x - d(e(x)) \|^2$$

Enter: Uncertainty

Bayesian Latent Variable Modeling



$$p(X, Z|\theta) = p(X|Z, \theta)p(Z|\theta)$$

$$= \prod_{i=1}^{N} (x_i|z_i, \theta)p(z_i|\theta)$$

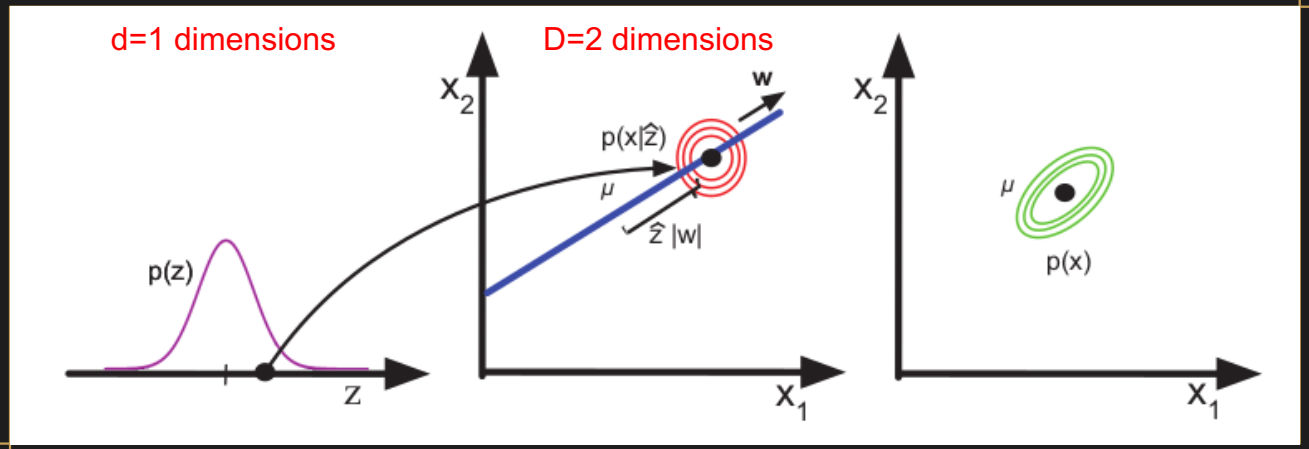
eg.,

$$p(Z|\theta) \sim \mathcal{N}(0, I)$$

$$p(X, Z|\theta) \sim \mathcal{N}(f(Z), cI)$$

A Probabilistic Take on PCA

Find the latent variable z that "summarizes" the data x



THE MODEL

Joint distribution is:

Has analytical solution! <3

D x d

$$p(X, Z|\theta) = \prod_{i=1}^{n} p(x_i|z_i, \theta)p(z_i|\theta) = \prod_{i=1}^{n} \mathcal{N}(x_i|Vz_i + \mu, \sigma^2 I).\mathcal{N}(z_i|0, I)$$

And what if the Subspaces are not Linear?

PCA

Linear transformation only.

Nonlinearity?

Can we make the transformation here nonlinear?

$$p(X, Z|\theta) = \prod_{i=1}^{n} p(x_i|z_i, \theta)p(z_i|\theta) = \prod_{i=1}^{n} \mathcal{N}(x_i|Vz_i + \mu, \sigma^2 I).\mathcal{N}(z_i|0, I)$$

PRIOR ON LATENT

↓

Will keep this simple.

↓

Note z's are independent.

GENERATOR OF SUBSPACE

↓

Can try complicated model

↓

NN

Variational Auto-Encoders

Simply a deep generalization of PCA:

PCA:

$$p(X,Z|\theta) = \prod_{i=1}^{n} p(x_i|z_i,\theta)p(z_i|\theta) = \prod_{i=1}^{n} \underbrace{\mathcal{N}(x_i|Vz_i+\mu,\sigma^2 I)}_{\text{Simple generator model}}.\underbrace{\mathcal{N}(z_i|0,I)}_{\text{Simple Prior}}$$

VAE:

$$p(X,Z|\theta) = \prod_{i=1}^{n} p(x_i|z_i,\theta)p(z_i|\theta) = \prod_{i=1}^{n} \underbrace{\left(\prod_{j=1}^{D} \mathcal{N}\left(x_{ij}|\mu_j(z_i),\sigma_j^2(z_i)\right)\right)}_{\text{“Complex” generator (well, also a Gaussian) which is parametrized by a NN}} \underbrace{\mathcal{N}(z_i|0,I)}_{\text{Same simple i.i.d Prior}}$$

Highly non-linear functions

The Intractability Problem

$$p(X, Z|\theta) = \prod_{i=1}^{n} p(x_i|z_i, \theta)p(z_i) = \prod_{i=1}^{n} \left(\prod_{j=1}^{D} \mathcal{N}\left(x_{ij}|\mu_j(z_i), \sigma_j^2(z_i)\right) \right) \mathcal{N}(z_i|0, I)$$

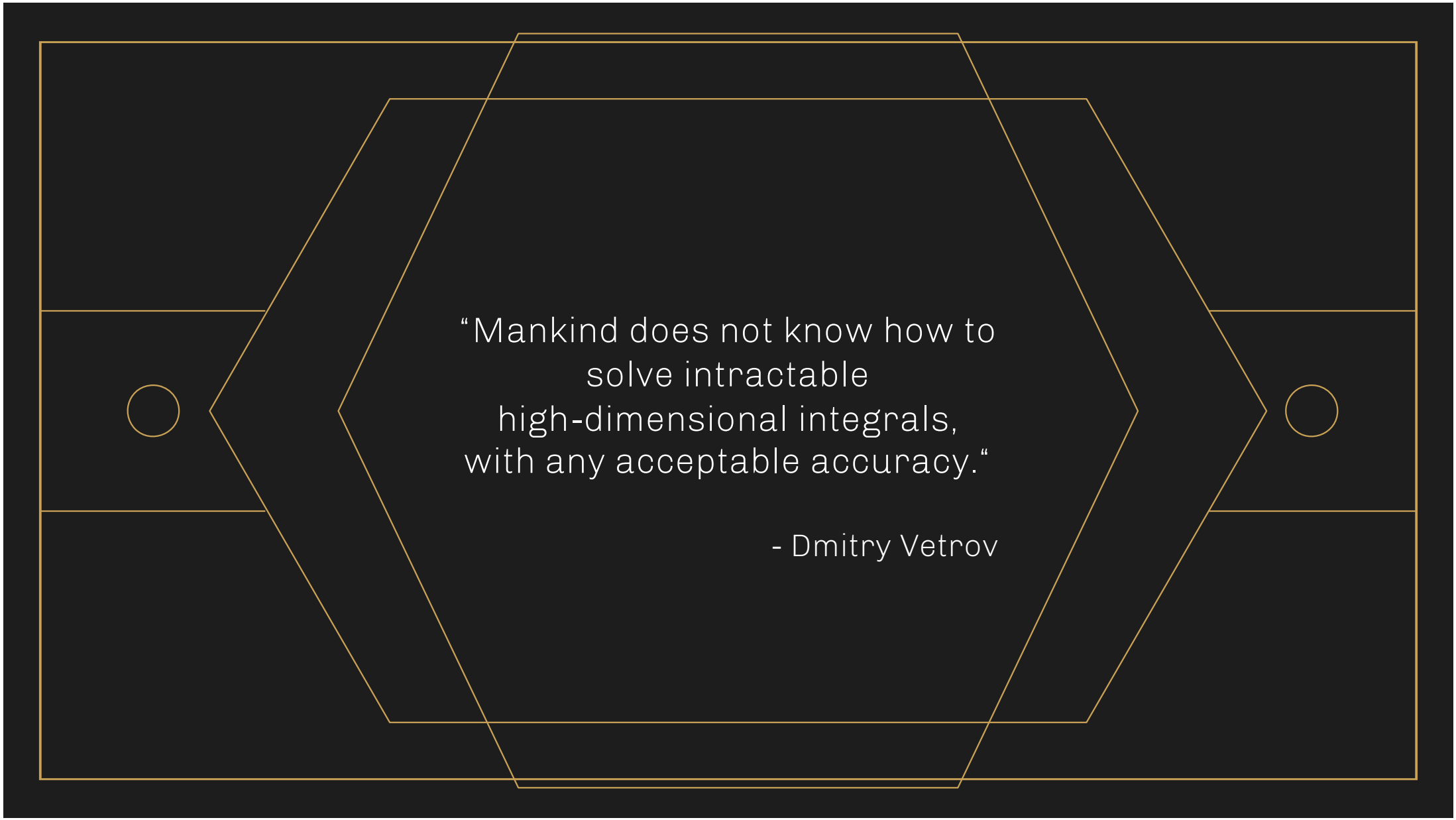
Highly non-linear functions of z,
this is not tractable anymore.

No analytical solution.

(Is a latent model, so) EM algorithm? Requires we compute (in the E-step):

$$p(Z) = \prod_{i-1}^{n} p(z_i|x_i, \theta) = \prod_{i=1}^{n} \frac{p(x_i|z_i, \theta)p(z_i)}{\int p(x_i|z_i, \theta)p(z_i)dz_i}$$

*Variational inference as the solution to intractability.


“Mankind does not know how to
solve intractable
high-dimensional integrals,
with any acceptable accuracy.“
- Dmitry Vetrov

OUR EXPERT TEAM

SAMPLING

eg. Markov Chain Monte Carlo
Sampling the exact posterior

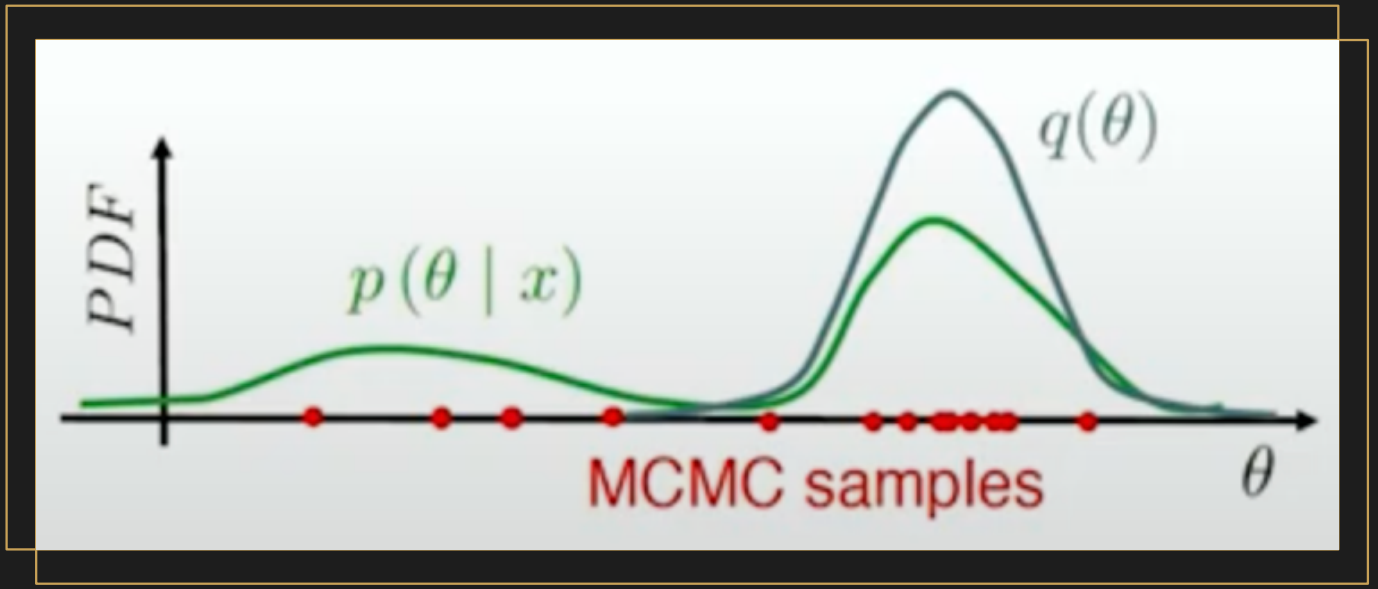
- Unbiased
- But slow

VARIATIONAL INFERENCE

Learn an approximation for the posterior

- Fast
- But biased

OUR EXPERT TEAM



VARIATIONAL INFERENCE

$$\text{min. } KL(q(\theta)||p(\theta|x)) = \int q(\theta) \log \frac{q(\theta)}{p(\theta|x)} d\theta$$

Want to approximate $p(\theta|x)$ with $q(\theta)$

But how to measure goodness of fit?
Kullback-Leibler divergence:

$$\text{min. } KL(q(\theta)||p(\theta|x)) = \int q(\theta) \log \frac{q(\theta)}{p(\theta|x)} d\theta$$

?

?

VARIATIONAL INFERENCE

Learn an approximation for the posterior

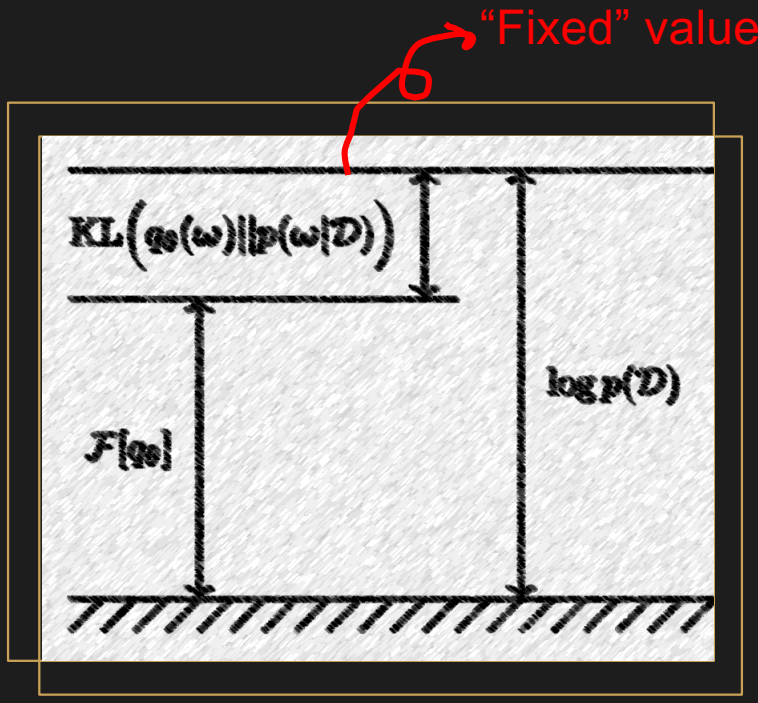
- Fast
- But biased

VARIATIONAL INFERENCE

$$\log p(x) = \int q(\theta) \log p(x)\, d\theta = \int q(\theta) \log \frac{p(x,\theta)}{p(\theta|x)}\, d\theta$$

$$= \int q(\theta) \log \frac{p(x,\theta) q(\theta)}{p(\theta|x) q(\theta)}\, d\theta$$

$$= \underbrace{\int q(\theta) \log \frac{p(x,\theta)}{q(\theta)}\, d\theta}_{\boldsymbol{L(q(\theta))},\ \text{evidence lower bound}} + \underbrace{\int q(\theta) \log \frac{q(\theta)}{p(\theta|x)}\, d\theta}_{\boldsymbol{KL(q(\theta)||p(\theta|x))},\ \text{the Kullback-Leibler divergence}}$$



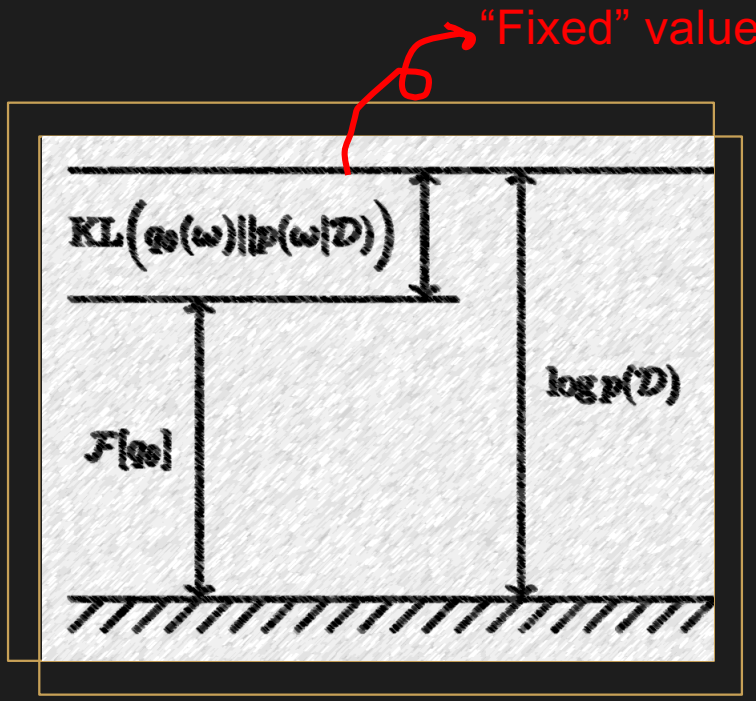
MAXIMIZING ELBO

$$L(q(\theta)) = \int q(\theta) \log \frac{p(x,\theta)}{q(\theta)} d\theta = \int q(\theta) \log \frac{p(x|\theta)p(\theta)}{q(\theta)} d\theta$$

$$= \int q(\theta) \log p(x|\theta) d\theta + \int q(\theta) \log \frac{p(\theta)}{q(\theta)} d\theta$$

$$= \underbrace{\mathbb{E}_{q(\theta)} \log p(x|\theta)}_{\text{Maximizing the log likelihood}} - \underbrace{KL(q(\theta)||p(\theta))}_{\text{Closest to prior}}$$

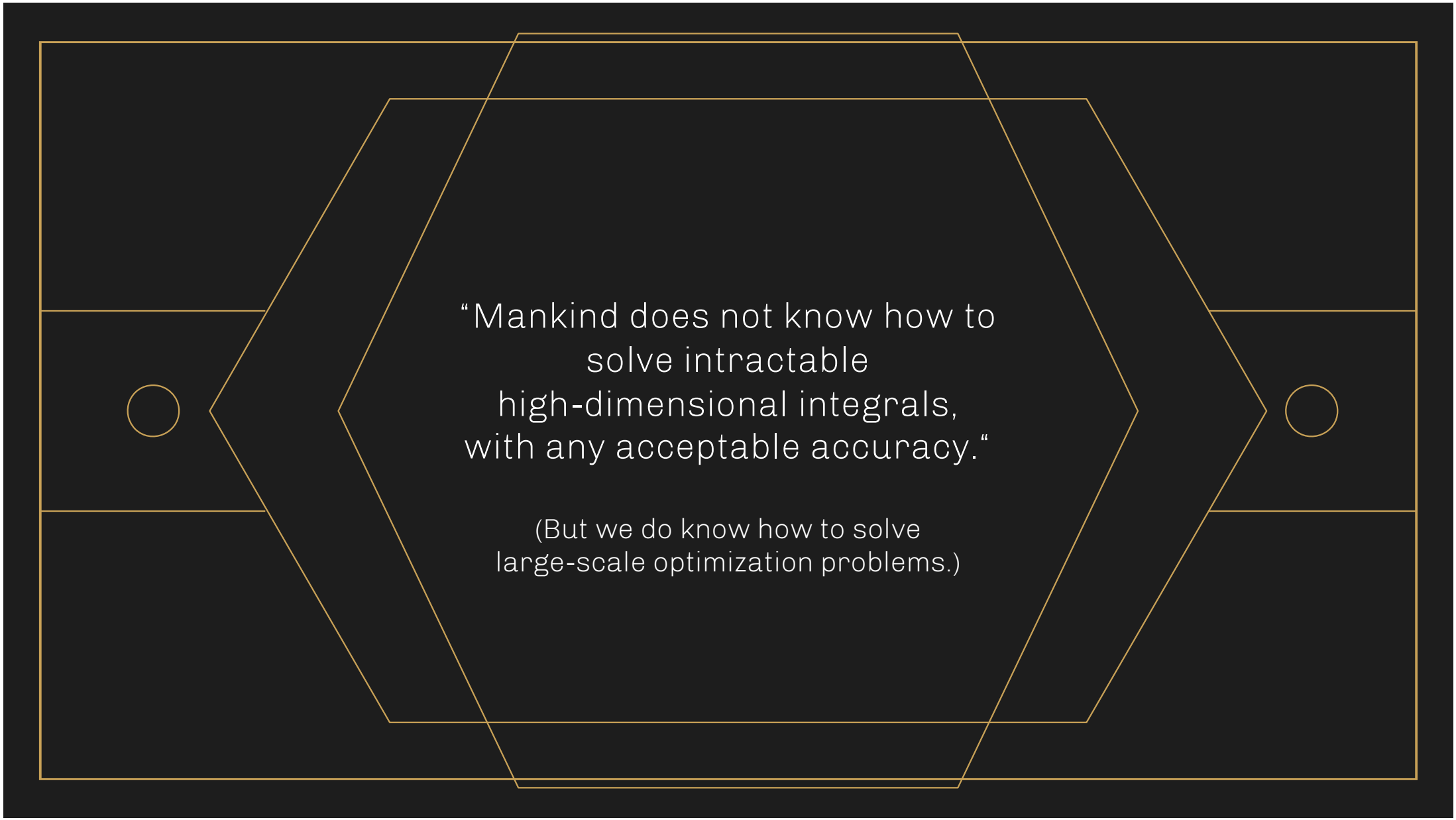
Will be the form of our Loss Function!



The Variational Approximation

A parametric approximation! Restrict the function to parametric family,

$$q(z_i|x_i, \phi) \approx p(z_i|x_i, \theta)$$


“Mankind does not know how to
solve intractable
high-dimensional integrals,
with any acceptable accuracy.“
(But we do know how to solve
large-scale optimization problems.)

The Variational Approximation

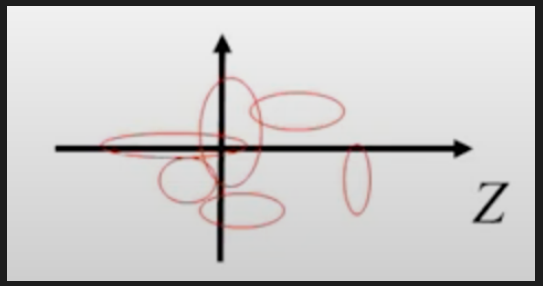
A parametric approximation! Restrict the function to parametric family,

$$q(z_i|x_i, \phi) \approx p(z_i|x_i, \theta)$$

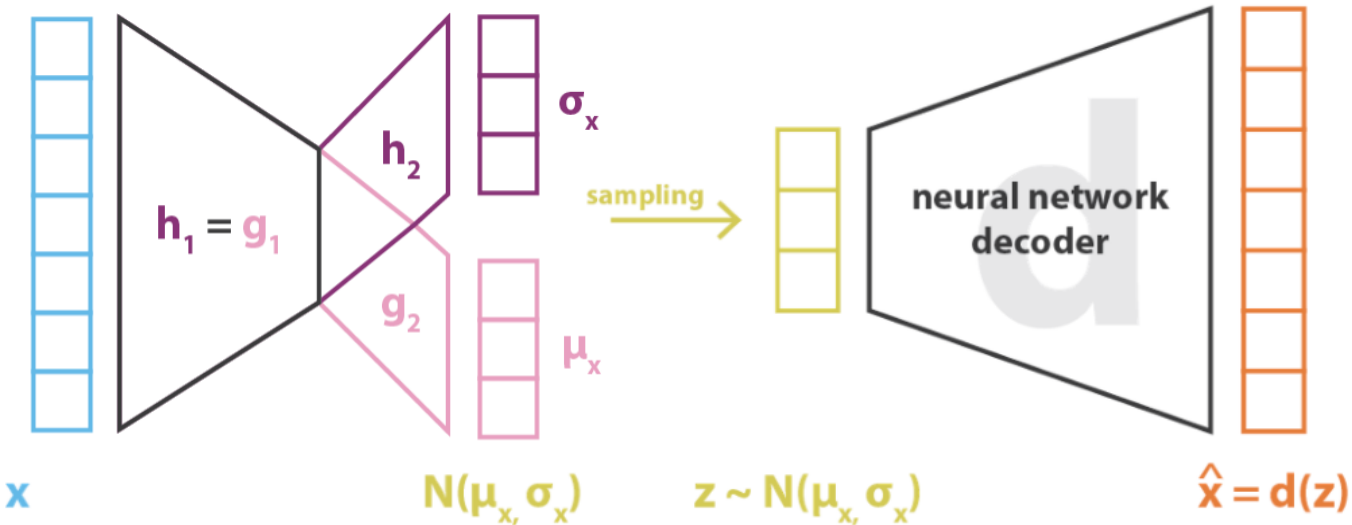
Let the approximate function has the form of *Factorized Gaussians*:

$$q(z_i|x_i, \phi) = \prod_{j=1}^{d} \mathcal{N}(z_{ij}|\mu_j(x_i), \sigma_j^2(x_i))$$

whose mean and variance are given by *another neural network* parametrized by ϕ.

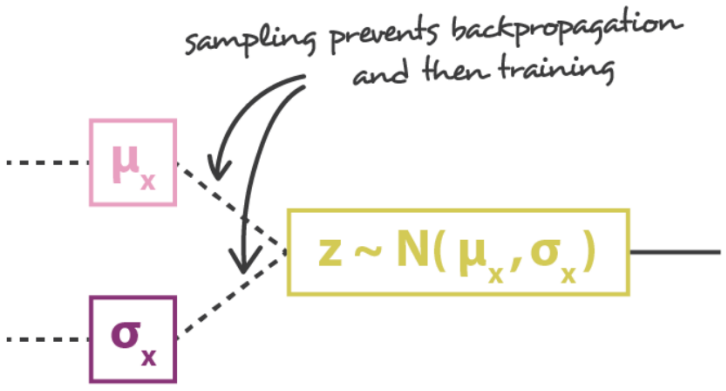


The Eventual Architecture

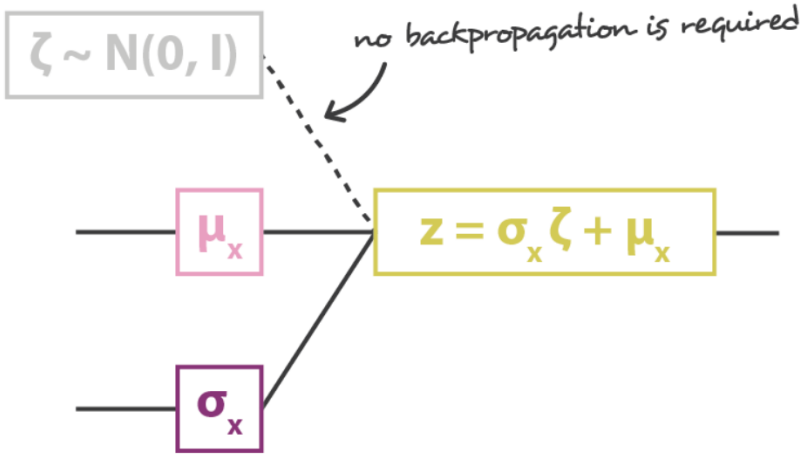


$$\text{loss} = \| x - \hat{x} \|^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

One Final Addition: Reparametrization Trick

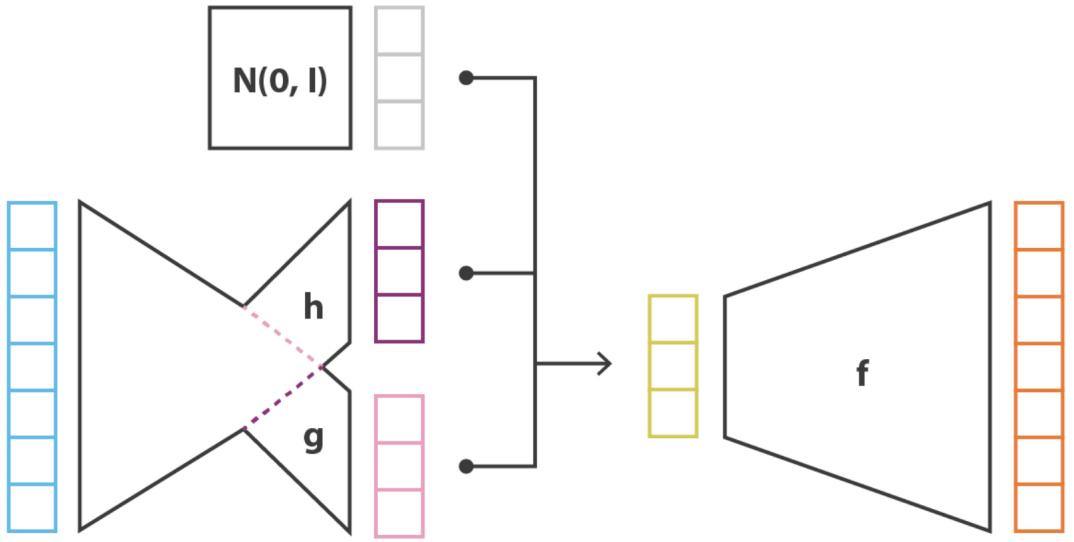


sampling without reparametrisation trick



sampling with reparametrisation trick

One Final Addition: Reparametrization Trick



x

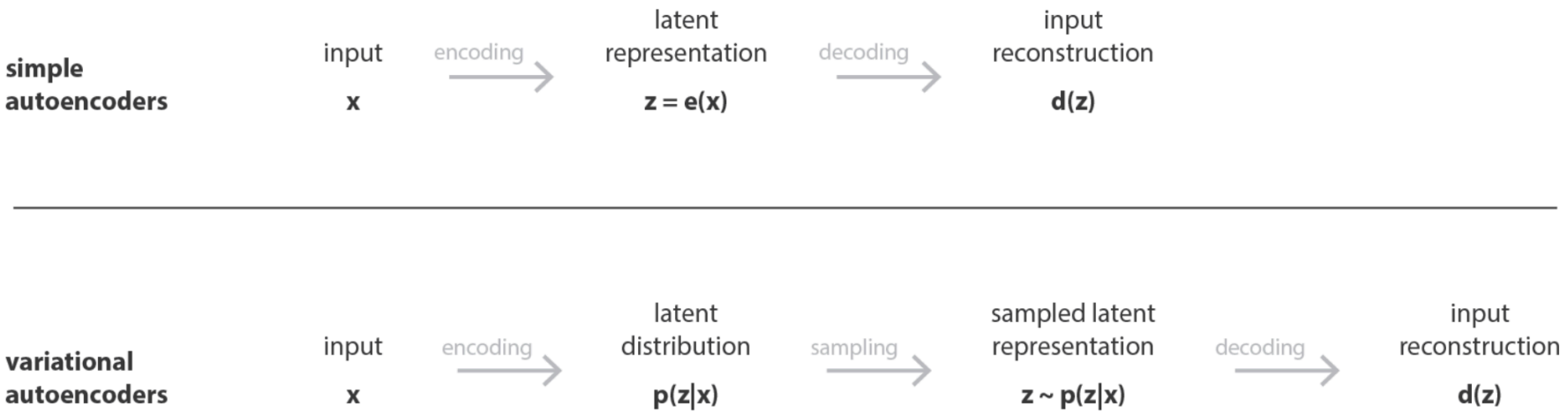
$\mu_x = g(x)$
$\sigma_x = h(x)$
$\zeta \sim N(0, I)$

$z = \sigma_x \zeta + \mu_x$

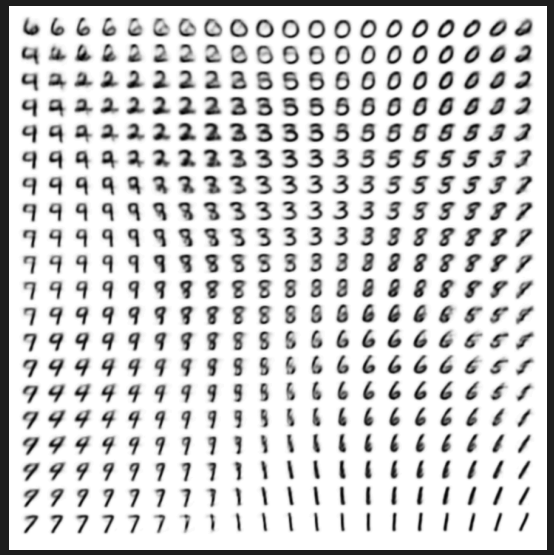
$\hat{x} = f(z)$

$$\text{loss} = C \, || \, x - \hat{x} \, ||^2 + KL[\, N(\mu_x, \sigma_x), N(0, I) \,] = C \, || \, x - f(z) \, ||^2 + KL[\, N(g(x), h(x)), N(0, I) \,]$$

ENTER: UNCERTAINTY



I CAN USE THIS HOW?



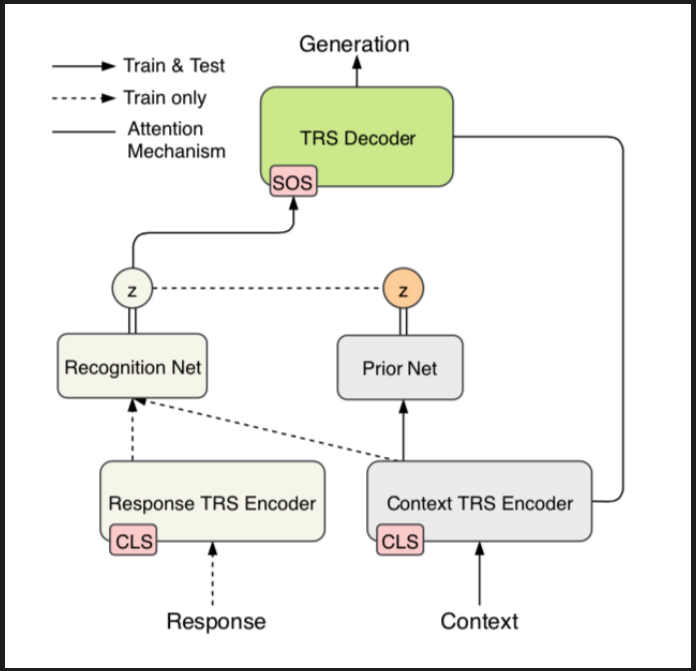
VAEs

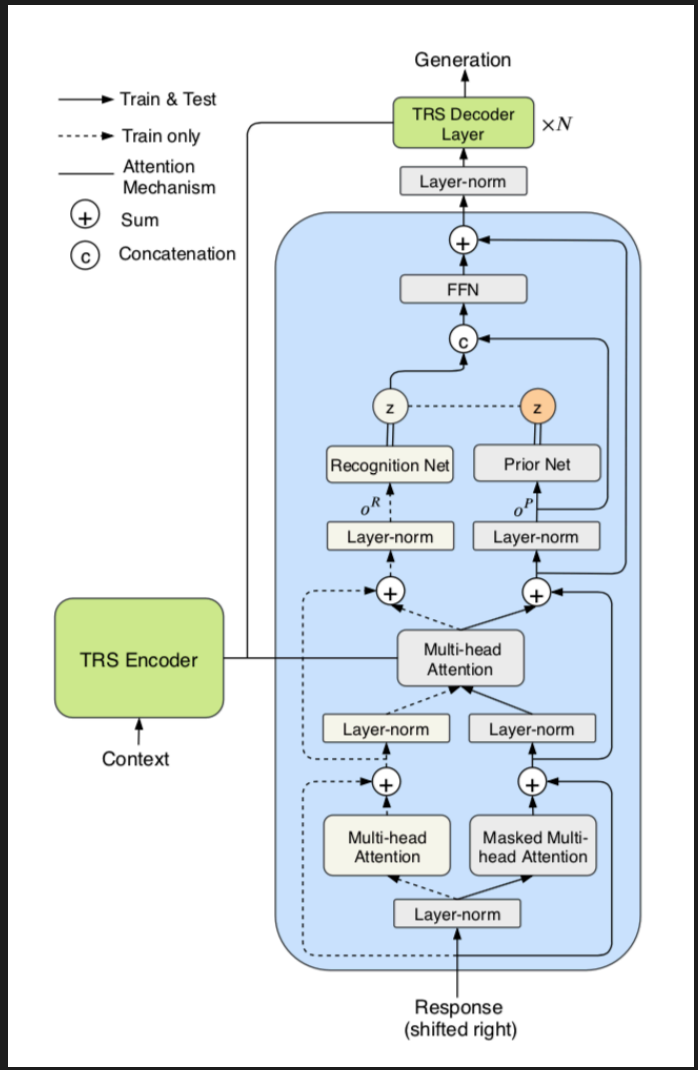
GENERATIVE MODEL
UNSUPERVISED LEARNING
OF MANIFOLDS

UNCERTAINTY IN
REPRESENTATIONS

Infusing The Transformer with VAEs

Lin et al. (2020), Variational Transformers for Diverse Response Generation





Every research is something missing.

What are we missing here?

WHAT IS MISSING HERE?

ASSUMPTION 1

Gaussian latents.

Normalizing Flows.

ASSUMPTION 2

I.I.D. priors.

Conditional Neural Process.

THANKS!

(Hopefully some remaining time for)

Questions?