In-Context Learning as Bayesian Inference

Tea Talk

Vincent Dutordoir

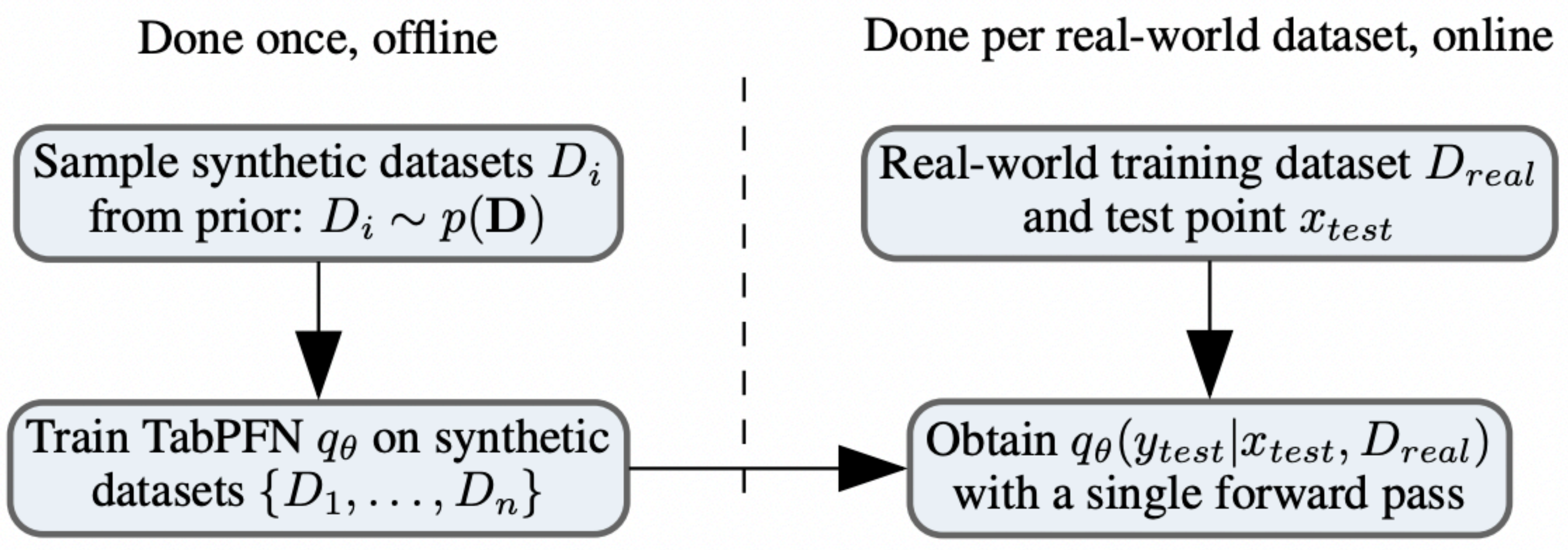
UNIVERSITY OF CAMBRIDGE

4th Nov 2022

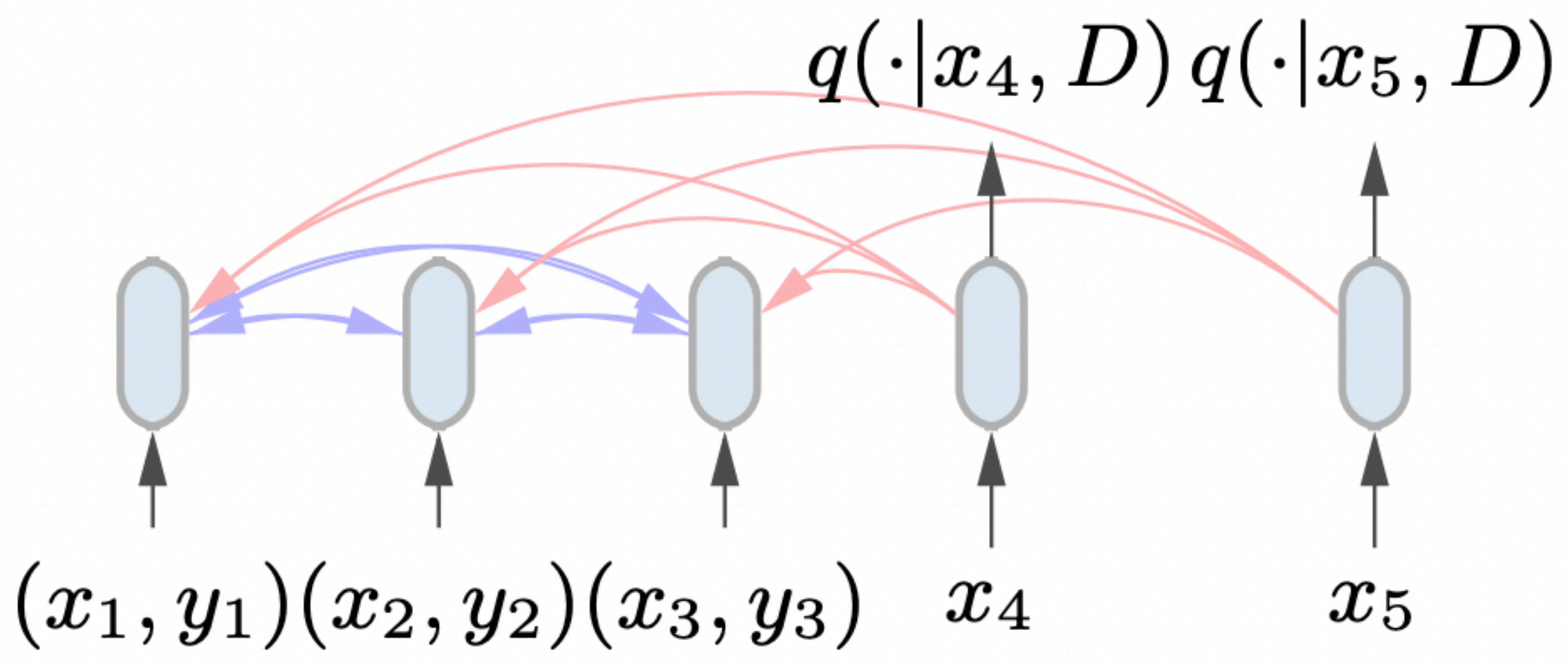
12 Oct 2022

TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second

Noah Hollmann[*,1,2] **Samuel Müller**[*,1] **Katharina Eggensperger**[1] **Frank Hutter**[1,3]
[1] University of Freiburg, [2] Charité University Medicine Berlin
[3] Bosch Center for Artificial Intelligence [*] Equal contribution.
Correspondance to `noah.hollmann@charite.de` & `muellesa@cs.uni-freiburg.de`



(a) Prior-fitting and inference

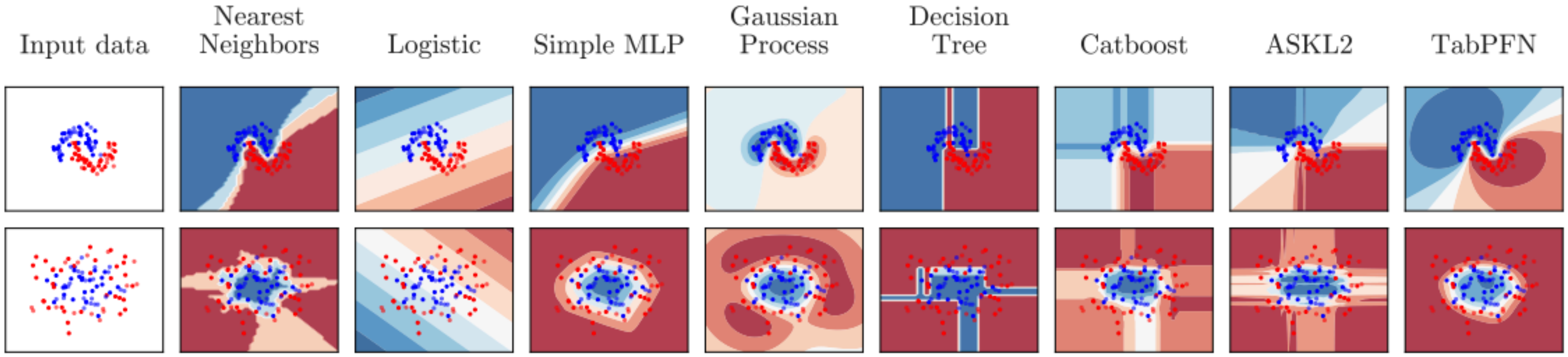


(b) Architecture and attention mechanism

12 Oct 2022

TabPFN: A Transformer That Solves Small Tabular Classification Problems in a Second

Noah Hollmann[*,1,2] **Samuel Müller**[*,1] **Katharina Eggensperger**[1] **Frank Hutter**[1,3]
[1] University of Freiburg, [2] Charité University Medicine Berlin
[3] Bosch Center for Artificial Intelligence [*] Equal contribution.
Correspondance to noah.hollmann@charite.de & muellesa@cs.uni-freiburg.de



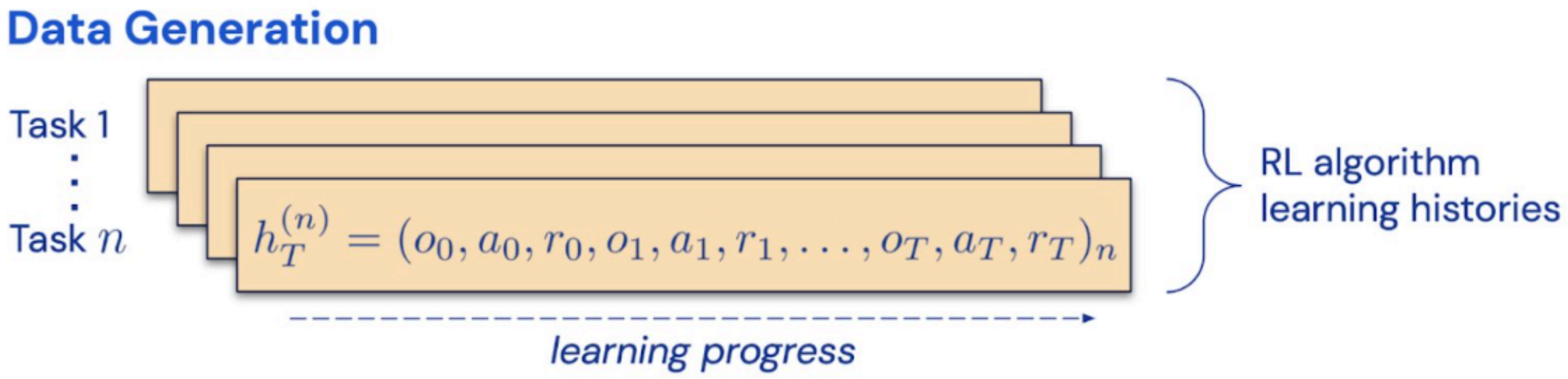
25 Oct 2022

In-context Reinforcement Learning with Algorithm Distillation

DeepMind

First, we collect a dataset of learning histories from an RL algorithm trained on diverse tasks.

This can be *any* RL algorithm - it can be doing gradient updates, replay, planning, can be on or off-policy, model-free or model- based.

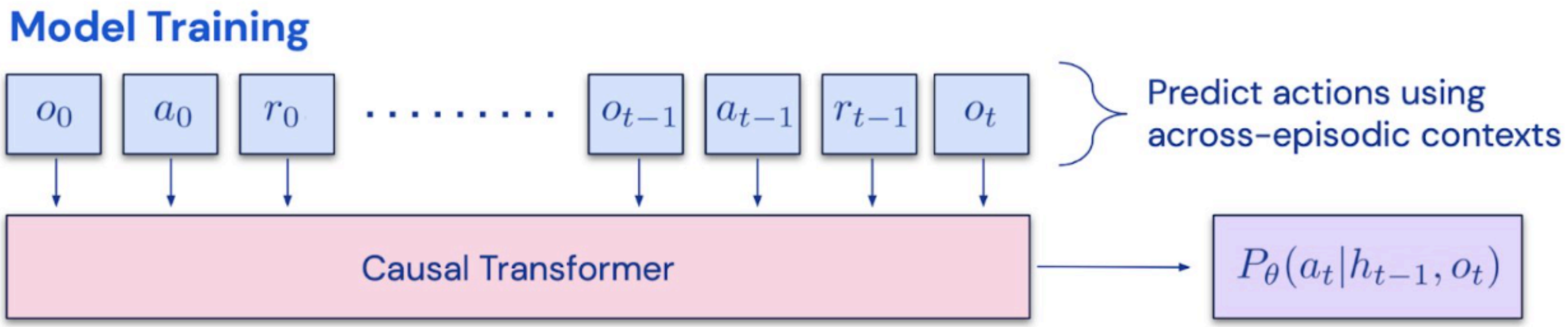


In-context Reinforcement Learning with Algorithm Distillation

DeepMind

Next, train a transformer to predict actions from the entire learning history preceding the current timestep.

Policy *improves* throughout RL training, to predict actions accurately, transformer needs to



An Explanation of In-context Learning as Implicit Bayesian Inference

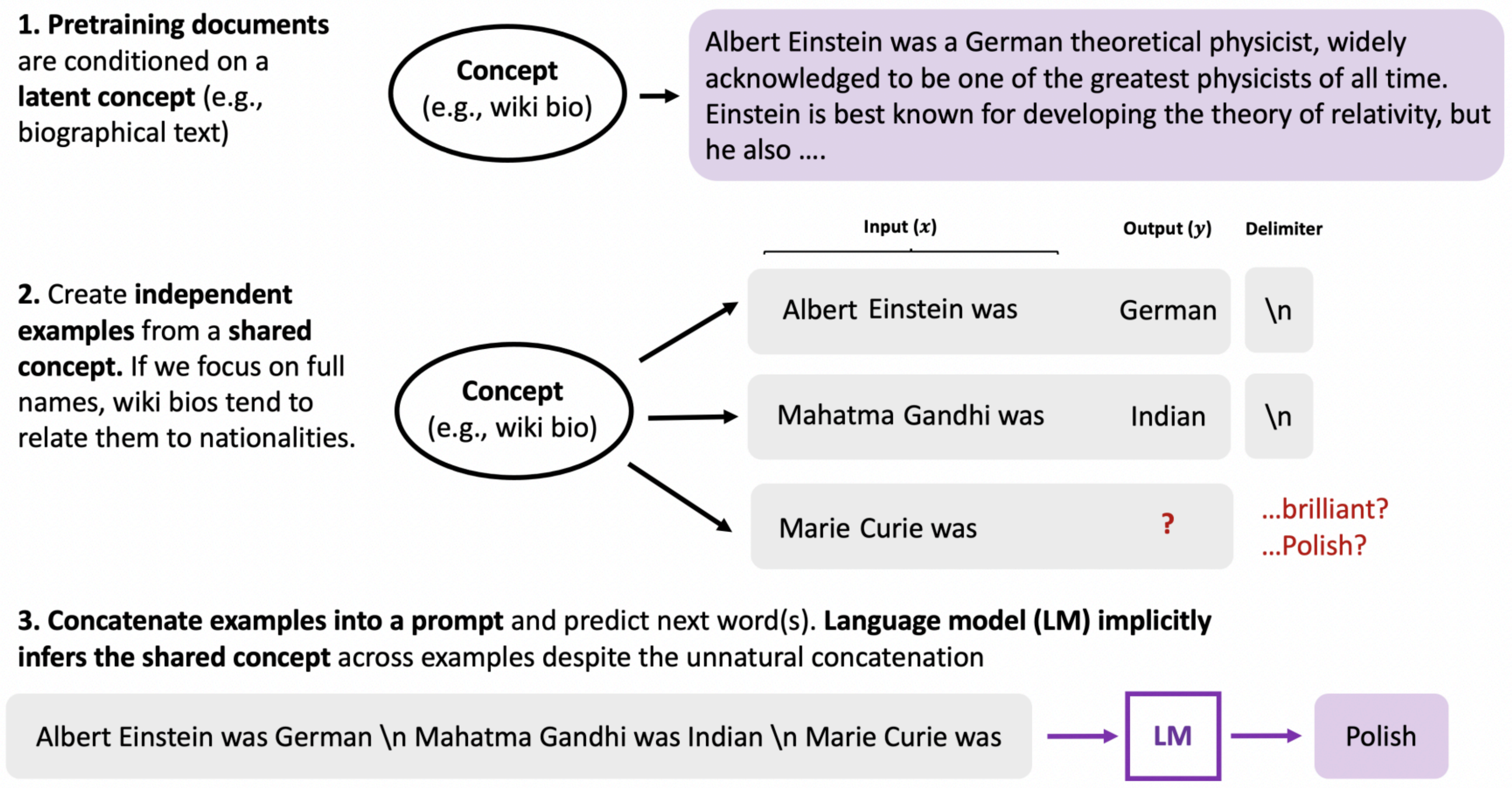
Sang Michael Xie
Stanford University
xie@cs.stanford.edu

Aditi Raghunathan
Stanford University
aditir@stanford.edu

Percy Liang
Stanford University
pliang@cs.stanford.edu

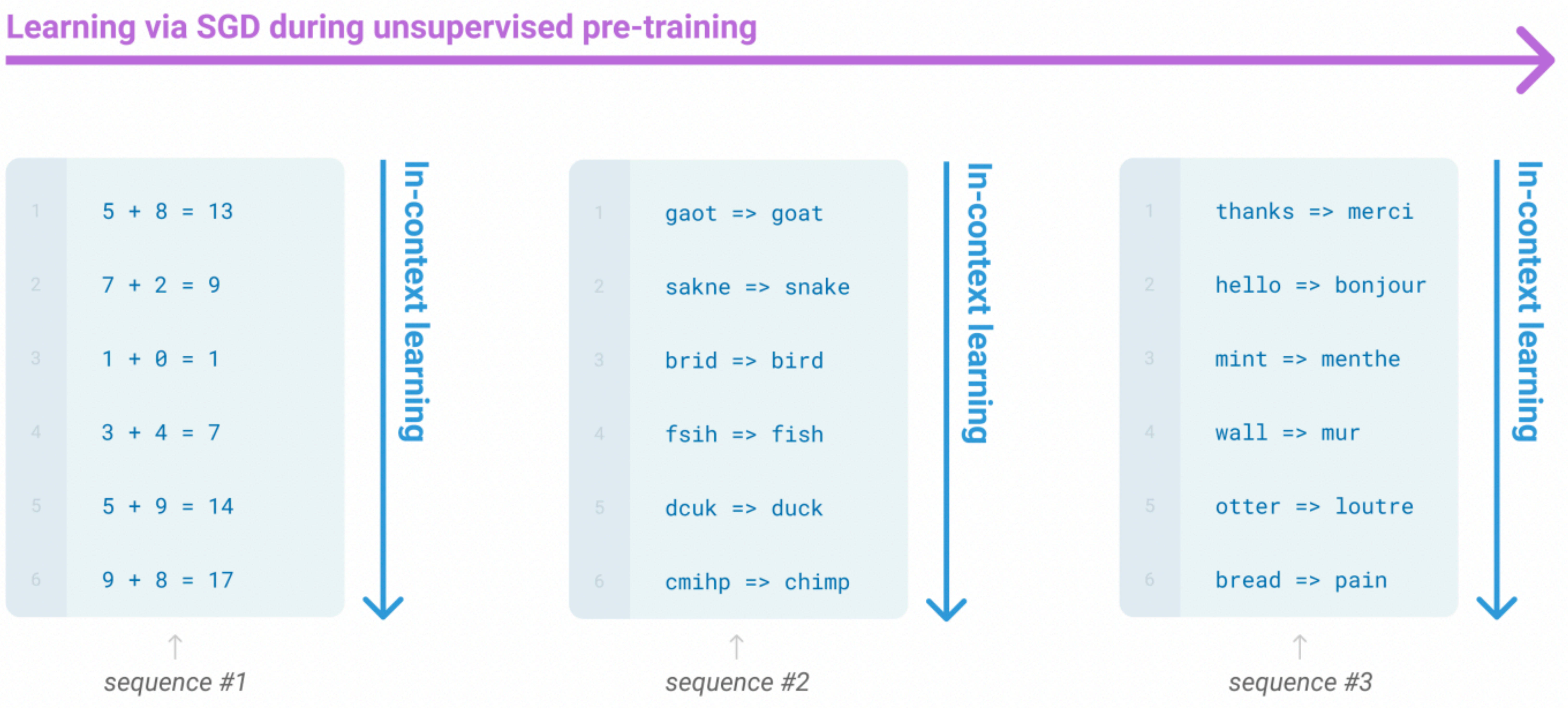
Tengyu Ma
Stanford University
tengyuma@cs.stanford.edu

Brown et al. Language Models are Few-Shot Learners. 2020


1. Pretraining documents are conditioned on a **latent concept** (e.g., biographical text)
Concept (e.g., wiki bio)
Albert Einstein was a German theoretical physicist, widely acknowledged to be one of the greatest physicists of all time. Einstein is best known for developing the theory of relativity, but he also ....
Input (x)
Output (y)
Delimiter
2. Create **independent examples** from a **shared concept.** If we focus on full names, wiki bios tend to relate them to nationalities.
Concept (e.g., wiki bio)
Albert Einstein was
German
\n
Mahatma Gandhi was
Indian
\n
Marie Curie was
?
...brilliant?
...Polish?
3. Concatenate examples into a prompt and predict next word(s). **Language model (LM) implicitly infers the shared concept** across examples despite the unnatural concatenation
Albert Einstein was German \n Mahatma Gandhi was Indian \n Marie Curie was
LM
Polish

Pretraining distribution. In our framework, a latent concept θ from a family of concepts Θ defines a distribution over observed tokens o from a vocabulary $\mathcal{O}$. To generate a document, we first sample a concept from a prior $p(\theta)$ and then sample the document given the concept. Each pretraining document is a length T sequence:

$$p(o_1, \ldots, o_T) = \int_{\theta \in \Theta} p(o_1, \ldots, o_T | \theta) p(\theta) d\theta. \tag{2}$$



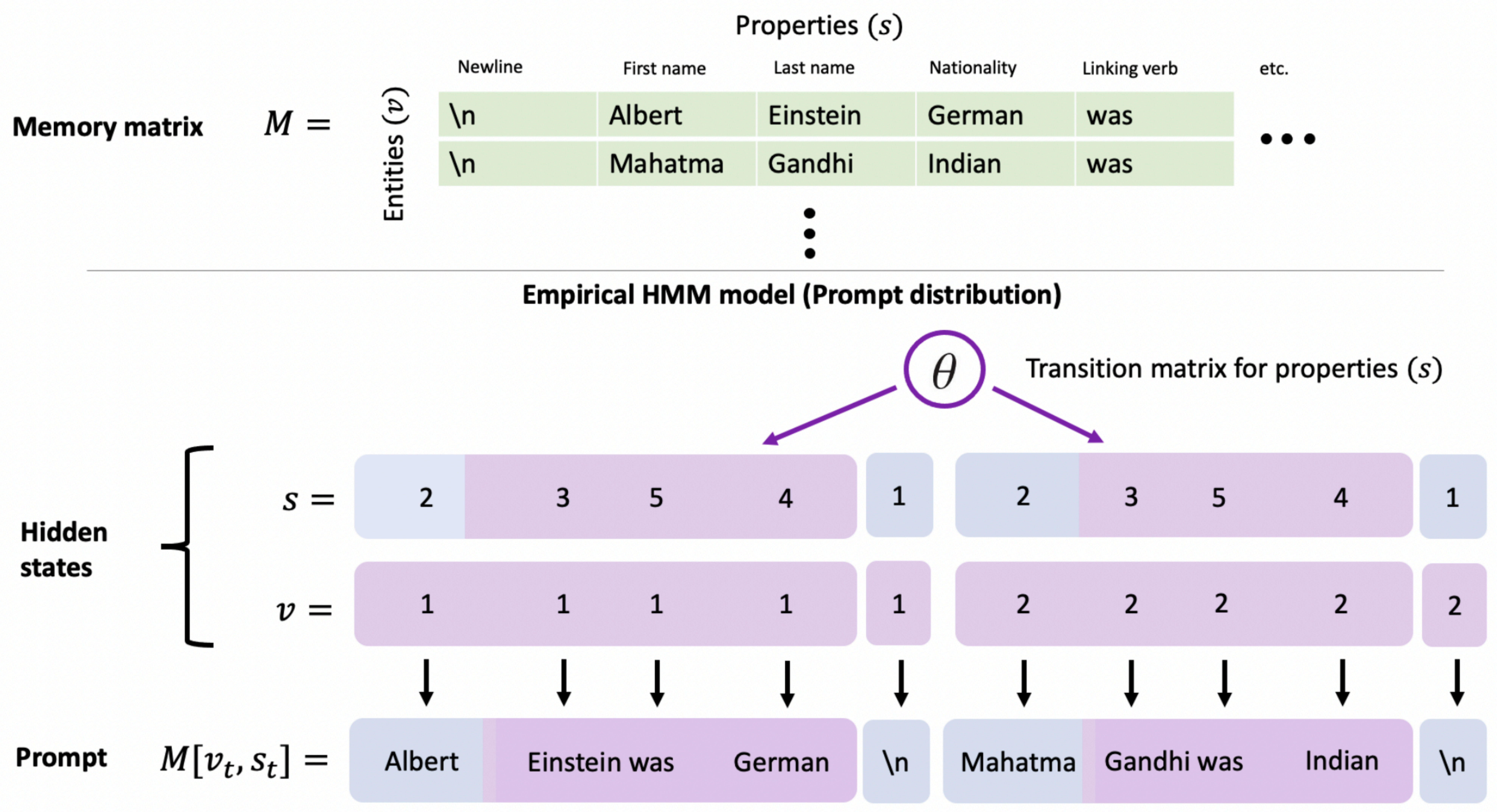
$$p(\text{output}|\text{prompt}) = \int_{\text{concept}} p(\text{output}|\text{concept}, \text{prompt})p(\text{concept}|\text{prompt})d(\text{concept}).$$

Theorem 1. *Assume the assumptions in Section 2.1 hold. If Condition 1 holds, then as $n \to \infty$ the prediction according to the pretraining distribution is*

$$\arg\max_y \; p(y|S_n, x_{test}) \to \arg\max_y \; p_{prompt}(y|x_{test}). \tag{15}$$

Thus, the in-context predictor f_n achieves the optimal 0-1 risk: $\lim_{n\to\infty} L_{0\text{-}1}(f_n) = \inf_f \; L_{0\text{-}1}(f)$.

$$[S_n, x_{\text{test}}] = [x_1, y_1, o^{\text{delim}}, x_2, y_2, o^{\text{delim}}, \dots, x_n, y_n, o^{\text{delim}}, x_{\text{test}}] \sim p_{\text{prompt}}.$$


Properties (s)
Newline
First name
Last name
Nationality
Linking verb
etc.
Memory matrix
$M =$
Entities (v)
\n
Albert
Einstein
German
was
\n
Mahatma
Gandhi
Indian
was
Empirical HMM model (Prompt distribution)
θ
Transition matrix for properties (s)
Hidden states
$s =$
2 3 5 4 1 2 3 5 4 1
$v =$
1 1 1 1 1 2 2 2 2 2
Prompt
$M[v_t, s_t] =$
Albert
Einstein was
German
\n
Mahatma
Gandhi was
Indian
\n

Result

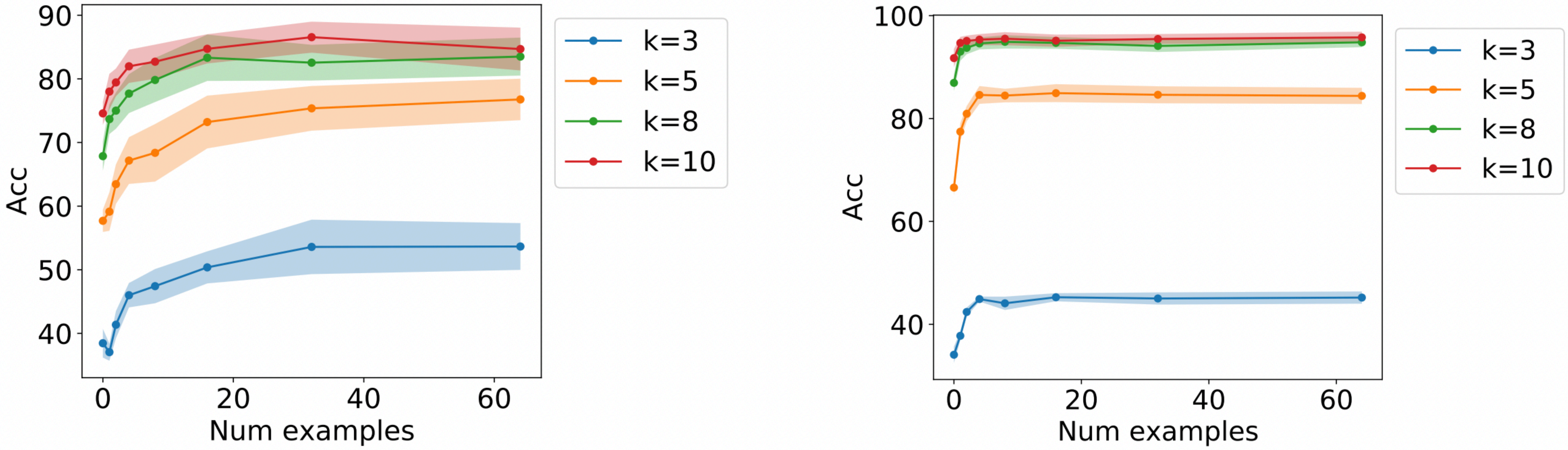


Figure 3: In-context accuracy (95% intervals) of Transformers (left) and LSTMs (right) on the GINC dataset. Accuracy increases with number of examples n and length of each example k.

Thank you

Questions

- No free lunch?