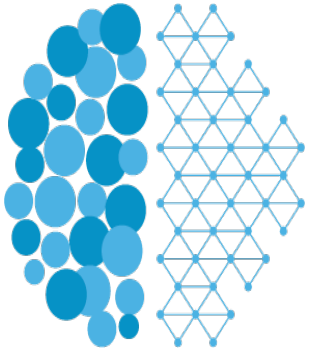
A Pot-Pourri of Recent Advances at MILA

Yoshua Bengio

July 6, 2018

Facebook AI Research, New York



MILA

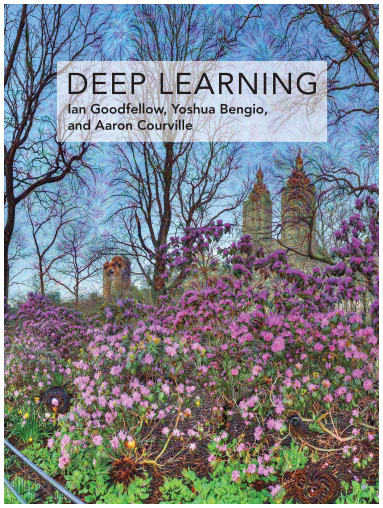
CIFAR

CANADIAN INSTITUTE FOR ADVANCED RESEARCH

Université de Montréal

IVADO

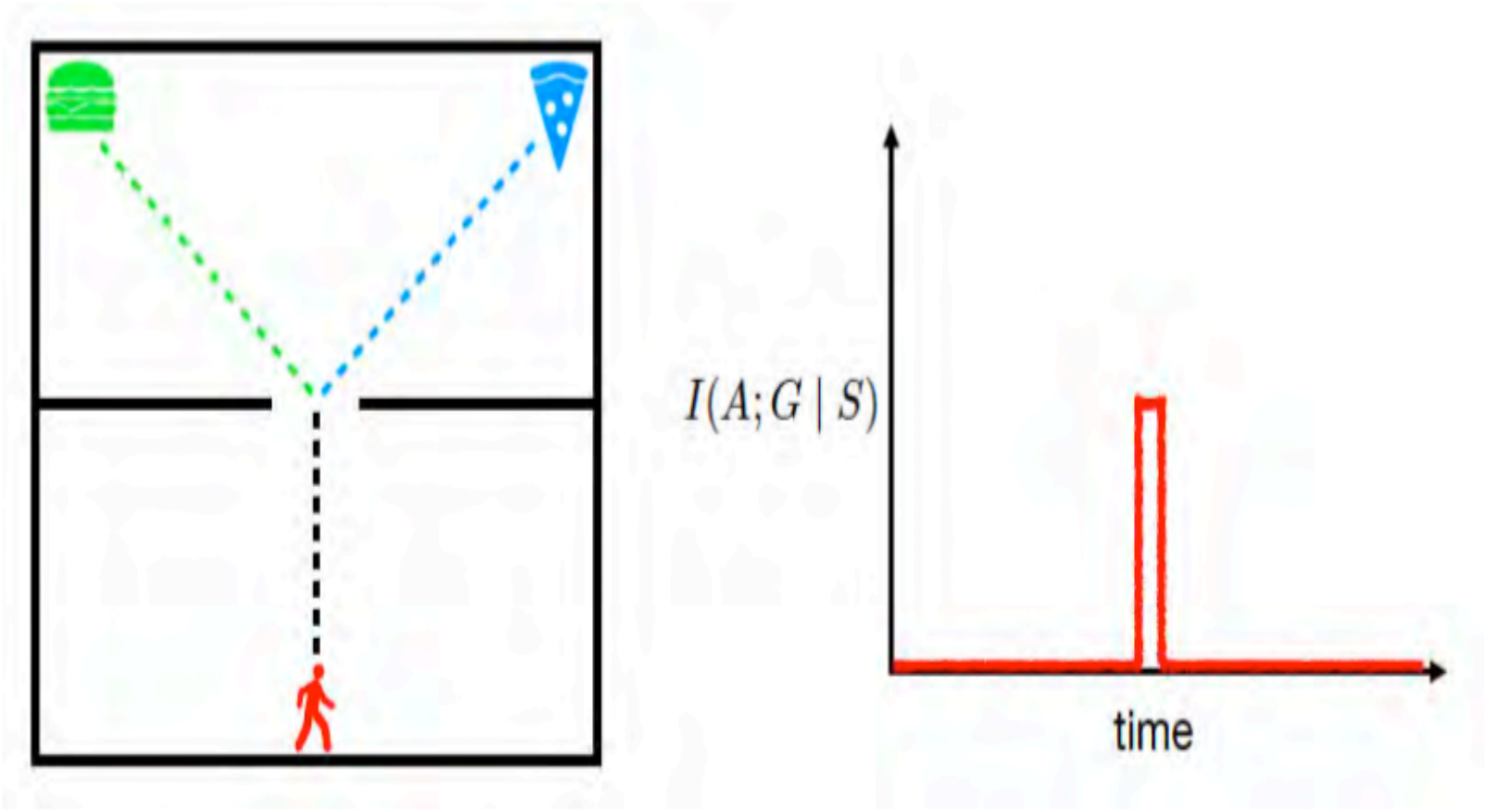
PLUG: **Deep Learning**, MIT Press, chapters for free online



InfoBot: Identifying Decision States Using Information Bottleneck

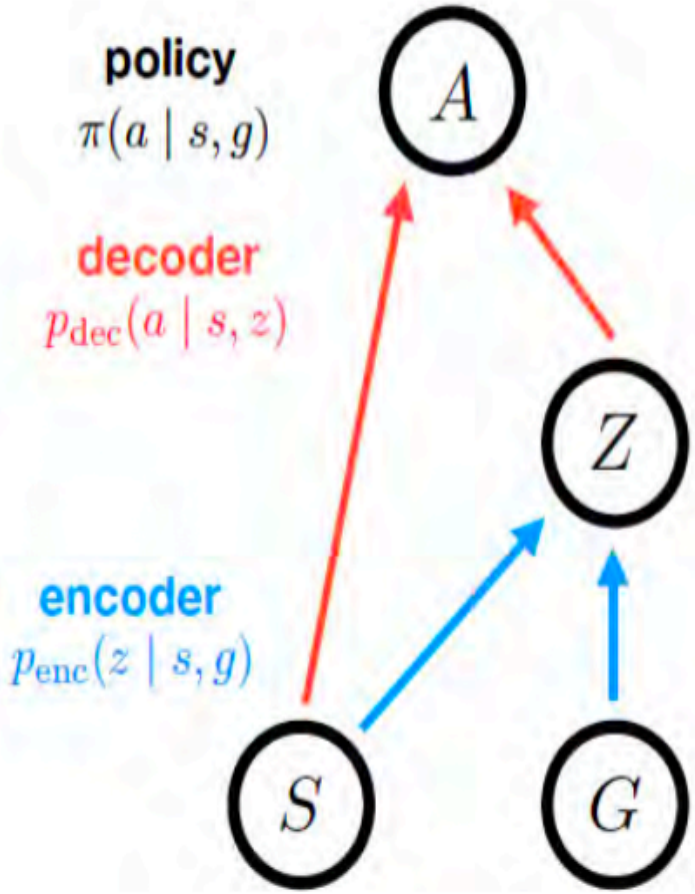
Anirudh Goyal, Riashat Islam, Zafarali Ahmed, Doina Precup, Matthew Botvinick, Hugo Larochelle, Sergey Levine and Yoshua Bengio

Relevant Goal Information



Information identifies useful subgoals
Grounding Subgoals in Information Transitions, 2011

InfoBot: Identifying Decision State



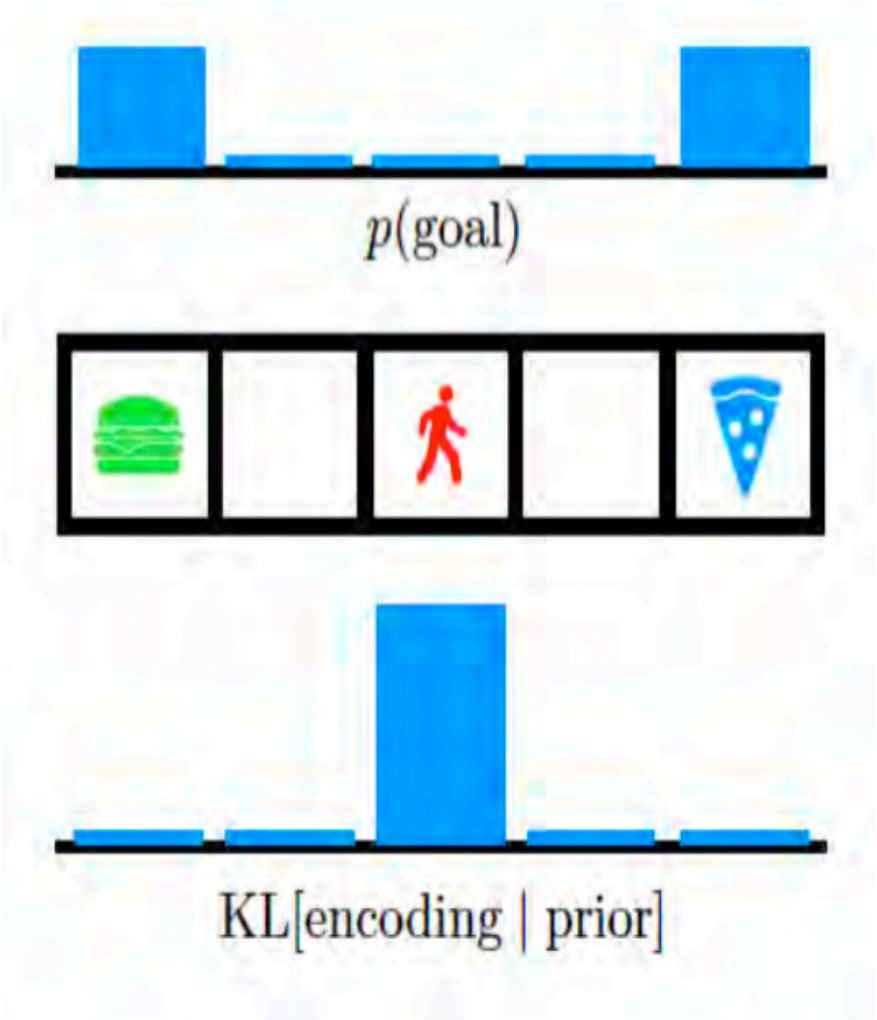
Variational information minimization b/w action and goal given state

$$I(A;G \mid S) \leq I(Z;G \mid S)$$
$$\leq \sum_g p(g) \sum_s p(s \mid g)\, \mathrm{KL}[p_{\text{enc}}(z \mid s, g) \mid r(z)]$$

sample a goal — *sample trajectory* — *penalize encoder for departures from prior*

Agent pays the price for querying the goal.

“Querying” the goal state



Using KL for Structured Exploration

- High KL == “Interesting State”
- Train Primitives to go to “High KL” states to query the goal.
- Incentivize agents to go to “High KL states” == Use KL as intrinsic motivation

Decision States v/s Bottleneck States

Distinction b/w

- **bottleneck states and**
- **decision states**

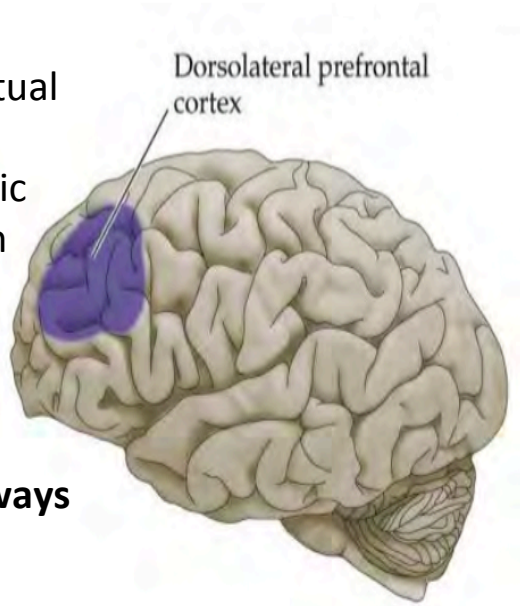
- **Decision states** ⇒ linked to available information to an agent
- **Bottleneck states** ⇒ Based on MDP connectivity structure.

Concept of decision states

- Decision states are not binary, it's more of a continuum.
- Some states are less decision-ey, and other states are more decision-ey.

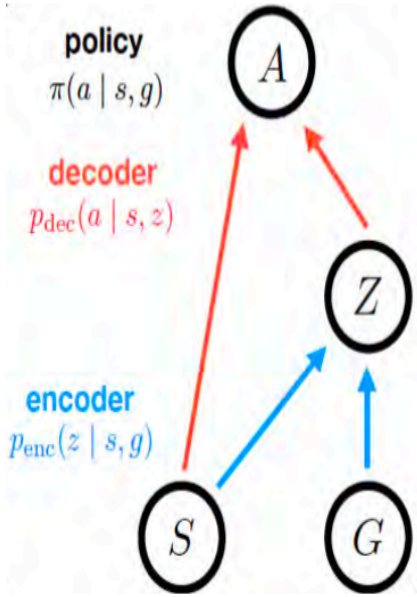
Fundamental distinction between automatic and controlled action selection.

- **Automatic Responding** - Perceptual inputs directly trigger actions.
- **Controlled Behaviour -** Automatic responding is overridden to align behaviour with the goal.



Infobot architecture contains 2 pathways

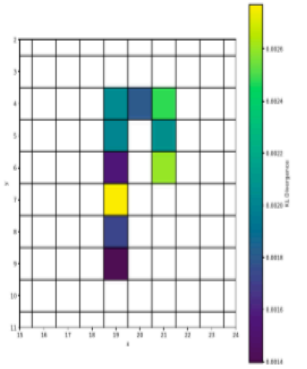
- Goal to Action ("controlled")
- State to Action ("Automatic responding")



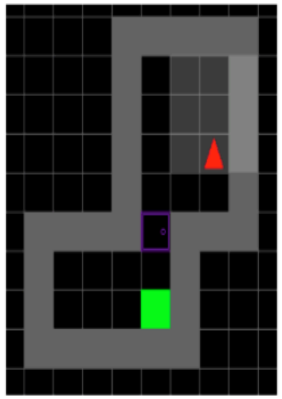
(Left) Diagram credit - Matthew Botvinick

Identifying Decision Points

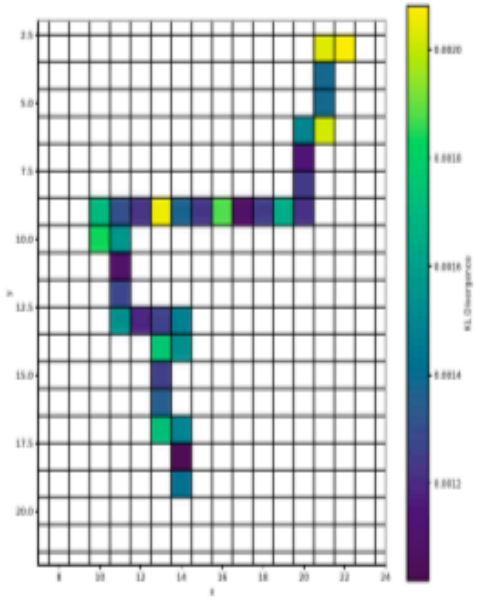
- Agent gets a partial view (POMDP)
- Policy trained on smaller maze (left), generalizes well to bigger mazes (right)



(a) KL Maps on the Maze



(b) Simple Grid world with two rooms.

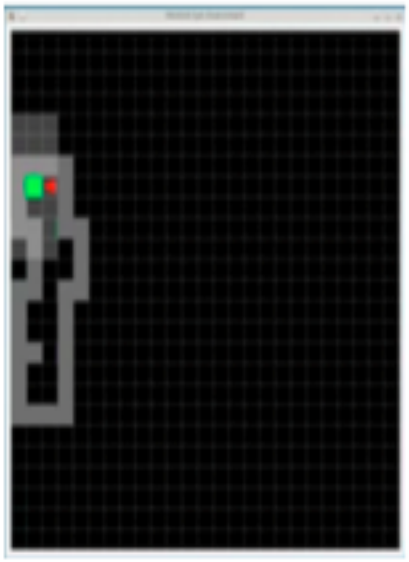


(a) KL Maps on a Large Maze

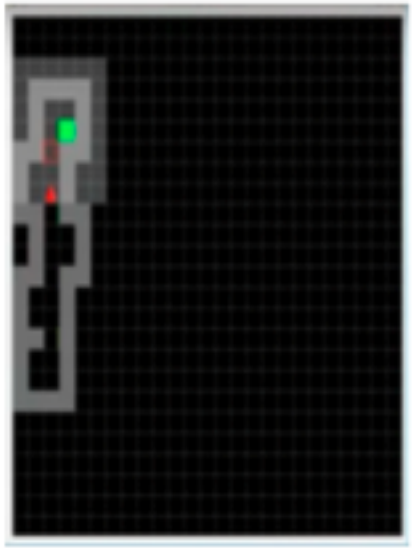
(b) Grid world with 10 rooms

Better generalization

- Wall following strategy. NXSY - Grid with X no. of rooms, of atmost size Y.



(a) MultiRoom-N4-S4

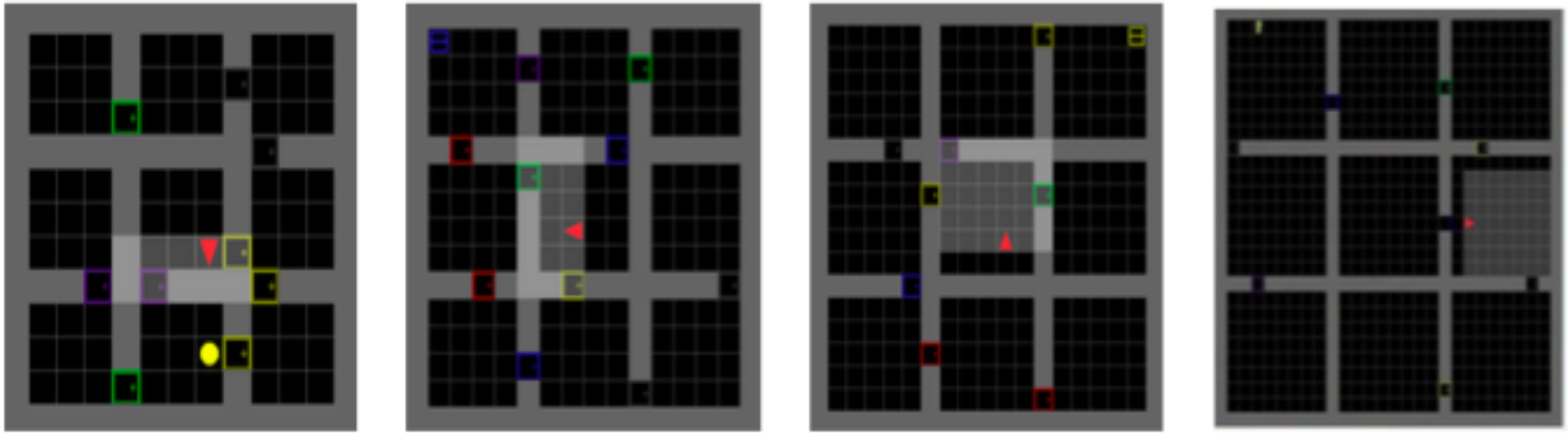


(b) MultiRoom-N5-S4

(c) MultiRoom-N6-S8

(d) MultiRoom-N12-S10

Better Generalization



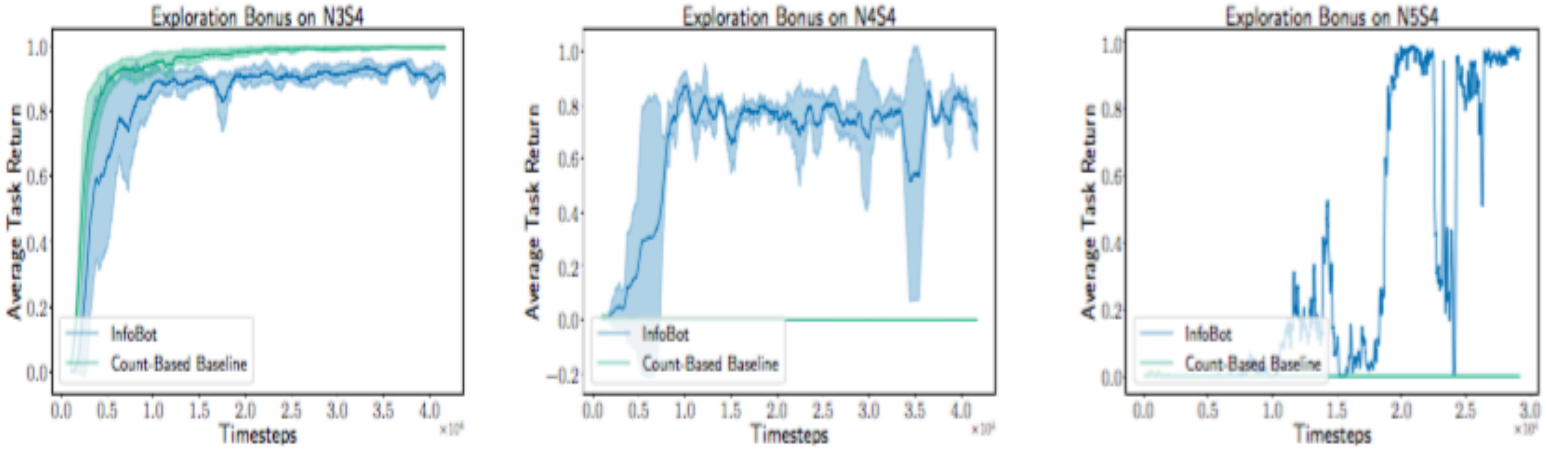
(a) FindObjS5 (b) FindObjS6 (c) FindObjS7 (d) FindObjS10

Train Task	Minigrid-FindObjS7-v0	Minigrid-FindObjS10-v0
Minigrid-FindObjS5-v0 (a2c baseline)	56%	36%
Minigrid-FindObjS7-v0 (a2c baseline)	62%	40%
Minigrid-FindObjS5-v0 (Infobot + No KL cost)	44%	24%
Minigrid-FindObjS5-v0 (Infobot)	78%	61%

Table 1: Generalization of the agent to larger grids in Minigrid-FindObjSY envs.

Structured Exploration - Use KL as exploration Bonus

NXSY - Grid with X number of rooms, of atmost size Y (procedurally generated)



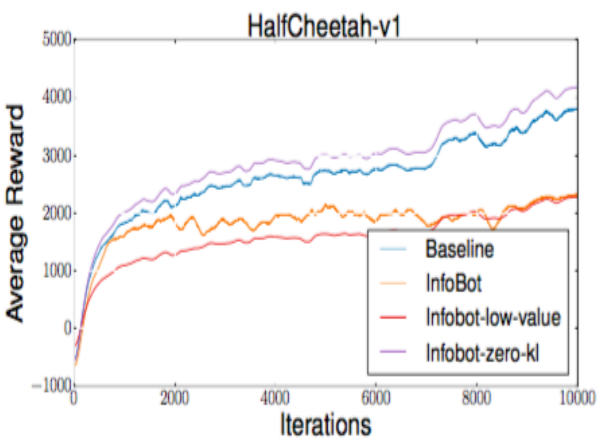
Method	MiniGrid-MultiRoom-N3-S4	MiniGrid-MultiRoom-N5-S4
A2c baseline	0%	0%
TRPO + VIME	54%	0%
Count based exploration	**95%**	0%
A2c + KL exploration	90%	**85%**

Table 2: Comparison of InfoBot's exploration strategy with a count-based exploration method as well as VIME.

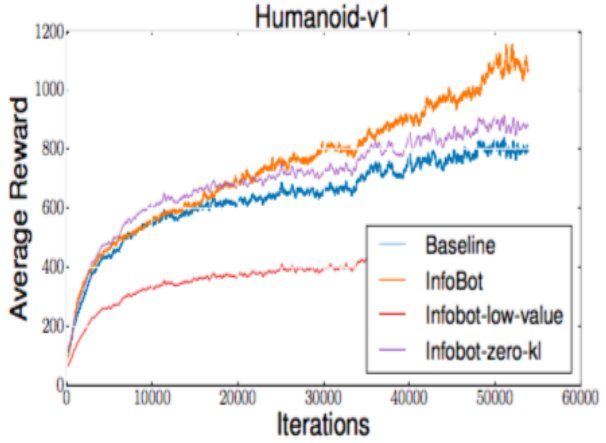
Better Exploratory Policy - Continuous Control Tasks

- **Use high value states as goals (Recall Traces: Backtracking Model)**

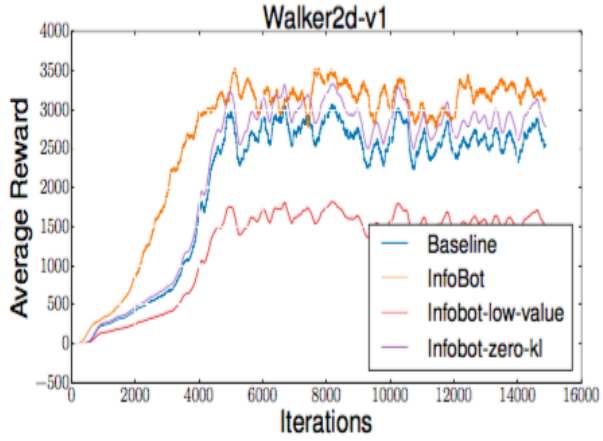
- **Baseline -** Proximal policy optimization (PPO)
- **Infobot-low-value** = Use low value states as goal.
- **Infobot-zero-KL -** coeff. Of KL loss is 0.



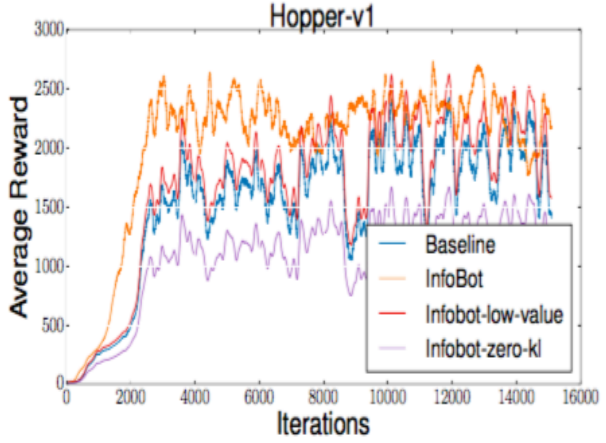
(a) Half-Cheetah



(b) Humanoid



(c) Walker



(d) Hopper

Disentangling optimization and generalization

- The traditional ML picture is that optimization and generalization are neatly separated aspects
- That makes theory easier to handle, separately
- Unfortunately not the case
- SGD variants influence optimization AND generalization

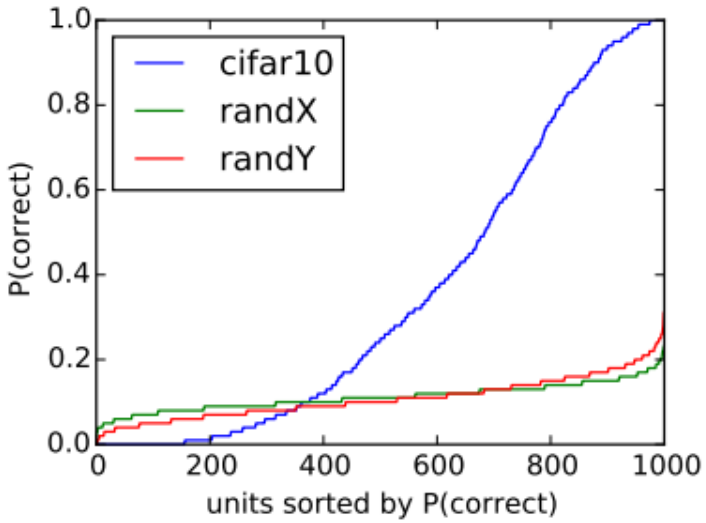
Memorization in Deep Networks

Mostly from preprint arXiv:1706.05394
Devansh Arpit, Stanisław Jastrzębski, Nicolas Ballas, David Krueger, Emmanuel Bengio, Maxinder S Kanwal, Tegan Maharaj, Asja Fischer, Aaron Courville, Yoshua Bengio, Simon Lacoste-Julien

Memorization in Deep Networks

- Deep networks trained with SGD generalize well due to its implicit regularization effect (Zhang et al 2016)
- Deep networks achieve ~100% train accuracy on random data (Zhang et al 2016)
- Do deep networks also memorize real data?

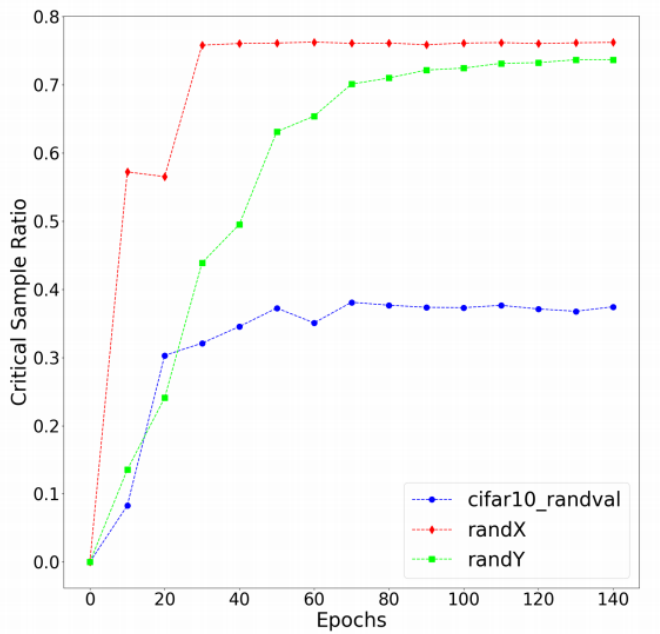
Real data has Dominant Patterns



Fraction of times each of 1000 samples is classified correctly after 1 epoch across 100 runs

- Real data: some samples are learned first.
- Random data: samples are learned in arbitrary order.

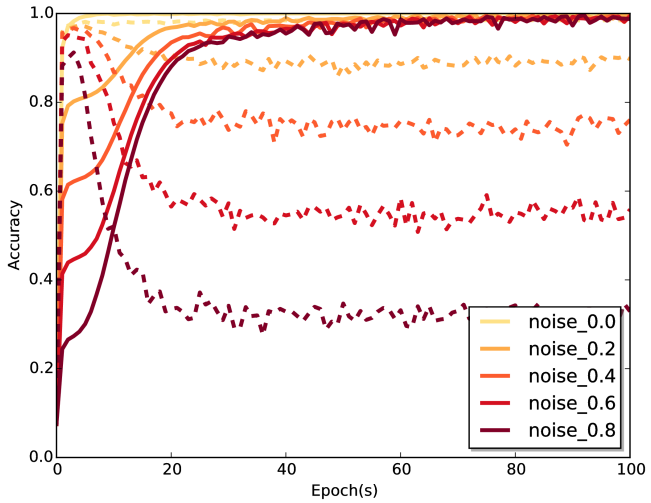
Larger Margin on Real data



Critical sample ratio = fraction of samples which have adversarial examples in their vicinity

- Real data: distance from decision boundary is large
- Random data: distance from decision boundary is small

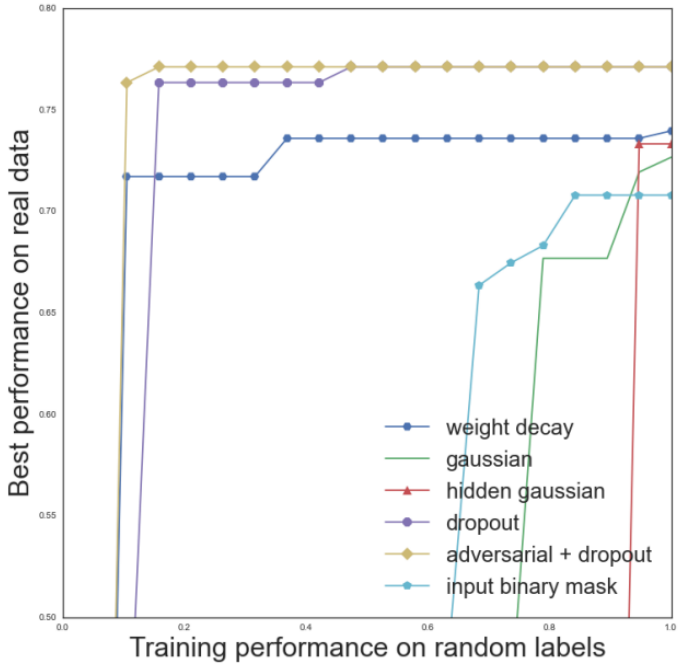
Patterns come First



Train (full) and validation (dotted) accuracy on MNIST during training with noisy labels

- Validation accuracy peaks before falling
- Patterns in real data learned before overfitting noise

Regularization Hinders Memorization



Best validation performance (picked across hyper parameter grid) on real data vs. training performance on noise labels for the same model, for different regularizers.

- Dropout is best at hindering memorization
- Maintains performance on real data for reduced memorization on random data.

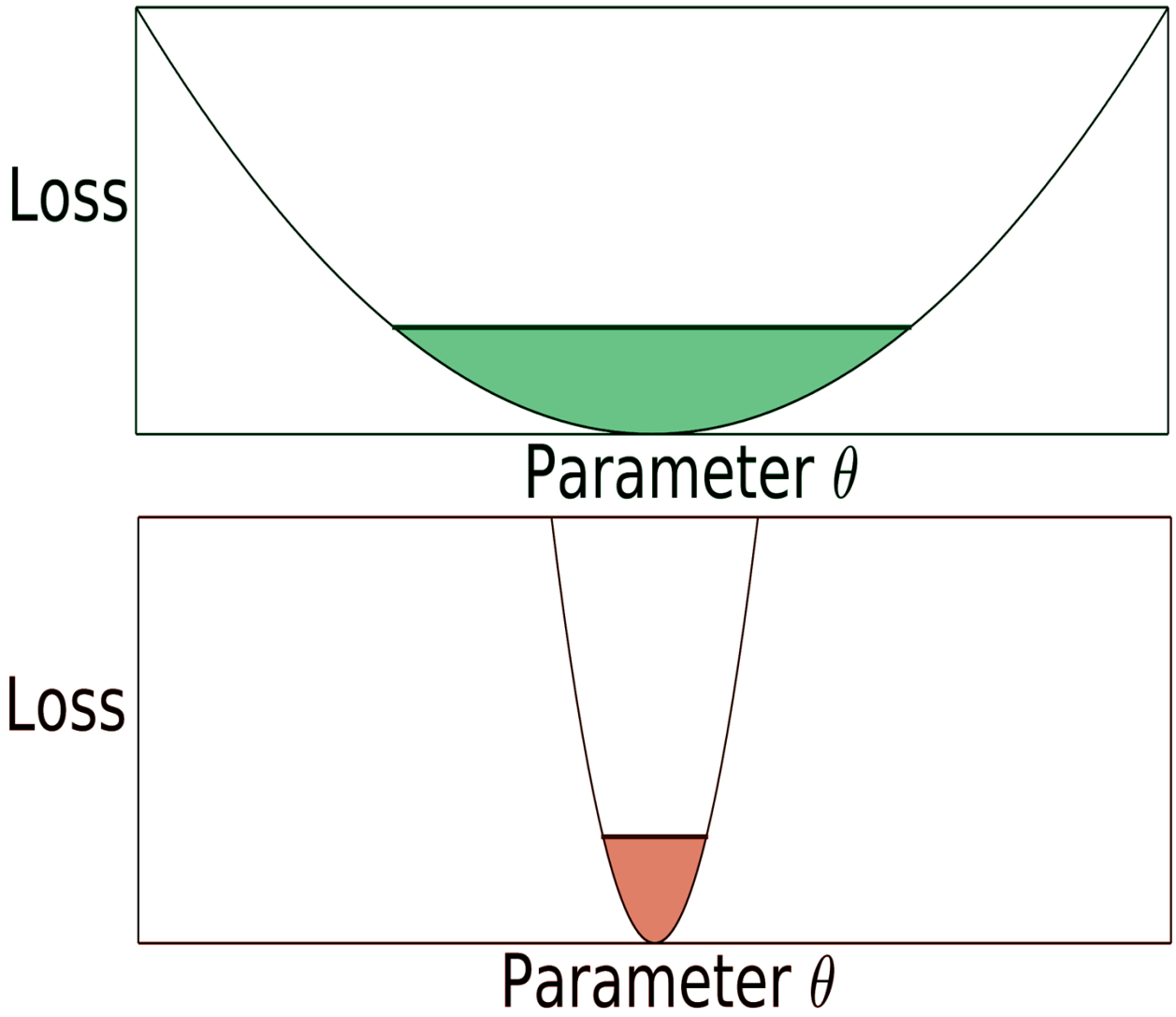
Take Home Message

- **DNNs learn patterns before memorizing noise**
- **Regularization hinders memorization**

On the relevance of loss function geometry for generalization

Laurent Dinh, Razvan Pascanu,
Samy Bengio, Yoshua Bengio

Flatness



Reparametrization

$$\eta = g^{-1}(\theta) \qquad L_\eta(\eta) = L\left(g(\eta)\right)$$

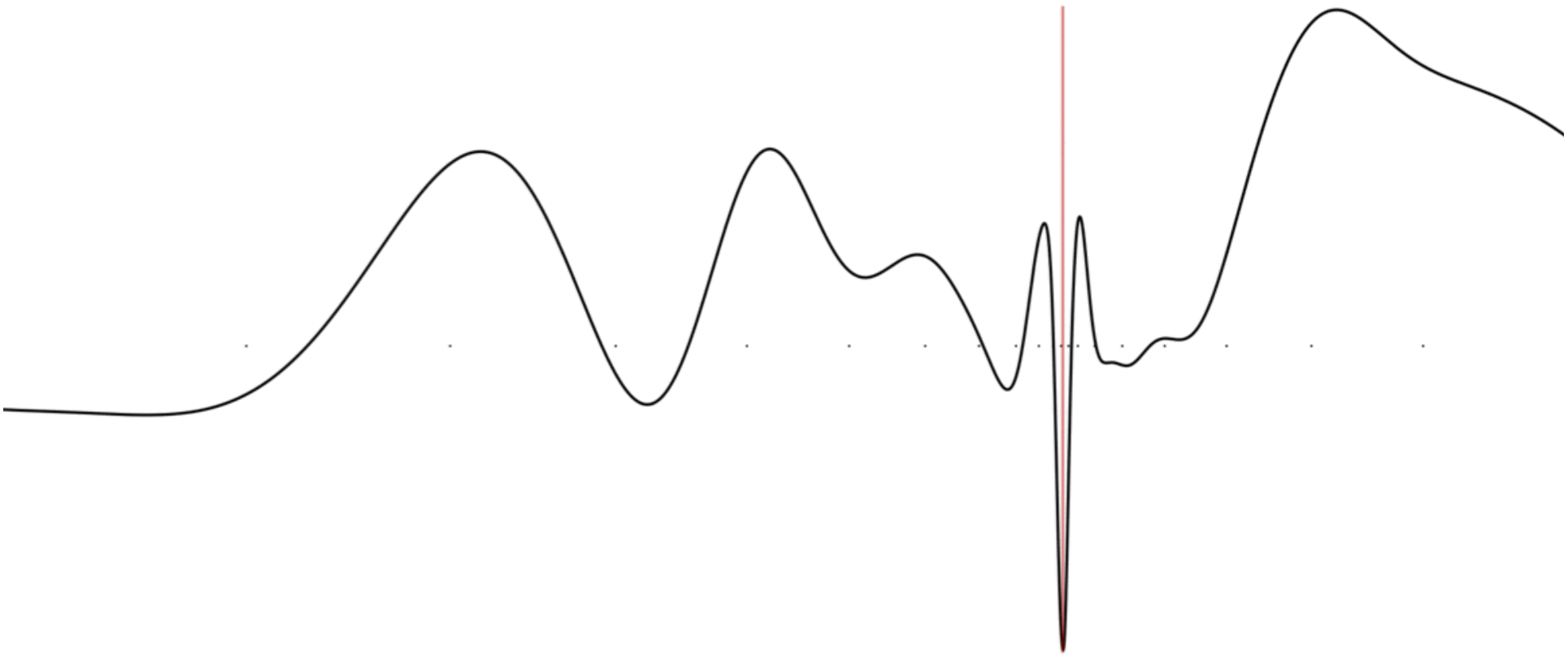
-
- Differentiation at critical point

$$(\nabla^2 L_\eta)(\eta) = (\nabla g)(\eta)^T (\nabla^2 L)\left(g(\eta)\right)(\nabla g)(\eta)$$

- Flat minima $\xrightarrow{g}$ Sharp minima

  Sharp minima $\xrightarrow{g}$ Flat minima

Reparametrization



Sharp minima can generalize
Flat minima can poorly generalize

Eppur, si muove!

And yet, it moves

Factors influencing Minima in SGD

Mostly from preprint arXiv:1711.04623

Stanisław Jastrzębski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, Amos Storkey

Behavior of SGD

- Small mini-batch finds wider minima (Keskar et al 2016)
- What dynamics/factors govern the quality of minima found by SGD?

SGD as Stochastic Differential Equation

- Mini-batch gradient $\mathbf{g}^{(S)}(\boldsymbol{\theta})$ (due to CLT), batch size S:

- SGD with learning rate η is described by:

$$\mathbf{g}^{(S)}(\boldsymbol{\theta}) = \underbrace{\mathbf{g}(\boldsymbol{\theta})}_{\frac{\partial L(\theta)}{\partial \theta}} + \frac{1}{\sqrt{S}}\Delta\mathbf{g}(\boldsymbol{\theta}), \text{ where } \Delta\mathbf{g}(\boldsymbol{\theta}) \sim N(0, \mathbf{C}(\boldsymbol{\theta}))$$

- Continuous stochastic differential equation (SDE) form: (Li et al 2017)

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \boldsymbol{g}^{(S)}(\boldsymbol{\theta})$$

$$\frac{d\boldsymbol{\theta}}{dt} = -\eta \boldsymbol{g}(\boldsymbol{\theta}) + \frac{\eta}{\sqrt{S}}\mathbf{B}(\boldsymbol{\theta})\mathbf{f}(t)$$

If small enough learning rate, ie. small steps

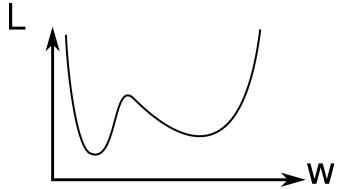
Note: $\mathbf{C}(\boldsymbol{\theta}) = \mathbf{B}(\boldsymbol{\theta})^{\mathsf{T}}\mathbf{B}(\boldsymbol{\theta})$

Equilibrium Distribution of SGD

- The equilibrium distribution of this SDE is given by:
- ~Inverse relation between loss and density

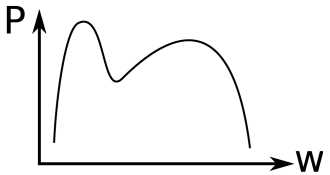
$$P(\boldsymbol{\theta}) = P_0 \exp\left(-\frac{2L(\boldsymbol{\theta})}{n\sigma^2}\right)$$

- Noise n controls the granularity of the equilibrium distribution



$n = \eta/S$

n=10



n=1

Note: η = learning rate, S = batch size, σ^2 = fixed isotropic gradient variance

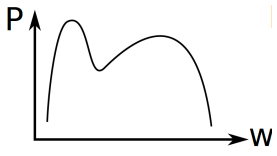
Implications of the Theory

- Probability of ending in a minima A described by Hessian $\mathbf{H}_A$:

$$p_A \propto \frac{1}{\sqrt{\det \mathbf{H}_A}} \exp\left(-\frac{2}{n\sigma^2} L_A\right)$$

- In general, minima with larger volume is favored more (simply because it has higher probability mass)
- Higher noise n prioritizes width (volume) over depth
- Final equilibrium distribution is unchanged when learning rate and batch size are scaled proportionally $\eta \to \beta\eta$, $S \to \beta S$

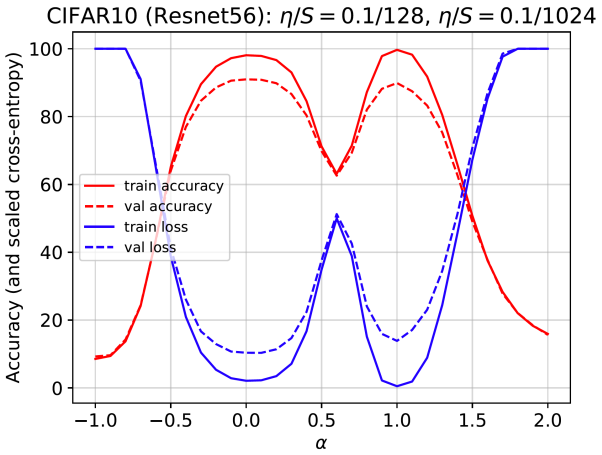
$$P(\boldsymbol{\theta}) = P_0 \exp\left(-\frac{2L(\boldsymbol{\theta})}{n\sigma^2}\right)$$



Note: n = η/S, η = learning rate, S = batch size, σ^2 = fixed isotropic gradient variance

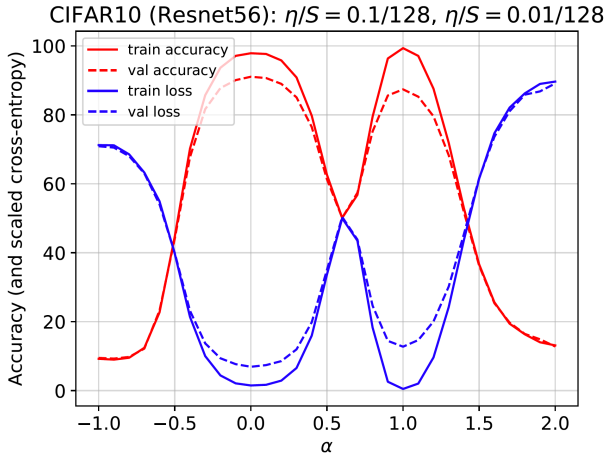
Smaller Noise – Sharper Bowl

- Interpolation between apparent minima found by SGD at large/small noise level



α = 0: baseline noise level

α = 1: small noise due to large batch size

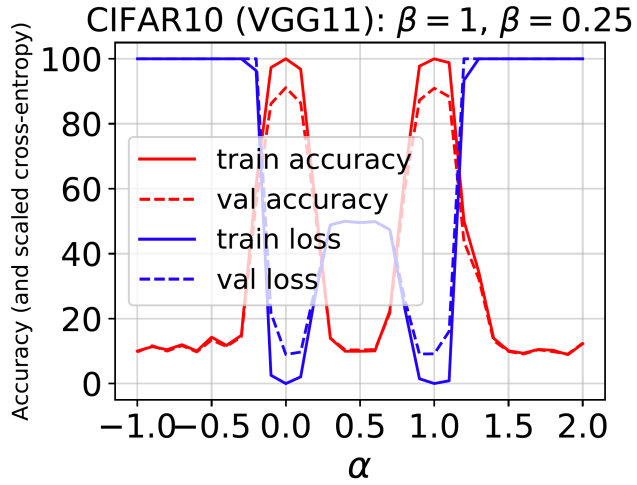


α = 0: baseline noise level

α = 1: small noise due to small learning rate

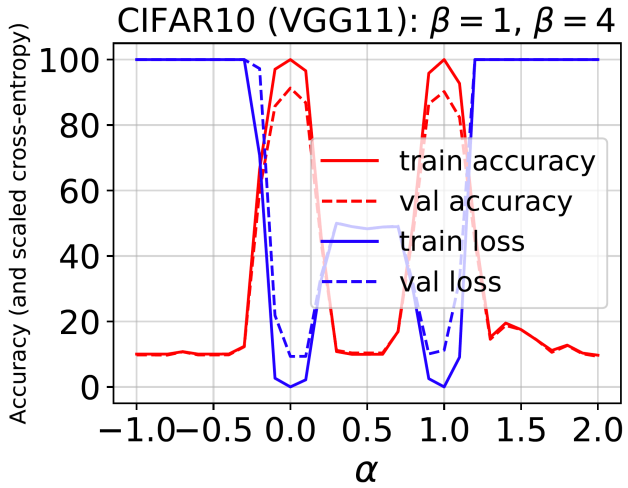
Equal noise – Equal width

- Interpolation between minima found by SGD at the same noise level (due to different learning rate/batch-size configurations)



α = 0: baseline noise level η/S

α = 1: same noise level 0.25η/ 0.25S

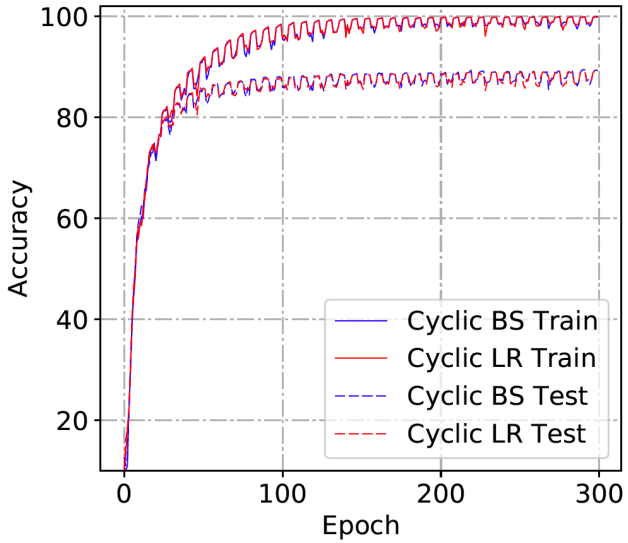


α = 0: baseline noise level η/S

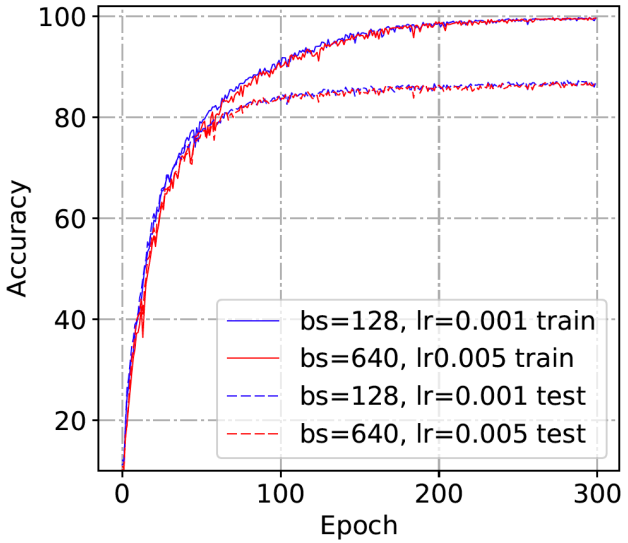
α = 1: same noise level 4η/4S

Same Noise - Same Learning Dynamics

- Theory talks about final equilibrium distribution but seems to apply along trajectory as well
- But even learning dynamics is similar when learning rate and batch size are scaled proportionally $\eta \to \beta\eta, S \to \beta S$



Cyclic Learning Rate and Cyclic Batch-size



Constant Learning Rate and Constant Batch-size

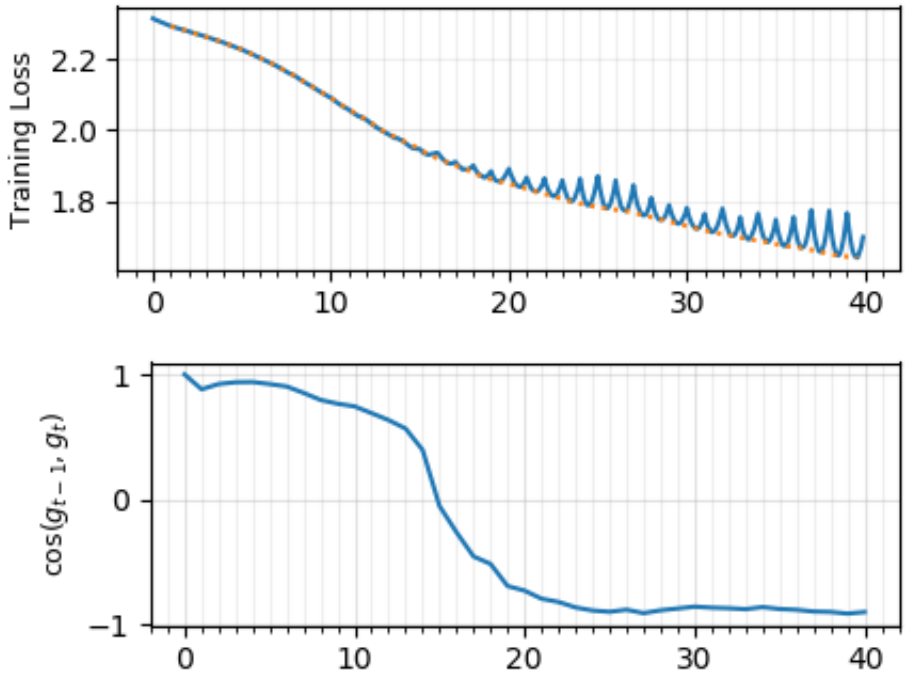
Take Home Messages

- **DNNs learn patterns before memorizing noise**
- **Regularization hinders memorization**
- **The quality of final minima and learning dynamics is similar when learning rate and batch size are scaled proportionally**
- **Larger noise favors large volume minima over deep ones**
- **Larger noise (e.g. due to BS or l.rate) hinders memorization**

A Walk with SGD

Xing, Arpit, Tsirigotis & Bengio ArXiv:1802.08770

- Interpolate in parameter space between minibatch SGD updates and see convex shape
- After initial phase, updates bounce off valley floor, which monotonically improves, traversing larger distances with smaller batch sizes (BS)
- Learning rate: height from floor
- BS: exploration noise
- Pure GD gets stuck on floor, while SGD finds flatter regions, which generalize better



Sharpest Directions Along the SGD Trajectory
(Jastrzębski, Kenton, Ballas, Fischer, Bengio, Storkey)

- Even at the beginning of training, a high learning rate or small batch size influences SGD to visit flatter loss regions.
- the largest eigenvalues appears to always follow a similar pattern, with a fast increase in the early phase and a decrease thereafter, where the peak value is determined by the learning rate and batch size.
- altering the learning rate just in the direction of the eigenvectors associated with the largest eigenvalues, SGD can be steered towards regions which are an order of magnitude sharper but correspond to models with similar generalization, confirming that curvature of the endpoint found by SGD is not predictive of its generalization properties.

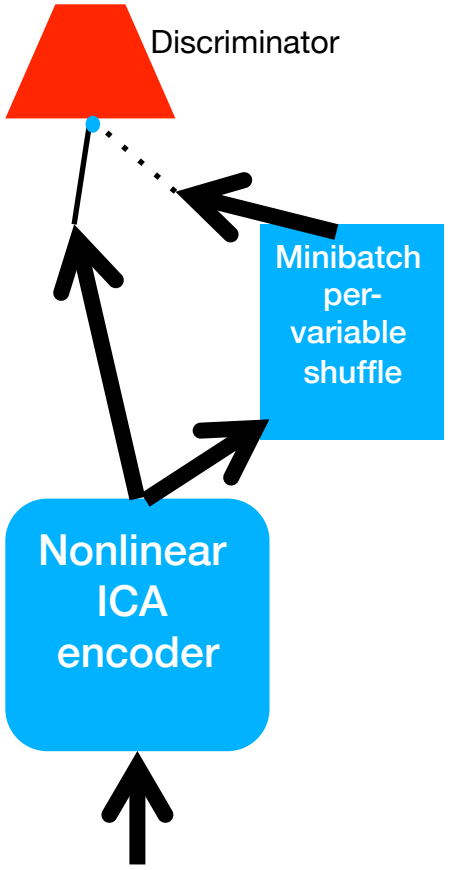
Using a discriminator to optimize **independence**, mutual information or entropy

Brakel & Bengio ArXiv:1710.05050

- Train a discriminator to separate between pairs (A,B) coming from P(A,B) and pairs coming from P(A) P(B)

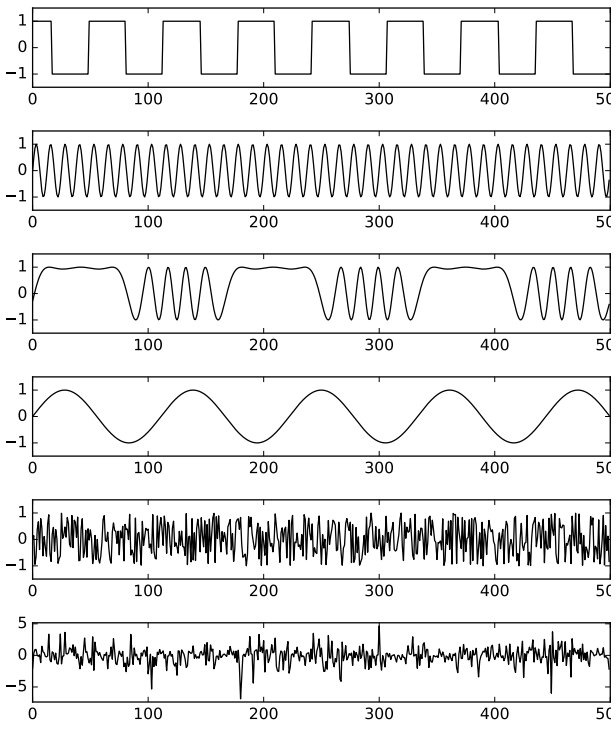
- Generalize this to measuring **independence** of all the outputs of a representation function (encoder). Maximize independence by backpropagating the independence score into the encoder

→ NON-LINEAR ICA.

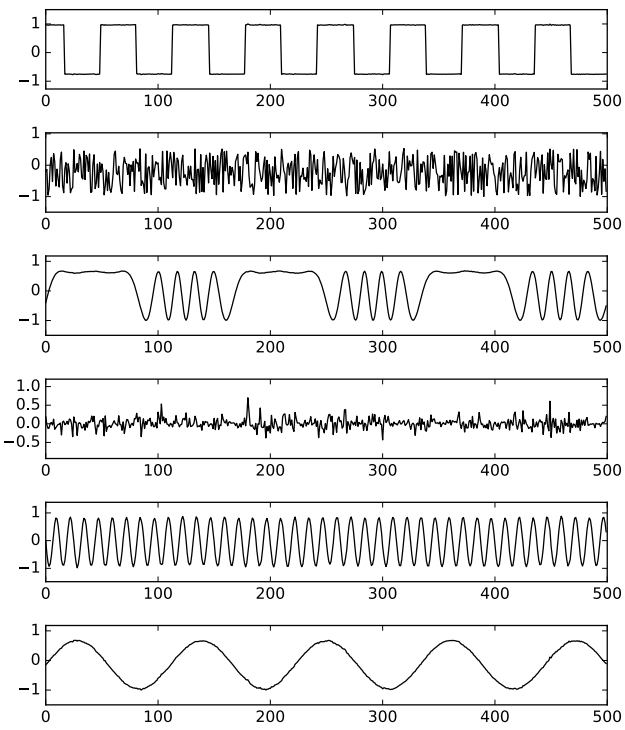


Non-Linear Independent Component Analysis Results

- Sources were either mixed linearly or non-linearly, independent components recovered in both cases

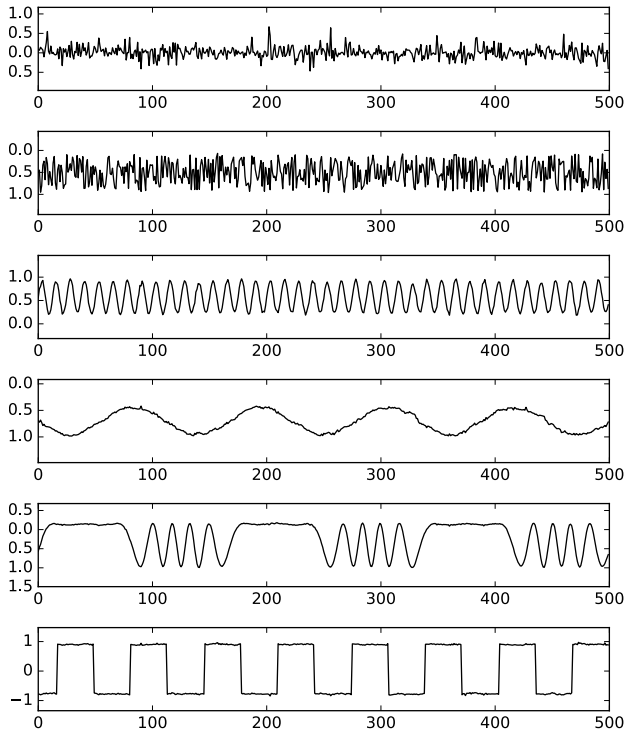


(a) Source signals.



(b) Anica reconstructions $\rho_{\max} = .997$.

Linearly mixed



(a) Anica PNL reconstructions $\rho_{\max} = .997$.

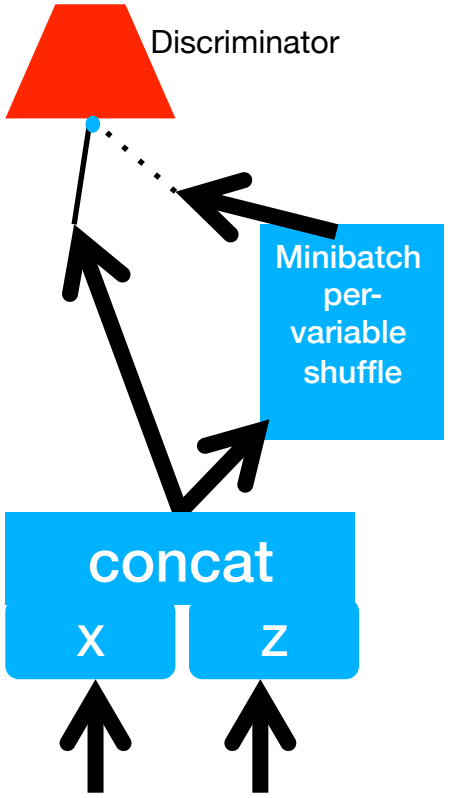
Nonlinearly mixed

Using a discriminator to optimize independence, mutual information or entropy

MINE: Mutual Information Neural Estimator

Belghazi et al ArXiv:1801.04062

Same architecture, but with a twist in the training objective which provides an asymptotically consistent estimator of mutual independence



Mutual information, KL divergence and Donsker-Varadhan Representation

[Belghazi et. al., 2018]

Mutual information: measure of dependence btwn 2 variables

$$I(X;Z) = \mathcal{D}_{KL}(\mathbb{P}_{X,Z} || \mathbb{P}_X \otimes \mathbb{P}_Z) = \mathbb{E}_{\mathbb{P}_{X,Z}} \left[\log \left(\frac{p(x,z)}{p(x)p(z)} \right) \right]$$

$$I(X;Z) = H(X) + H(Z) - H(X,Z) = D_{KL}(\mathbb{P}_{XZ} \,||\, \mathbb{P}_X \otimes \mathbb{P}_Z)$$

(Donsker & Varadhan, 1983):

$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) = \sup_{T:\Omega \to \mathbb{R}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T])$$

Optimal T:

$$T^* = \log \frac{d\mathbb{P}}{d\mathbb{Q}} + C$$

With suboptimal T:

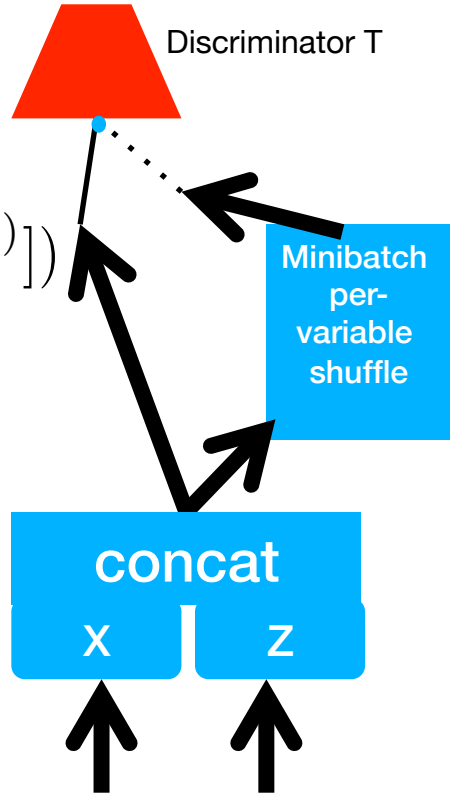
$$D_{KL}(\mathbb{P} \,||\, \mathbb{Q}) \geq \sup_{T \in \mathcal{F}} \mathbb{E}_{\mathbb{P}}[T] - \log(\mathbb{E}_{\mathbb{Q}}[e^T])$$

MINE: Estimator of MI

Given two r.v. X & Z and samples of their joint & marginals:

$$\widehat{I(X;Z)}_n = \mathbb{E}_{\hat{\mathbb{P}}^{(n)}_{XZ}}[T_{\hat{\theta}_n}(x,z)] - \log(\mathbb{E}_{\hat{\mathbb{P}}^{(n)}_{X} \otimes \hat{\mathbb{P}}^{(n)}_{Z}}[e^{T_{\hat{\theta}_n}(x,z)}])$$

where discriminator T is optimized to maximize the rhs

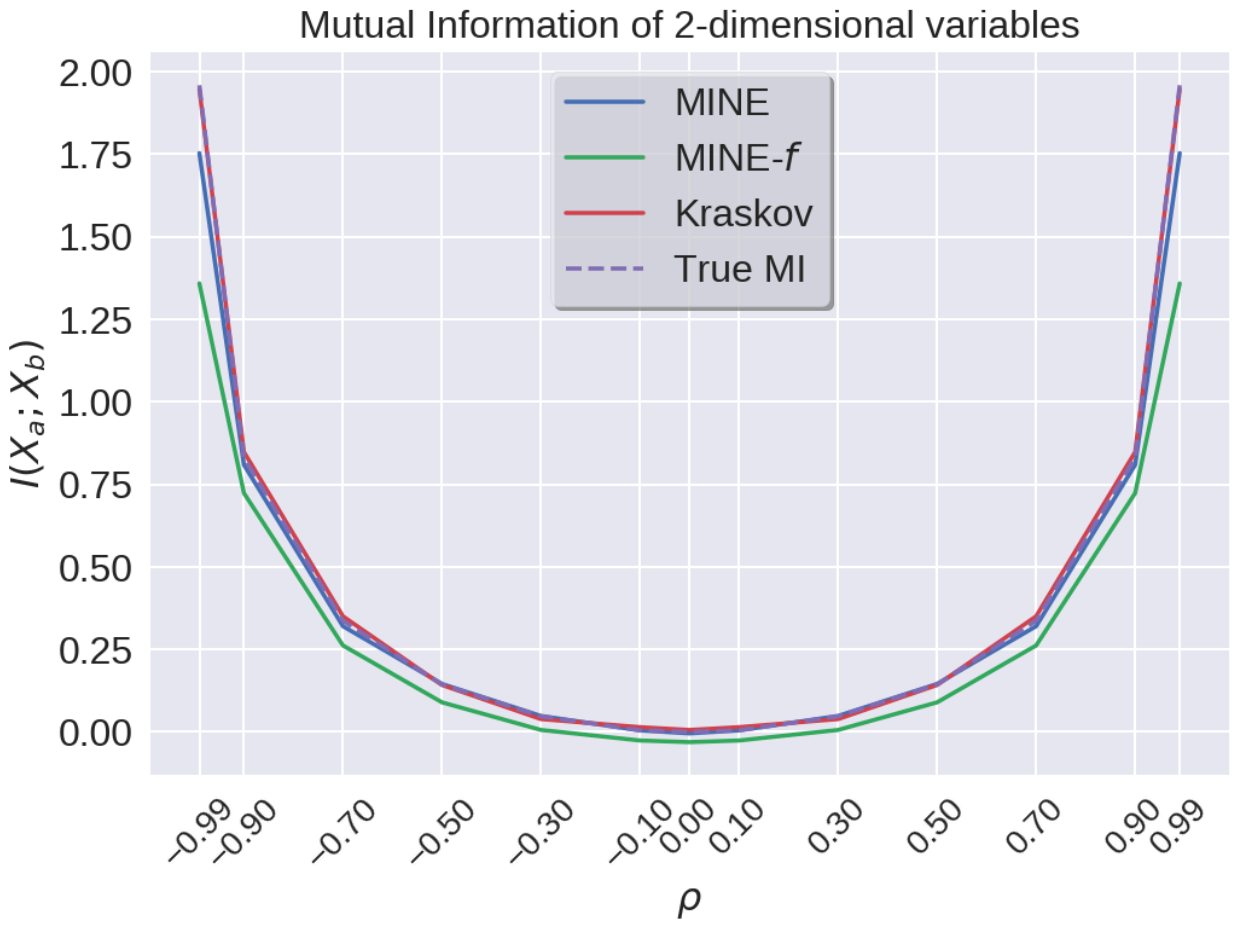


MINE: Consistency

Theorem: there exists a neural net architecture such that for all $\epsilon > 0$ there exists an integer *N* s.t.

$$\forall n \geq N, \quad |I(X,Z) - \widehat{I(X;Z)}_n| \leq \epsilon \text{ with probability one}$$

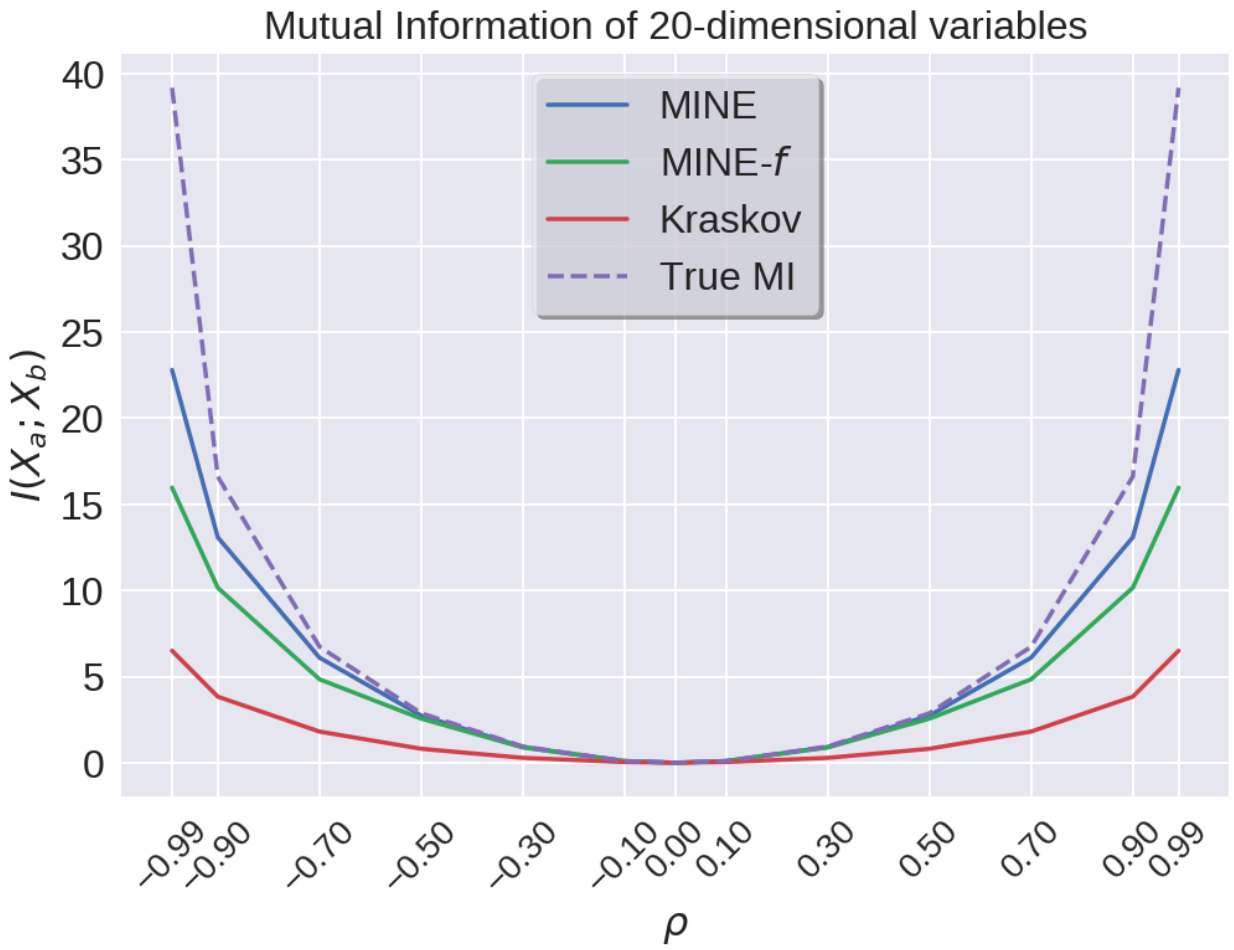
Demonstration of estimation



MI between 2 Gaussians

[Belghazi et. al., 2018]

Demonstration of estimation



MI between 2 Gaussians

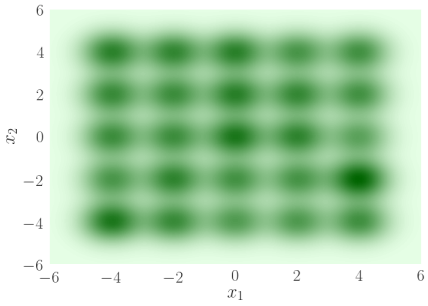
[Belghazi et. al., 2018]

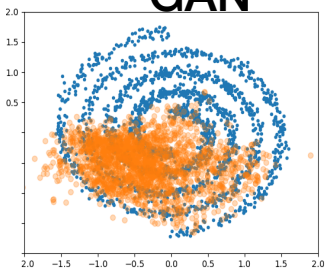
Maximizing ENTROPY: avoid GAN mode dropping by max MI(X,Z)

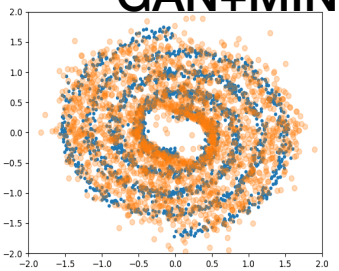
GAN

GAN+MINE

Ground Truth





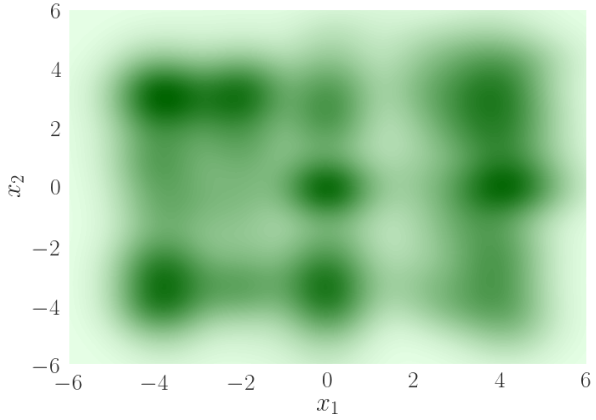


since *H(X|Z)=0*

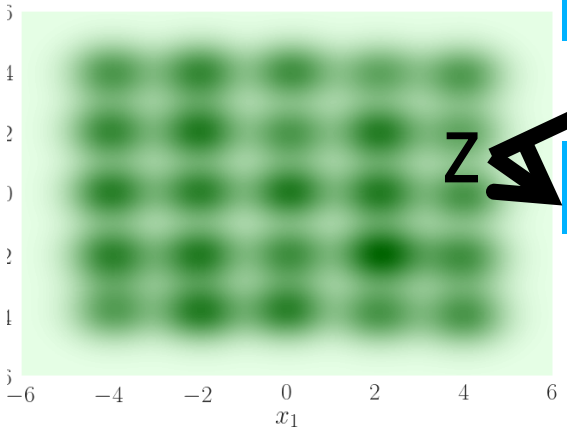
I(X;Z)=H(X)

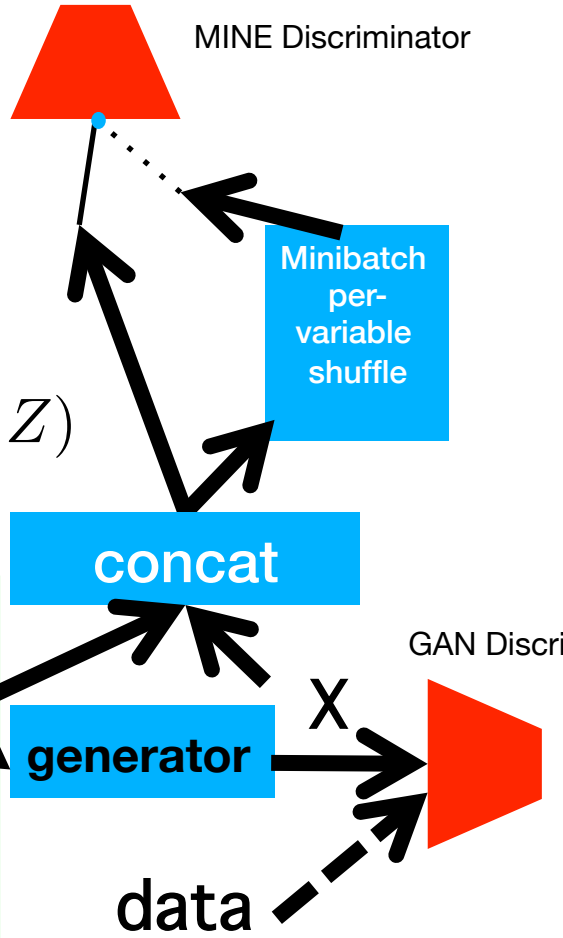
$$I(X;Z) := H(X) - H(X \mid Z)$$

GAN



GAN+MINE





Manifold Mixup

Alex Lamb*, Vikas Verma*, Chris Beckham, Aaron Courville, Ioannis Mitliagkas, Yoshua Bengio

How to improve generalization

Conjecture: *some* novel points which are off of the manifold contain combinations of attributes which we've seen during training.

-Examples:

-A deer with wheels instead of legs

-Tokyo city but with the eiffel tower in the skyline.

-A blue pig.

Manifold Mixup

-Augment the training set by randomly combining attributes.

-How to combine attributes?

  -Take a convex combination of the representations.

  -I.e. King - Queen + Child == Boy

-What attributes? **In the latent space of a deep net!**

  -Earlier layers cover more of the network.

  -Later layers make it easier to span the space.

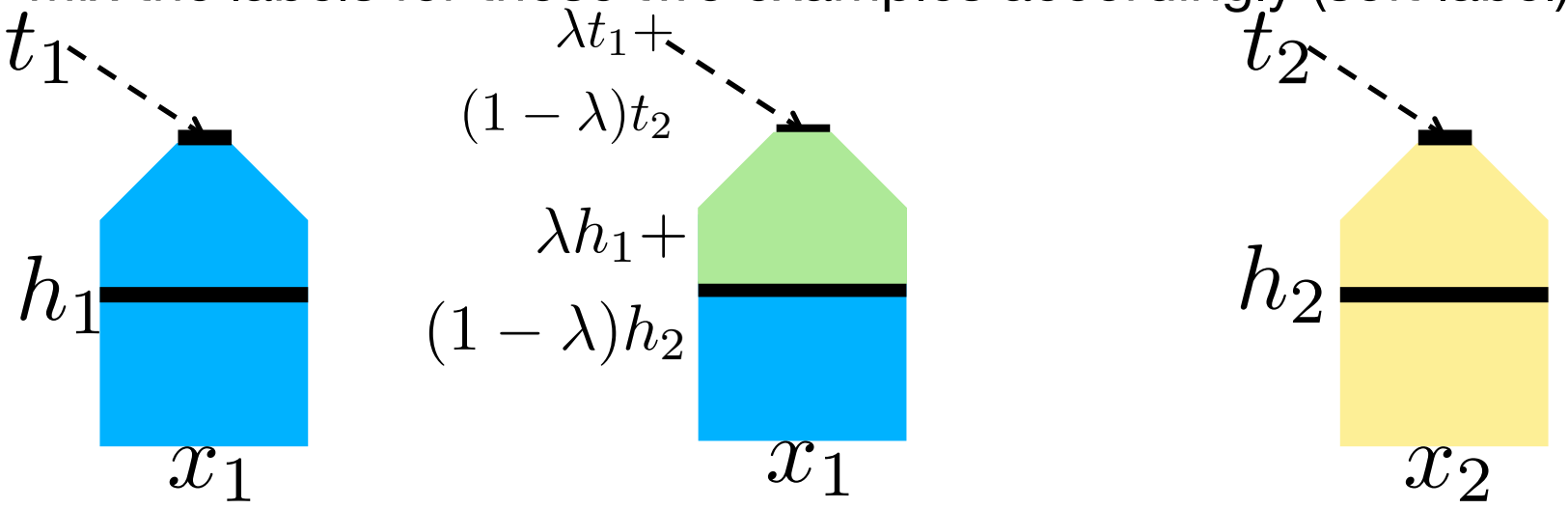
Details of Manifold Mixup

-On each update, pick a random layer uniformly (including the input).

-Sample $\lambda \sim \mathrm{Beta}(\alpha, \alpha)$

-Mix between two random examples from the minibatch at that layer with coeff λ.

-Mix the labels for those two examples accordingly (soft label).



Results - Classification

Model	Test Acc	Test NLL
PreActResNet18		
No Mixup	94.88	0.2646
Input Mixup ($\alpha = 1.0$) (Zhang et al., 2017)	96.10	n/a
Input Mixup ($\alpha = 1.0$) (ours)	96.498	0.1945
Manifold Mixup ($\alpha = 2.0$)	**97.104**	**0.1407**
PreActResNet152		
No Mixup	95.797	0.1994
Input Mixup ($\alpha = 1.0$)	96.844	0.2312
Manifold Mixup ($\alpha = 2.0$)	97.238	0.1419
Manifold Mixup all layers ($\alpha = 6.0$)	**97.622**	**0.0957**

CIFAR - 10

Model	Test Acc	Test NLL
PreActResNet18		
No Mixup (Zhang et al., 2017)	74.4	n/a
No Mixup (ours)	75.32	1.284
Input Mixup ($\alpha = 1.0$) (Zhang et al., 2017)	78.9	n/a
Manifold Mixup ($\alpha = 2.0$)	**78.95**	**0.913**
PreActResNet34		
Input Mixup ($\alpha = 1.0$)	77.208	1.085
Manifold Mixup ($\alpha = 2.0$)	**79.609**	**0.930**

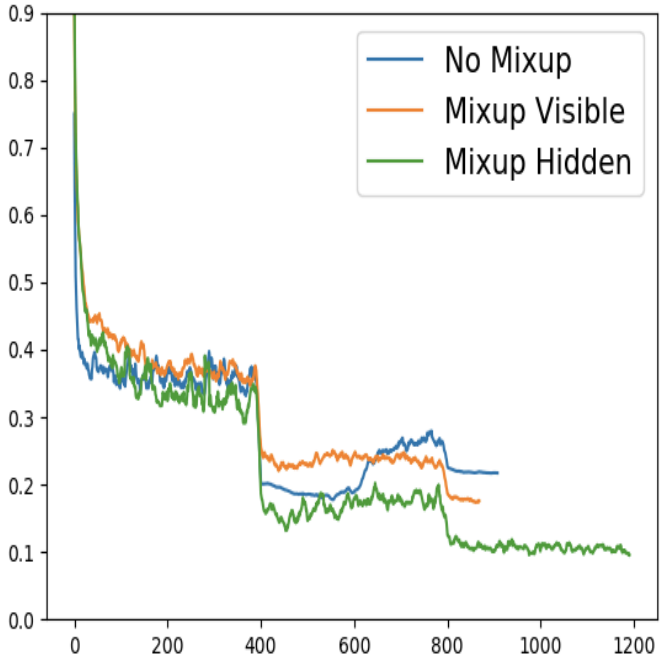
CIFAR - 100

-ShakeShake is 97.14% on CIFAR-10.
-Best or close-to-best results on CIFAR-10 that don't use complicated and expensive hyperparameter search procedure (like AutoAugment or Neural Architecture Search)

Results - Likelihood

-Likelihood is WAY better with manifold mixup. Means it's less confident when it makes wrong predictions.

Negative Log-Likelihood (lower is better)



Epoch

Results - Novel Deformations

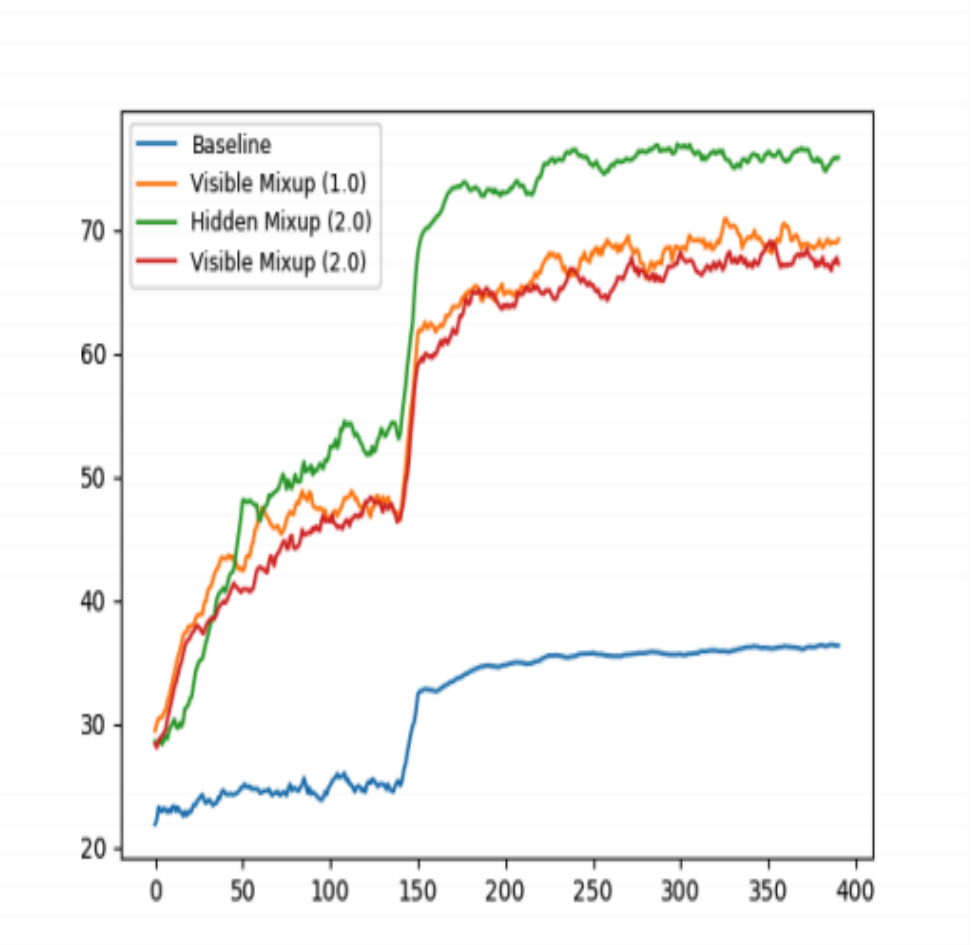
Test Set Deformation	No Mixup Baseline	Input Mixup α=1.0	Input Mixup α=2.0	*Manifold Mixup* α=2.0
Rotation U(−20°,20°)	52.96	55.55	56.48	**60.08**
Rotation U(−60°,60°)	26.77	28.47	27.53	**33.78**
Shearing U(−28.6°, 28.6°)	55.92	58.16	60.01	**62.85**
Shearing U(−57.3°, 57.3°)	35.66	39.34	39.7	**44.27**
Zoom In (80% rescale)	47.95	52.18	50.47	**52.7**
Zoom Out (140% rescale)	19.34	41.81	42.02	**45.29**
Zoom Out (160% rescale)	11.12	25.48	25.85	**27.02**

CIFAR-100

Results - Adversarial

CIFAR10 Models	FGSM ϵ=0.03
Adv. Training (Madry)	60.30
Adversarial Training + Fortified Networks	81.80
Baseline (ours)	36.32
Input Mixup ($\alpha = 1.0$)	71.51
Manifold Mixup ($\alpha = 2.0$)	77.50

CIFAR100 Models	FGSM ϵ=0.03
Input Mixup ($\alpha = 1.0$)	40.7
Manifold Mixup ($\alpha = 2.0$)	44.96



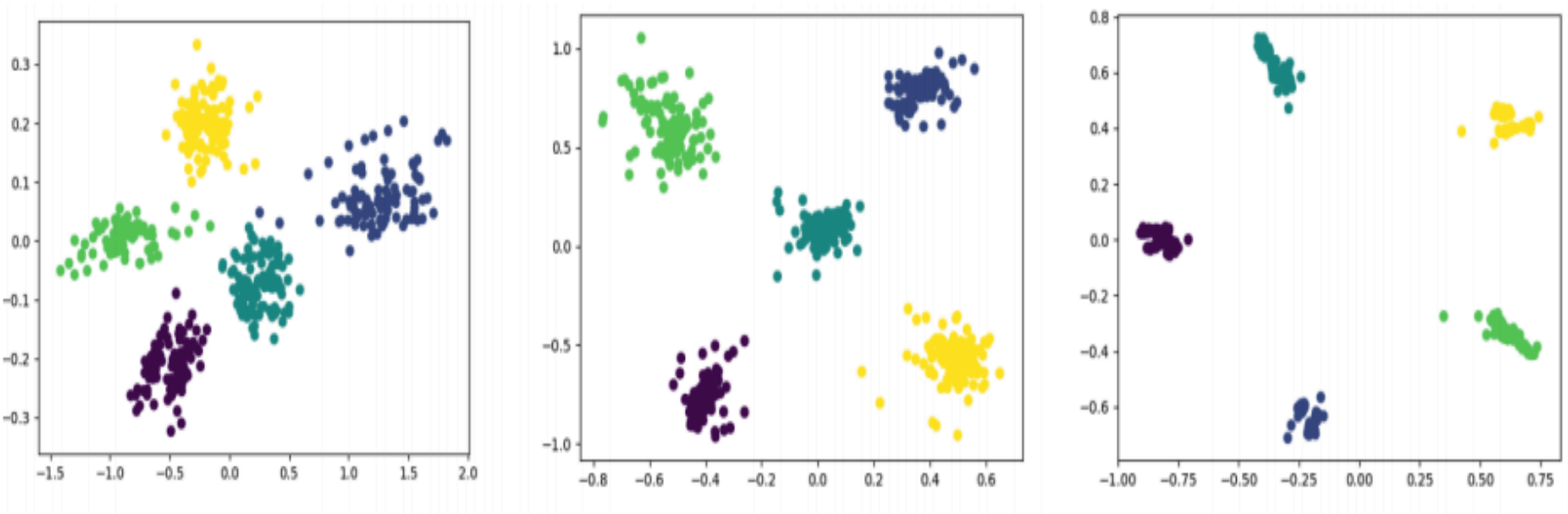
Semi-supervised Learning

Table 4: Results on semi-supervised learning on CIFAR-10 (4k labels) and SVHN (1k labels) (in test error %). All results use the same standardized architecture (WideResNet-28-2). Each experiment was run for 5 trials. † refers to the results reported in (Oliver et al., 2018)

SSL Approach	CIFAR-10	SVHN
Supervised †	20.26 ± 0.38	12.83 ± 0.47
Mean-Teacher †	15.87 ± 0.28	5.65 ± 0.47
VAT †	13.86 ± 0.27	5.63 ± 0.20
VAT-EM †	13.13 ± 0.39	$\mathbf{5.35 \pm 0.19}$
Semi-supervised Input Mixup	10.71 ± 0.44	6.54 ± 0.62
Semi-supervised *Manifold Mixup*	$\mathbf{10.26 \pm 0.32}$	5.70 ± 0.48

-Not too close to SOTA (CIFAR-10 down to 5%).
-For computational reasons, may still be preferable over VAT.

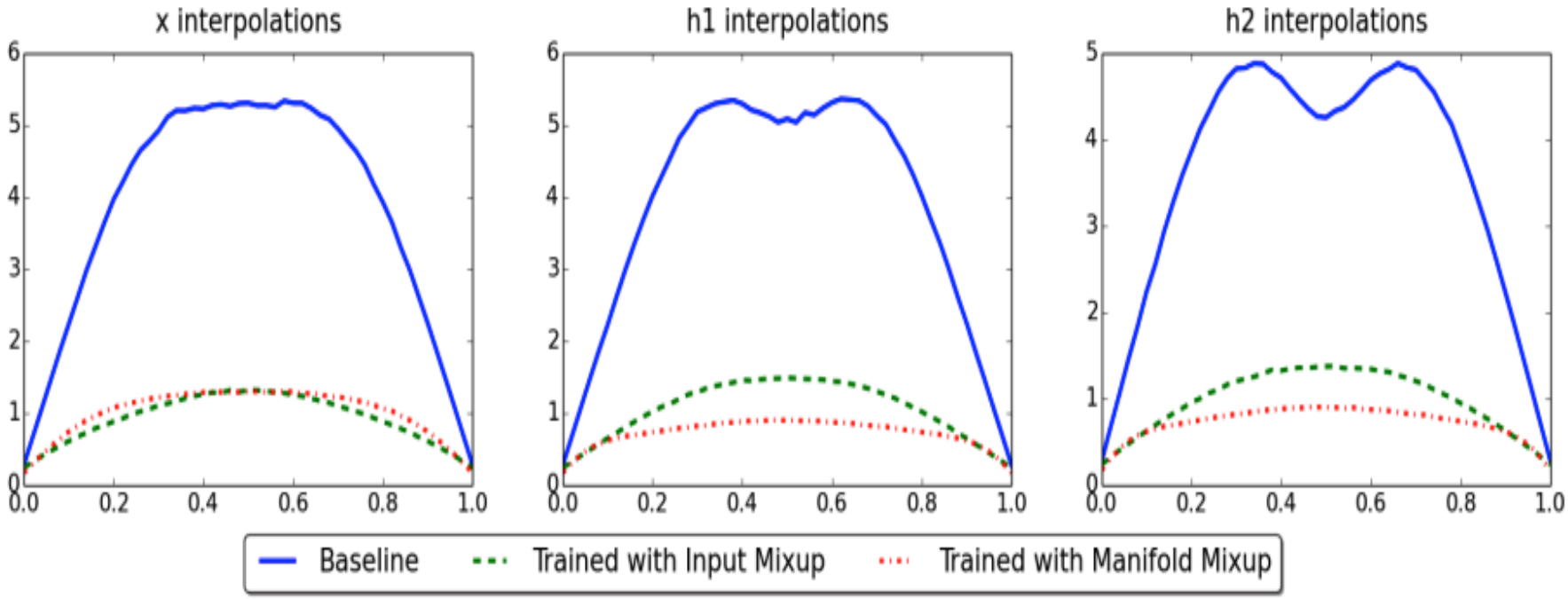
Analysis - How are representations changed?



-If we block gradients at the mixing points - Manifold Mixup no longer helps!

-Manifold Mixup is changing our representations to make interpolations less likely to collide.

Analysis - interpolations



-So manifold mixup helps a lot along hidden space interpolations.

-But input mixup helps a lot with hidden interpolations too. Why?

Visualizing interpolations

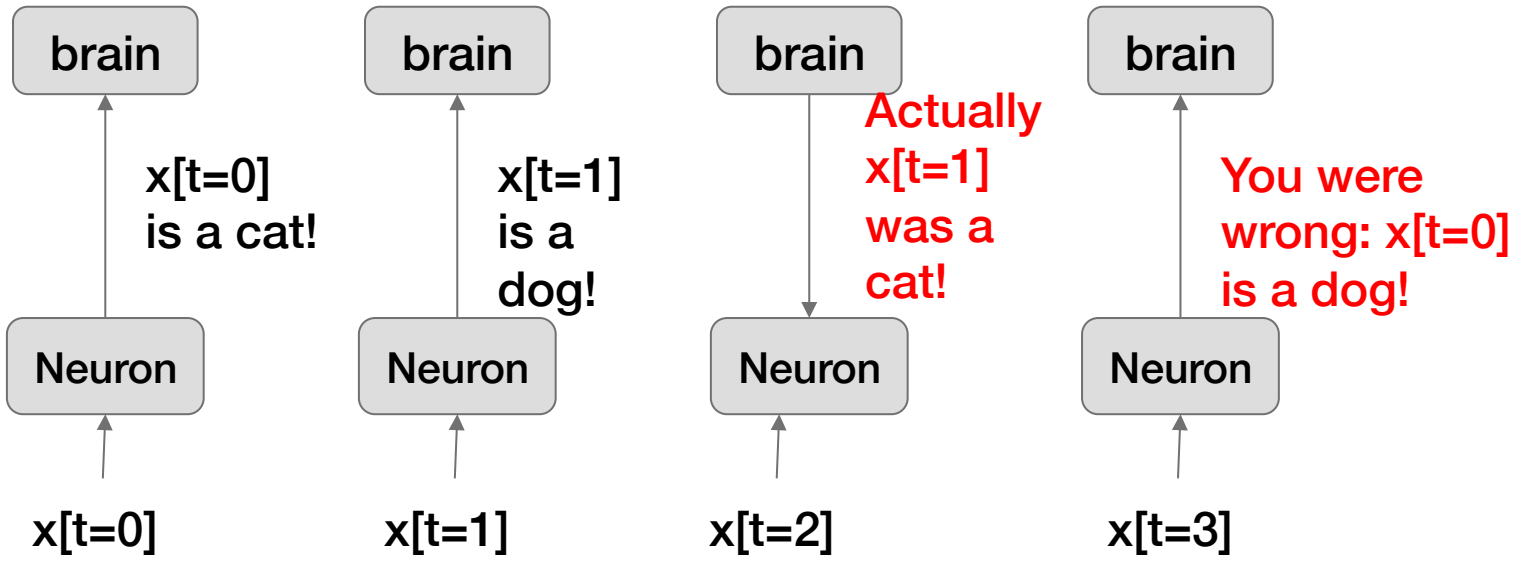
Figure 3: **Interpolations in the input space** with a mixing rate varied from 0.0 to 1.0.

Figure 4: **Interpolations in the hidden states** (using a small convolutional network trained to predict the input from the output of the second resblock). The interpolations in the hidden states show a better blending of semantically relevant features, and more of the images are visually consistent.

Biological Plausibility

-Let's say that you're a neuron and your job is recognizing animals.

-Another part of the brain uses your outputs, but takes a variable amount of time to return a feedback signal.



Current Practical Value

-Applying Manifold Mixup is rather straightforward.

-Requires (essentially) no additional computation or memory.

-Competitive with virtual adversarial training for semi-supervised.

-Provides significant gains in classification.

Montreal Institute for Learning Algorithms

MILA

Université de Montréal