

# Tackling Data Scarcity in Deep Learning

Anima Anandkumar & Zachary Lipton

email:

[anima@caltech.edu](mailto:anima@caltech.edu), [zlipton@cmu.edu](mailto:zlipton@cmu.edu)

shenanigans:

[@AnimaAnandkumar](#) [@zacharylipton](#)



**Carnegie  
Mellon  
University**

# Outline

- **Introduction / Motivation**

- **Part One**

- Deep Active Learning

- Active Learning Basics
    - Deep Active Learning for Named Entity Recognition (ICLR 2018) <https://arxiv.org/abs/1707.05928>
    - Active Learning w/o the Crystal Ball (*forthcoming* 2018)
    - How transferable are the datasets collected by active learners? (*arXiv* 2018) <https://arxiv.org/abs/1807.04801>
    - Connections to RL — Efficient exploration with BBQ Nets (AAAI 2018) <https://arxiv.org/abs/1608.05081>

- More realistic modeling of interaction

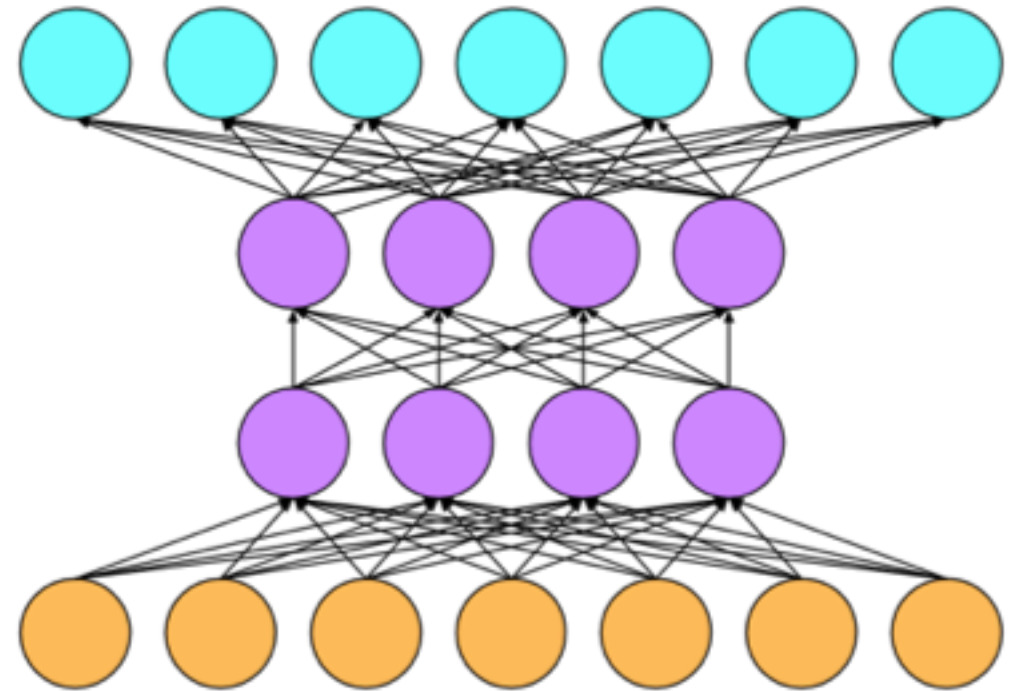
- Active Learning with Partial Feedback (*arXiv* 2018) <https://arxiv.org/abs/1802.07427>
    - Learning From Noisy, Singly Labeled Data (*ICLR* 2018) <https://arxiv.org/abs/1712.04577>

- **Part Two (Anima)**

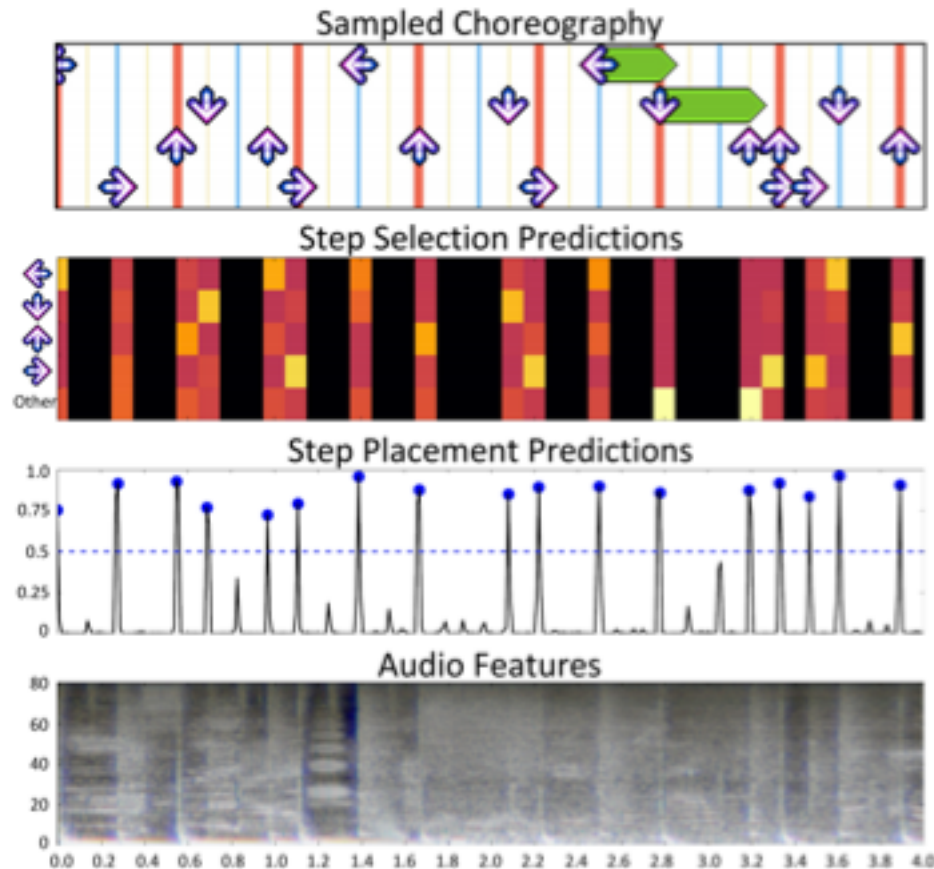
- Data Augmentation w/ Generative Models
  - Semi-supervised learning
  - Domain Adaptation
  - Combining Symbolic and Function Evaluation Expressions

# Deep Learning

- Powerful tools for building predictive models
- Breakthroughs:
  - Handwriting recognition ([Graves 2008](#))
  - Speech Recognition ([Mohamed 2009](#))
  - Drug Binding Sites ([Dahl 2012](#))
  - Object recognition ([Krizhevsky 2012](#))
  - Atari Game Playing ([Mnih 2013](#))
  - Machine Translation ([Sutskever 2014](#))
  - AlphaGO ([Silver 2015](#))



# Less well-known applications deep learning...



<https://arxiv.org/abs/1703.06891>

# Contributors to Success

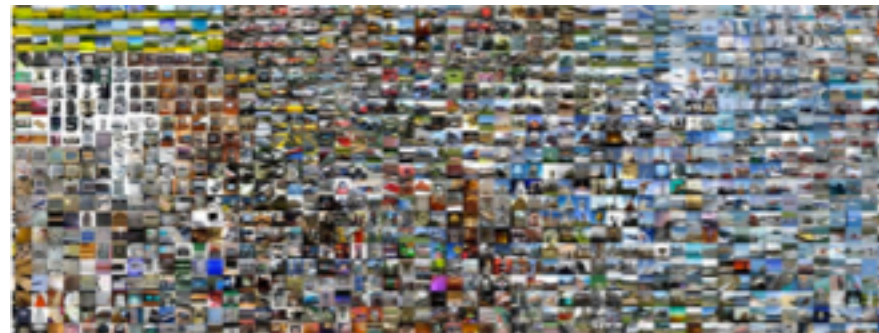
- Algorithms  
*(what we'd like to believe)*



- Computation



- Data



# Still, Big Problems Remain

- DL requires **BIG DATA**, often prohibitively expensive to collect
- Supervised models make **predictions** but we want to take **actions**
- Supervised learning **doesn't know why** a label applies
- In general, these models break under **distribution shift**
- DRL is impressive but brittle, suffers **high sample complexity**
- Modeling causal mechanisms sounds right, but we **lack tools**

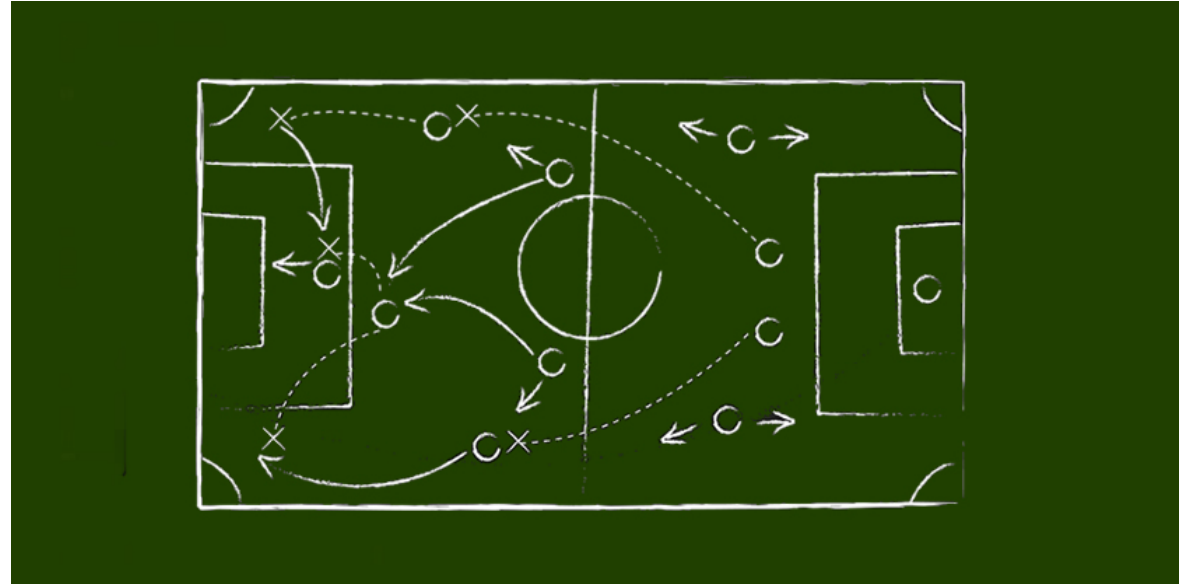
# Just How Data-Hungry are DL Systems?

- CV systems trained on ImageNet (**1M+** images)
- ASR (speech) systems trained on **11,000+ hrs** of annotated data
- OntoNotes (English) NER dataset contains **625,000** annotated words



# Strategies to Cope with Scarce Data

- Data Augmentation
- Semi-supervised Learning
- Transfer Learning
- Domain Adaptation
- **Active Learning**



# Considerations

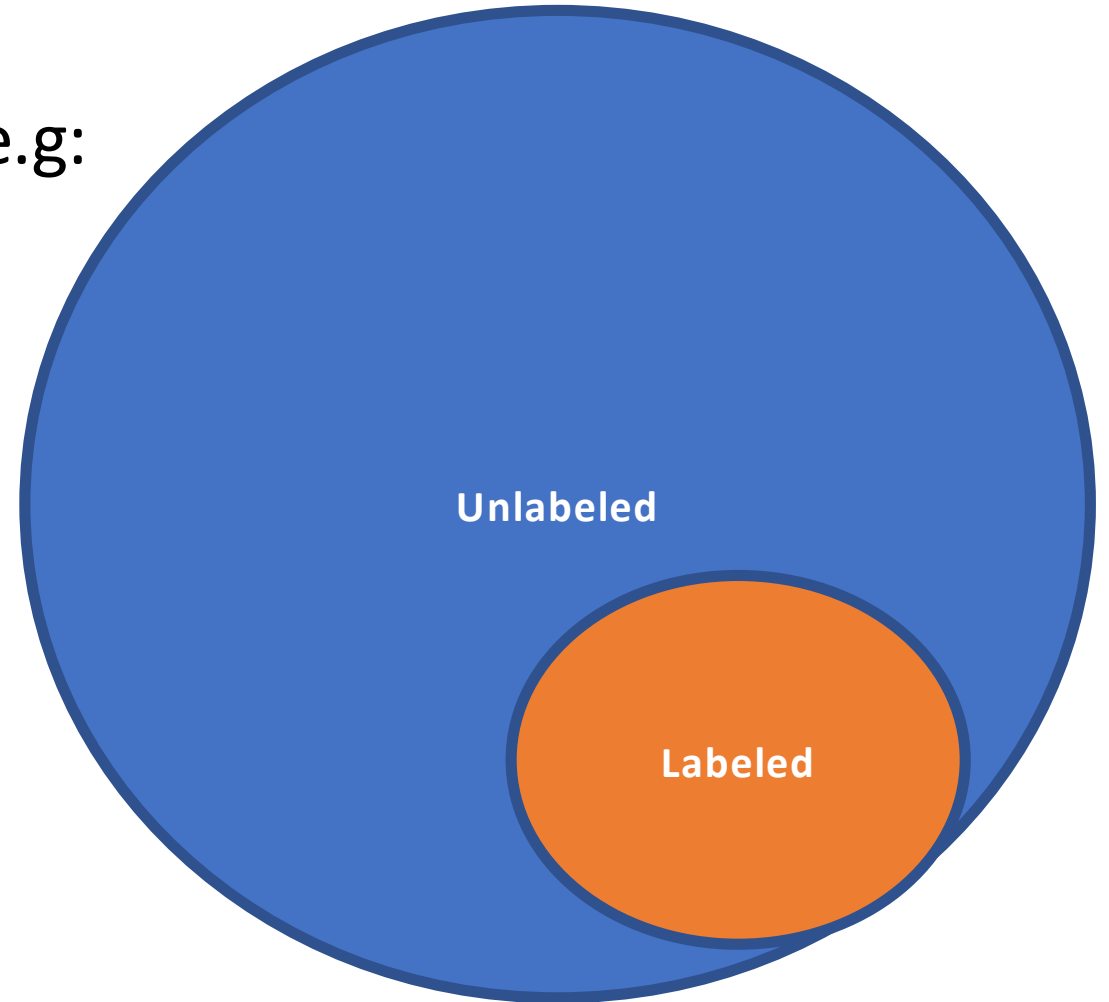
- Are examples  $\mathbf{x}$  scarce or just labels  $y$ ?
- Do we have access to annotators to **interactively** query labels?
- Do we have access lots of labeled data for **related tasks**?
- *Just how related* are the tasks?

# Semi-Supervised learning

- Use **both labeled & unlabeled** data, e.g:
  - Learn representation (AE) w all data, learn classifier w labeled data
  - Apply classification loss on *labeled*, regularizer on *all* data

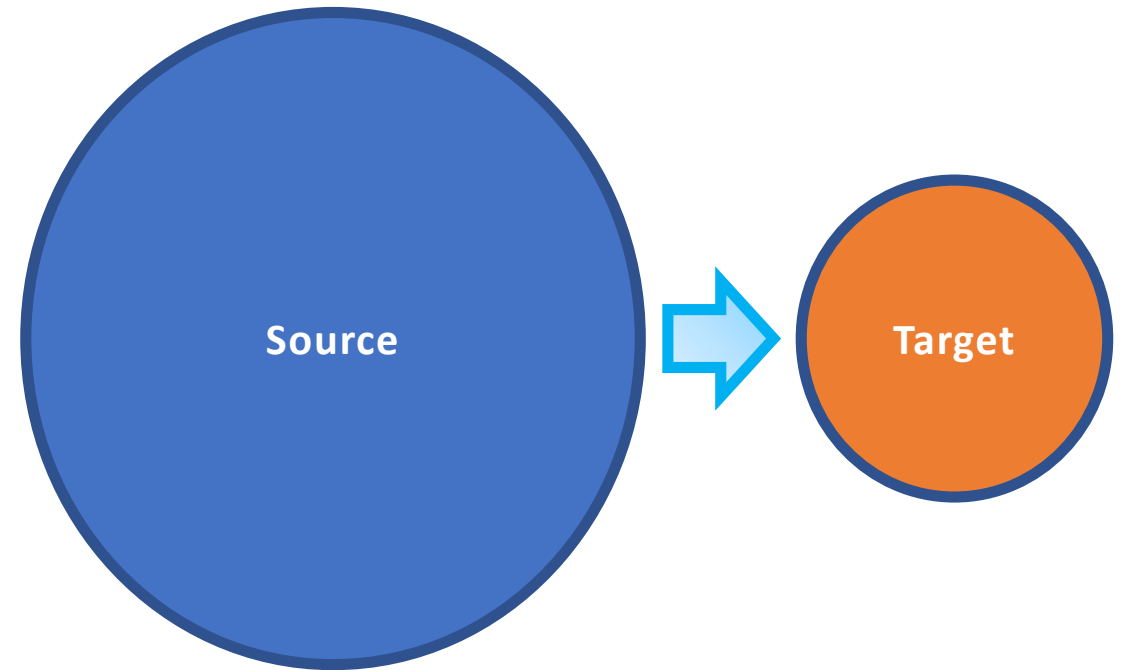
$$L = C(x, y, \theta) + R(x, \theta)$$

- Current SOA: Consistency-based training ([Laine](#), [Athiwaratkun](#))



# Transfer Learning

- $|D_{\text{source}}| \gg |D_{\text{target}}|$   
→ pre-train on source task
- Strangely effective, even across very different tasks
- Intuition: **transferable features**  
([Yosinski 2015](#))
- Requires some labeled target data
- Common practice, poorly understood



# Domain Adaptation

- **Labeled source** data, **unlabeled target** data
- When some invariances may not need target distribution labels
- Formal Setup
  - Distributions
    - Source distribution  $p(\mathbf{x}, y)$
    - Target distribution  $q(\mathbf{x}, y)$
  - Data
    - Training examples  $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n) \sim p(\mathbf{x}, y)$
    - Test examples  $(\mathbf{x}'_1, \dots, \mathbf{x}'_m) \sim q(\mathbf{x})$
  - Objective
    - Predict well on the test distribution, **WITHOUT** seeing any labels  $y_i \sim q(y)$

# Mission Impossible

- What if  
 $Q(Y=1 | \mathbf{x}) = 1 - P(Y=1 | \mathbf{x})$ ?
- Must make assumptions...
- Absent assumptions, DA is impossible ([Ben-David 2010](#))



# Label shift (aka target shift)

- Assume  $p(\mathbf{x}, y)$  changes, but the conditional  $p(\mathbf{x}|y)$  is fixed

$$q(y, \mathbf{x}) = q(y)p(\mathbf{x}|y)$$

- Makes anticausal assumption,  $y$  causes  $\mathbf{x}$ !
- Diseases cause symptoms, objects cause sensory data!
- But how can we estimate  $q(y)$  without any samples  $y_i \sim q(y)$ ?

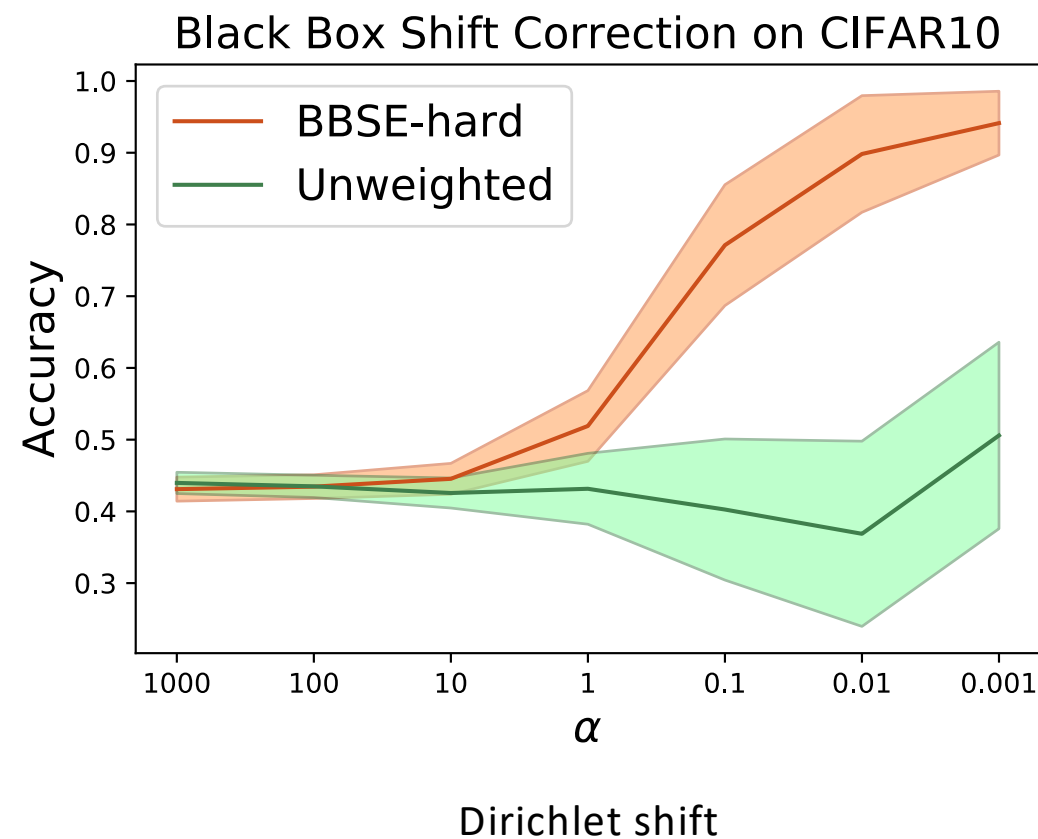
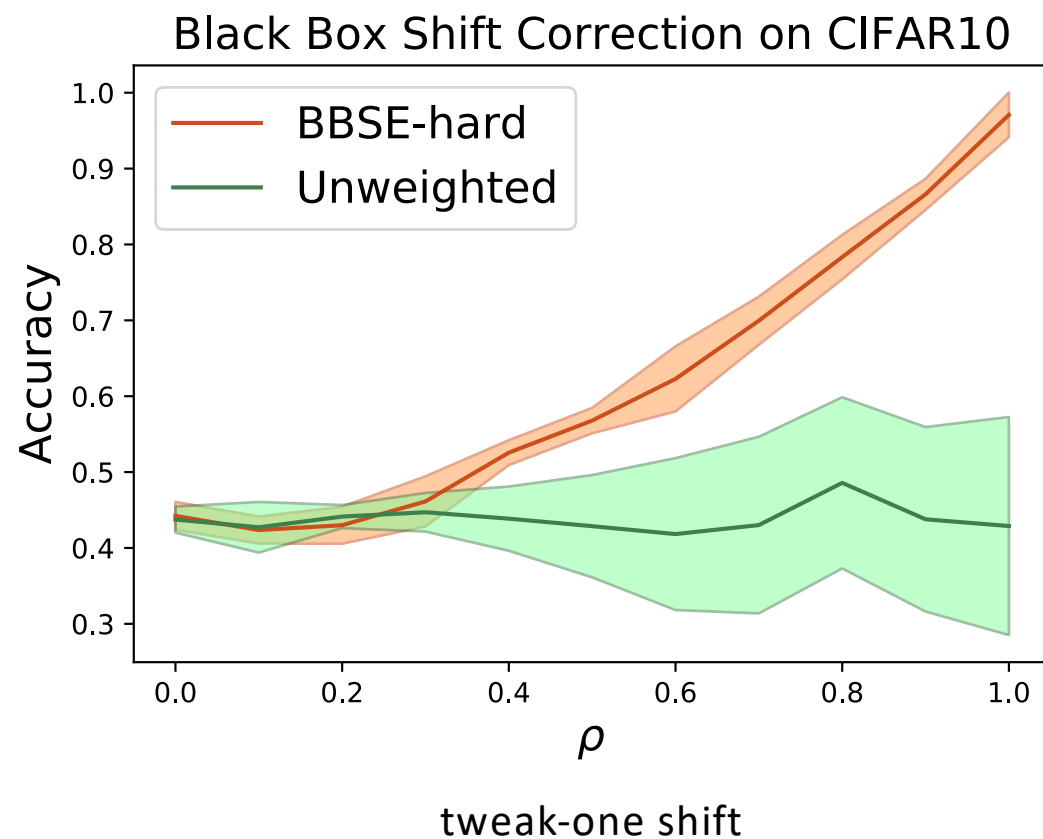
# Black box shift estimation

- Because  $C_{\hat{y}|y}$  is same on P and Q, we can solve for  $q(y)$  by solving a linear system
- We just need:
  1. Empirical C matrix converges
  2. Empirical C matrix invertible
  3. Expected  $f(x)$  converges

$$\begin{matrix} & y \\ & \boxed{C_{\hat{y}|y}} \\ \hat{y} & \end{matrix} \cdot q(y) = \begin{matrix} \hat{y} \\ \boxed{\phantom{0}} \\ Q \end{matrix}$$

P

# Can estimate shift, (CIFAR 10)



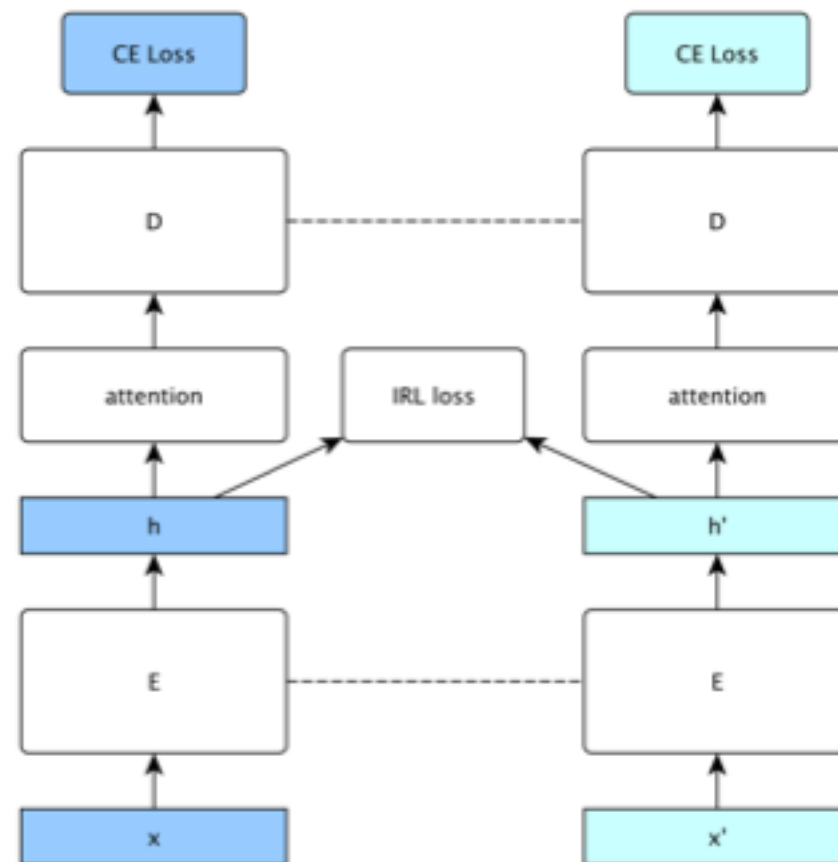
# Other Domain Adaptation Variations

- Covariate shift  $p(y|\mathbf{x}) = q(y|\mathbf{x})$   
([Shimodaira 2000](#), [Gretton 2007](#), [Sugiyama 2007](#), [Bickel 2009](#))
- Divergence  $d(p||q) < \varepsilon$   
 $\lambda$ -shift ([Mansour 2013](#)) f-divergences ([Hu 2018](#))
- Data augmentation: assumed invariance to rotations, crops, etc.  
([Krizhevsky 2012](#))
- Multi-condition training in speech ([Hirsch 2000](#))
- Adversarial examples: assumed invariance to  $l_p$  norm perturbations  
([Goodfellow 2014](#))

# Noise-Invariant Representations ([Liang 2018](#))

- Noisy examples not just **same class**, they're (to us) the **same image**
- Penalize difference in latent representations

	CER on Noisy Data					
	Base	Data Aug.	Adv.	Logit	IRL-E	IRL-C
Error on test-clean	6.5%	6.4%	6.5%	5.1%	3.5%	<b>3.3%</b>
In-domain (6SNRdB)	27.8%	10.8%	16.5%	8.7%	6.0%	<b>5.7%</b>
In-domain (12SNRdB)	13.5%	7.8%	12.1%	6.2%	4.2%	<b>4.1%</b>
Impulse Convolve	24.1%	21.0%	28.3%	47.6%	18.0%	<b>13.8%</b>
Speech (6SNRdB)	91.5%	32.0%	67.7%	33.0%	16.4%	<b>14.1%</b>
Speech (12SNRdB)	77.8%	15.2%	34.7%	11.1%	7.6%	<b>6.8%</b>
Volume (+6 dB)	6.5%	6.4%	9.8%	5.1%	3.6%	<b>3.5%</b>
Volume (-6 dB)	6.5%	6.3%	9.6%	5.0%	3.6%	<b>3.5%</b>
Telephony	14.2%	12.2%	21.3%	10.3%	7.1%	<b>6.4%</b>

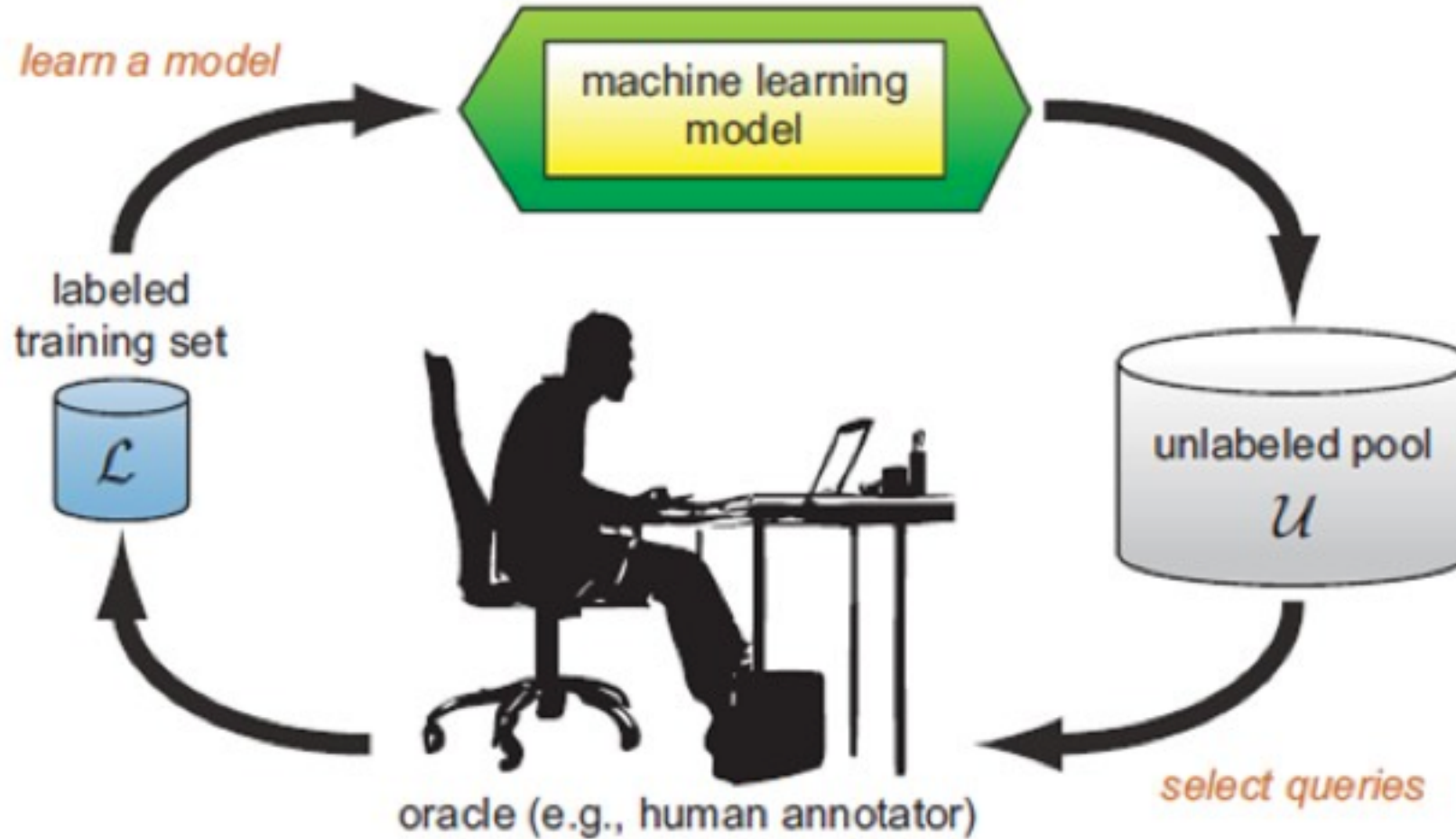


# Outline

- Deep Active Learning
  - **Active Learning Basics**
  - Deep Active Learning for Named Entity Recognition (ICLR 2018)  
<https://arxiv.org/abs/1707.05928>
  - Active Learning w/o the Crystal Ball (under review)
  - How transferable are the datasets collected by active learners? (in prep)
  - Connections to RL — Efficient exploration with BBQ Nets (AAAI 2018)  
<https://arxiv.org/abs/1608.05081>
- More realistic dive into interactive mechanisms
  - Active Learning with Partial Feedback <https://arxiv.org/abs/1802.07427>
  - Learning From Noisy, Singly Labeled Data (ICLR 2018)  
<https://arxiv.org/abs/1712.04577>

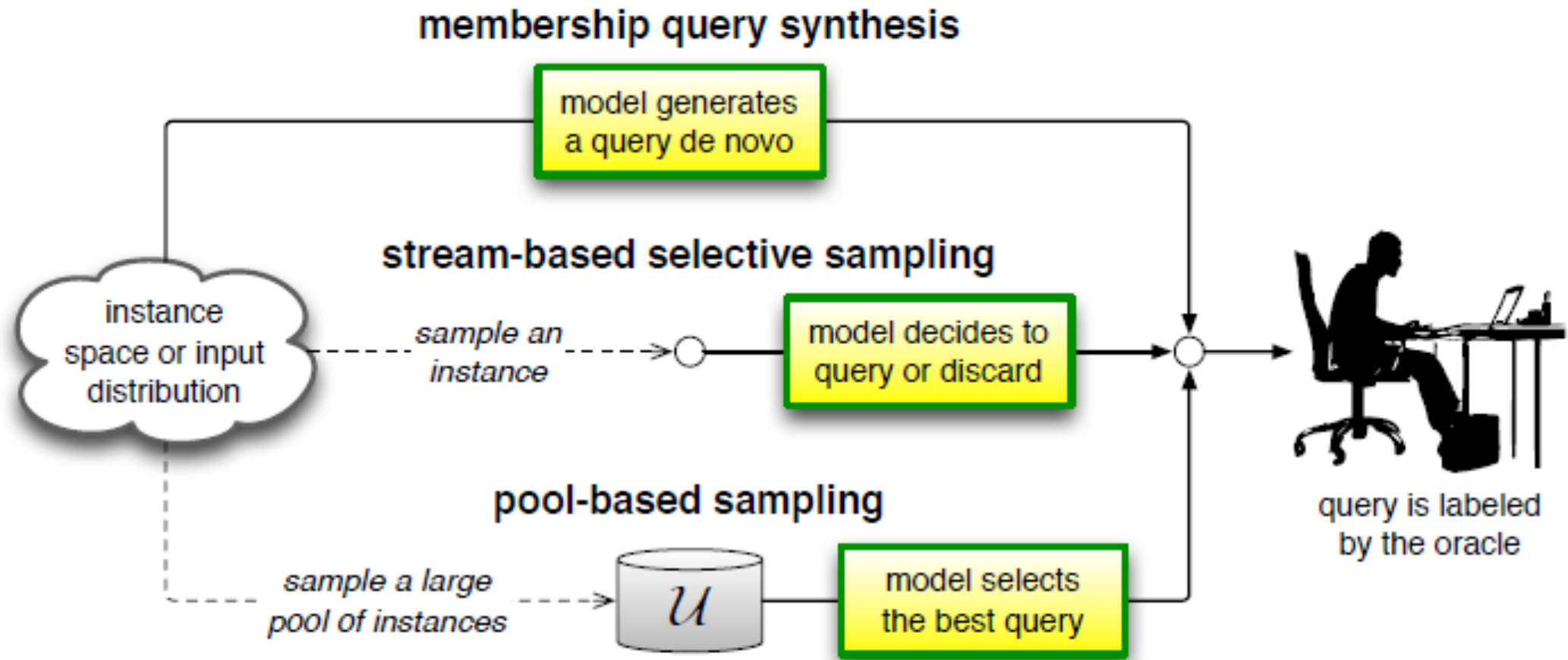
# Active Learning Basics

# Active Learning



[Image credit: Settles, 2010](#)

# Design decision: *pool-*, *stream-*, *de novo*-based



# Other considerations

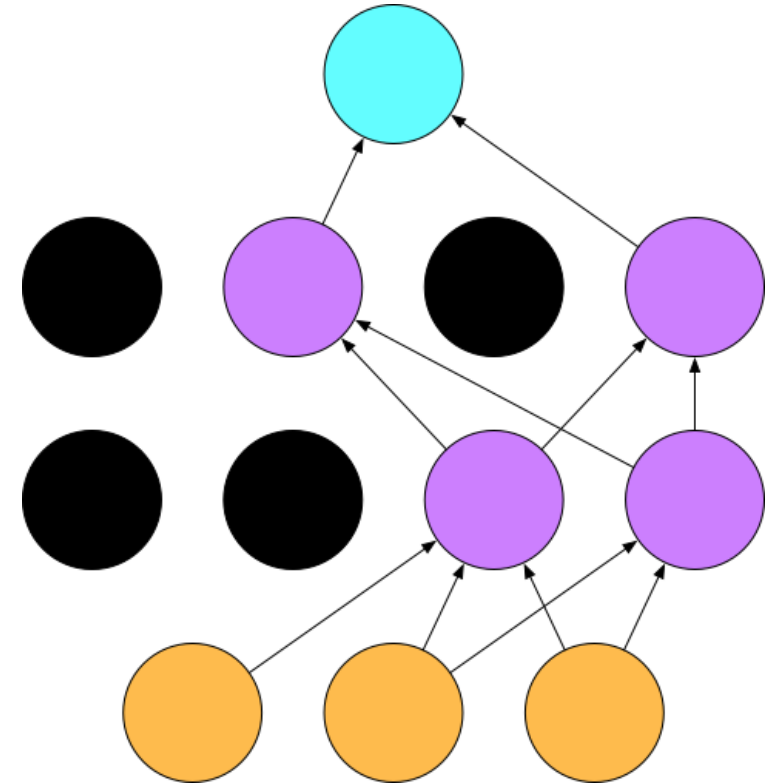
- **Acquisition function** — How to choose samples?
- **Number of queries per round** — Tradeoff b/w computation/accuracy
- **Fine-tuning vs training from scratch between rounds**  
Fine-tuning more efficient, but danger of overfitting earlier samples
- **Must get things right the first time!**

# Acquisition functions

- Uncertainty based sampling
  - Least confidence  $-\max_k \hat{y}_k$
  - Maximum entropy  $-\sum_k \hat{y}_k \log \hat{y}_k$
- Bayesian Active Learning by Disagreement (BALD) ([Houlsby 2011](#))
  - Sample multiple times from a stochastic model
  - Look at the consensus (plurality) prediction
  - Estimate confidence = the percentage of votes agreeing on that prediction

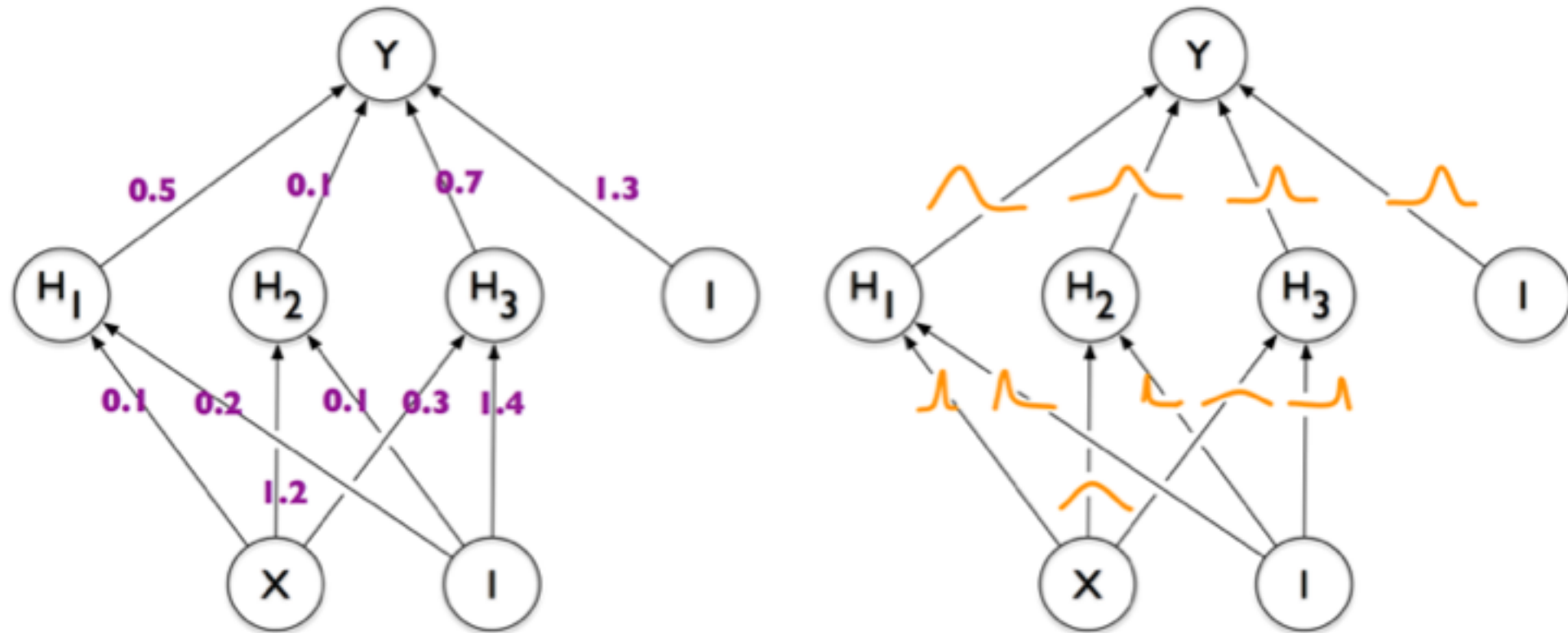
# The Dropout Method ([Gal 2017](#))

- Train with dropout
- Sample  $n$  independent dropout masks
- Make forward pass w each dropout mask
- Assess *confidence* based on agreement



<https://arxiv.org/abs/1703.02910>

# Bayes-by-Backpropagation (weight uncertainty)



<https://arxiv.org/abs/1505.05424>

# Bayes-by-Backprop gives useful uncertainty estimates

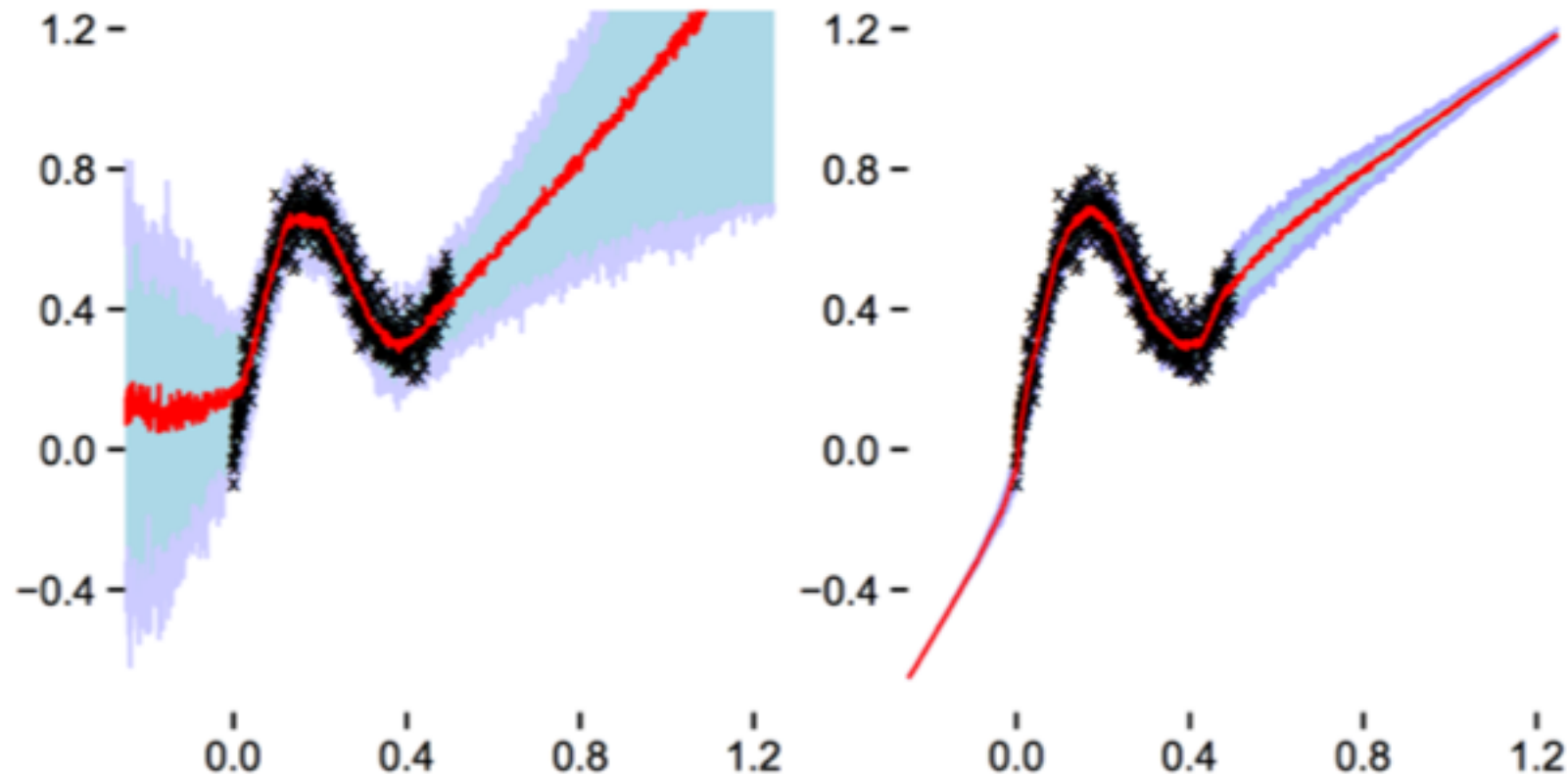


Figure from *Weight Uncertainty* (Blundell et al. 2015)

# Optimizing variational parameters

$$\begin{aligned}\theta^* &= \arg \min_{\theta} \text{KL}[q(\boldsymbol{w}|\theta) || P(\boldsymbol{w}|\mathcal{D})] \\ &= \arg \min_{\theta} \int_{-\infty}^{\infty} q(\boldsymbol{w}|\theta) \ln \frac{q(\boldsymbol{w}|\theta)}{P(\boldsymbol{w})P(\mathcal{D}|\boldsymbol{w})} d\boldsymbol{w} \\ &= \arg \min_{\theta} \text{KL}[q(\boldsymbol{w}|\theta) || P(\boldsymbol{w})] - \mathbb{E}_{q(\boldsymbol{w}|\theta)}[\ln P(\mathcal{D}|\boldsymbol{w})]\end{aligned}$$

# Deep Active Learning for Named Entity Recognition

Yanyao Shen, Hyokun Yun, Zachary C. Lipton, Yakov Kronrod, Anima Anandkumar

<https://arxiv.org/abs/1707.05928>

# Named Entity Recognition

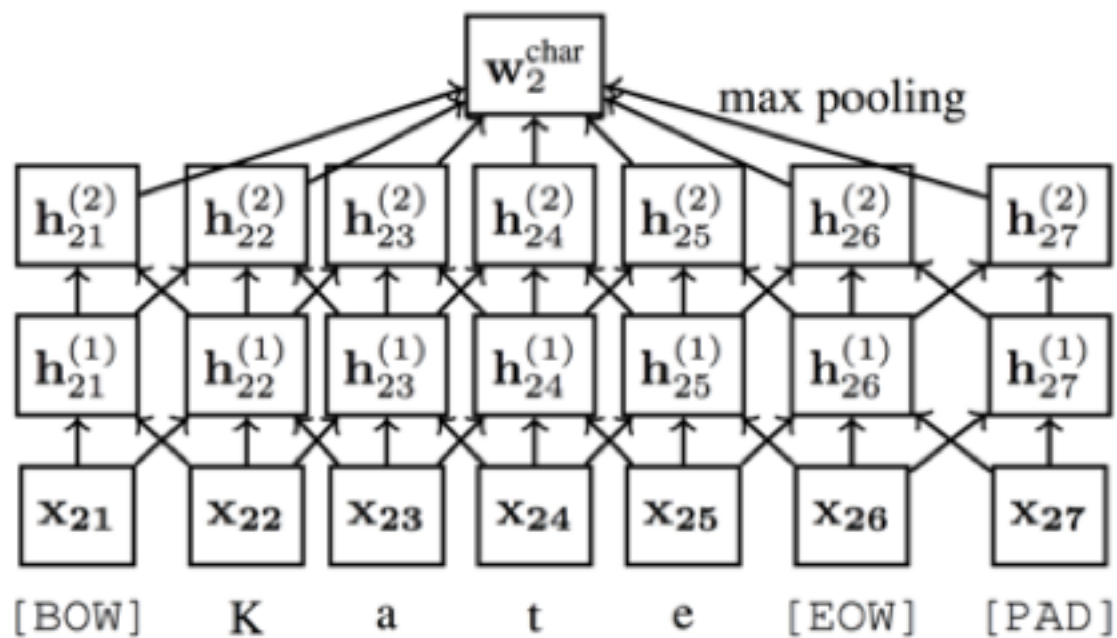
In 1917, Einstein applied the general theory of relativity to model the large-scale structure of the universe. He was visiting the United States when Adolf Hitler came to power in 1933 and did not go back to Germany, where he had been a professor at the Berlin Academy of Sciences. He settled in the U.S., becoming an American citizen in 1940. On the eve of World War II, he endorsed a letter to President Franklin D. Roosevelt alerting him to the potential development of "extremely powerful bombs of a new type" and recommending that the U.S. begin similar research. This eventually led to what would become the Manhattan Project. Einstein supported defending the Allied forces, but largely denounced using the new discovery of nuclear fission as a weapon. Later, with the British philosopher Bertrand Russell, Einstein signed the Russell-Einstein Manifesto, which highlighted the danger of nuclear weapons. Einstein was affiliated with the Institute for Advanced Study in Princeton, New Jersey, until his death in 1955.

Tag colours:

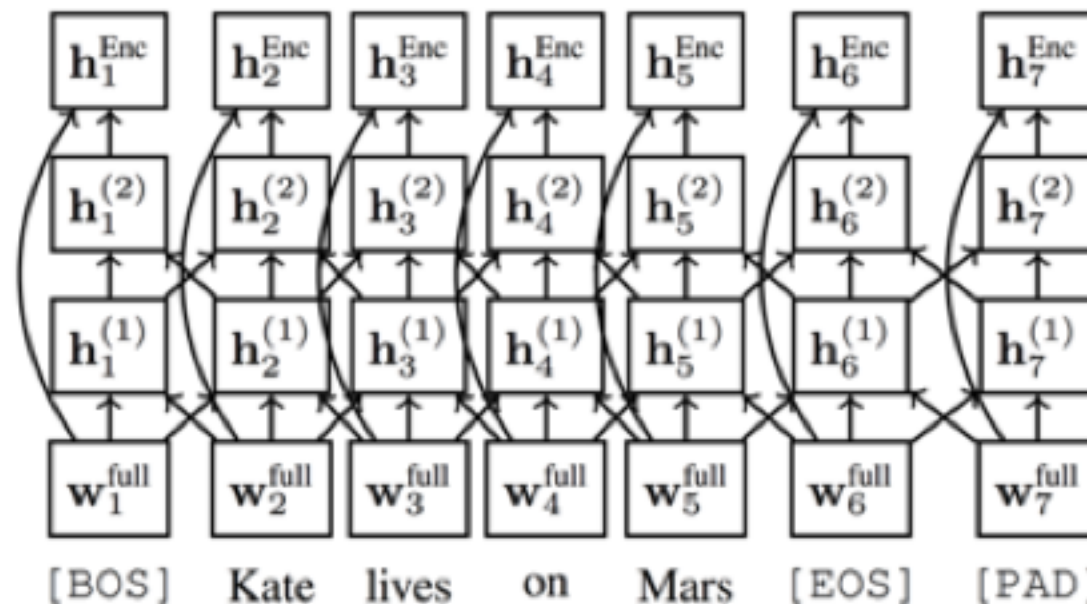
LOCATION TIME PERSON ORGANIZATION MONEY PERCENT DATE

# Modeling - Encoders

Word embedding



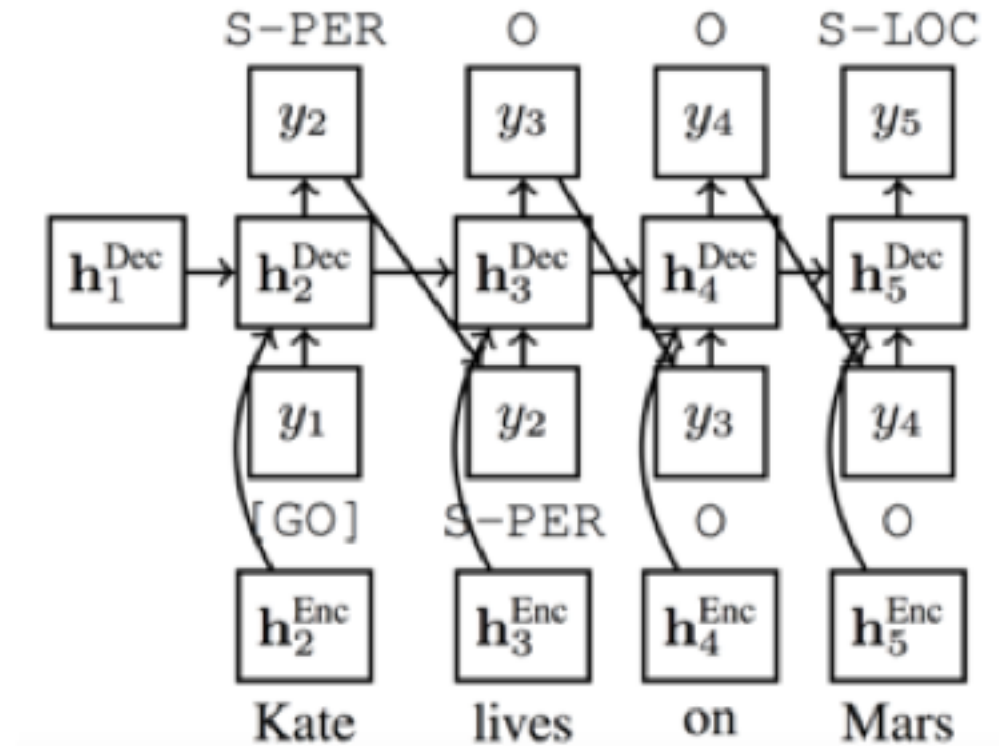
Sentence encoding



$$w_i^{\text{full}} := (w_i^{\text{char}}, w_i^{\text{emb}})$$

# Tag Decoder

- Each tag conditioned on
  1. Current sentence representation
  2. Previous decoder state
  3. Previous decoder output
- Greedy decoding
  1. For OntoNotes NER wide beam gives little advantage
  2. Faster, necessary for active learning



# Active learning heuristics

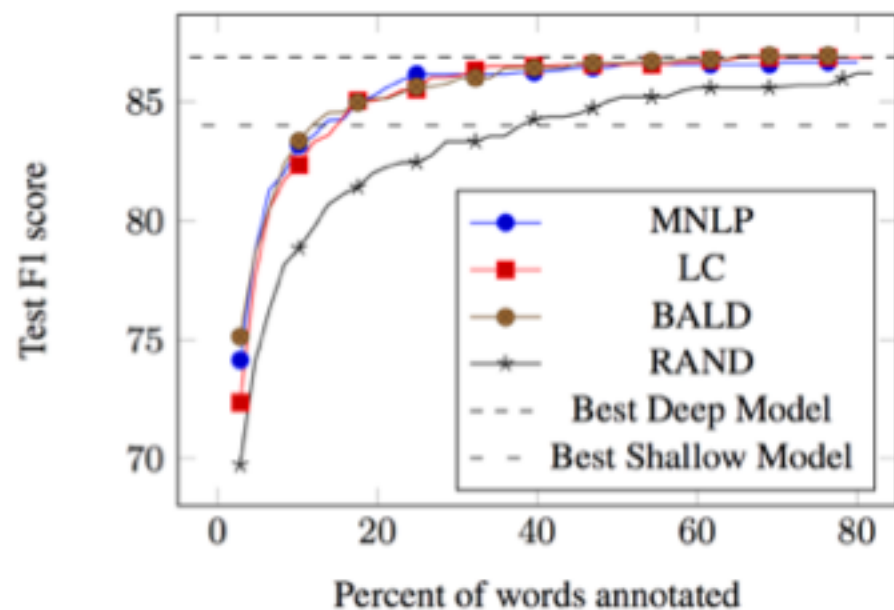
**Normalized maximum log probability**

$$\max_{y_1, \dots, y_n} \frac{1}{n} \sum_{i=1}^n \log \mathbb{P}[y_i \mid y_1, \dots, y_{n-1}, \{\mathbf{x}_{ij}\}]$$

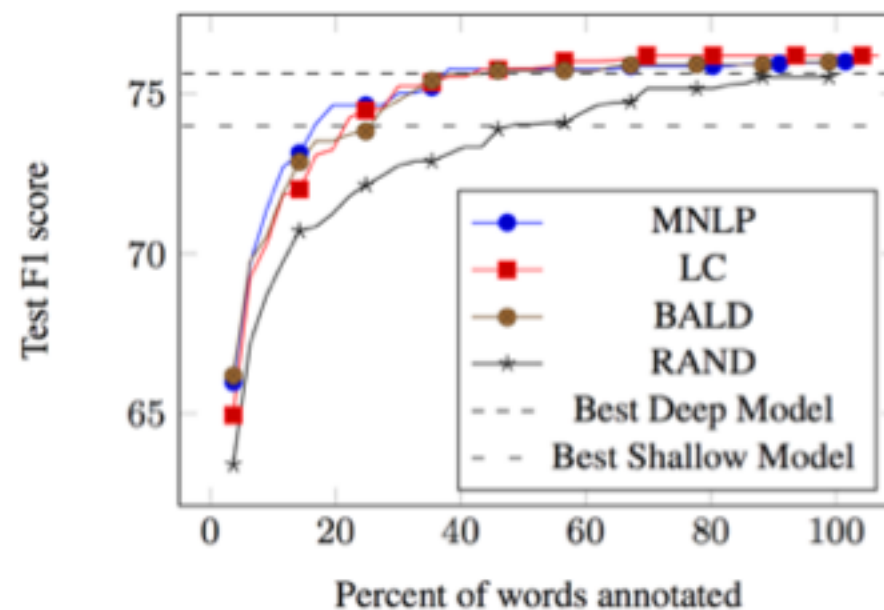
**Bayesian active learning by disagreement (BALD)**

$$f_i = 1 - \frac{\max_y |\{m : \operatorname{argmax}_{y'} \mathbb{P}^m[y_i = y'] = y\}|}{M}$$

# Results — 25% samples, 99% performance



(a) OntoNotes-5.0 English



(b) OntoNotes-5.0 Chinese

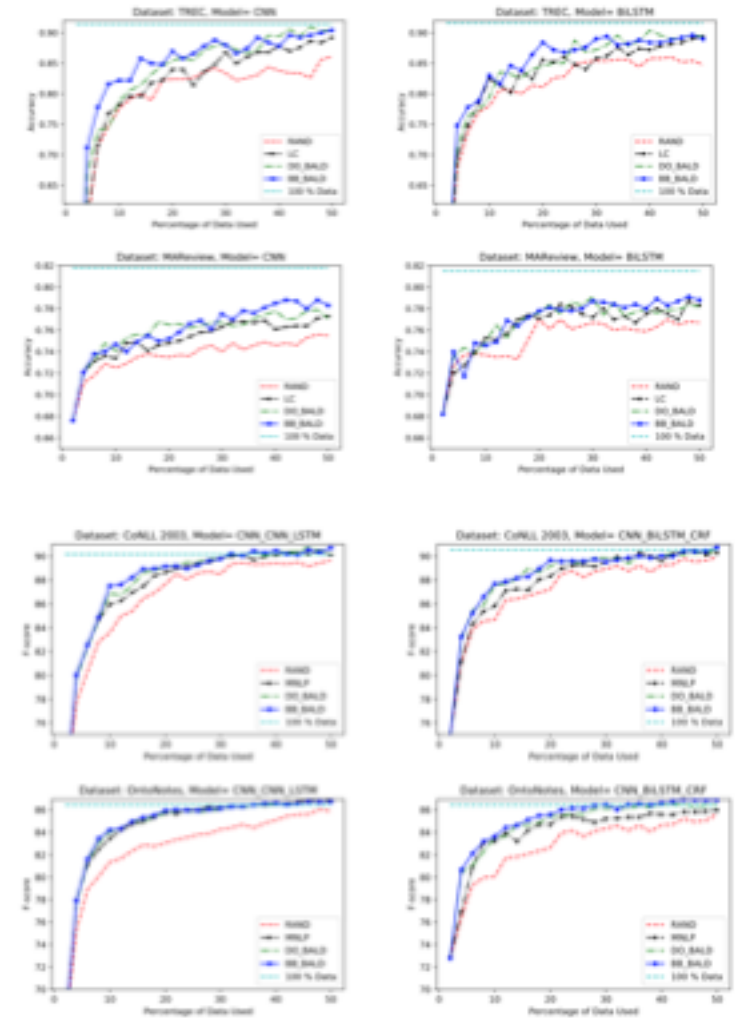
# Problems!

- Active learning sounds **great on paper**
- But...
  1. Paints a cartoonish picture of annotation
  2. Hindsight is 20/20, **but not our foresight**
  3. In reality, can't run 4 strategies & **retrospectively** pronounce a winner
  4. Can't use full set of labels to pick architecture, hyperparameters
  5. Supervised learner **can mess up** – active learner **must be right** 1<sup>st</sup> time

# Active Learning without the Crystal Ball

(work with Aditya Siddhant)

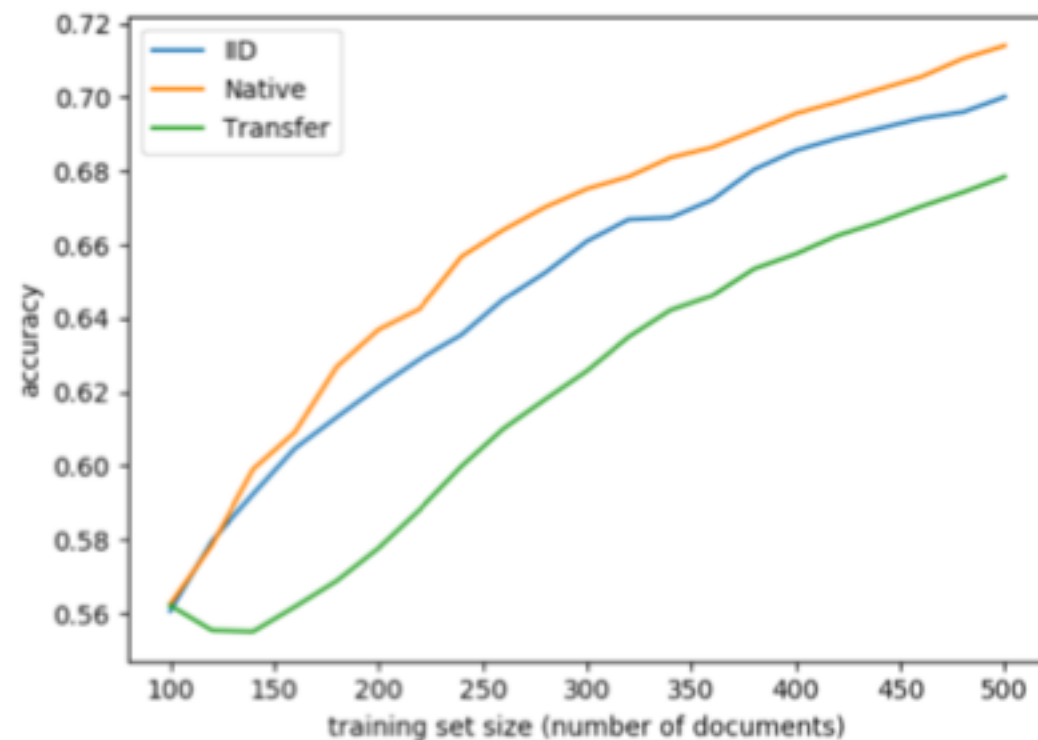
- Simulated active learning shows results on 1 problem, 1-2 **datasets**, with 1 **model**
- **Peeks at data for hyperparameters** of inherited architectures
- We look across settings to see:  
**does consistent story emerge?**
- **Surprisingly, BALD performs best across wide range of NLP problems**
- **Both Dropout & Bayes-by-Backprop work**



# How Transferable are Active Sets Across Learners?

(w David Lowell & Byron Wallace)

- Datasets tend to have a longer shelf-life than models
- When model goes stale, will active set transfer to new models?
- **Answer is dubious**
- Sometimes **outperforms**, but often **underperforms** i.i.d. data



# Other approached & research directions

- Pseudo-label when confident, actively query when not ([Wang 2016](#))
- Select based on representativeness (select for a diverse samples)
- Select based on expected magnitude of the gradient ([Zhang 2017](#))

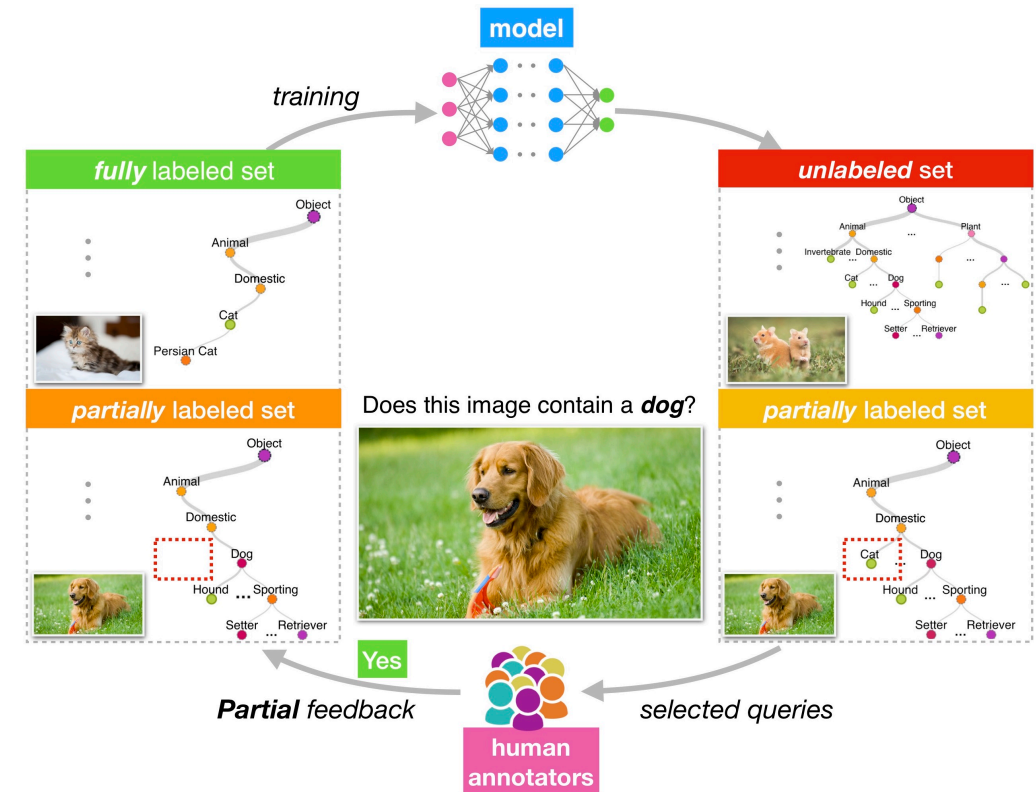
# Active Learning with Partial Feedback

Peiyun Hu, Zachary C. Lipton, Anima Anandkumar, Deva Ramanan

<https://arxiv.org/pdf/1802.07427.pdf>

# Opening the black annotation black box

- Traditional active learning papers ignore how the sausage gets made
- Real labeling procedures not atomic
- Annotation requires asking simpler (often binary questions):  
*Does this image contain a dog?*
- Cost  $\propto$  [# of questions asked]



# How was ImageNet Created

- Step 1: get a list of categories
- Step 2: use Google Image Search to filter per-category candidates
- Step 3: Crowdsource with **binary feedback**

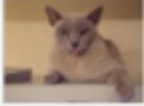

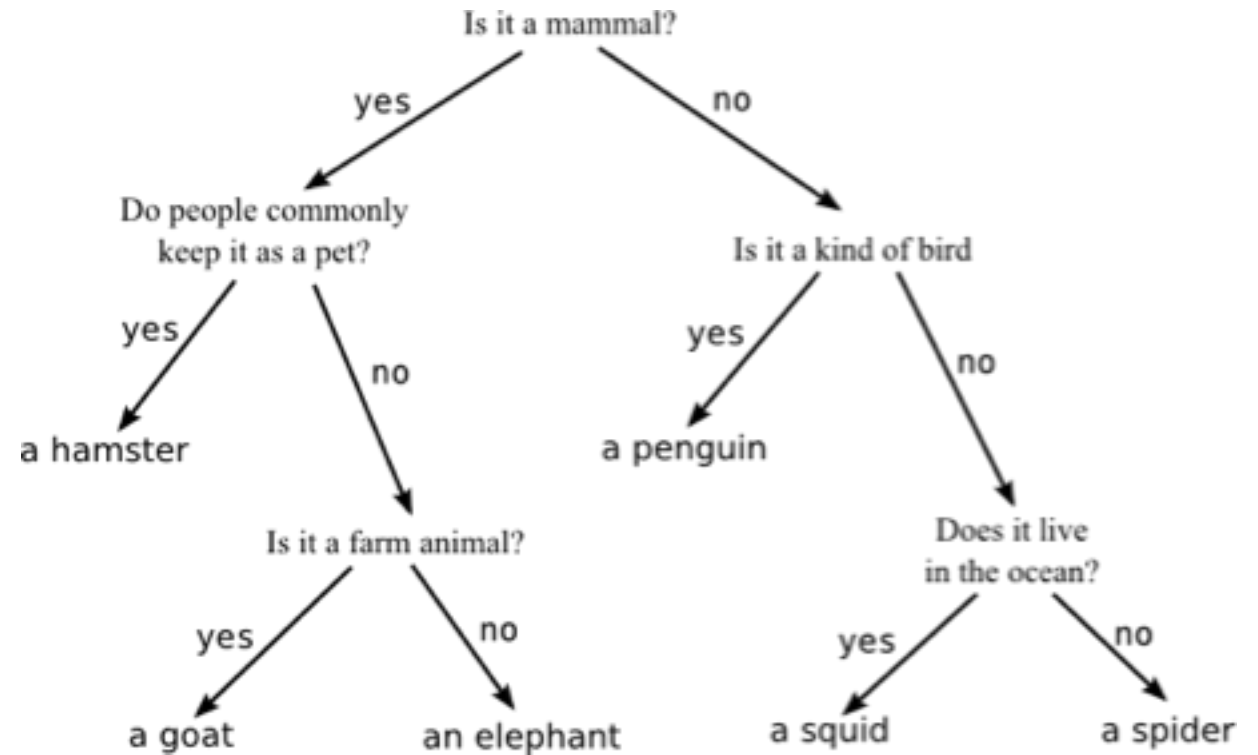
							
User 1	Y	Y	Y	#Y	# N	Conf Cat	Conf BCat
User 2	N	Y	Y	0	1	0.07	0.23
User 3	N	Y	Y	1	0	0.85	0.69
User 4	Y	N	Y	1	1	0.46	0.49
User 5	Y	Y	Y	2	0	0.97	0.83
User 6	N	N	Y	0	2	0.02	0.12
				3	0	0.99	0.90
				2	1	0.85	0.68

Figure 7: **Left:** Is there a Burmese cat in the images? Six randomly sampled users have different answers. **Right:** The confidence score table for “Cat” and “Burmese cat”. More votes are needed to reach the same degree of confidence for “Burmese cat” images.

# More realistic active learning scenario

- Producing an exact label is a lot like playing 20 questions
- Could start at top of hierarchy and drill down
- Better – use current beliefs to decide *which questions to ask!*



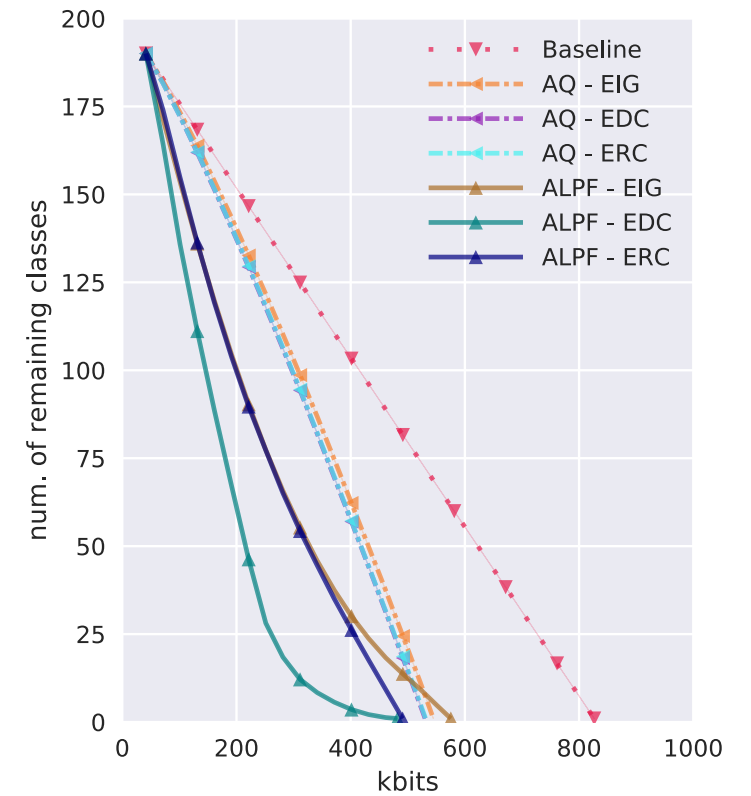
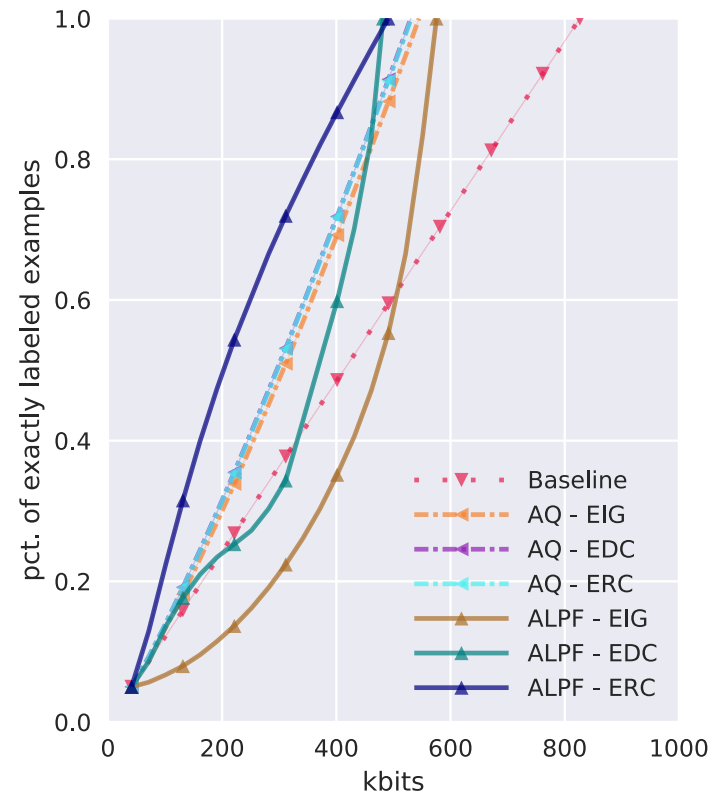
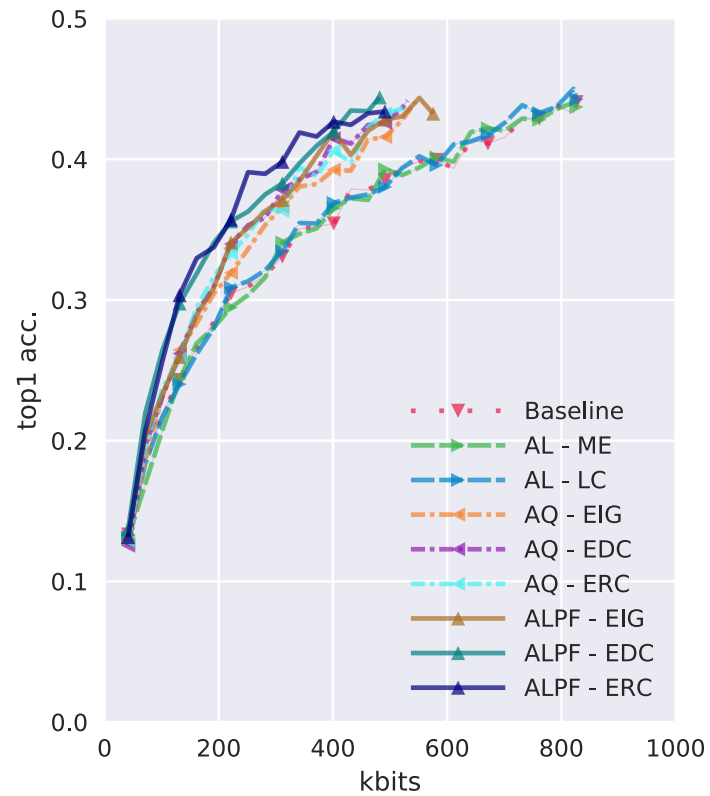
# Active learning with partial feedback

- Given: unlabeled data, a hierarchy of classes (say, a tree)
- Choose **questions**: (example, class) pairs
- The class can be a superclass, e.g. internal nodes in hierarchy
- Annotator gives binary feedback
- Train on partial labels
- Best classifier under fixed budget
- Annotate a dataset with fewer queries

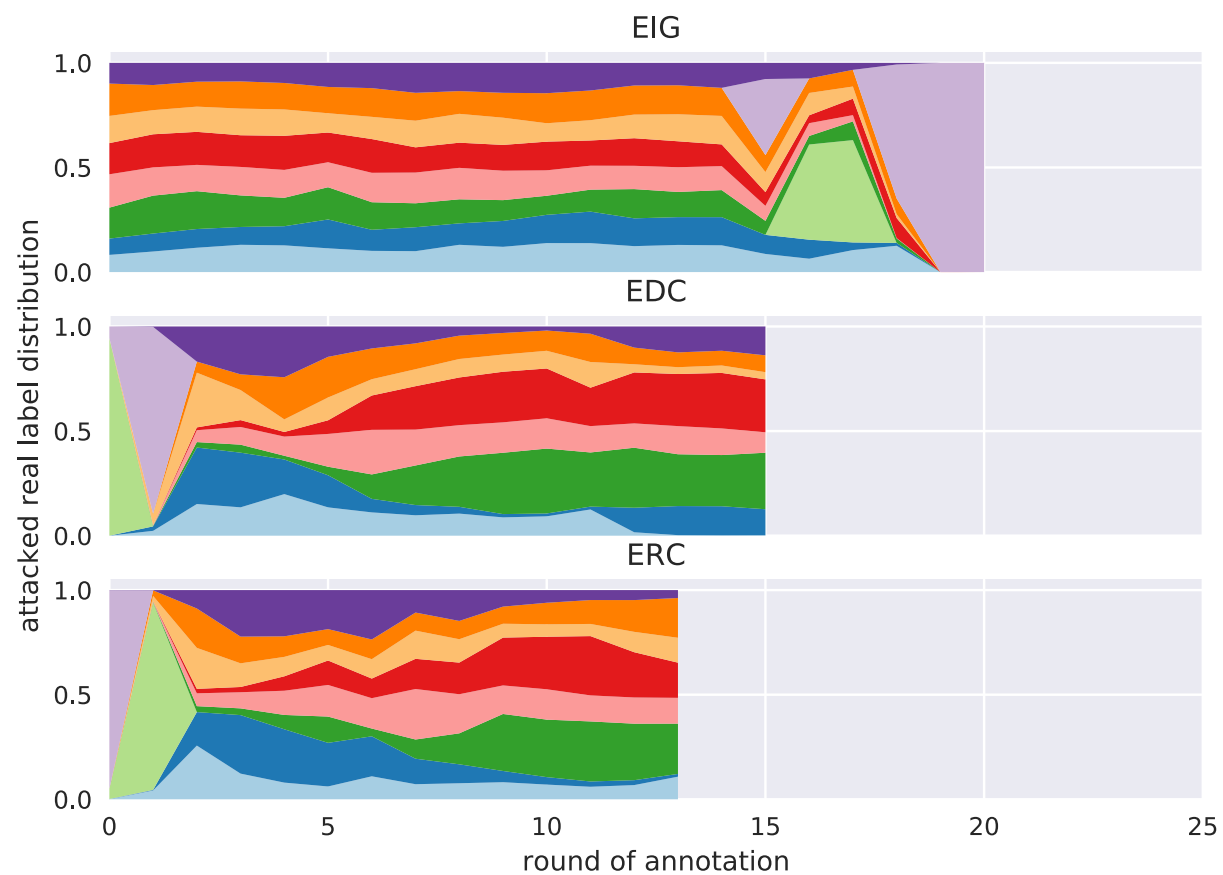
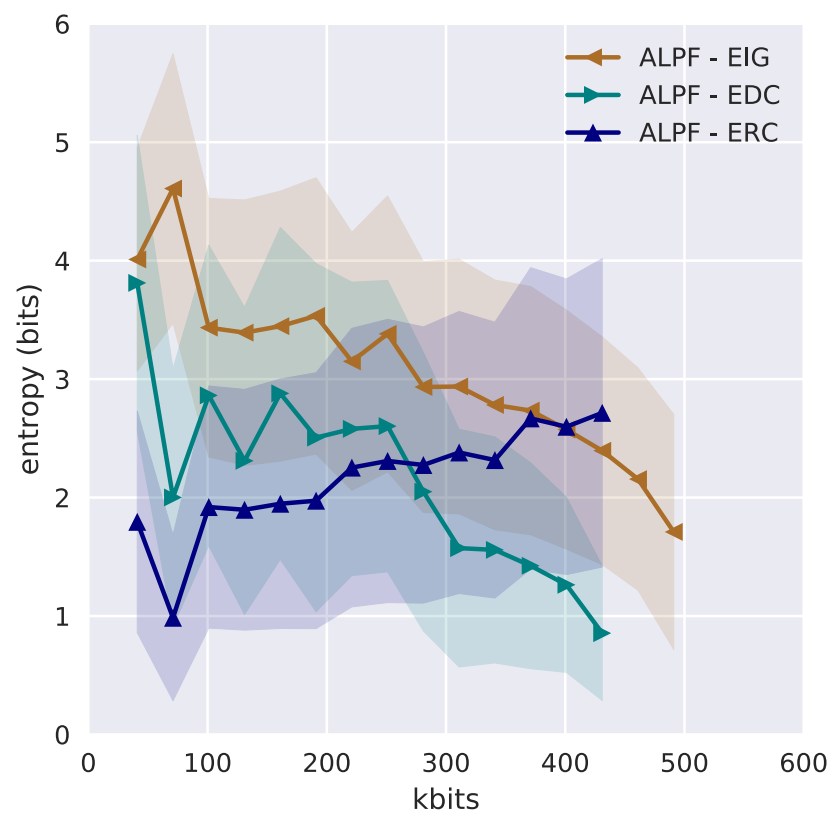
# Three sampling strategies

- Expected information gain (EIG)
- Expected decrease in potential classes (EDC)
- Expected number of remaining classes (ERC)

# Quantitative results



# Qualitative analysis










# Learning from Noisy Singly-Labeled Data

Ashish Khetan, Zachary C. Lipton, Anima Anandkumar

<https://arxiv.org/abs/1712.04577>

# Classical Crowdsourcing Setup

- Redundant labeling to overcome noise
- Task: aggregate intelligently
- Naive baseline: **majority vote**
- Can do better with EM
- **Classic algos ignore features**
- Given 1 label/ex. all workers perfect!

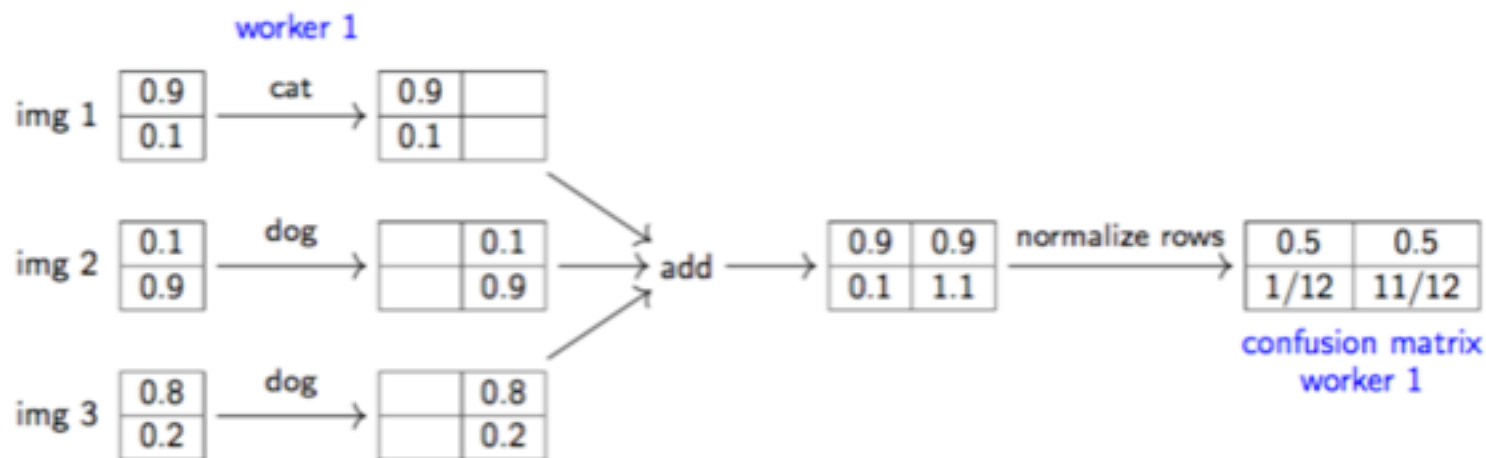
						
	✓		✓		×	
	✓	×			×	
			✓	×		×
		×	✓		×	
	×			×		×
		✓		✓		✓
majority voting	✓	×	✓	×	×	×

training data for supervised learning

## Expectation Maximization (EM)

- Initialize expected ground-truth labels by majority voting (MV)
- repeat:
  - ▶ Estimate **confusion matrices**: MLE given **expected ground truth labels**
  - ▶ Estimate **expected ground-truth labels**: MLE given **confusion matrices**

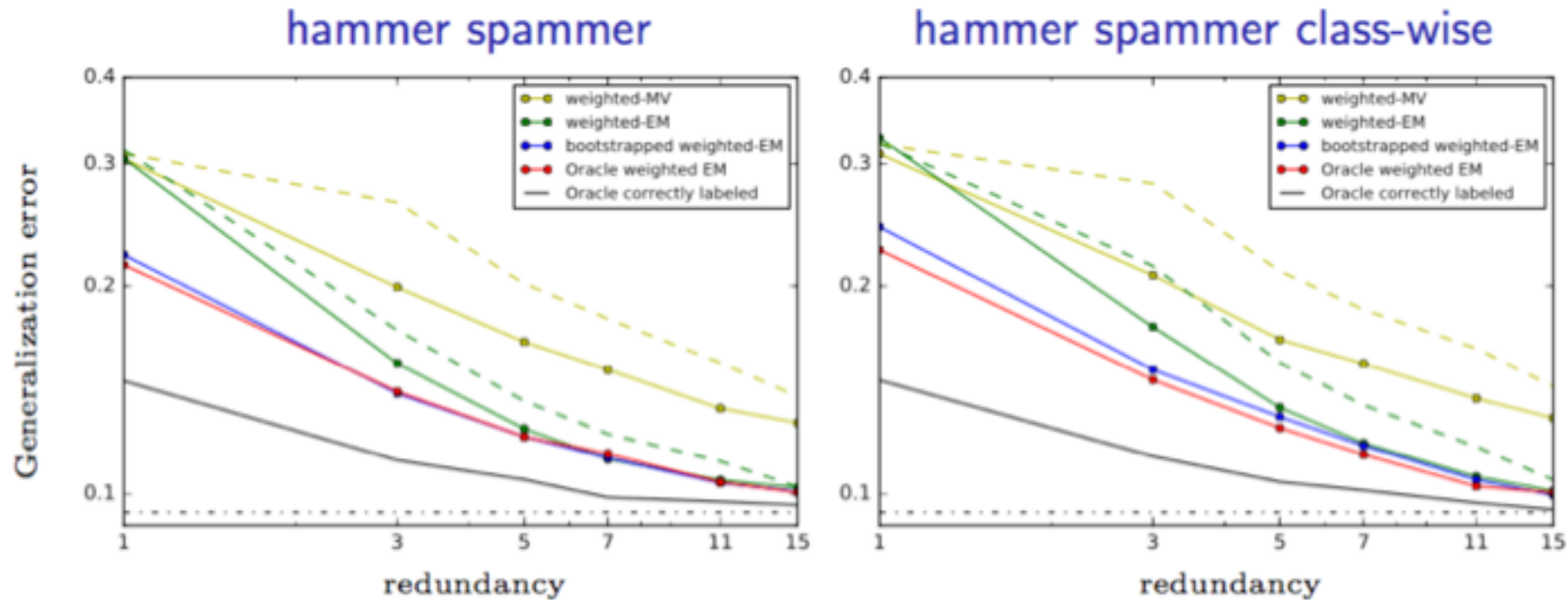
Concretely: Estimate confusion matrices



# Bootstrapping EM

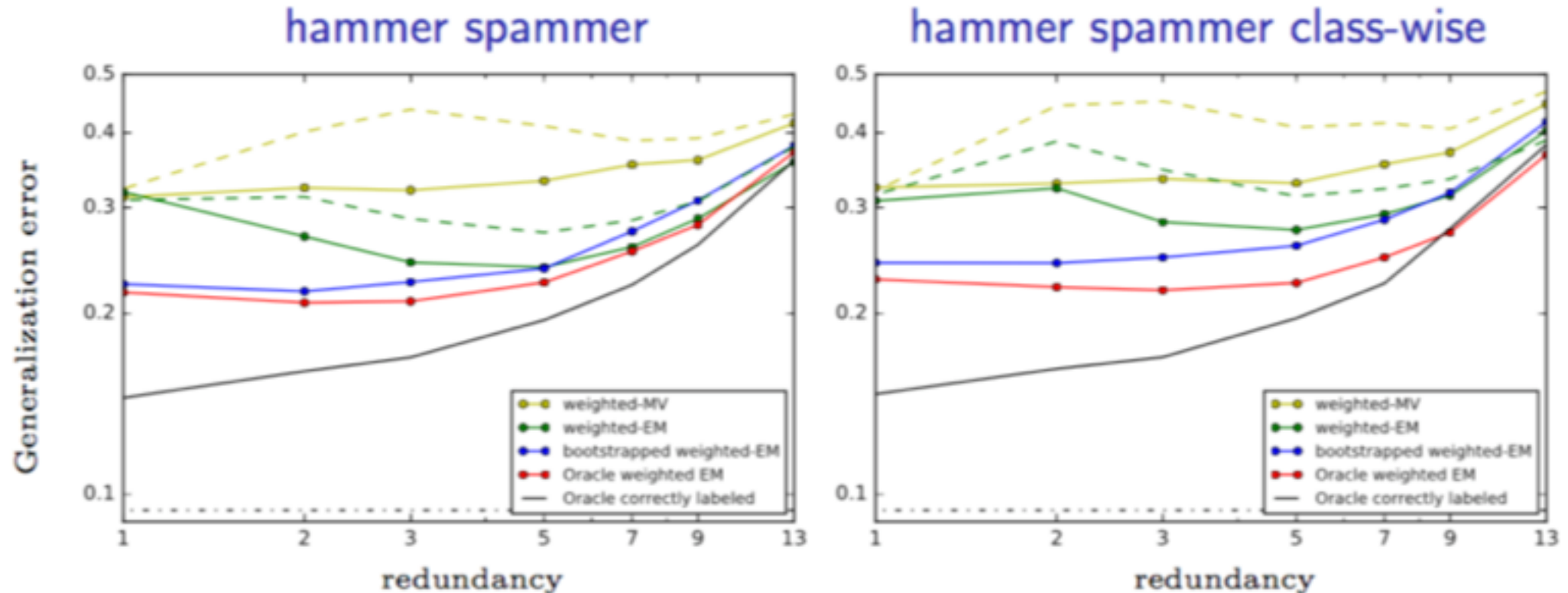
- **Insight:** Learned model agrees with workers more when they are right
- Learning algorithm:
  1. Aggregate labels by weighted majority (no-op if singly labeled)
  2. Train model
  3. Use predictions to estimate worker confusion matrices
  4. Given the estimated confusion matrices, retrain model with probability-weighted loss function

# CIFAR10 Results – Varying Redundancy



# Fixed Annotation Budget (CIFAR10)

## Label Once and Move On!

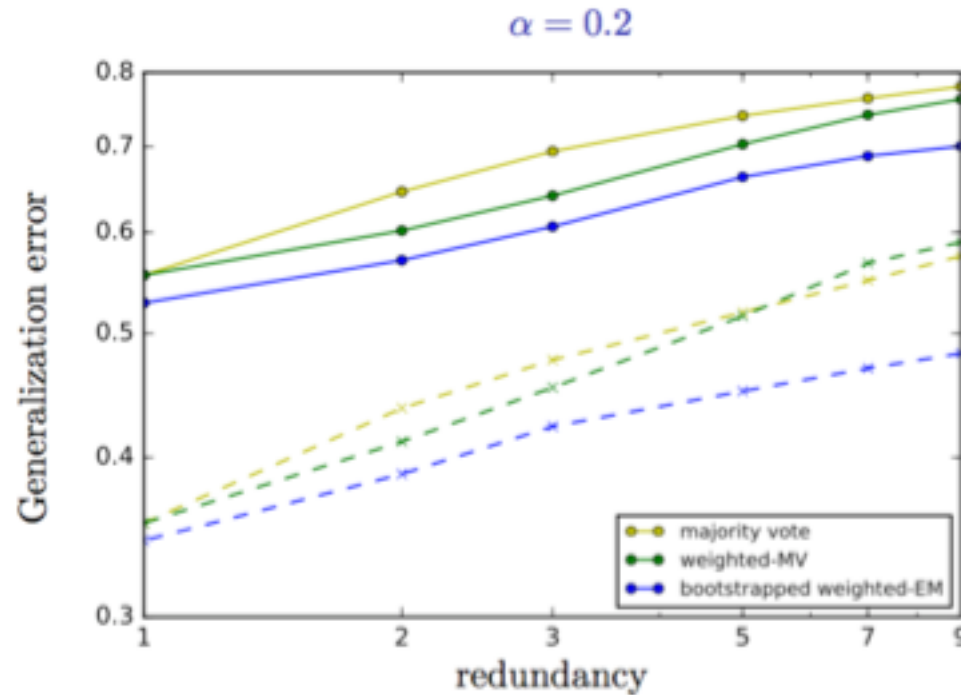


Probability-weighted EM: Optimal redundancy 3/5

Probability-weighted bootstrapped EM: Optimal redundancy 1

# Fixed annotation budget (ImageNet): label once and move on!

Fixed annotation budget  
hammer spammer class-wise



Probability-weighted bootstrapped EM: Optimal redundancy 1

# Efficient Exploration for Dialogue Policy Learning w. BBQ-Networks

Zachary C. Lipton, Xiujun Li, Lihong Li, Jiangeng Gao, Faisal Ahmed, Li Deng

<https://arxiv.org/abs/1608.05081>

# Chatbots



InfoBot

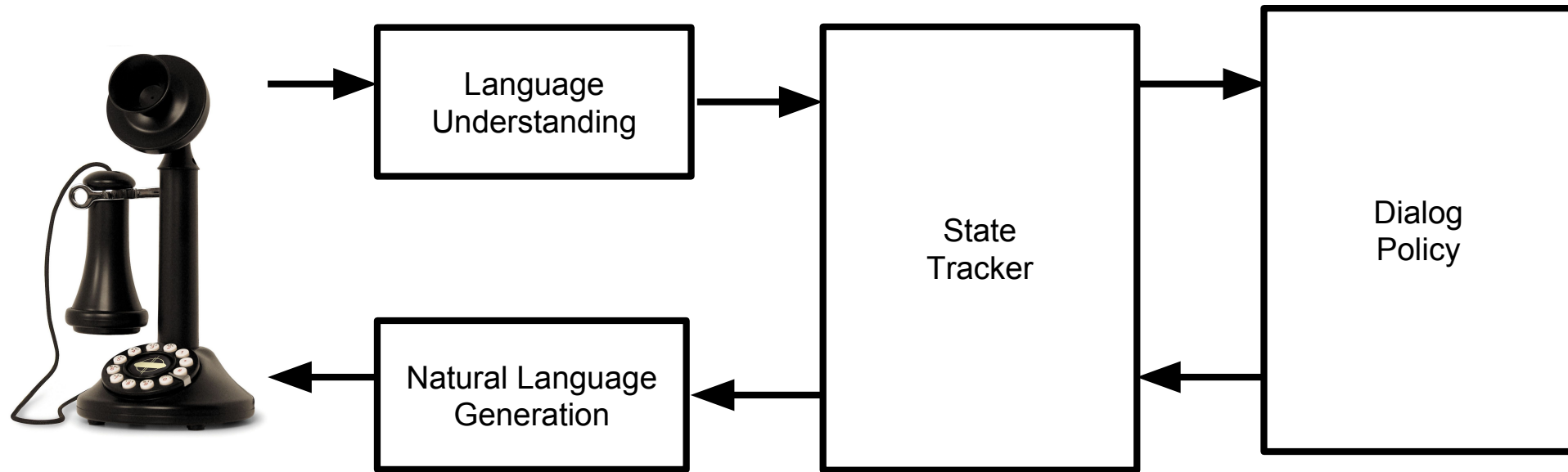


Task Completion



Chit-Chat

# Typical dialogue system architecture



# Dialogue-Act Representations

- **Semantic representation of dialog utterances:**

Agent: greeting()

User: request(ticket, numberofpeople=2)

Agent: request(city)

User: inform(city=Seattle)

Agent: request(genre)

User: inform(action)

Agent: inform(multiplechoice={...})

User: inform(moviename=Our Kind of Traitor)

Agent: inform(taskcomplete, theater=Cinemark Lncln Sq)

User: thanks()

- **Mapping from-to NLP handled by LU and NLG components**

# Deep Reinforcement Learning



Agent



Environment



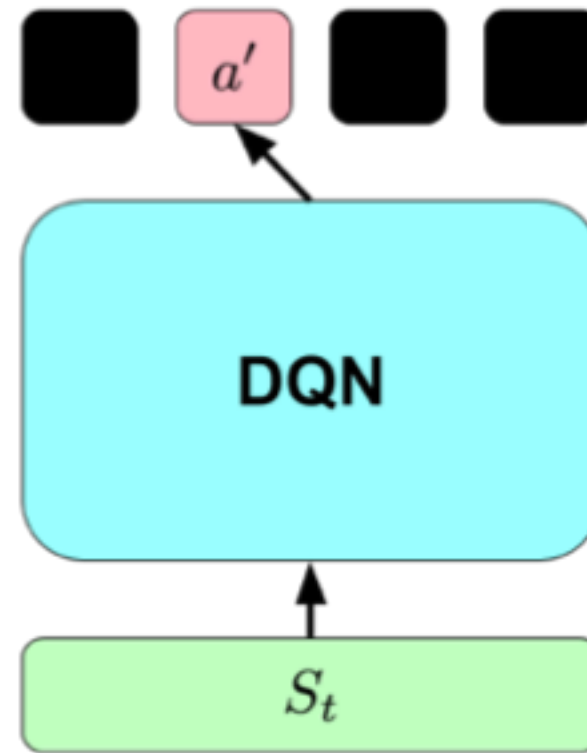
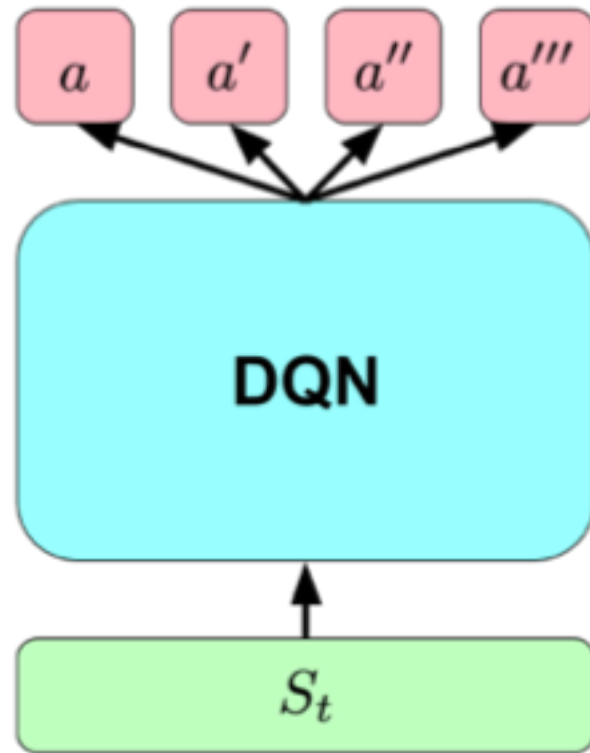
Actions



Rewards

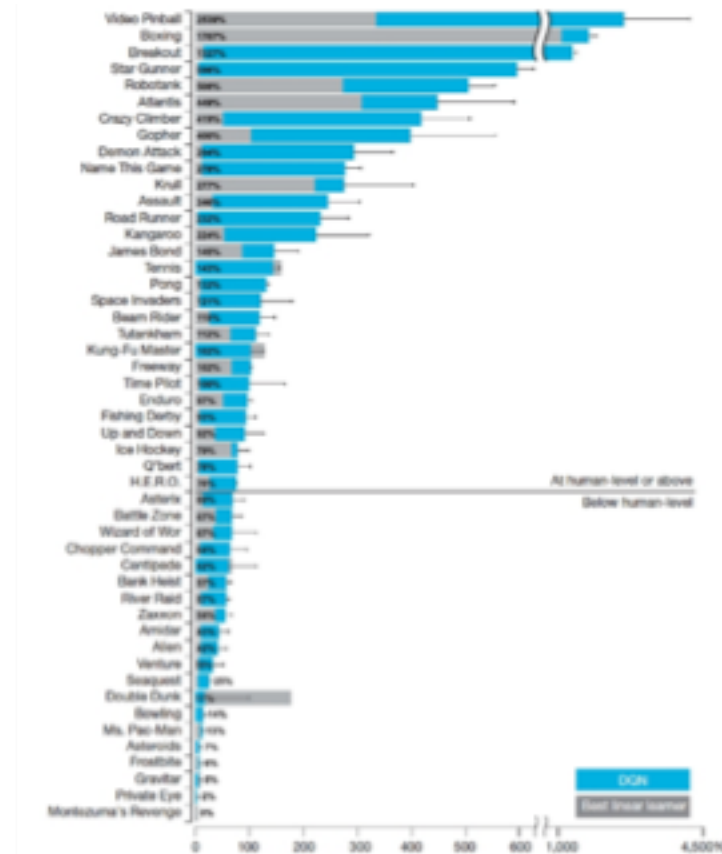
# Deep Q-Networks

For problems with many states and actions, must approximate Q function



# DQNs are awesome at games ....

... but take squillions of interactions to train.



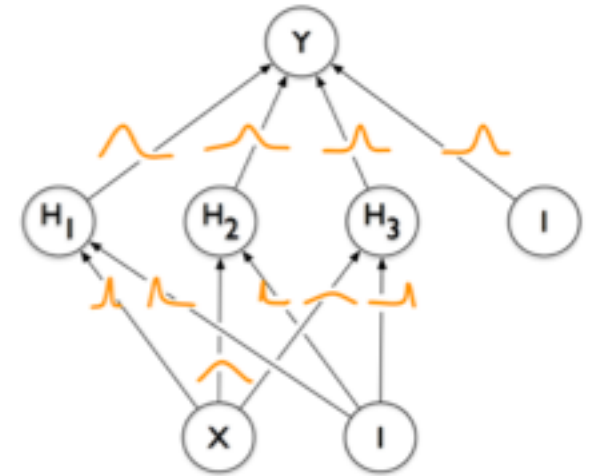
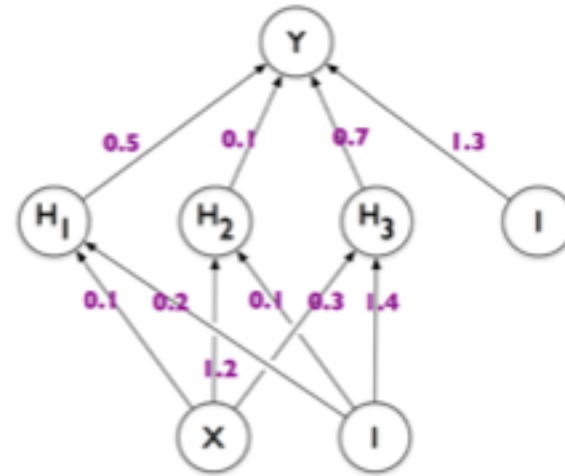
# Thompson Sampling

- Alternative to ( $\epsilon$ -greedy)
- Choose each action according to **probability** that it's the best
- Requires estimating **uncertainty**
- **Conundrum:**
  - Neural networks get **best predictions**, want to use them for estimating Q
  - OTOH, other approaches better-established for **estimating uncertainty**
- **Solution:**
  - Extract uncertainty estimates from neural networks

# BBQ-Networks

- **At train time:**

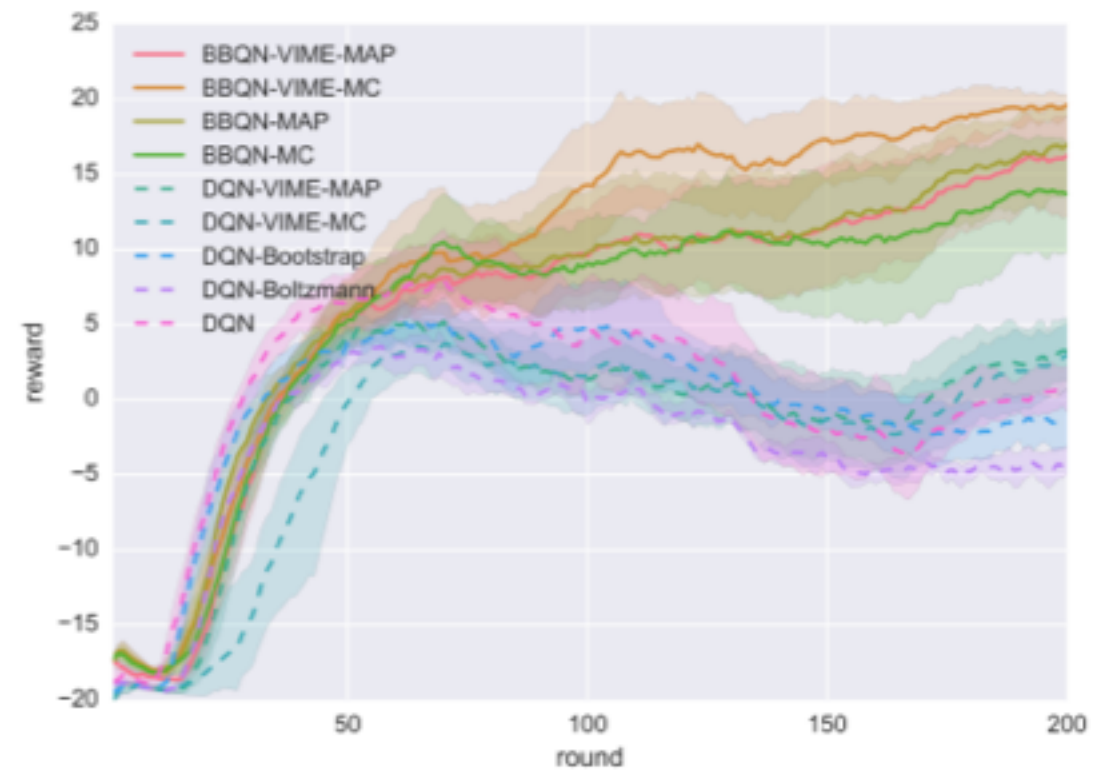
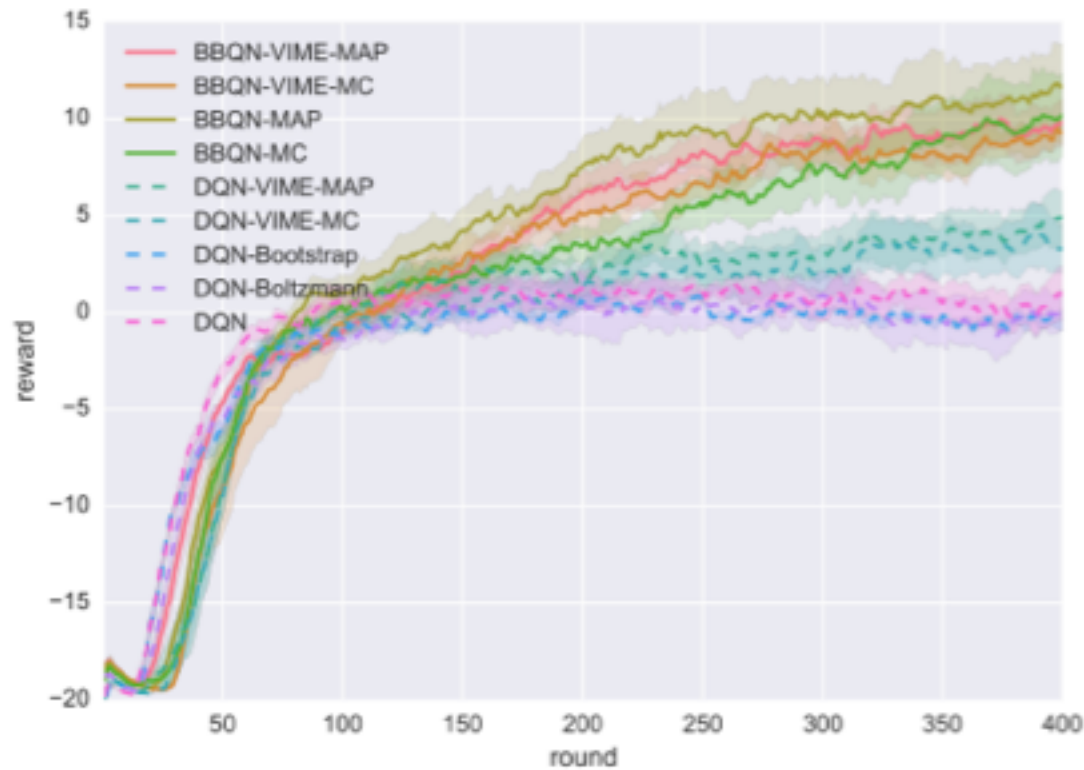
1. Sample weights from  $q(w)$
2. Make forward pass
3. Generate TD target using MAP estimate from target network
4. Update parameters with reparameterisation trick



- **Action time**

1. Sample weights from  $q(w)$
2. Choose best action for that sample (Thompson sampling)

# Results for static (left) & domain extension (right)



# Thanks!

- **Stay in touch**

Interested in this work? Let's talk!

- **Contact**

[zlipton@cmu.edu](mailto:zlipton@cmu.edu)

- **Papers**

[Deep Active Learning for NER](#) (ICLR 2018)

Deep Bayesian Active Learning for NLP (forthcoming)

[How Transferable are the Active Sets](#) (arXiv 2018)

[Active Learning with Partial Feedback](#) (arXiv 2018)

[Learning from noisy Singly-Labeled Data](#) (ICLR 2018)

[BBO-networks](#) (AAAI 2018)

- **Acknowledgments**

Yanyao Shen Hyokun Yun, Ashish Khetan, Anima Anandkumar, Xiujun Li, Lihong Li, Jianfeng Gao, Li Deng, Peiyun Hu, Aditya Siddhant, David Lowell, Byron Wallace