10-301/10-601 Introduction to Machine Learning

Machine Learning Department
School of Computer Science
Carnegie Mellon University

In-context Learning

+

Reinforcement Learning:
Markov Decision Processes

Pat Virtue & Matt Gormley
Lecture 20
Mar. 25, 2026

PRE-TRAINING VS. FINE-TUNING

The Start of Deep Learning

- The architectures of modern deep learning have a long history:
  - 1960s: Rosenblatt's 3-layer multi-layer perceptron, ReLU)
  - 1970-80s: RNNs and CNNs
  - 1990s: linearized self-attention
- The spark for deep learning came in 2006 thanks to **pre-training** (e.g., Hinton & Salakhutdinov, 2006)

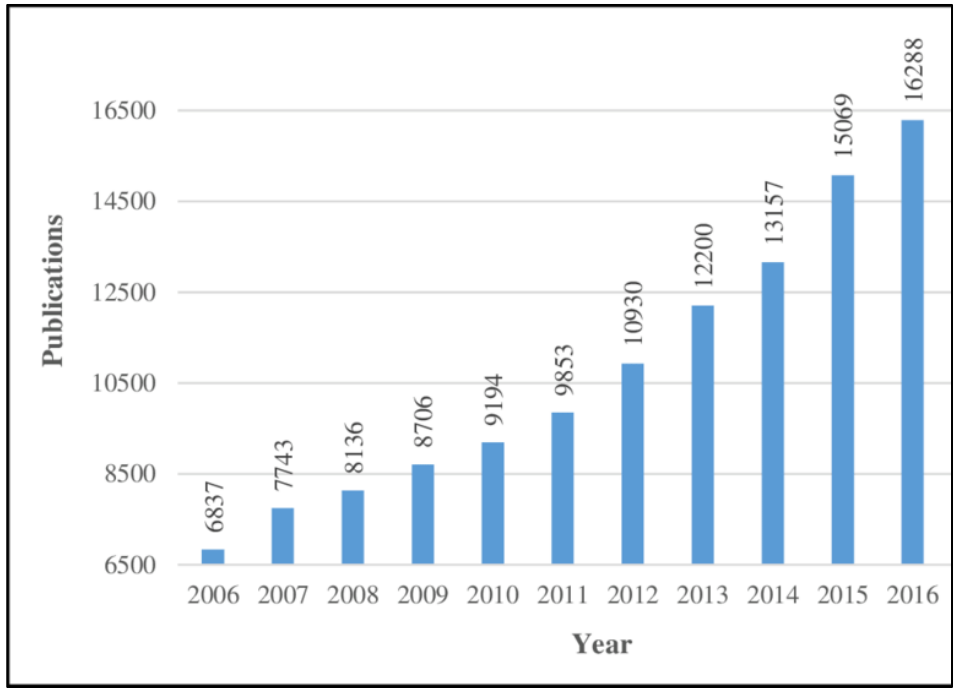
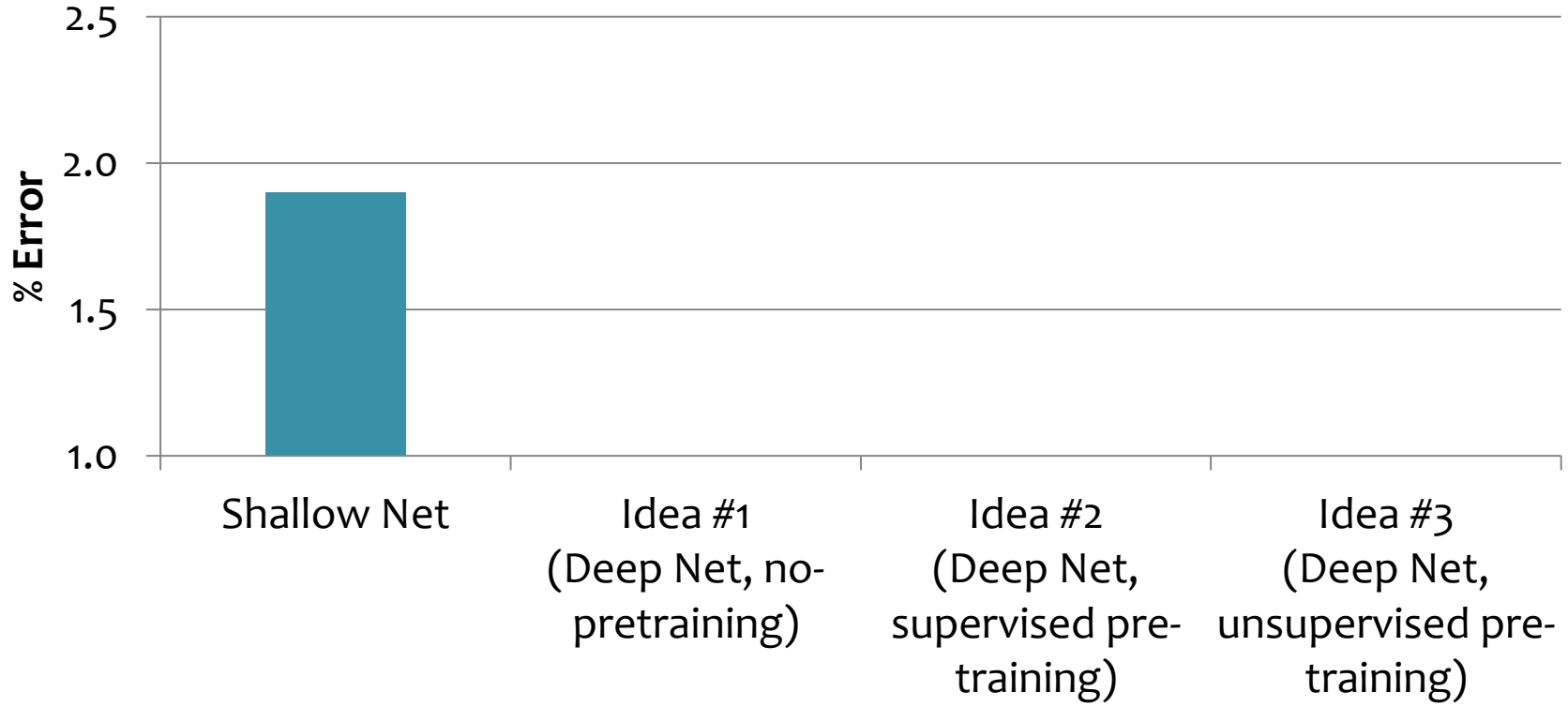


Figure from Vargas et al. (2017)

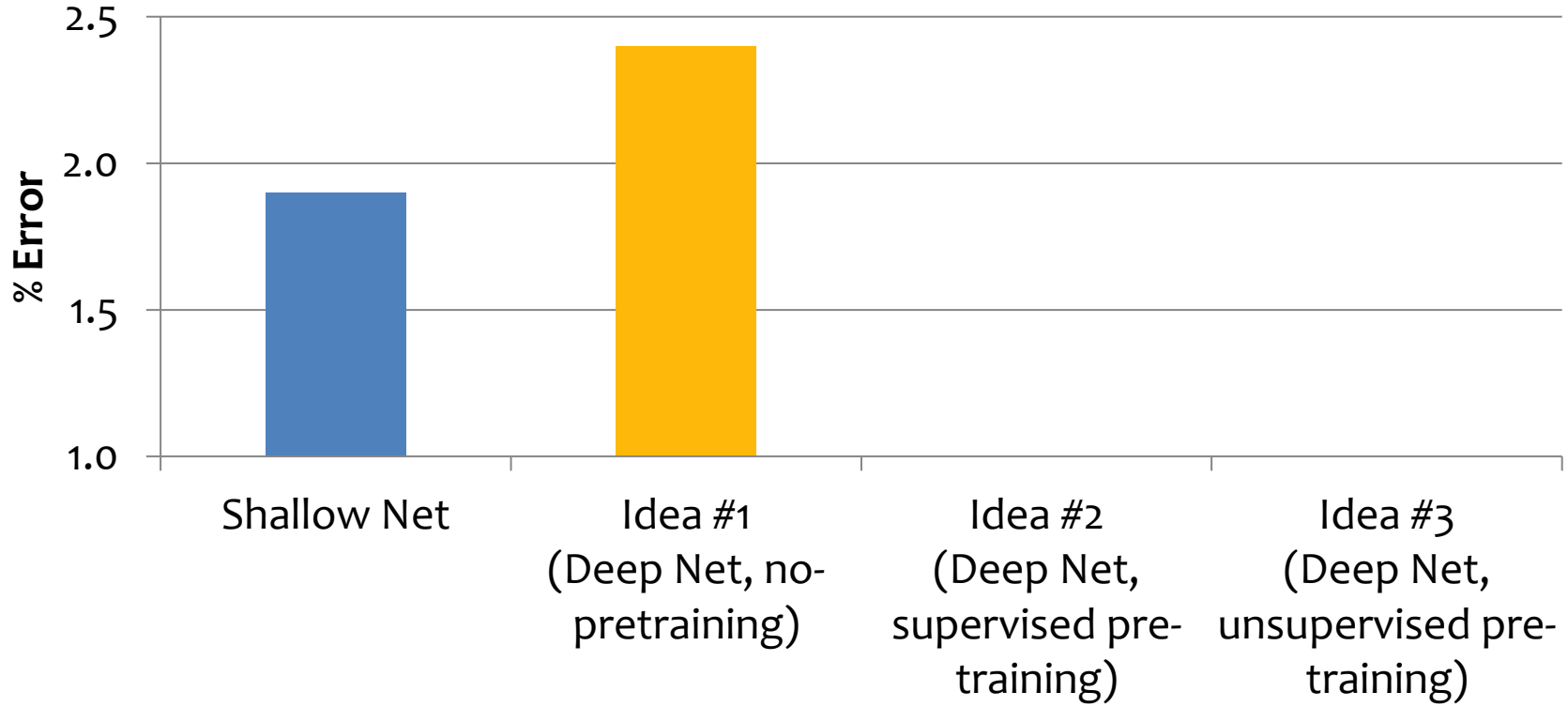
Pre-Training and Fine-Tuning on MNIST

- Results from Bengio et al. (2006) on MNIST digit classification task
- Percent error (lower is better)
- Some methods first do pre-training
- Every method includes fine-tuning on labeled data



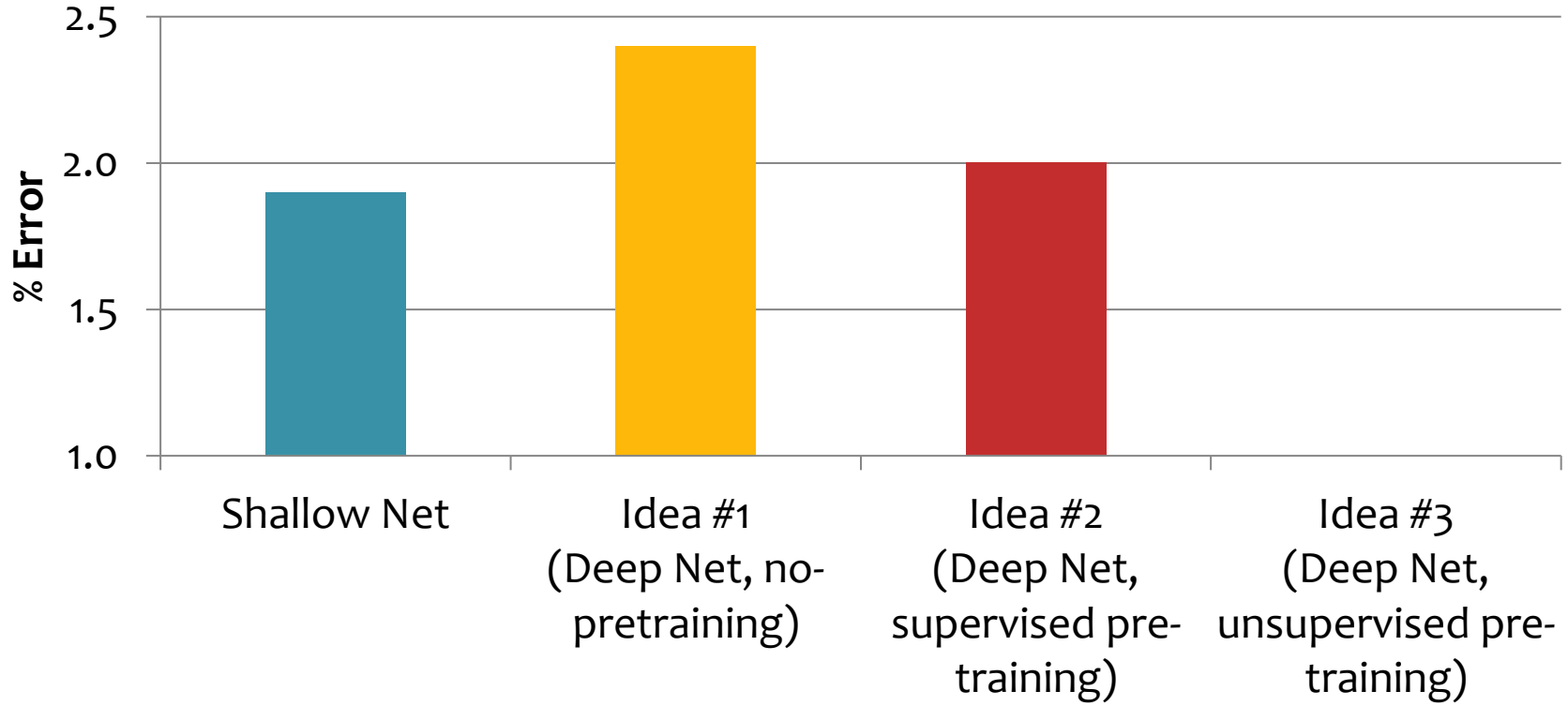
Pre-Training and Fine-Tuning on MNIST

- Results from Bengio et al. (2006) on MNIST digit classification task
- Percent error (lower is better)
- Some methods first do pre-training
- Every method includes fine-tuning on labeled data



Pre-Training and Fine-Tuning on MNIST

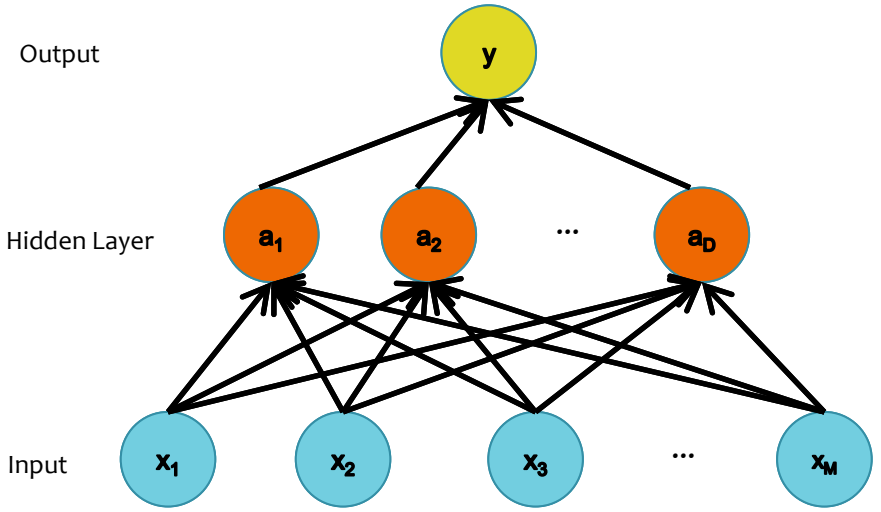
- Results from Bengio et al. (2006) on MNIST digit classification task
- Percent error (lower is better)
- Some methods first do pre-training
- Every method includes fine-tuning on labeled data



Unsupervised Autoencoder Pre-Training for Vision

Unsupervised pre-training of the first layer:

- What should it predict?
- What else do we observe?
- **The input!**

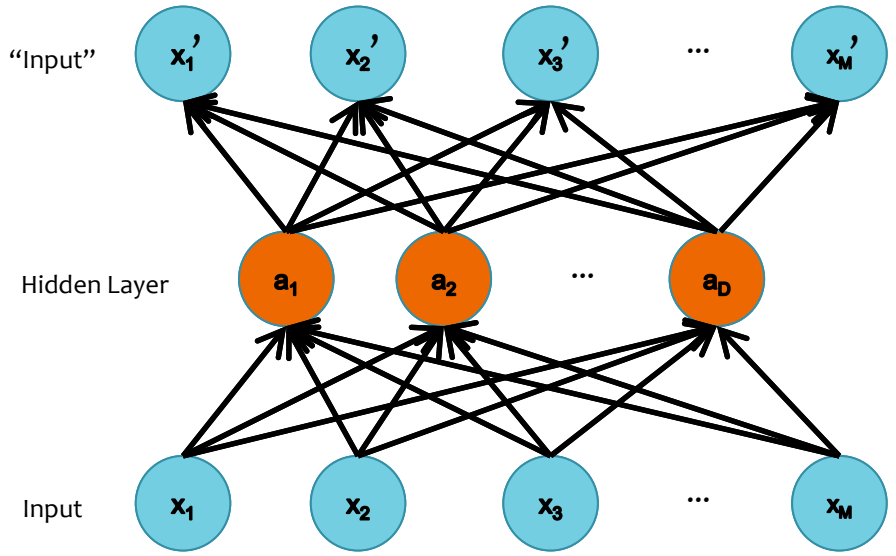


Unsupervised Autoencoder Pre-Training for Vision

Unsupervised pre-training of the first layer:

- What should it predict?
- What else do we observe?
- **The input!**

This topology defines an Auto-encoder.



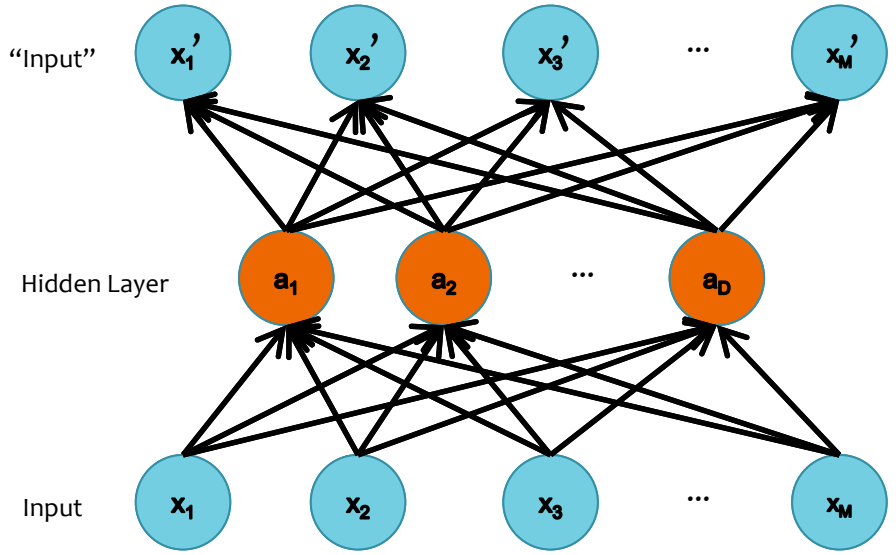
Unsupervised Autoencoder Pre-Training for Vision

Key idea: Encourage z to give small reconstruction error:

- x' is the *reconstruction* of x
- Loss = $|| x - \text{DECODER}(\text{ENCODER}(x)) ||^2$
- Train with the same backpropagation algorithm for 2-layer Neural Networks with x_m as both input and output.

DECODER: $x' = h(W'z)$

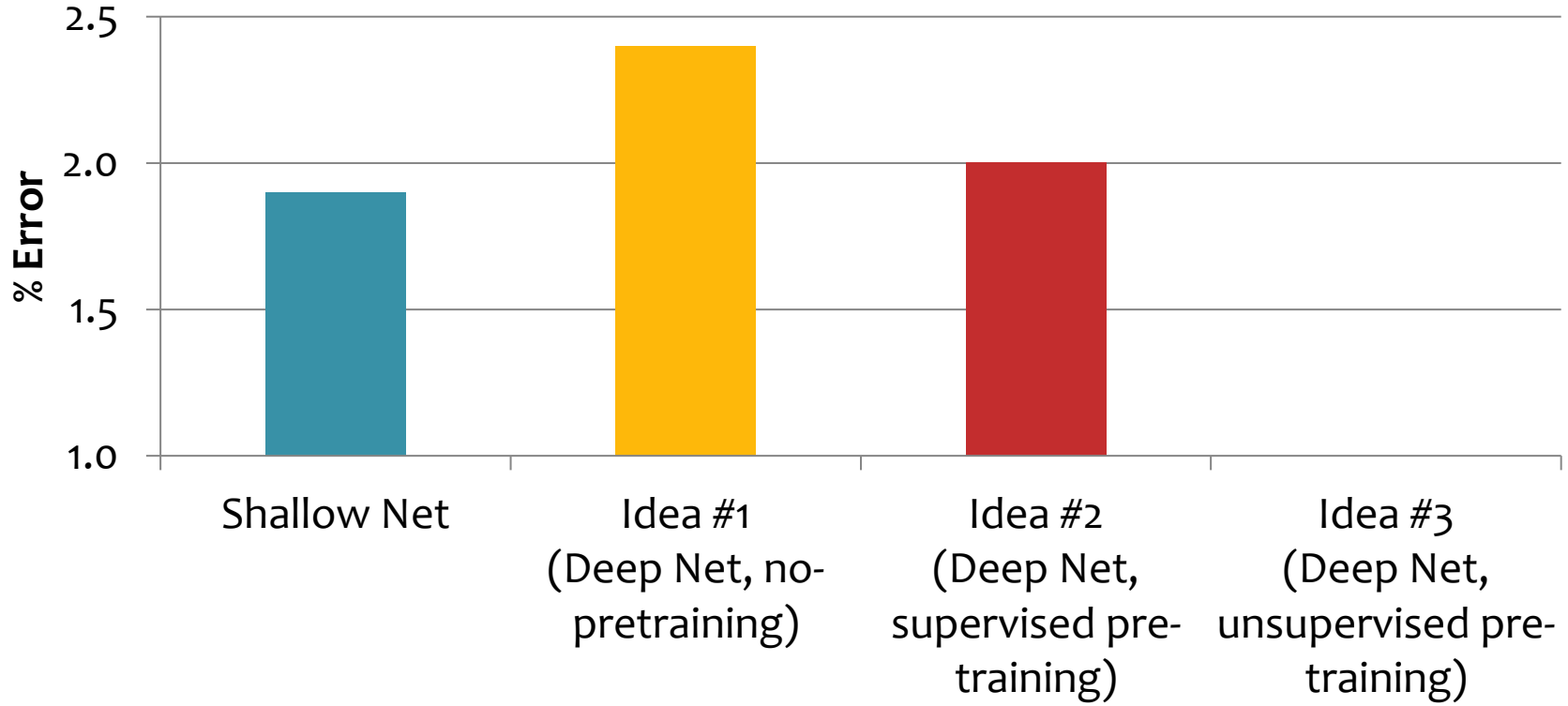
ENCODER: $z = h(Wx)$



Slide adapted from Raman Arora

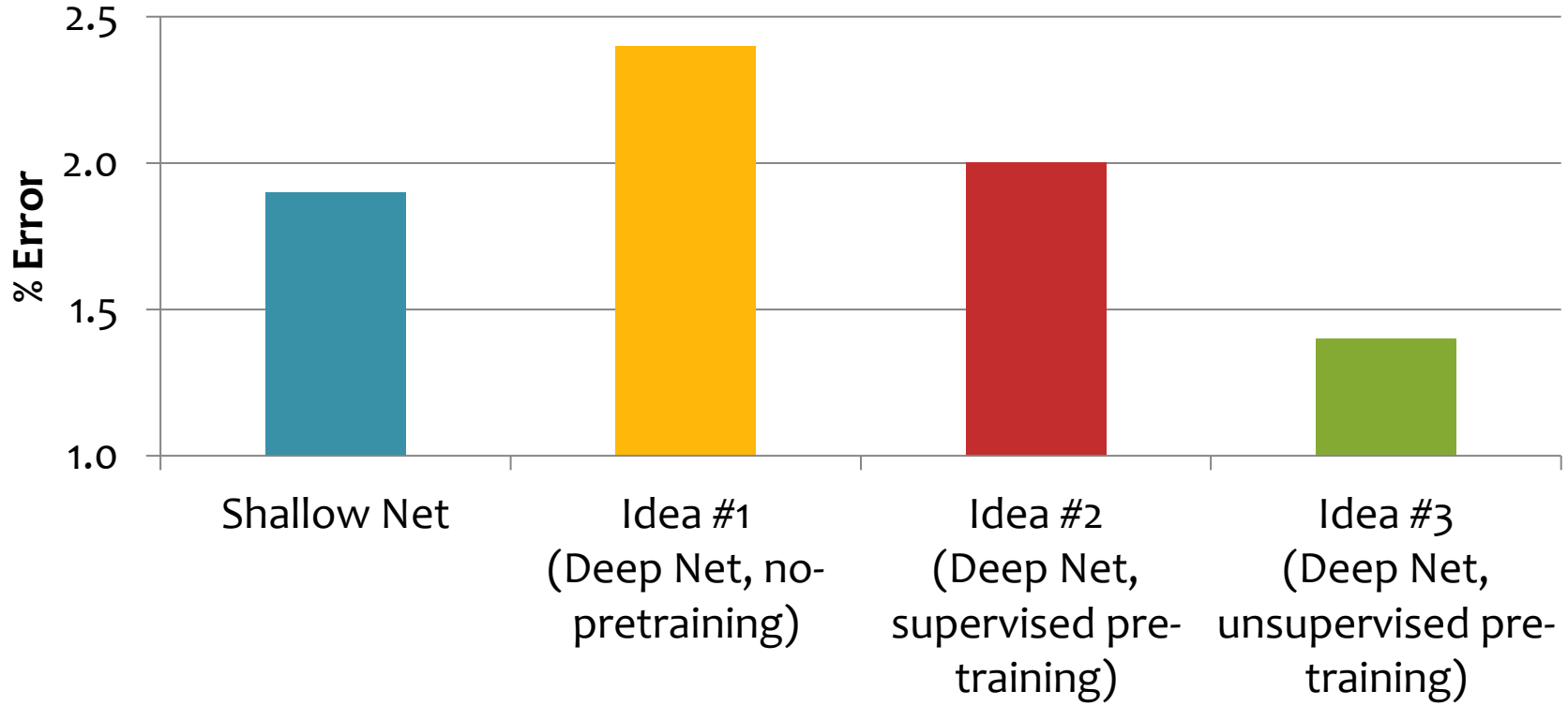
Pre-Training and Fine-Tuning on MNIST

- Results from Bengio et al. (2006) on MNIST digit classification task
- Percent error (lower is better)
- Some methods first do pre-training
- Every method includes fine-tuning on labeled data



Pre-Training and Fine-Tuning on MNIST

- Results from Bengio et al. (2006) on MNIST digit classification task
- Percent error (lower is better)
- Some methods first do pre-training
- Every method includes fine-tuning on labeled data



Supervised Pre-Training for Vision

- Nowadays, we tend to just do supervised pre-training on a massive labeled dataset
- Vision Transformer's success was largely due to using a much larger pre-training dataset

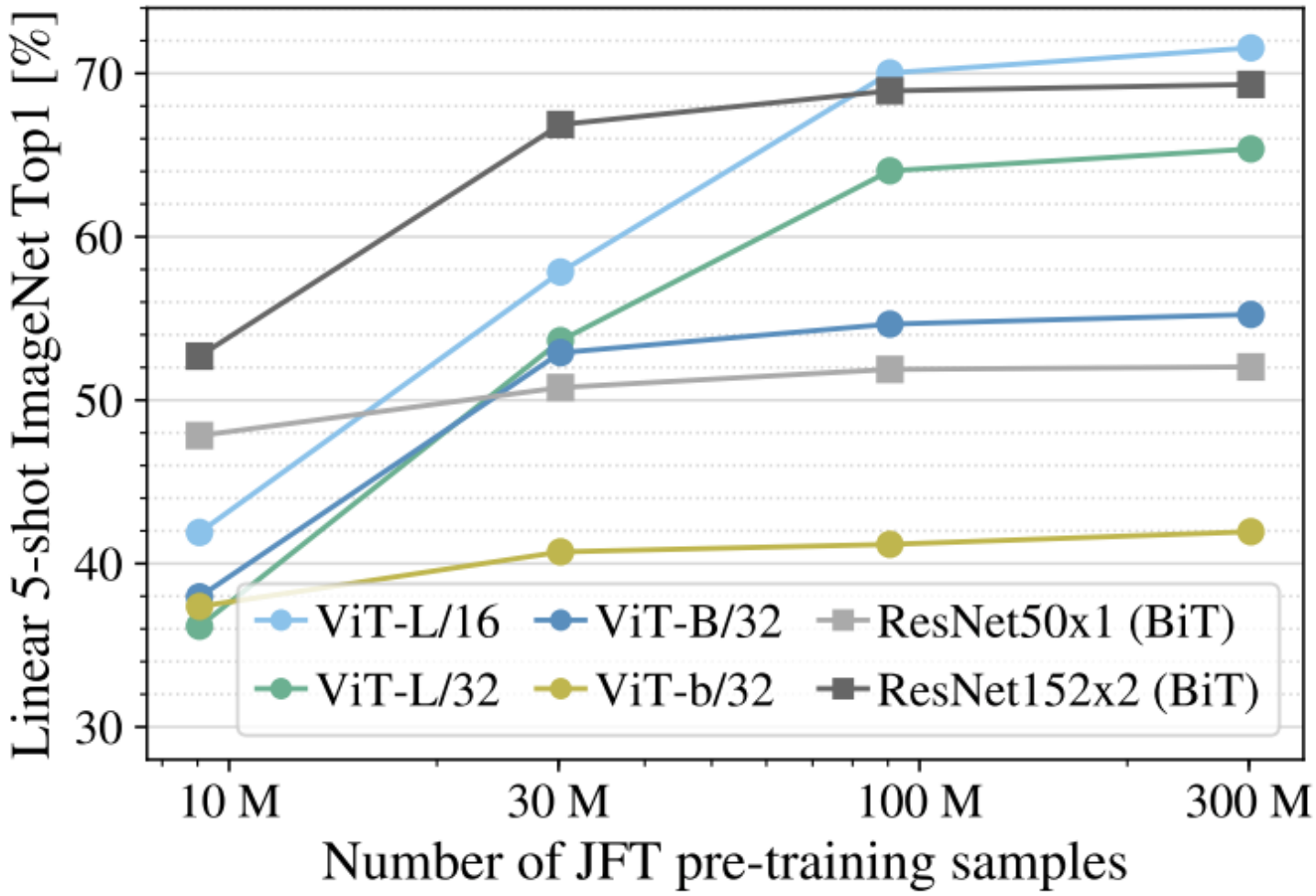
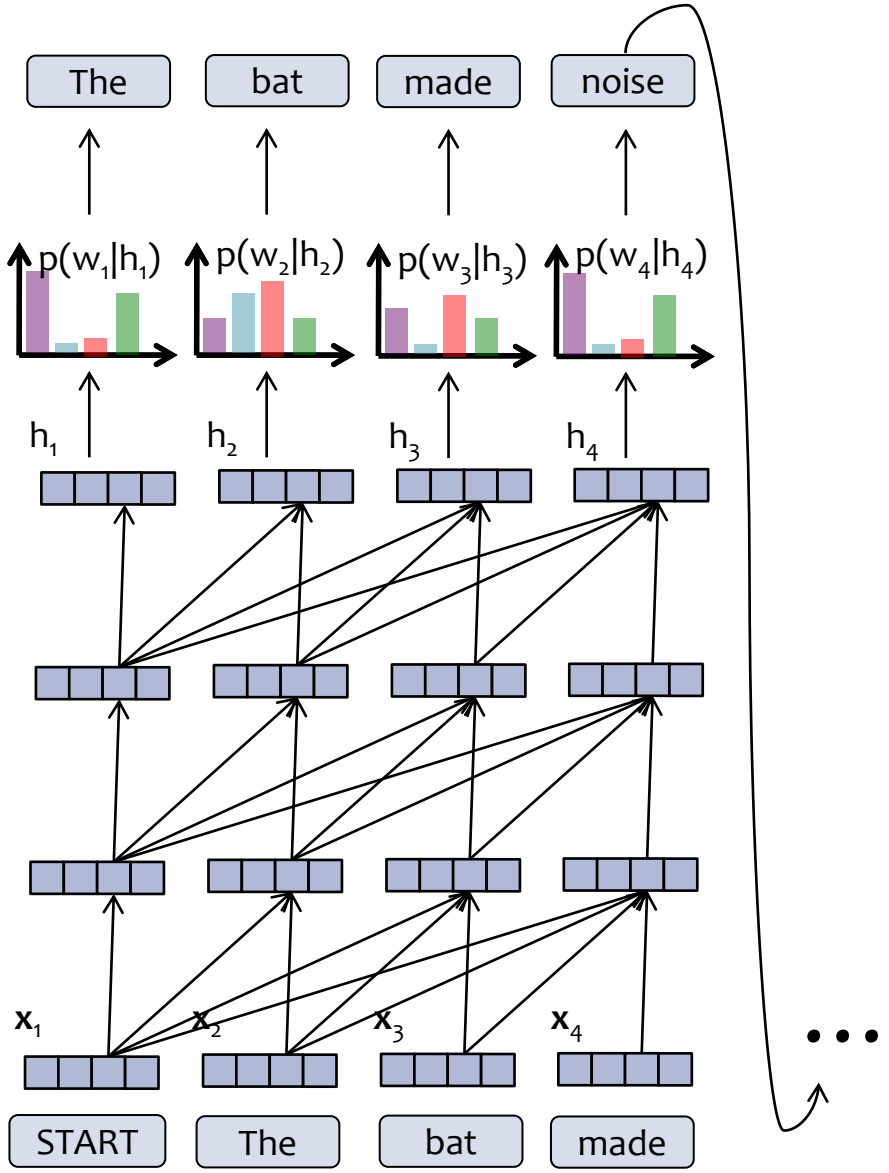


Figure from https://arxiv.org/pdf/2010.11929

Unsupervised Pre-Training for an LLM



Generative pre-training for a deep language model:

- each training example is an (unlabeled) sentence
- the objective function is the likelihood of the observed sentence

Practically, we can **batch** together many such training examples to make training more efficient

Training Data for LLMs

GPT-3 Training Data:

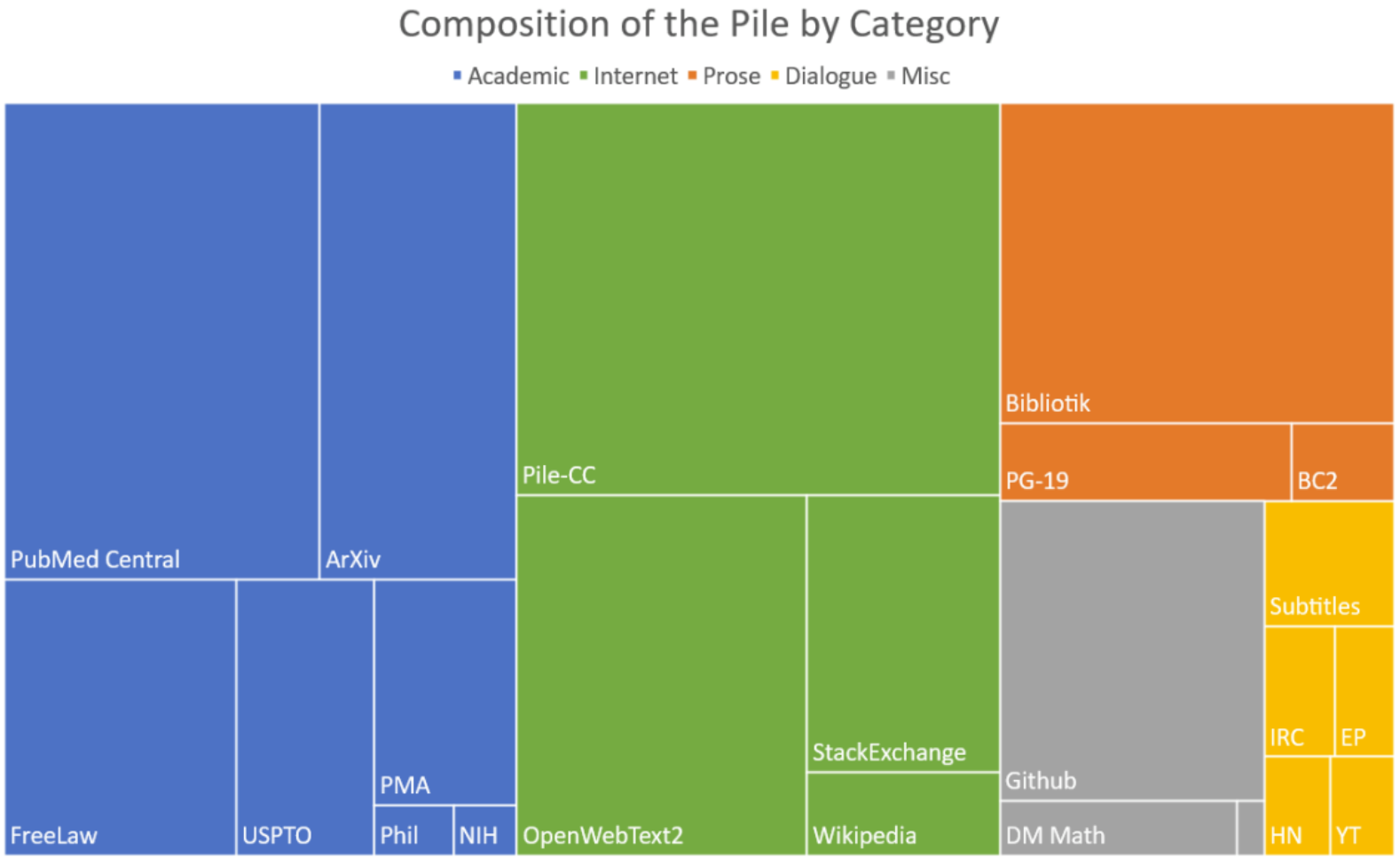
Dataset	Quantity (tokens)	Weight in training mix	Epochs elapsed when training for 300B tokens
Common Crawl (filtered)	410 billion	60%	0.44
WebText2	19 billion	22%	2.9
Books1	12 billion	8%	1.9
Books2	55 billion	8%	0.43
Wikipedia	3 billion	3%	3.4

Table from http://arxiv.org/abs/2005.14165

Training Data for LLMs

The Pile:

- An open source dataset for training language models
- Comprised of 22 smaller datasets
- Favors high quality text
- 825 Gb ≈ 1.2 trillion tokens



Pre-Training vs. Fine-Tuning

Definitions

Pre-training

- randomly initialize the parameters, then…
- *option A*: unsupervised training on very large set of unlabeled instances
- *option B*: supervised training on a very large set of labeled examples

Fine-tuning

- initialize parameters to values from pre-training
- (optionally), add a prediction head with a small number of randomly initialized parameters
- train on a specific task of interest by backprop

Example: Vision Models

Pre-training

- Example A: unsupervised autoencoder training on very large set of unlabeled images (e.g. MNIST digits)
- Example B: supervised training on a very large image classification dataset (e.g. ImageNet w/21k classes and 14M images)

Fine-tuning

- object detection, training on 200k labeled images from COCO
- semantic segmentation, training on 20k labeled images from ADE20k

Example: Language Models

Pre-training

- unsupervised pre-training by maximizing likelihood of a large set of unlabeled sentences such as…
- The Pile (800 Gb of text)
- Dolma (3 trillion tokens)

Fine-tuning

- MMLU benchmark: a few training examples from 57 different tasks ranging from elementary mathematics to genetics to law
- code generation, training on ~400 training examples from MBPP

POST TRAINING

RLHF

- Modern LLMs are not just fine-tuned, they are a variety of posting training stages that occur
- One of the most important is that of alignment, which can be done with algorithms like RLHF and DPO
- RLHF is a form of fine-tuning that uses *reinforcement learning* where the reward function is learned from human preferences

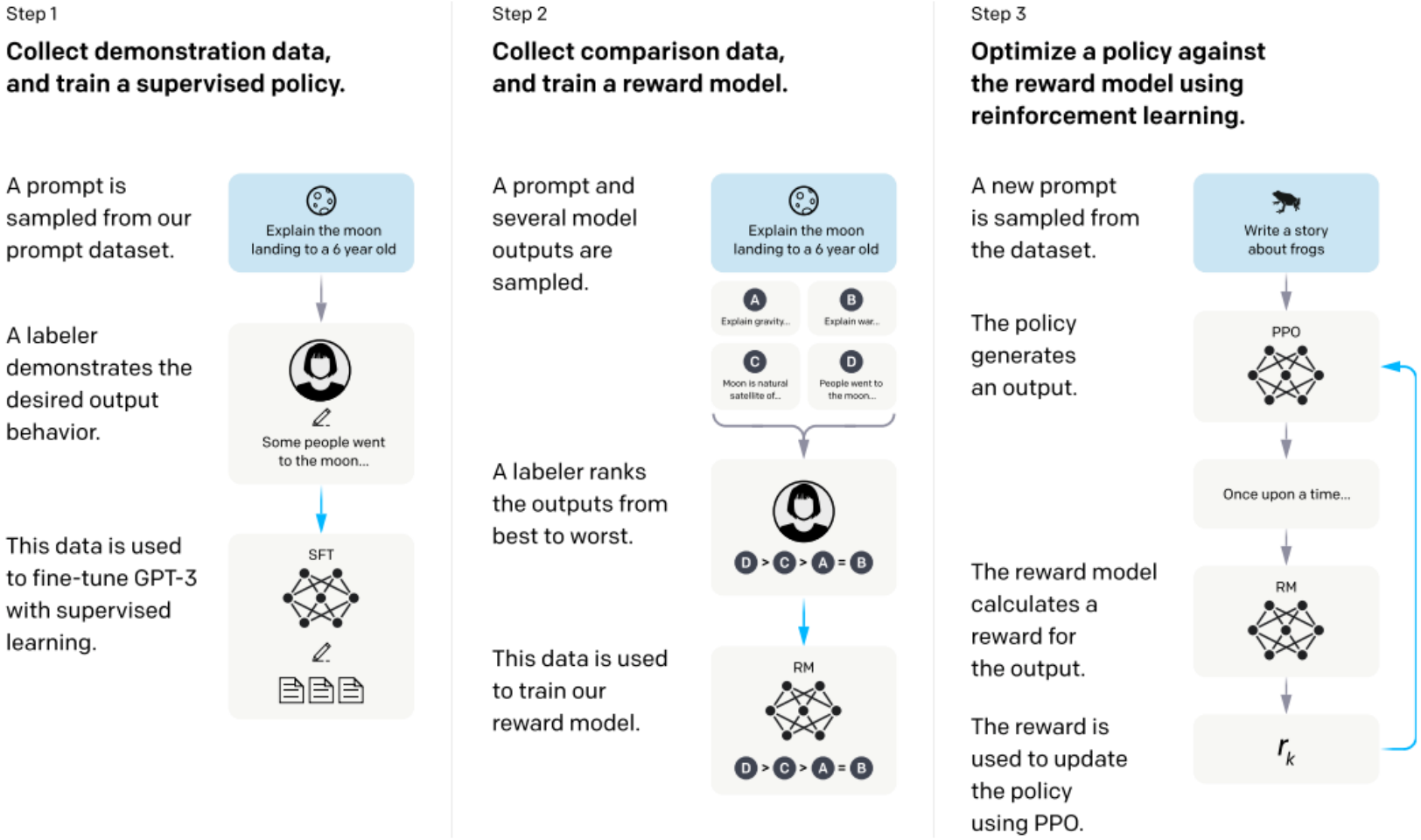


Figure from https://arxiv.org/pdf/2203.02155.pdf

PROMPTING

Zero-shot vs. Few-Shot

- ***Definition***: in **zero-shot learning** we assume that training data does not contain any examples of the labels that appear in the test data
- ***Definition***: in **few-shot learning** we assume that training data contains a handful (maybe two, three, or four) examples of each label

Prompting

- Language models are trained to maximize the likelihood of sequences in the training data
- Most condition on the previous tokens to generate the next tokens

Definition: An *autoregressive language model* defines a probability distribution over sequences $\mathbf{x}_{1:T}$ of the form:

$$p(\mathbf{x}_{1:T}) = \prod_{t=1}^{T} p(x_t \mid x_1, \ldots, x_{t-1})$$

- Key idea behind prompting: provide a prefix string to the such that its likely completion is the answer you want

text completion

Prompting Examples

GPT-3

Prompt (i.e. context)

```
The City
BY C. P. CAVAFY
TRANSLATED BY EDMUND KEELEY
[Poem text omitted]


SOME TREES
John Ashbery
[Poem text omitted]


Shadows on the Way
Wallace Stevens
```

Output (i.e. generated poem)

```
I must have shadows on the way
If I am to walk I must have
Each step taken slowly and alone
To have it ready made

And I must think in lines of grey
To have dim thoughts to be my guide
Must look on blue and green
And never let my eye forget
That color is my friend
And purple must surround me too

The yellow of the sun is no more
Intrusive than the bluish snow
That falls on all of us.  I must have
Grey thoughts and blue thoughts walk with me
If I am to go away at all.
```

Conditional Sampling:

To draw a sample, feed in a prompt (i.e. context) and then sample from the model conditioned on that prompt

Examples from GPT-3 paper

translation

Prompting Examples

Llama-2-70B

Prompt

El último dueño de esta propiedad había sido un hombre soltero, que alcanzó una muy avanzada edad, y que durante gran parte de su existencia tuvo en su hermana una fiel compañera y ama de casa.

English translation:

Conditional Sampling:

To draw a sample, feed in a prompt (i.e. context) and then sample from the model conditioned on that prompt

Model Output

The last owner of this property was a bachelor man who lived to a very advanced age, and who had his sister as a faithful companion and housekeeper for most of his life.

English source from Sense & Sensibility

The late owner of this estate was a single man, who lived to a very advanced age, and who for many years of his life, had a constant companion and housekeeper in his sister.

Text from https://huggingface.co/datasets/opus_books/viewer/en-es

question answering (entailment)

Prompting Examples

Llama-2-70B

Prompt

Rhodochiton is a genus of flowering plants within the family Plantaginaceae, native to southern Mexico and neighbouring Guatemala. They climb by means of twining leaf stalks. One of the three species, "Rhodochiton atrosanguineus", the purple bell vine, is grown as an ornamental plant. All three species are sometimes included in "Lophospermum".

Question: You can find the purple bell vine in more than one country. True, False, or Neither?

Model Output

Answer: True.

Explanation: The purple bell vine (Rhodochiton atrosanguineus) is native to southern Mexico and neighbouring Guatemala, which means it can be found in more than one country.

Correct Answer

True.

Text from https://huggingface.co/datasets/facebook/anli

summarization

Prompting Examples

Llama-2-70B

Prompt

Story: Jason listened to the weather and heard it was goingto be sunny. He thought the kids might like to goswimming. He gathered up the swimsuits, towelsand sunscreen. Jason and the kids got into the truckand drove to the beach. They spent the next 2 hoursplaying and splashing in the surf.

One-sentence Summary:

Model Output

Jason took the kids swimming at the beach after hearing the weather forecast, gathering necessary items and driving there.

Story text from https://arxiv.org/pdf/1910.00998.pdf

What else can a large LM (attempt to) do?

Using the idea of **prompts**, we can apply LMs to a **variety of different problems** in natural language processing.

In the **zero-shot setting**, we simply feed the context to the model and observe how it completes the sequence. (i.e. there is no additional training)

Answer fact-based questions:

Context →	Organisms require energy in order to do what?
Correct Answer →	mature and develop.
Incorrect Answer →	rest soundly.
Incorrect Answer →	absorb light.
Incorrect Answer →	take in nutrients.

Complete sentences logically:

Context →	My body cast a shadow over the grass because
Correct Answer →	the sun was rising.
Incorrect Answer →	the grass was cut.

Complete analogies:

Context →	lull is to trust as
Correct Answer →	cajole is to compliance
Incorrect Answer →	balk is to fortitude
Incorrect Answer →	betray is to loyalty
Incorrect Answer →	hinder is to destination
Incorrect Answer →	soothe is to passion

Reading comprehension:

Context →	anli 1: anli 1: Fulton James MacGregor MSP is a Scottish politician who is a Scottish National Party (SNP) Member of Scottish Parliament for the constituency of Coatbridge and Chryston. MacGregor is currently Parliamentary Liaison Officer to Shona Robison, Cabinet Secretary for Health & Sport. He also serves on the Justice and Education & Skills committees in the Scottish Parliament. Question: Fulton James MacGregor is a Scottish politican who is a Liaison officer to Shona Robison who he swears is his best friend. True, False, or Neither?
Correct Answer →	Neither
Incorrect Answer →	True
Incorrect Answer →	False

Examples from GPT-3

Zero-shot LLMs

- GPT-2 (1.5B parameters) for unsupervised prediction on various tasks
- GPT-2 models p(output | input, task)
  - translation: *(translate to french, english text, french text)*
  - reading comprehension: *(answer the question, document, question, answer)*
- Why does this work?

"I'm not the cleverest man in the world, but like they say in French: **Je ne suis pas un imbecile [I'm not a fool].**
In a now-deleted post from Aug. 16, Soheil Eid, Tory candidate in the riding of Joliette, wrote in French: "**Mentez mentez, il en restera toujours quelque chose**," which translates as, "**Lie lie and something will always remain.**"
"I hate the word '**perfume**,"' Burr says. 'It's somewhat better in French: '**parfum**.'
If listened carefully at 29:55, a conversation can be heard between two guys in French: "**-Comment on fait pour aller de l'autre coté? -Quel autre coté?**", which means "**- How do you get to the other side? - What side?**".
If this sounds like a bit of a stretch, consider this question in French: **As-tu aller au cinéma?**, or **Did you go to the movies?**, which literally translates as Have-you to go to movies/theater?
"**Brevet Sans Garantie Du Gouvernement**", translated to English: "**Patented without government warranty**".

Table 1. Examples of naturally occurring demonstrations of English to French and French to English translation found throughout the WebText training set.

Figures from Radford et al. (2019)

Zero-shot LLMs

- GPT-2 (1.5B parameters) for unsupervised prediction on various tasks
- GPT-2 models p(output | input, task)
  - translation: *(translate to french, english text, french text)*
  - reading comprehension: *(answer the question, document, question, answer)*
- Why does this work?

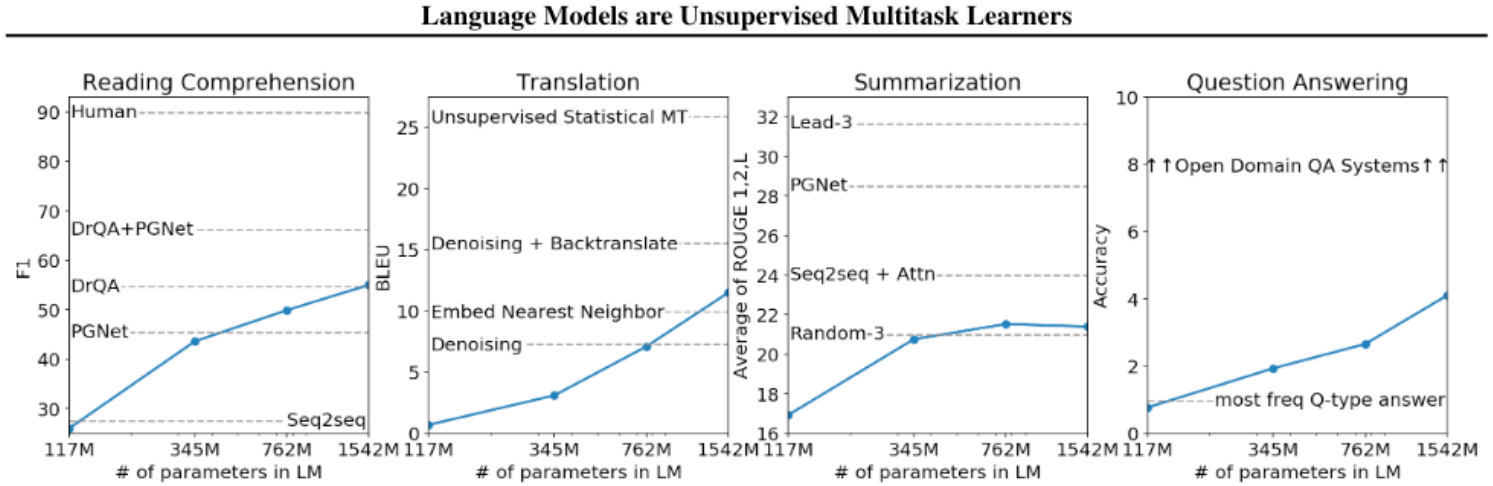


Figure 1. Zero-shot task performance of WebText LMs as a function of model size on many NLP tasks. Reading Comprehension results are on CoQA (Reddy et al., 2018), translation on WMT-14 Fr-En (Artetxe et al., 2017), summarization on CNN and Daily Mail (See et al., 2017), and Question Answering on Natural Questions (Kwiatkowski et al., 2019). Section 3 contains detailed descriptions of each result.

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

Figures from Radford et al. (2019)

IN-CONTEXT LEARNING

Few-shot Learning with LLMs

Suppose you have…

- a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ and N is rather small (i.e. few-shot setting)
- a very large (billions of parameters) pre-trained language model

There are two ways to "learn"

Option A: Supervised fine-tuning

- **Definition:** fine-tune the LLM on the training data using…
  - a standard supervised objective
  - backpropagation to compute gradients
  - your favorite optimizer (e.g. Adam)

Option B: In-context learning

- **Definition:**
  1. feed training examples to the LLM as a prompt
  2. allow the LLM to infer patterns in the training examples during inference (i.e. decoding)
  3. take the output of the LLM following the prompt as its prediction

Few-shot Learning with LLMs

Suppose you have…

- a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ and N is rather small (i.e. few-shot setting)
- a very large (billions of parameters) pre-trained language model

There are two ways to "learn"

This section!

Option A: Supervised fine-tuning

- **Definition:** fine-tune the LLM on the training data using…
  - a standard supervised objective
  - backpropagation to compute gradients
  - your favorite optimizer (e.g. Adam)
- **Pro:** fits into the standard ML recipe
- **Pro:** still works if N is large
- **Con:** backpropagation requires ~3x the memory and computation time as the forward computation
- **Con:** you might not have access to the model weights at all (e.g. because the model is proprietary)

Option B: In-context learning

- **Definition:**
  1. feed training examples to the LLM as a prompt
  2. allow the LLM to infer patterns in the training examples during inference (i.e. decoding)
  3. take the output of the LLM following the prompt as its prediction
- **Con:** the prompt may be very long and Transformer LMs require $O(N^2)$ time/space where N = length of context
- **Pro:** no backpropagation required and only one pass through the training data
- **Pro:** does not require model weights, only API access

Few-shot In-context Learning

- Few-shot learning can be done via in-context learning
- Typically, a task description is presented first
- Then a sequence of input/output pairs from a training dataset are presented in sequence

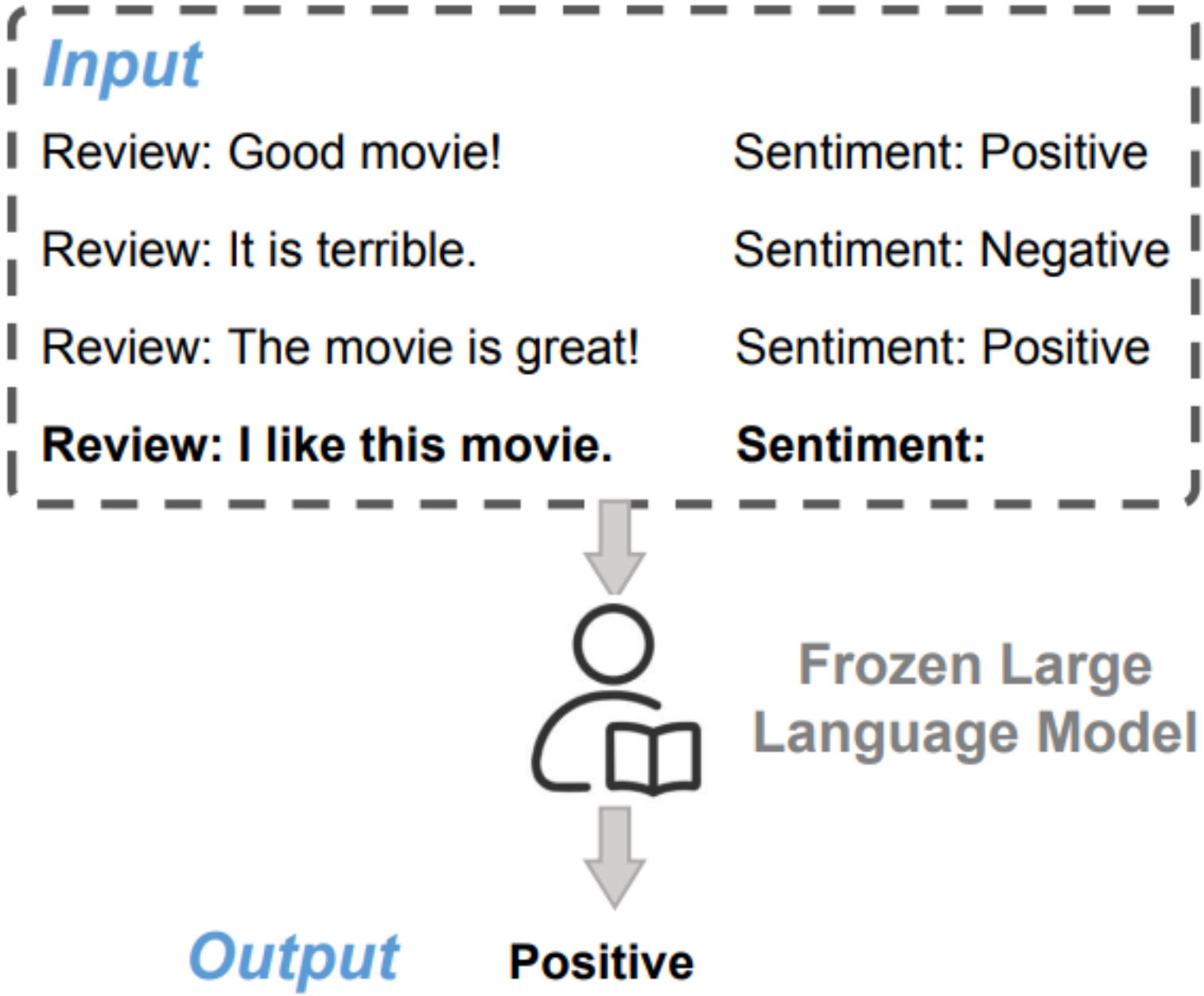


Figure from https://arxiv.org/pdf/2310.09881.pdf

Few-shot In-context Learning

- Few-shot learning can be done via in-context learning
- Typically, a task description is presented first
- Then a sequence of input/output pairs from a training dataset are presented in sequence

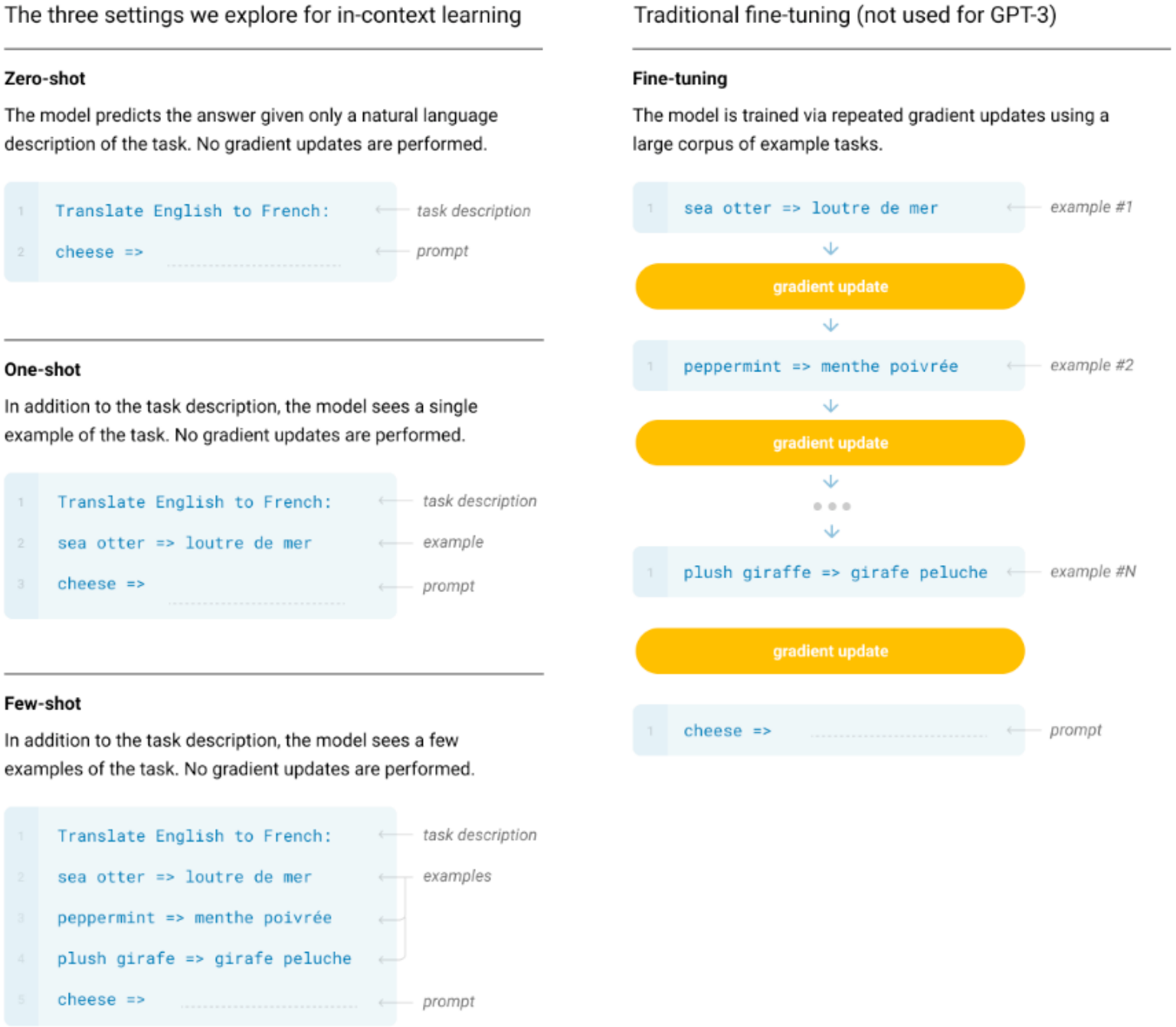


Figure from http://arxiv.org/abs/2005.14165

LEARNING PARADIGMS

Learning Paradigms

Paradigm	**Data**	
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$	$\mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$	
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$	
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$	
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector	

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$

Learning Paradigms

Paradigm	Data
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$

Learning Paradigms

Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ and can query $y^{(i)} = c^*(\cdot)$ at a cost

Learning Paradigms

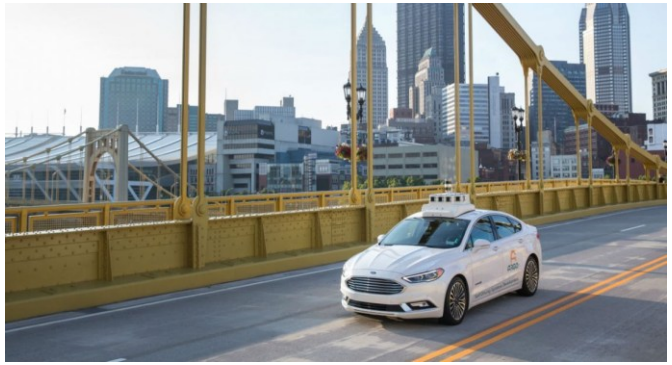
Paradigm	**Data**
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N} \qquad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$

Learning Paradigms

Paradigm	Data
Supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$ and $y = c^*(\cdot)$
$\hookrightarrow$ Regression	$y^{(i)} \in \mathbb{R}$
$\hookrightarrow$ Classification	$y^{(i)} \in \{1, \ldots, K\}$
$\hookrightarrow$ Binary classification	$y^{(i)} \in \{+1, -1\}$
$\hookrightarrow$ Structured Prediction	$\mathbf{y}^{(i)}$ is a vector
Unsupervised	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ $\quad \mathbf{x} \sim p^*(\cdot)$
Semi-supervised	$\mathcal{D} = \{\mathbf{x}^{(i)}, y^{(i)}\}_{i=1}^{N_1} \cup \{\mathbf{x}^{(j)}\}_{j=1}^{N_2}$
Online	$\mathcal{D} = \{(\mathbf{x}^{(1)}, y^{(1)}), (\mathbf{x}^{(2)}, y^{(2)}), (\mathbf{x}^{(3)}, y^{(3)}), \ldots\}$
Active Learning	$\mathcal{D} = \{\mathbf{x}^{(i)}\}_{i=1}^{N}$ and can query $y^{(i)} = c^*(\cdot)$ at a cost
Imitation Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}), (s^{(2)}, a^{(2)}), \ldots\}$
Reinforcement Learning	$\mathcal{D} = \{(s^{(1)}, a^{(1)}, r^{(1)}), (s^{(2)}, a^{(2)}, r^{(2)}), \ldots\}$

MDPS & REINFORCEMENT LEARNING

RL: Examples



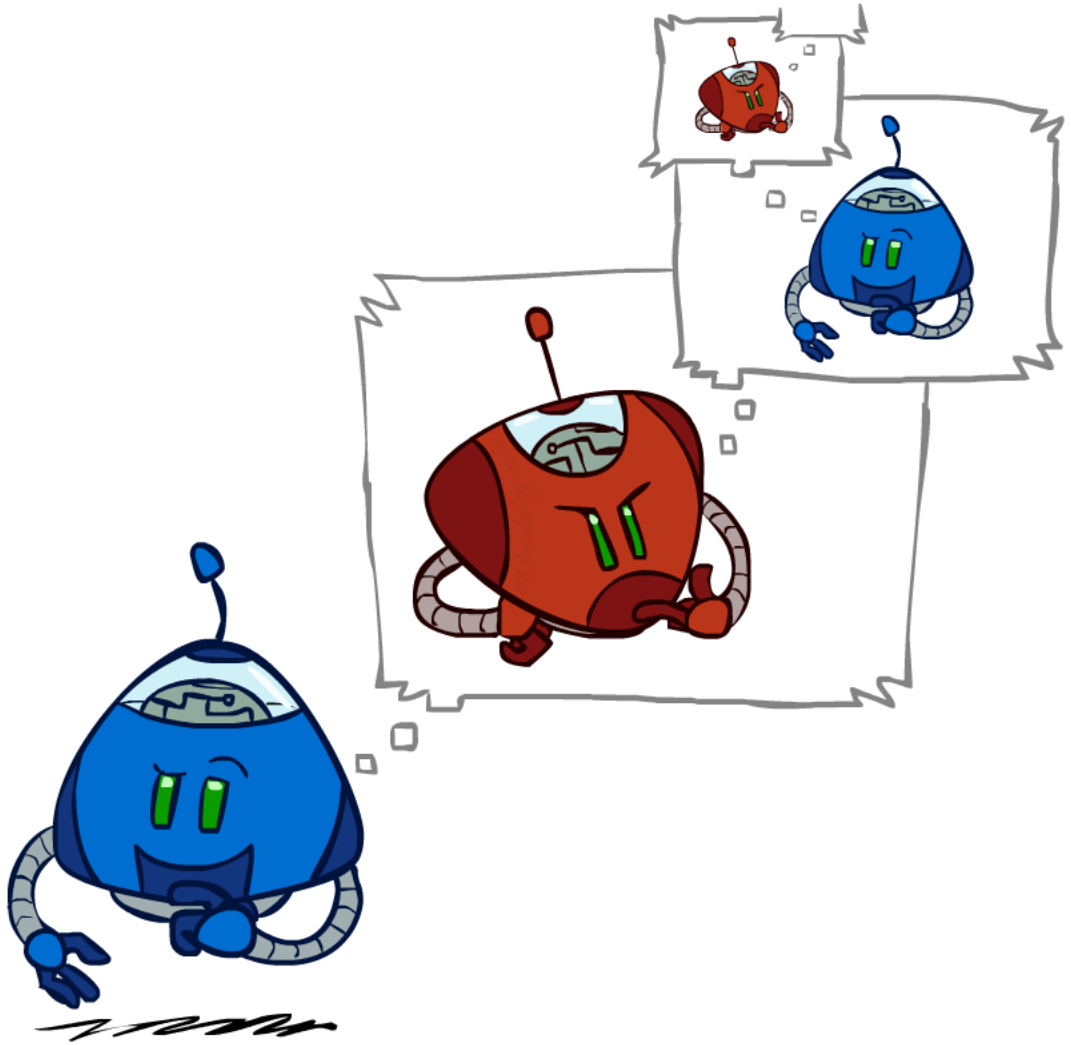
Source: https://www.cnet.com/news/boston-dynamics-robot-dog-spot-finally-goes-on-sale-for-74500/

Source: https://twitter.com/alphagomovie

Source: https://techobserver.net/2019/06/argo-ai-self-driving-car-research-center/

Source: https://www.wired.com/2012/02/high-speed-trading/

Adversarial Search



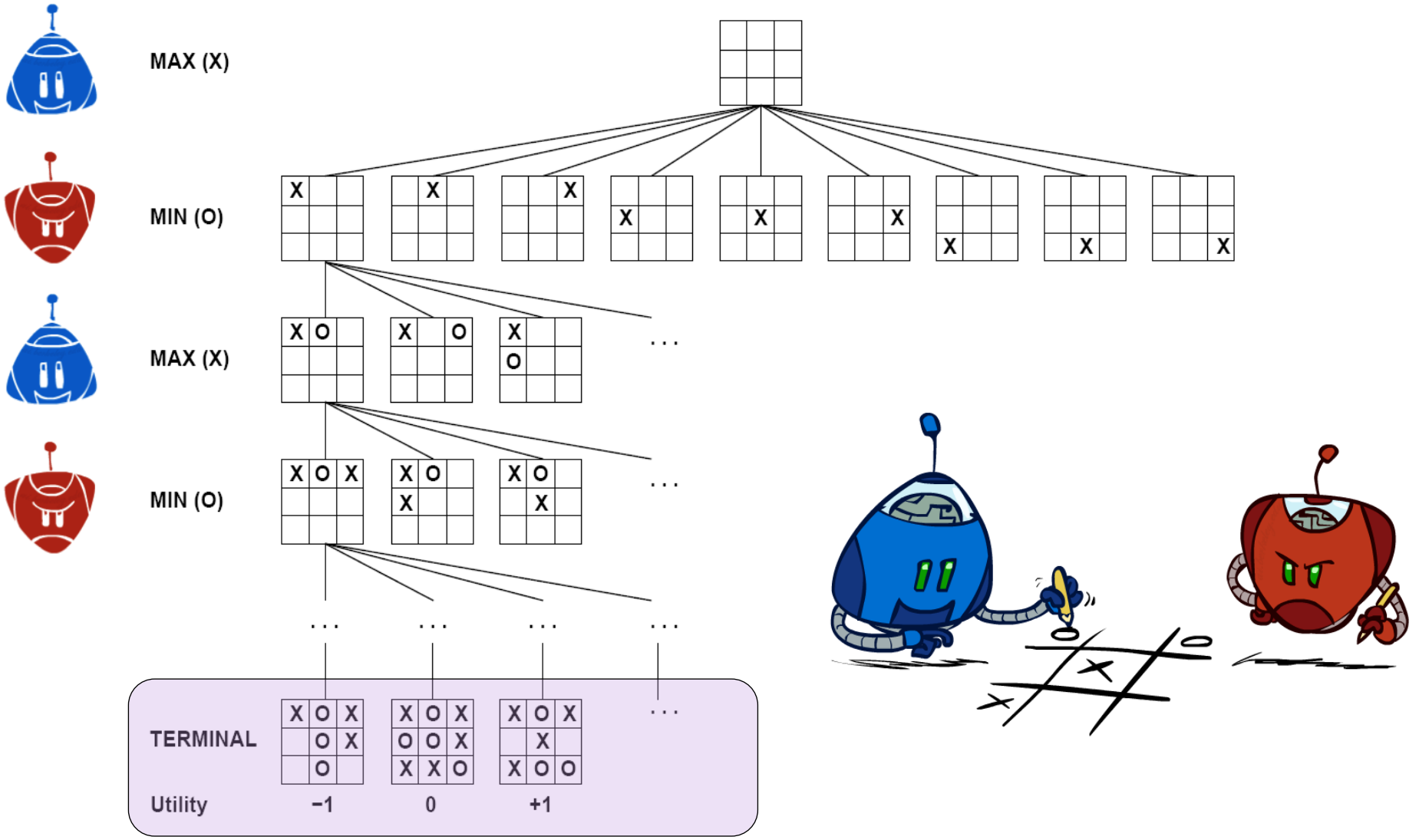
Slide credit: UC Berkeley AI

Minimax

States

Actions

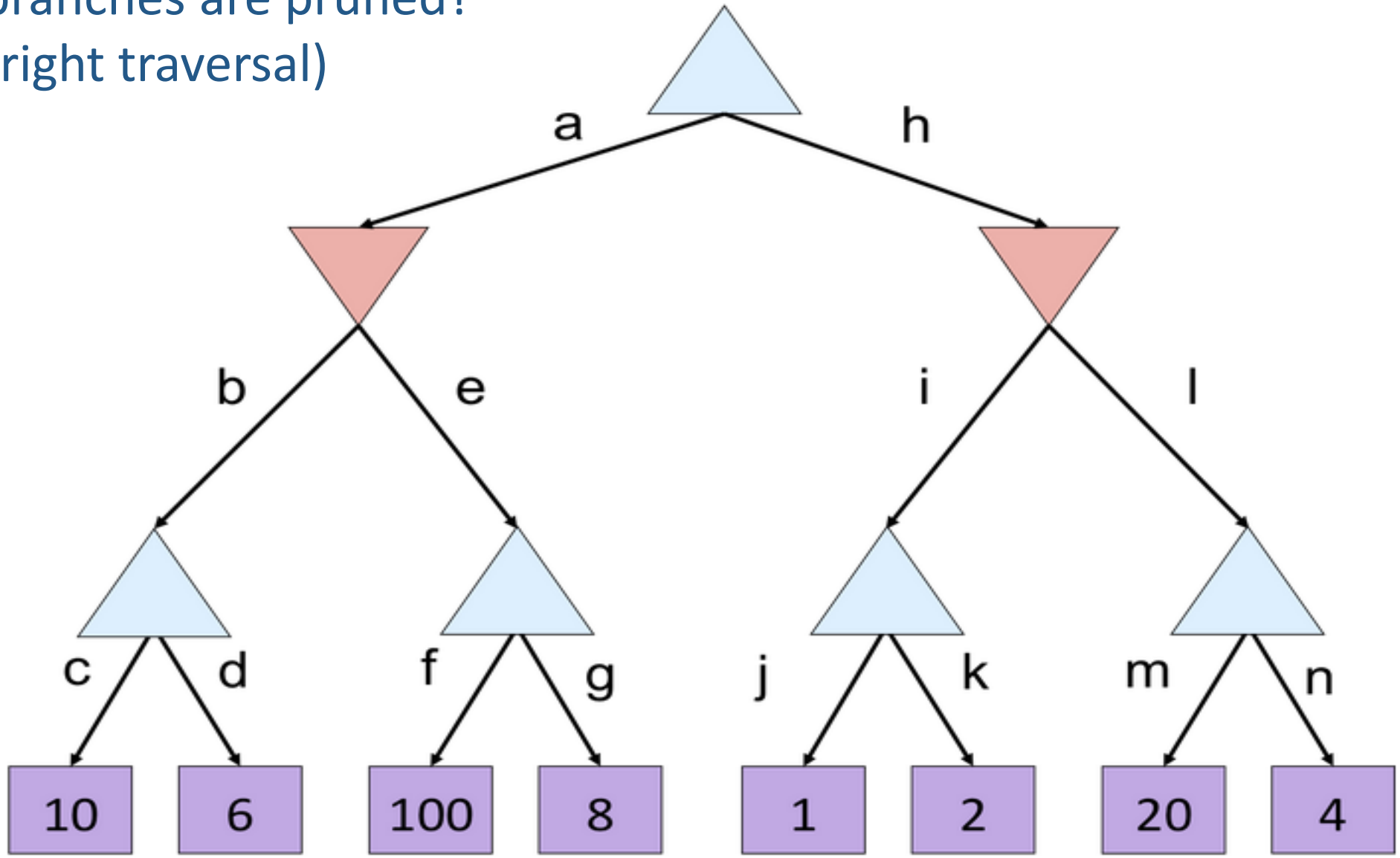
Values



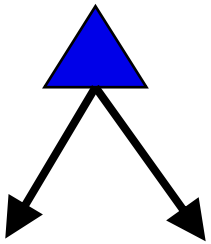
Slide credit: UC Berkeley AI

Example

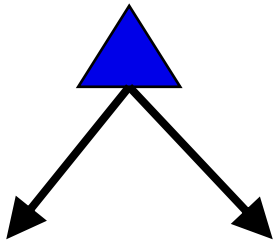
Which branches are pruned?
(Left to right traversal)



Minimax Notation



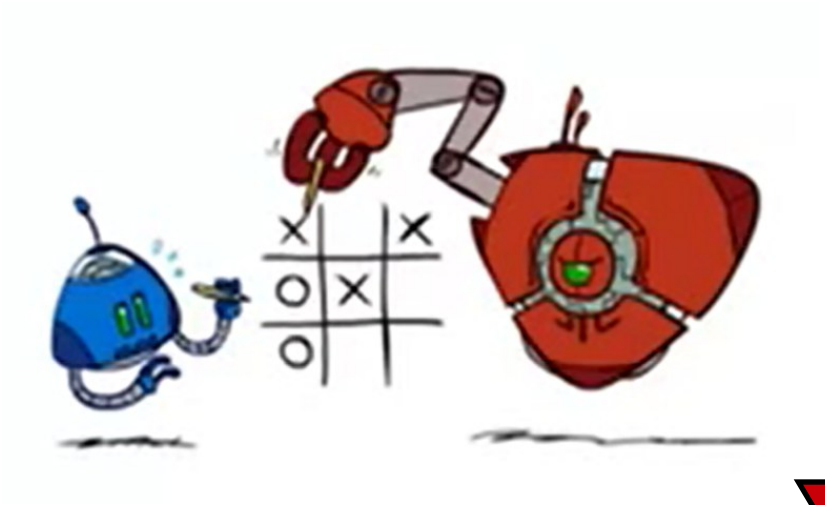
$$V(s) = \max_{a} V(s'),$$
$$\text{where } s' = result(s, a)$$

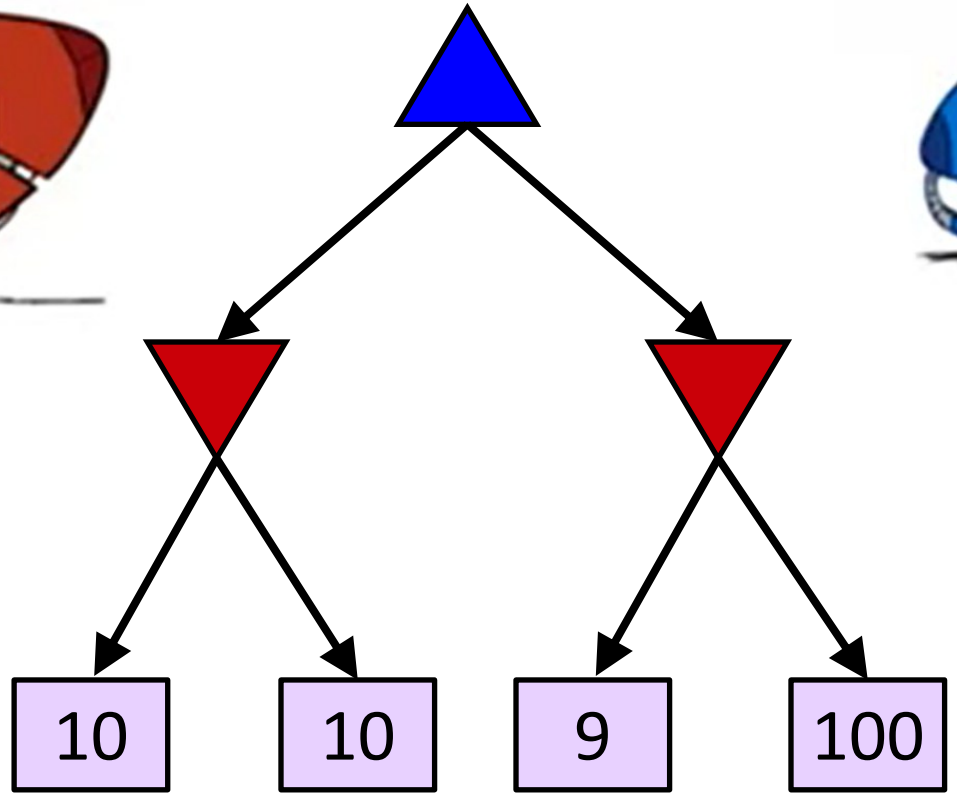


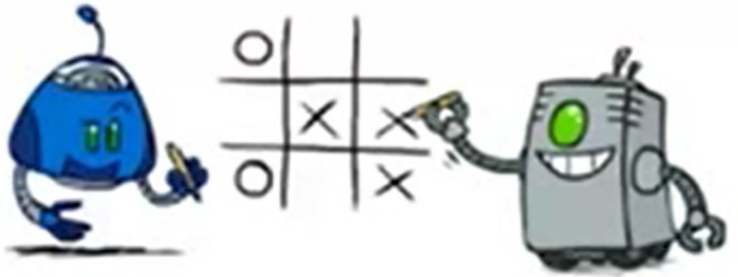
$$\hat{a} = \operatorname{argmax}_{a} V(s'),$$
$$\text{where } s' = result(s, a)$$

Modeling Assumptions

Know your opponent



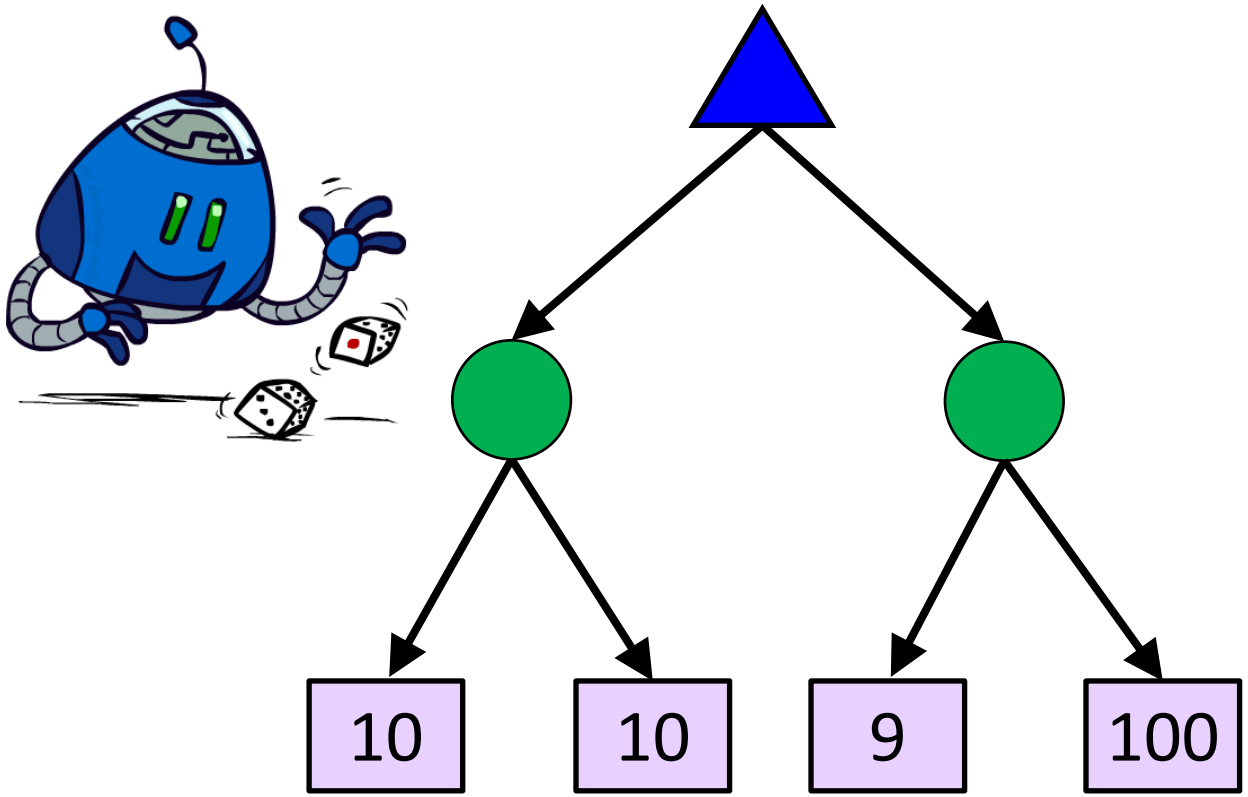




Slide credit: UC Berkeley AI

Modeling Assumptions

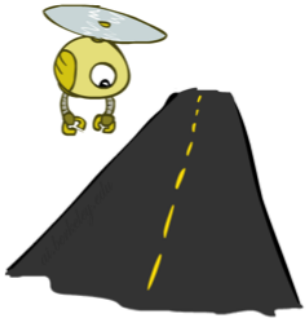
Chance nodes: Expectimax

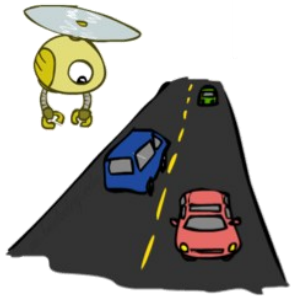


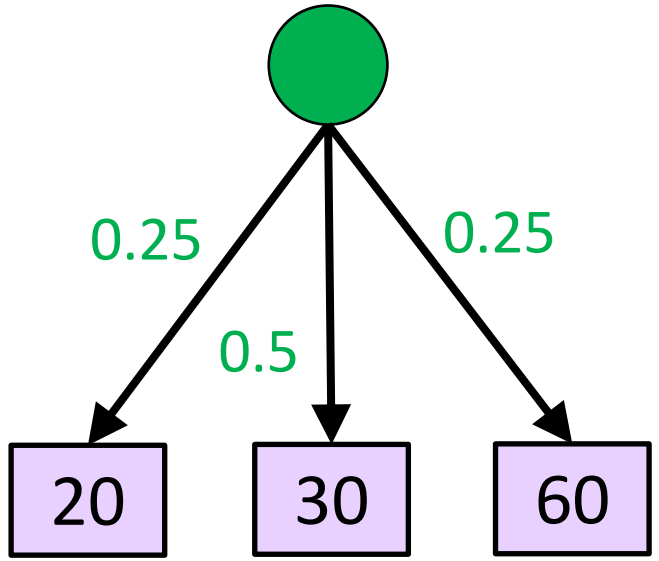
Slide credit: UC Berkeley AI

Expectations

Time:	20 min		30 min		60 min
	x	+	x	+	x
Probability:	0.25		0.50		0.25







Max node notation

$$V(s) = \max_a V(s'),$$

where $s' = result(s, a)$

Expectation node notation

$$V(s) = \sum_{s'} P(s')\, V(s')$$

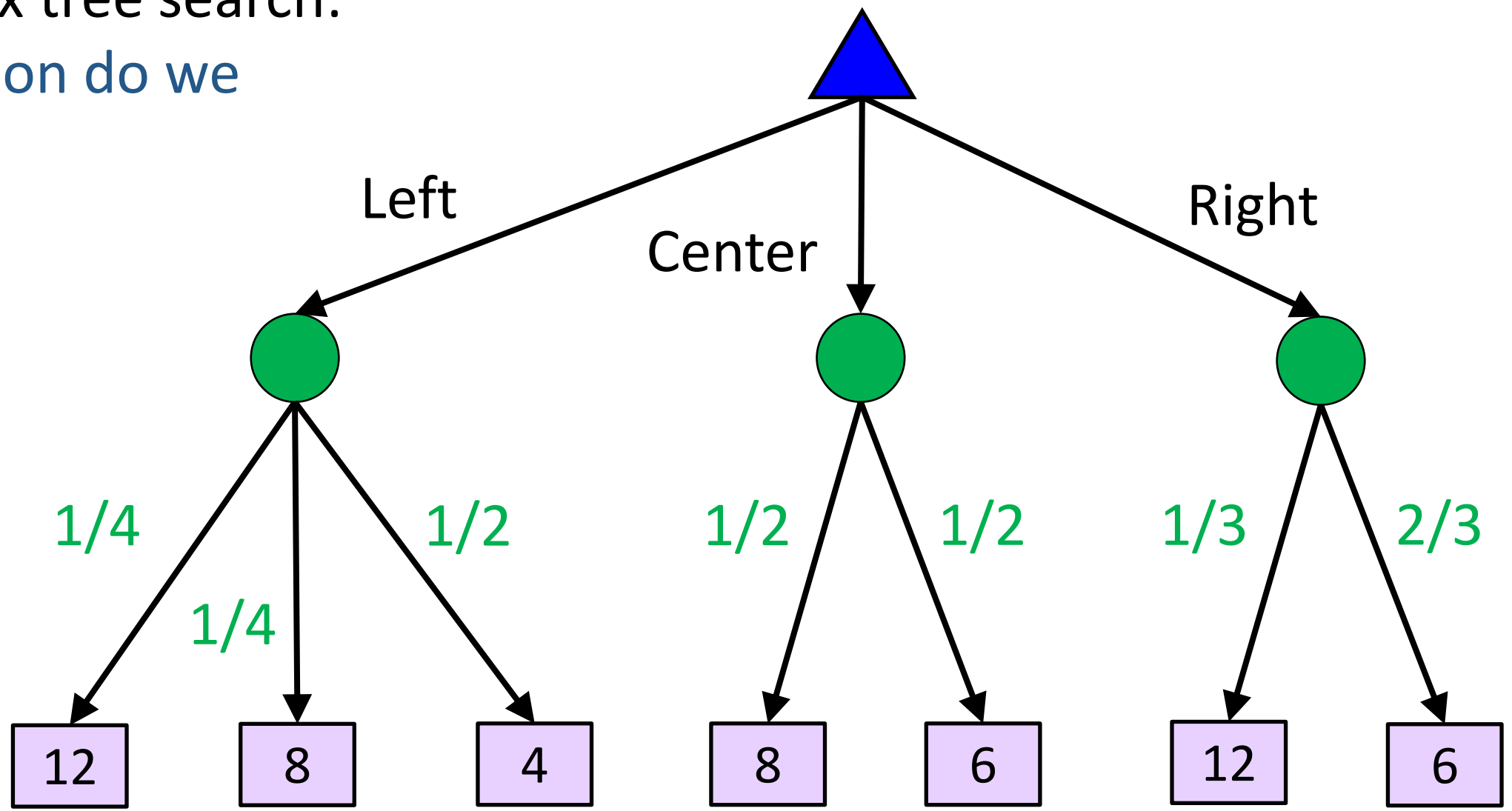
Slide credit: UC Berkeley AI

Question

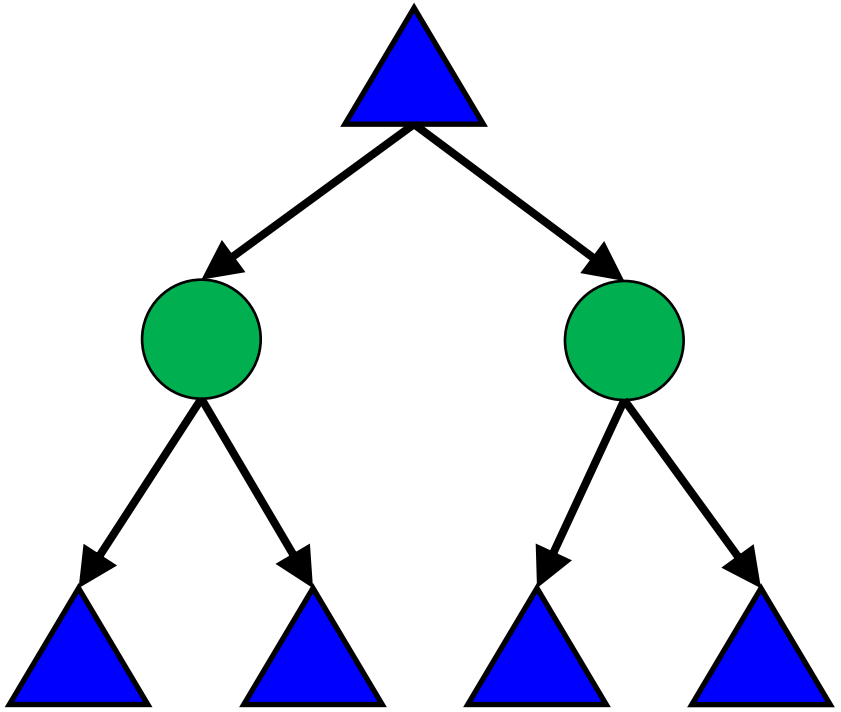
Expectimax tree search:

Which action do we choose?

A) Left
B) Center
C) Right
D) Eight



Expectimax Notation



$$V(s) = \max_{a} \sum_{s'} P(s'|s,a)\, V(s')$$

MDP Notation

Standard expectimax: $V(s) = \max_a \sum_{s'} P(s'|s,a)V(s')$

Bellman equations: $V^*(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^*(s')]$

Value iteration: $V_{k+1}(s) = \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V_k(s')], \quad \forall\, s$

Next Time: Expectimax → MDPs

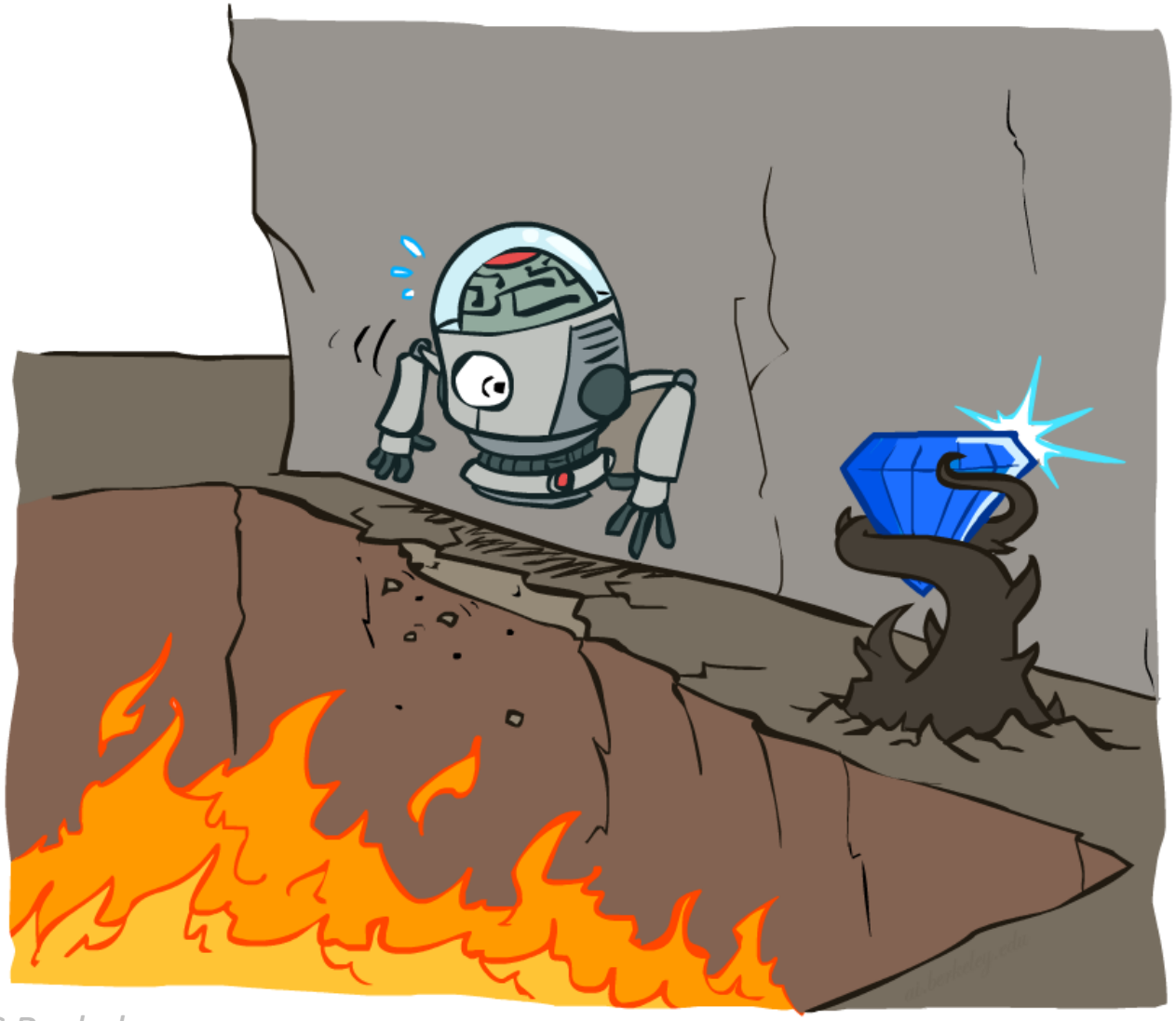


Image credit: *Ketrina Yim, UC Berkeley*