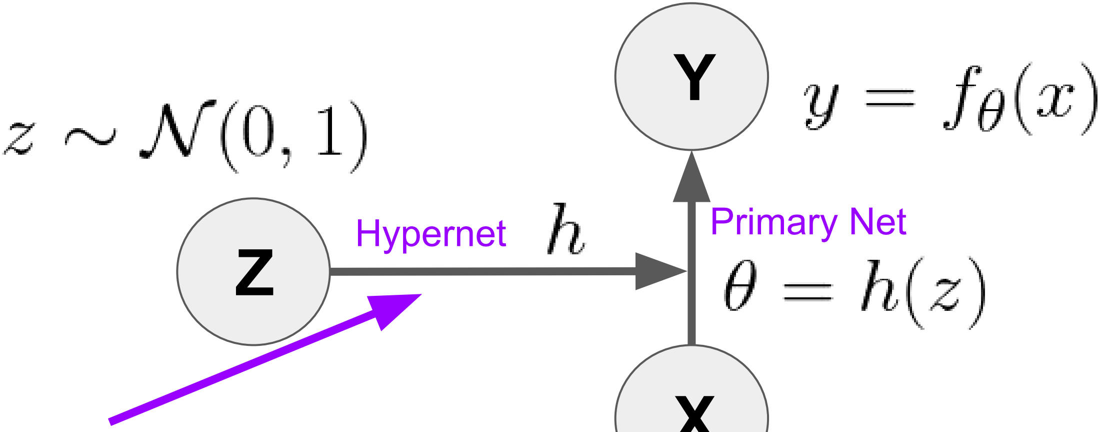
Bayesian Hypernetworks

David Krueger*, Chin-Wei Huang*
Riahsat Islam, Ryan Turner, Aaron Courville
MILA and McGill RLLAB

What’s a Bayesian Hypernet?

Hypernet: a DNN that generates params of another DNN (the “primary net”)

Task: predict y from x



Y

$y = f_\theta(x)$

$z \sim \mathcal{N}(0, 1)$

Z

Hypernet h

Primary Net

$\theta = h(z)$

X

Think GAN / VAE / Real NVP

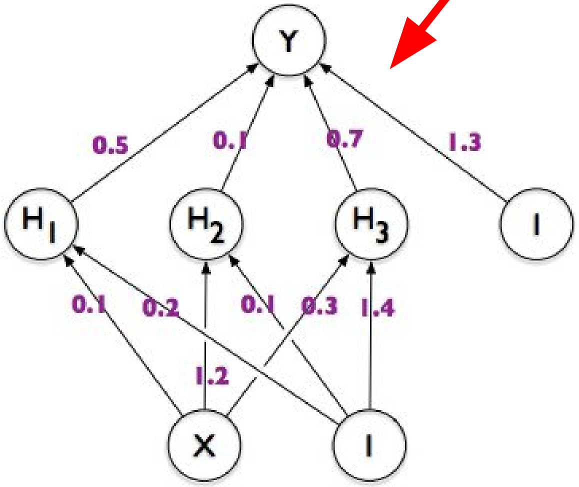
What is a Bayesian Neural Net?

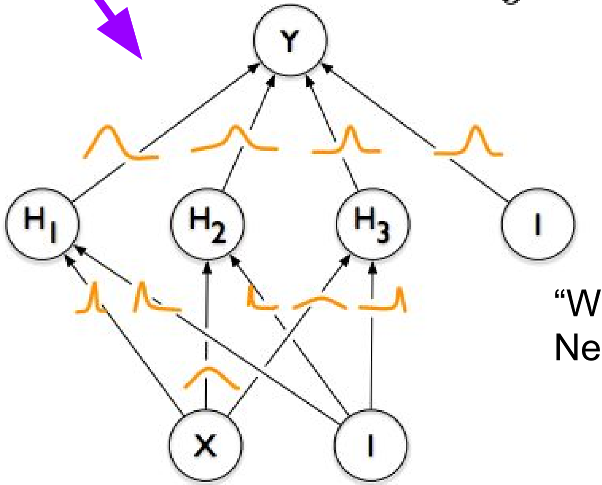
Bayes Rule: $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$

Predict using ensemble:

$$p(y|x) = \int p(y|x,\theta)p(\theta|\mathcal{D})d\theta$$

Argmax





“Weight Uncertainty in Neural Networks” - Blundell et al 2015

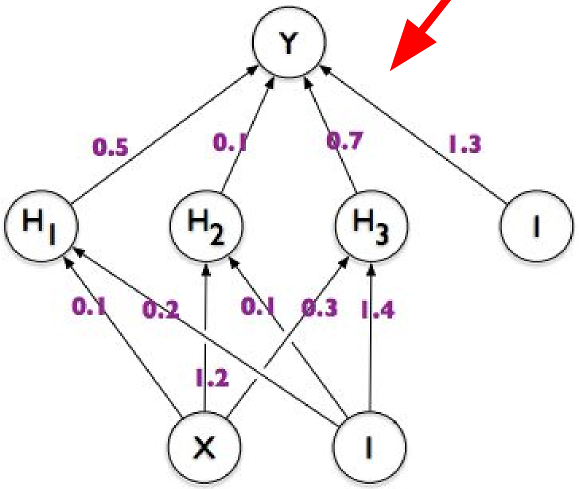
What’s **special** about Bayesian Neural Nets?

Bayes Rule: $p(\theta|\mathcal{D}) \propto p(\mathcal{D}|\theta)p(\theta)$

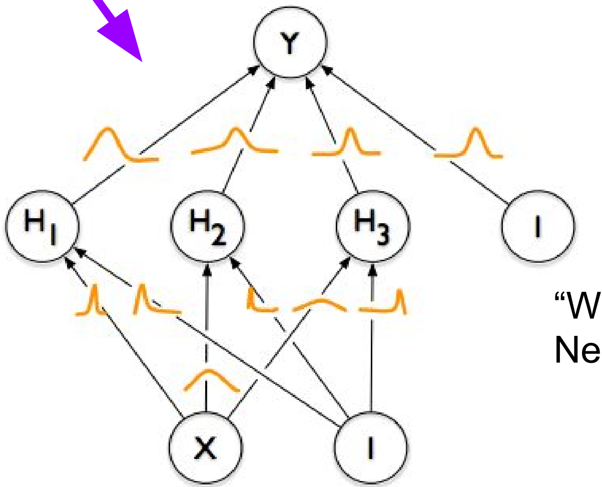
“Knows what it knows”

“Calibrated confidence”

“That’s my best guess”



“I’m 99% sure!”



“Weight Uncertainty in Neural Networks” - Blundell et al 2015

Example: self-driving cars

Q: Is there a person in the road?

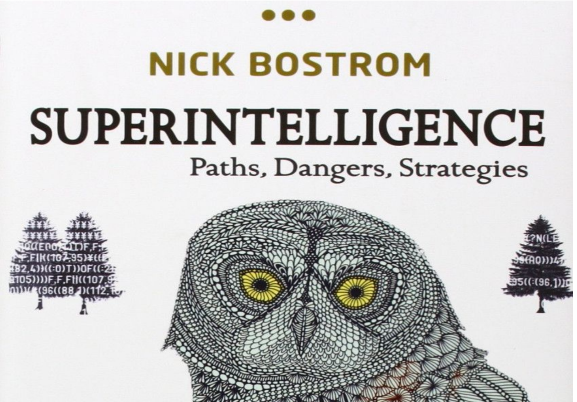
Car: No, and….

“That’s my best guess”

“I’m 51% sure!”

“I’m 99.999999% sure!”

Existential risk

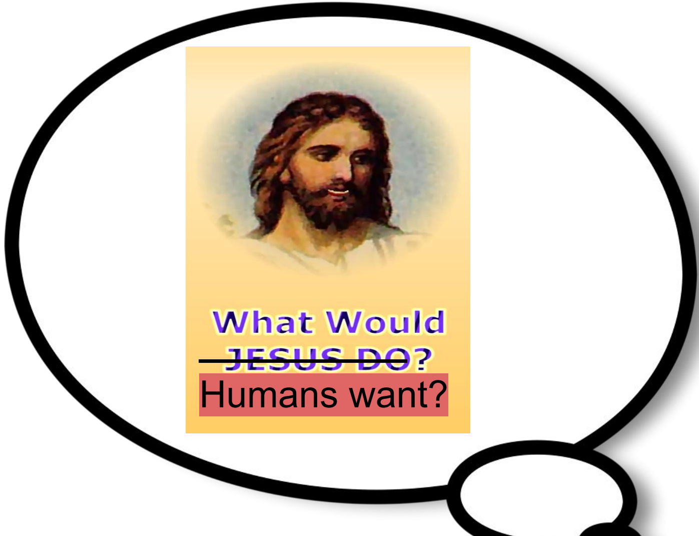

NICK BOSTROM

SUPERINTELLIGENCE

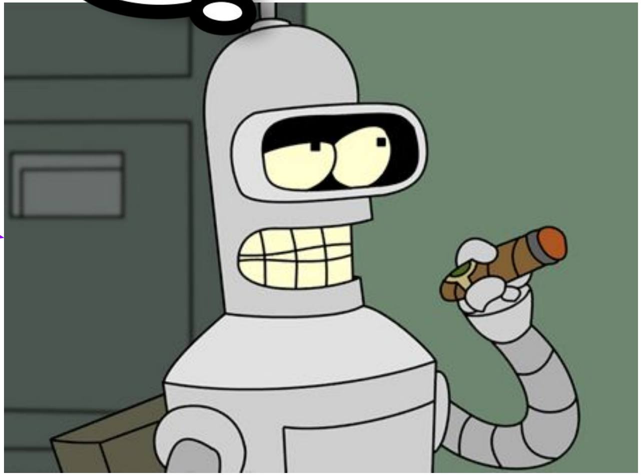
Paths, Dangers, Strategies

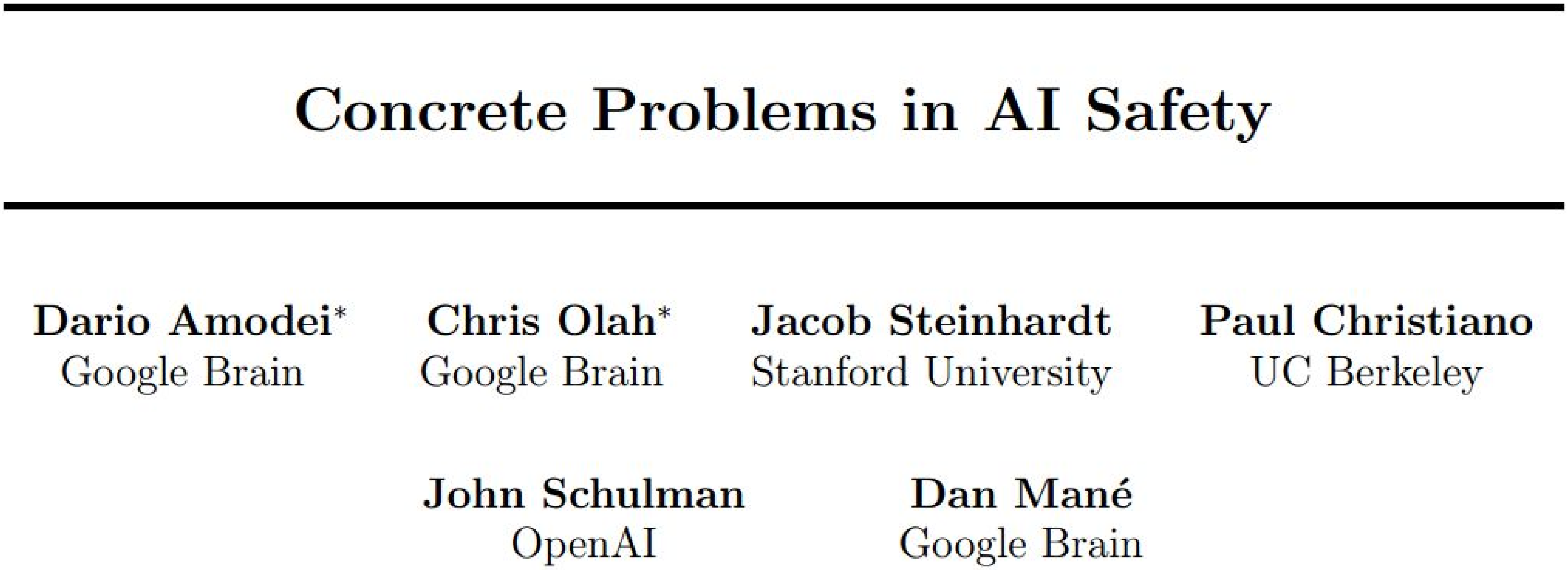
“Is the default outcome doom?”

AI Safety



KEEP CALM AND ASK A HUMAN





Concrete Problems in AI Safety

Dario Amodei*
Google Brain

Chris Olah*
Google Brain

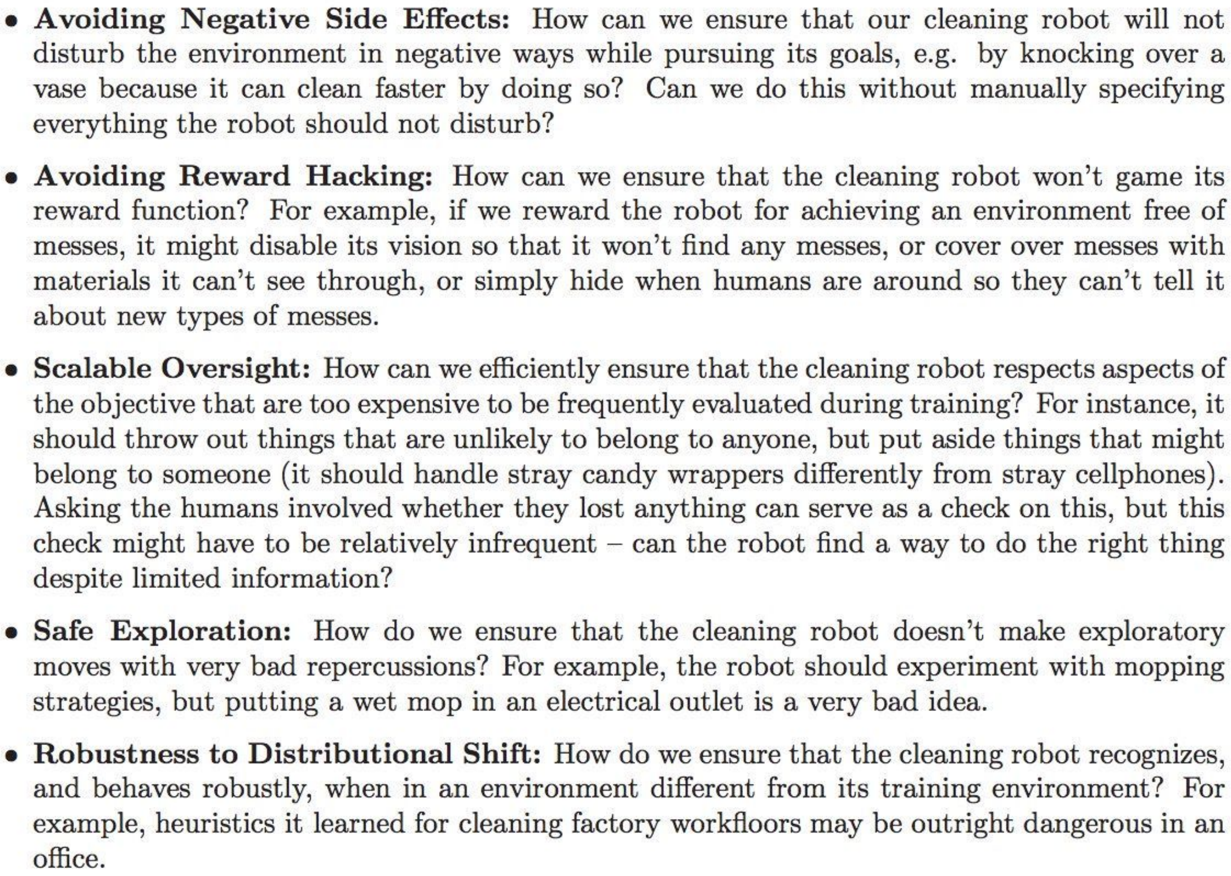
Jacob Steinhardt
Stanford University

Paul Christiano
UC Berkeley

John Schulman
OpenAI

Dan Mané
Google Brain

- Five “concrete problems”, calibrated confidence helps in 4/5



- **Avoiding Negative Side Effects:** How can we ensure that our cleaning robot will not disturb the environment in negative ways while pursuing its goals, e.g. by knocking over a vase because it can clean faster by doing so? Can we do this without manually specifying everything the robot should not disturb?

← Reward uncertainty

- **Avoiding Reward Hacking:** How can we ensure that the cleaning robot won't game its reward function? For example, if we reward the robot for achieving an environment free of messes, it might disable its vision so that it won't find any messes, or cover over messes with materials it can't see through, or simply hide when humans are around so they can't tell it about new types of messes.

← I don't know,
ask Tom Everrit

- **Scalable Oversight:** How can we efficiently ensure that the cleaning robot respects aspects of the objective that are too expensive to be frequently evaluated during training? For instance, it should throw out things that are unlikely to belong to anyone, but put aside things that might belong to someone (it should handle stray candy wrappers differently from stray cellphones). Asking the humans involved whether they lost anything can serve as a check on this, but this check might have to be relatively infrequent – can the robot find a way to do the right thing despite limited information?

← **active learning**

- **Safe Exploration:** How do we ensure that the cleaning robot doesn't make exploratory moves with very bad repercussions? For example, the robot should experiment with mopping strategies, but putting a wet mop in an electrical outlet is a very bad idea.

← safe exploration

- **Robustness to Distributional Shift:** How do we ensure that the cleaning robot recognizes, and behaves robustly, when in an environment different from its training environment? For example, heuristics it learned for cleaning factory workfloors may be outright dangerous in an office.

← **anomaly detection**

Technique

Variational Inference for Bayesian DNNs

- ELBO:

$$\mathcal{L}(\phi) = \mathbf{E}_{q_\phi(\theta)}[\log p(\mathcal{D}|\theta) + \log p(\theta) - \log q_\phi(\theta)]$$

$$\log p(\mathcal{D}) = \mathcal{L}(\phi) + KL(q_\phi(\theta)||p(\theta|\mathcal{D}))$$

constant **maximize** **minimize**

Encourages stochasticity!

- Examples:

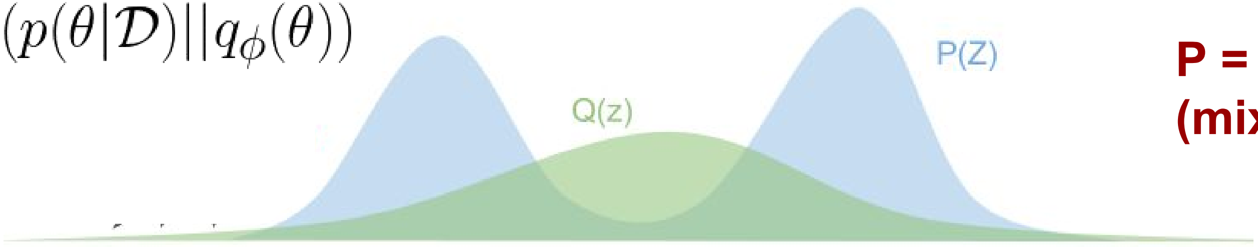
  Weight Uncertainty

  Variational Dropout / MC dropout

Problem with Variational Inference: KL divergence

Variational inference can **underestimate** uncertainty!

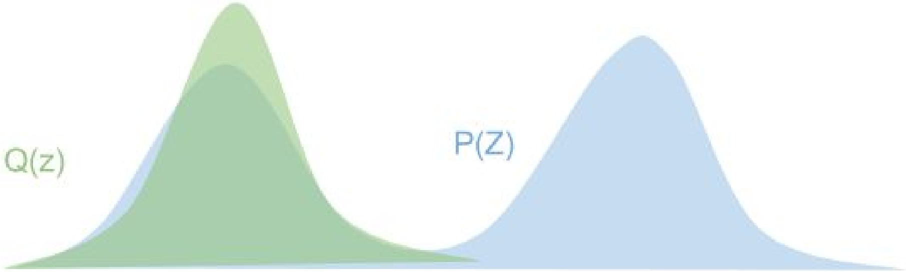
$$KL(p(\theta|\mathcal{D})||q_\phi(\theta))$$



P = true posterior (mixture of Gaussians)

$$KL(q_\phi(\theta)||p(\theta|\mathcal{D}))$$

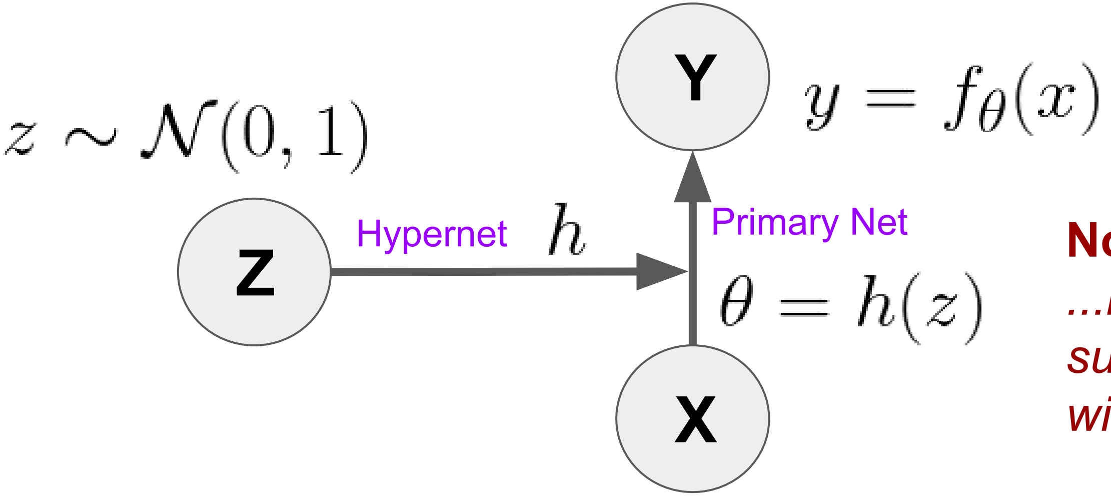
Q = variational approx (Gaussian)



Are Bayesian Hypernets the solution?

- **Previous work:** approximate posterior is factorial:
- **Use a DNN!**
  - ○ $\Rightarrow q(\theta|\mathcal{D})$ can be **dependent**, **multimodal**

$$q(\theta|\mathcal{D}) = \prod_i q(\theta_i|\mathcal{D})$$

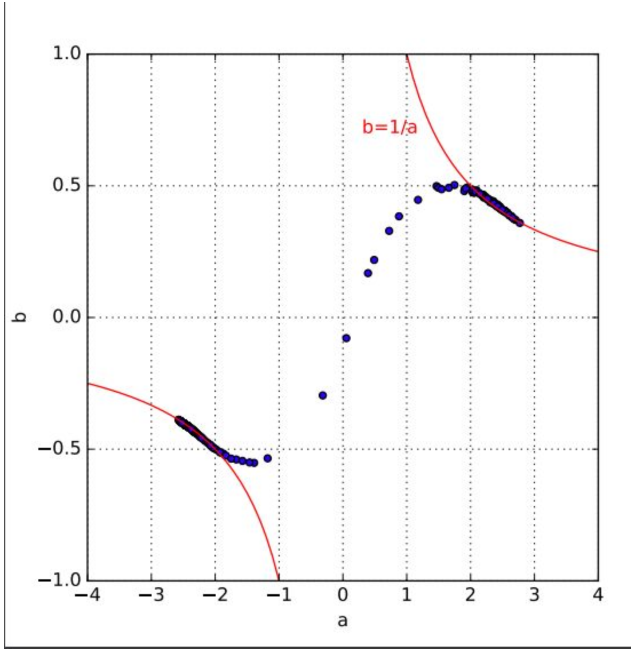


Note: h must be invertible!

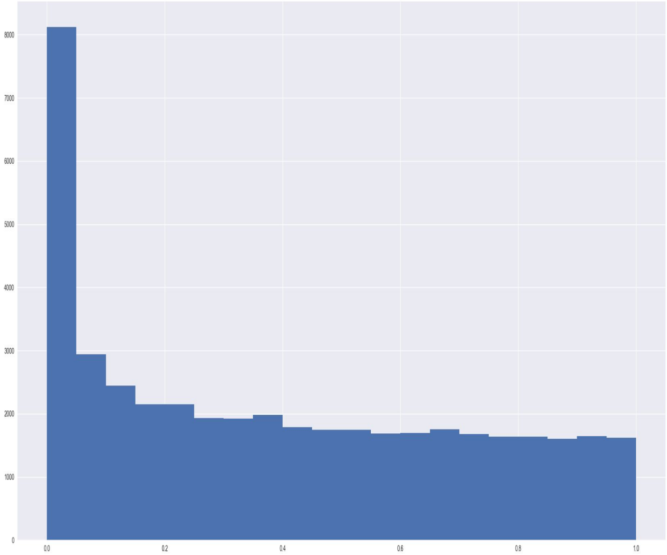
...but the image of h can be a subset of R^|theta|, unlike with NICE (generative model)

Some Qualitative Results:

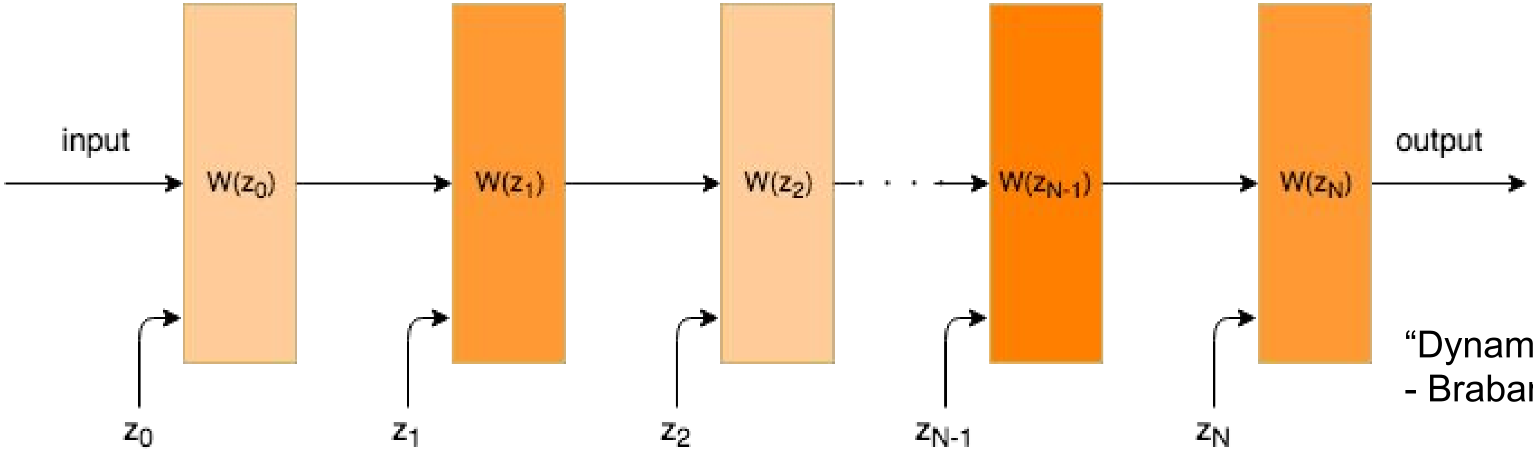
Multimodality

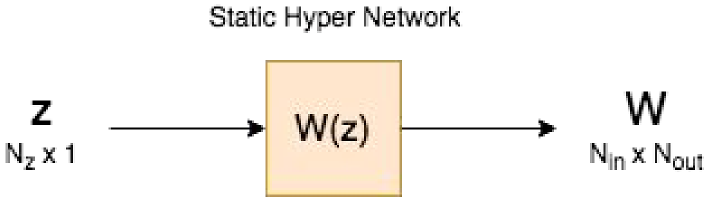


Correlation



Background: Hypernetworks



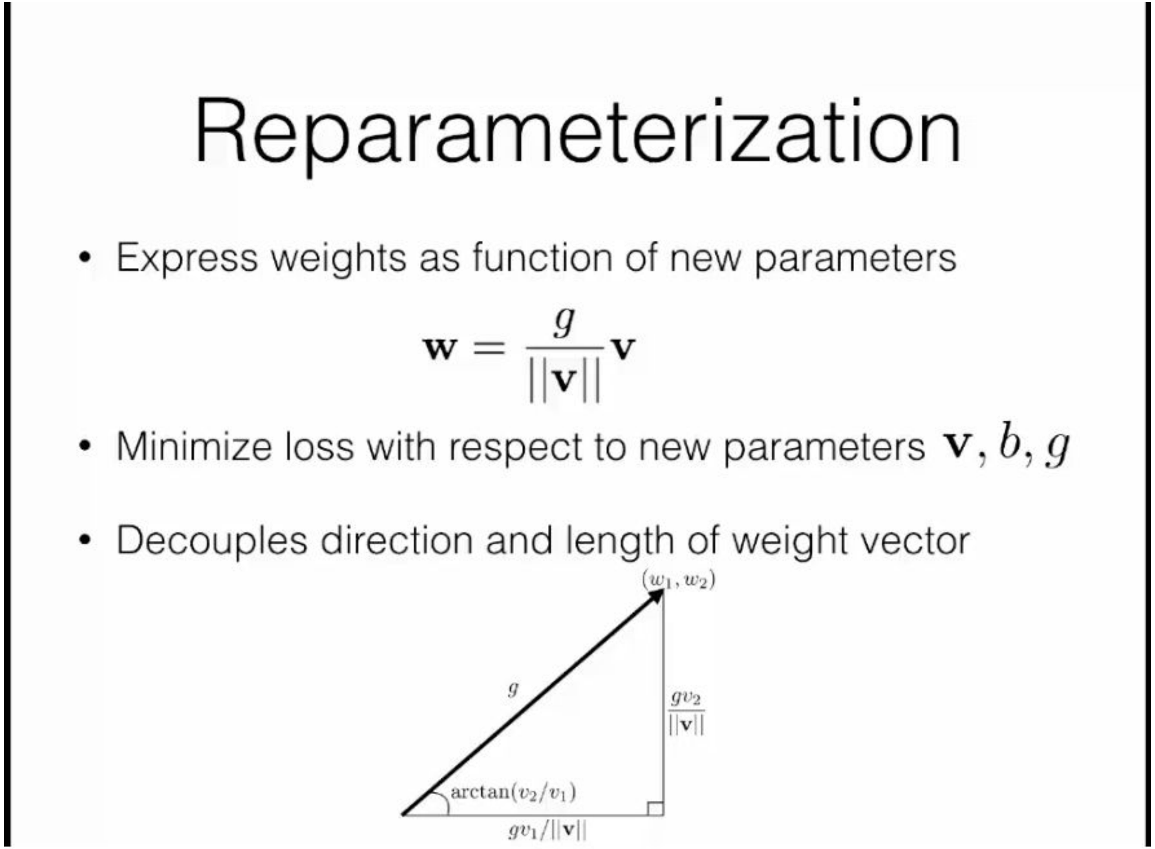


“Dynamic Filter Networks” - Brabandere et al. 2016

“Learning feed-forward one-shot learners” - Bertinetto et al. 2016

“HyperNetworks” - Ha et al. 2016

Background: Weight Normalization



Reparameterization

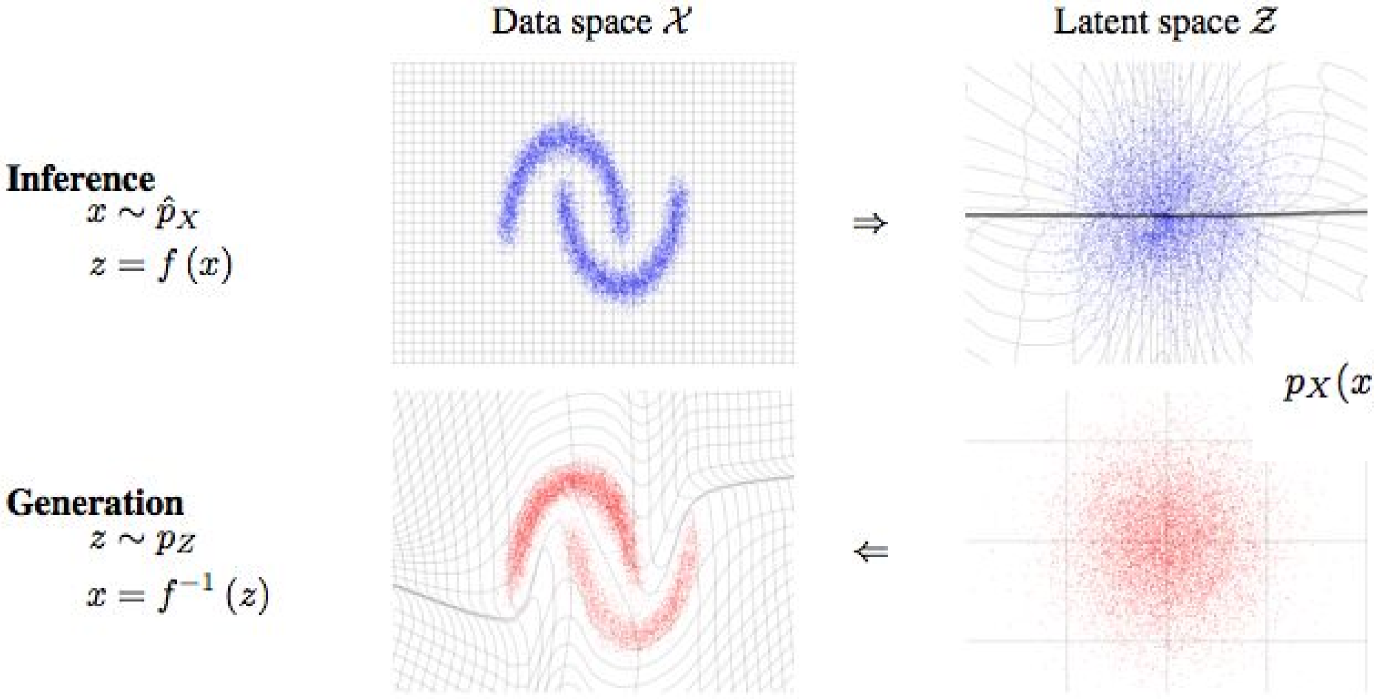
- Express weights as function of new parameters

$$\mathbf{w} = \frac{g}{||\mathbf{v}||}\mathbf{v}$$

- Minimize loss with respect to new parameters $\mathbf{v}, b, g$
- Decouples direction and length of weight vector

“Weight Normalization” -
Salimans and Kingma
(slide from NIPS 2016 talk)

Background: Invertible Deep Generative Models

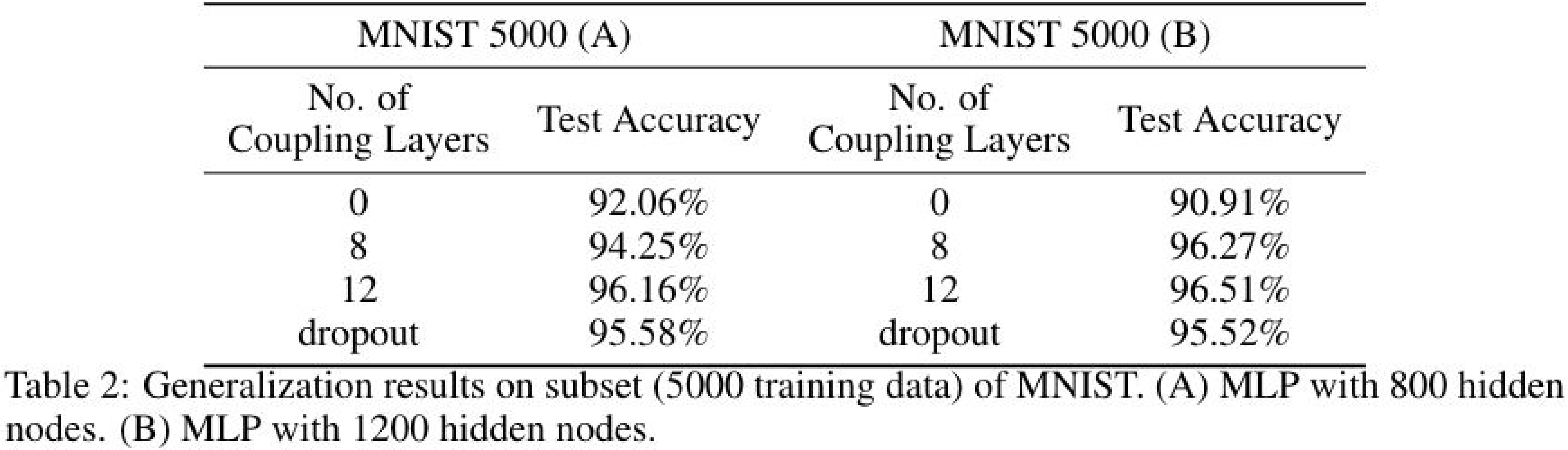


Key property: tractable likelihood (via **change of variable**):

$$p_X(x) = p_Z(z) \left| \det \left(\frac{\partial g(z)}{\partial z^T} \right) \right|^{-1}$$

(figure and equation: “Density Estimation via Real NVP” - Dinh et al. 2016)

Some results (5000 examples of MNIST):



MNIST 5000 (A)		MNIST 5000 (B)	
No. of Coupling Layers	Test Accuracy	No. of Coupling Layers	Test Accuracy
0	92.06%	0	90.91%
8	94.25%	8	96.27%
12	96.16%	12	96.51%
dropout	95.58%	dropout	95.52%

Table 2: Generalization results on subset (5000 training data) of MNIST. (A) MLP with 800 hidden nodes. (B) MLP with 1200 hidden nodes.

QUESTIONS?

Kill all humans??