Reinforcement Learning for Language Model Training

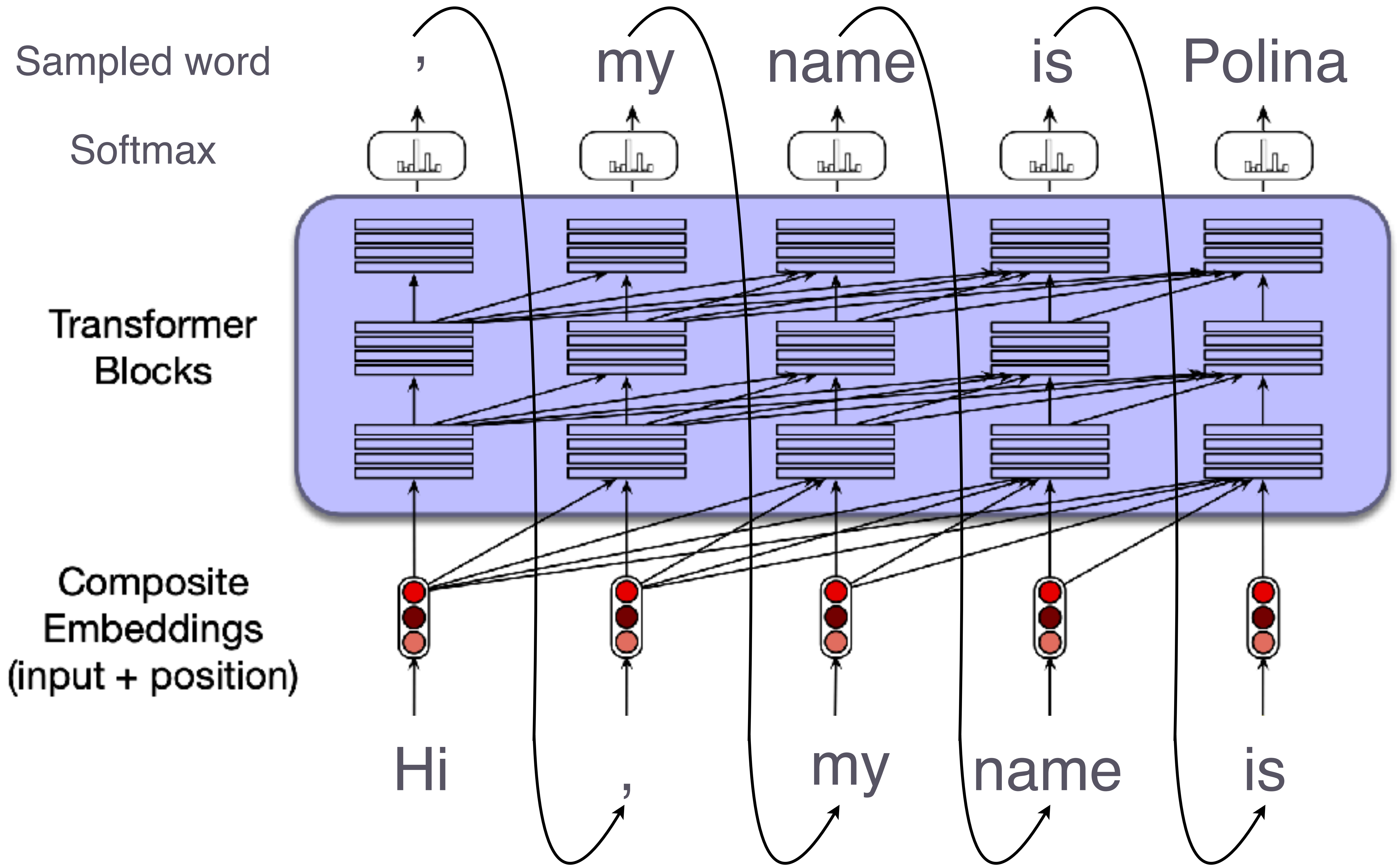
Polina Tsvilodub

Calibration & RL evaluation

RL4
LMT

Language models
Understanding mechanics

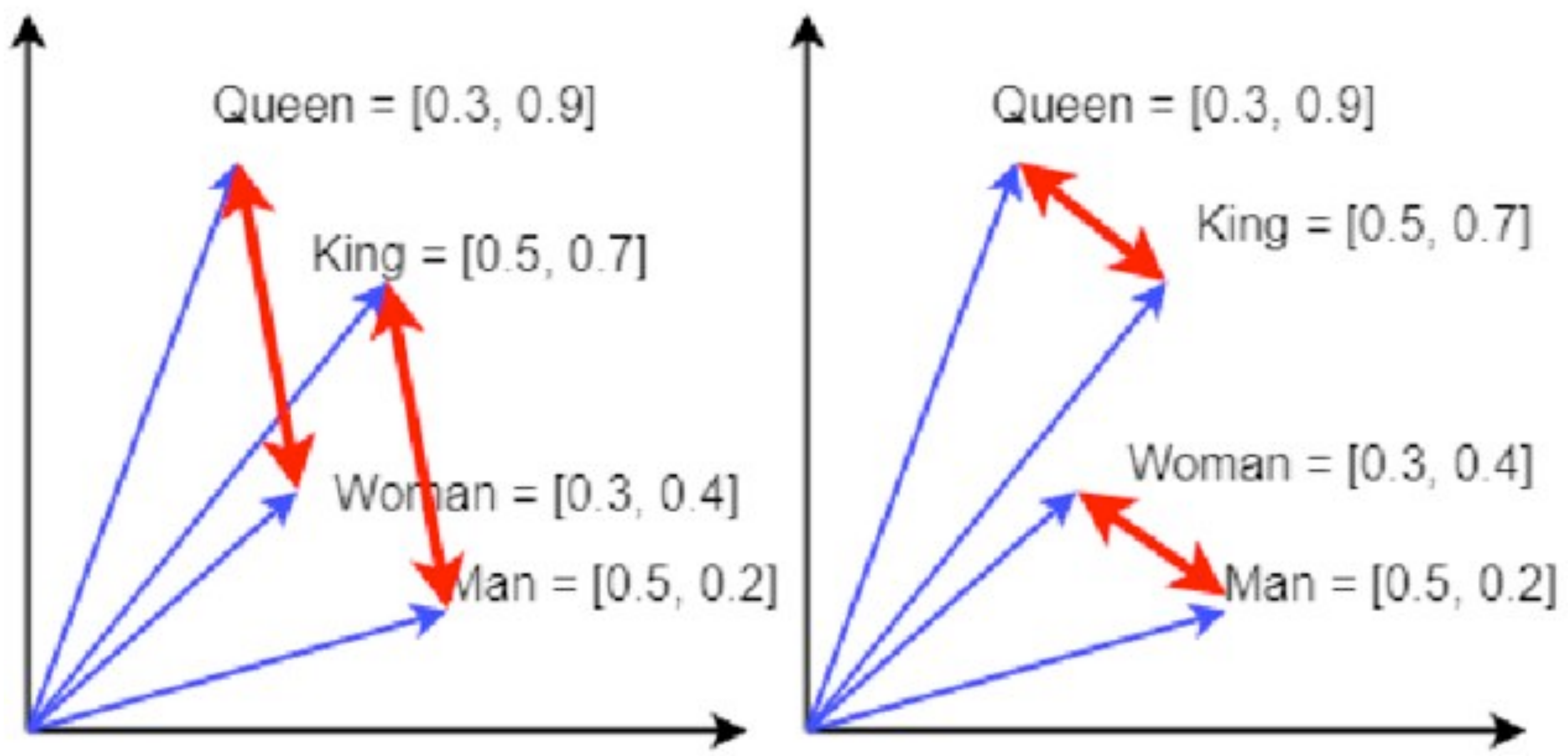
- what information in represented at different stages of processing?
- what information contributes to predicting the right answer?
- what (architectural) mechanisms extract important information?
- what (architectural) mechanisms are necessary for solving different tasks?
- how do we investigate systems involving RL fine-tuning?
- …



Embedding evaluation

Doctor - man + woman = ?

- pretrained **word embeddings** have been evaluated as semantic representations
  - vector arithmetic
  - $cos(w_1, w_2) = \dfrac{w_1 \cdot w_2}{||w_1||\ ||w_2||}$
- current models are decoder-only and use sub-word embeddings
  - semantic tasks often solved few-shot



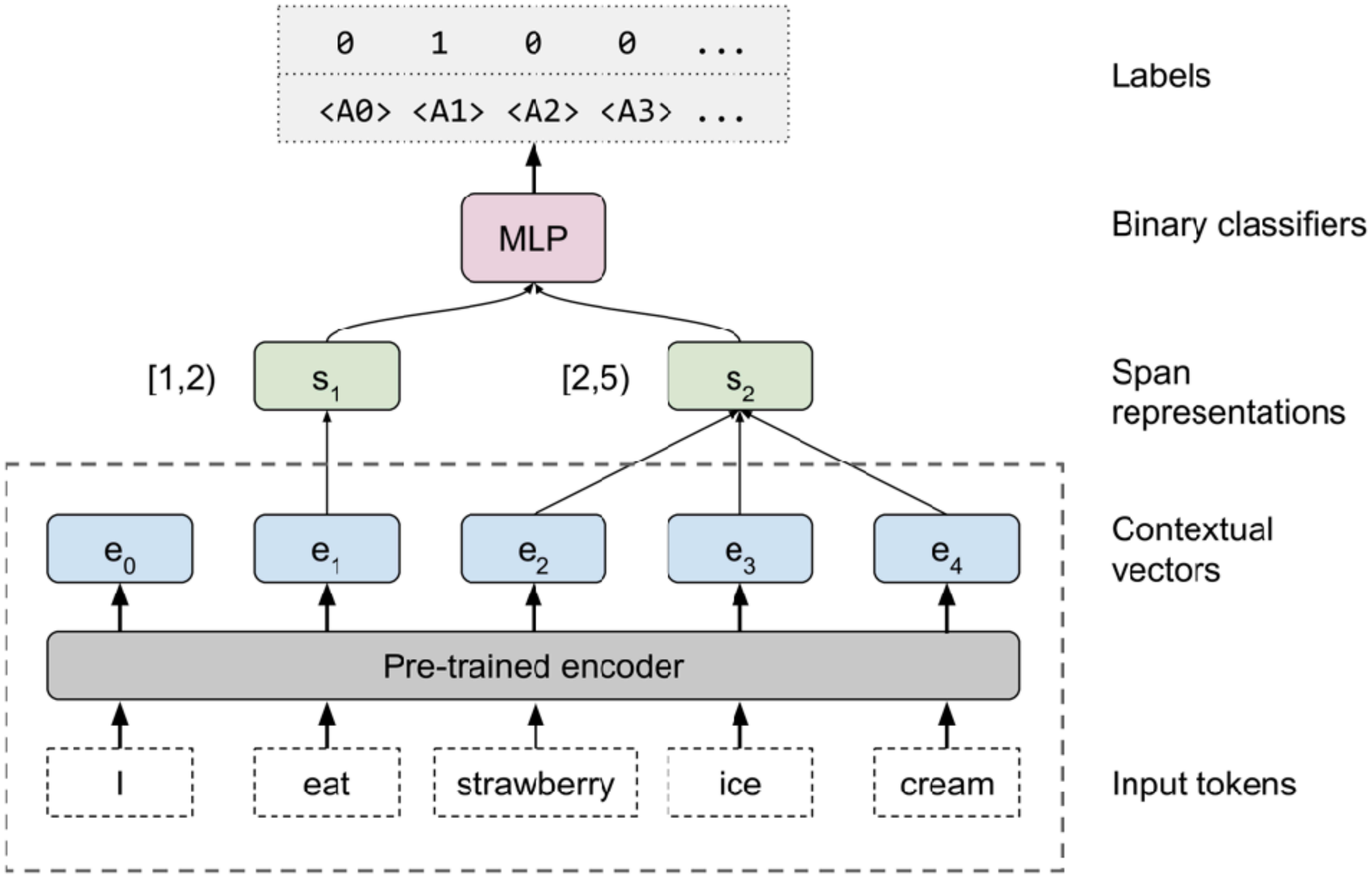
Bolukbasi et al. (2016), Mikolov et al. (2013), image src

Scalar mixing weights

which layers to combine information from

- consider L layers of stacked embeddings $H^{(0)}, \ldots, H^{(L)}$, input $w_1, \ldots, w_n$, vector $\left[\mathbf{h}_0^{(l)}, \ldots, \mathbf{h}_n^{(l)}\right]$ of word embeddings at layer l
- train scalar mixing weights $[s_0, \ldots, s_L]$ together with MLP classifiers for each layer to solve tasks (e.g., POS tagging) based on token representations:

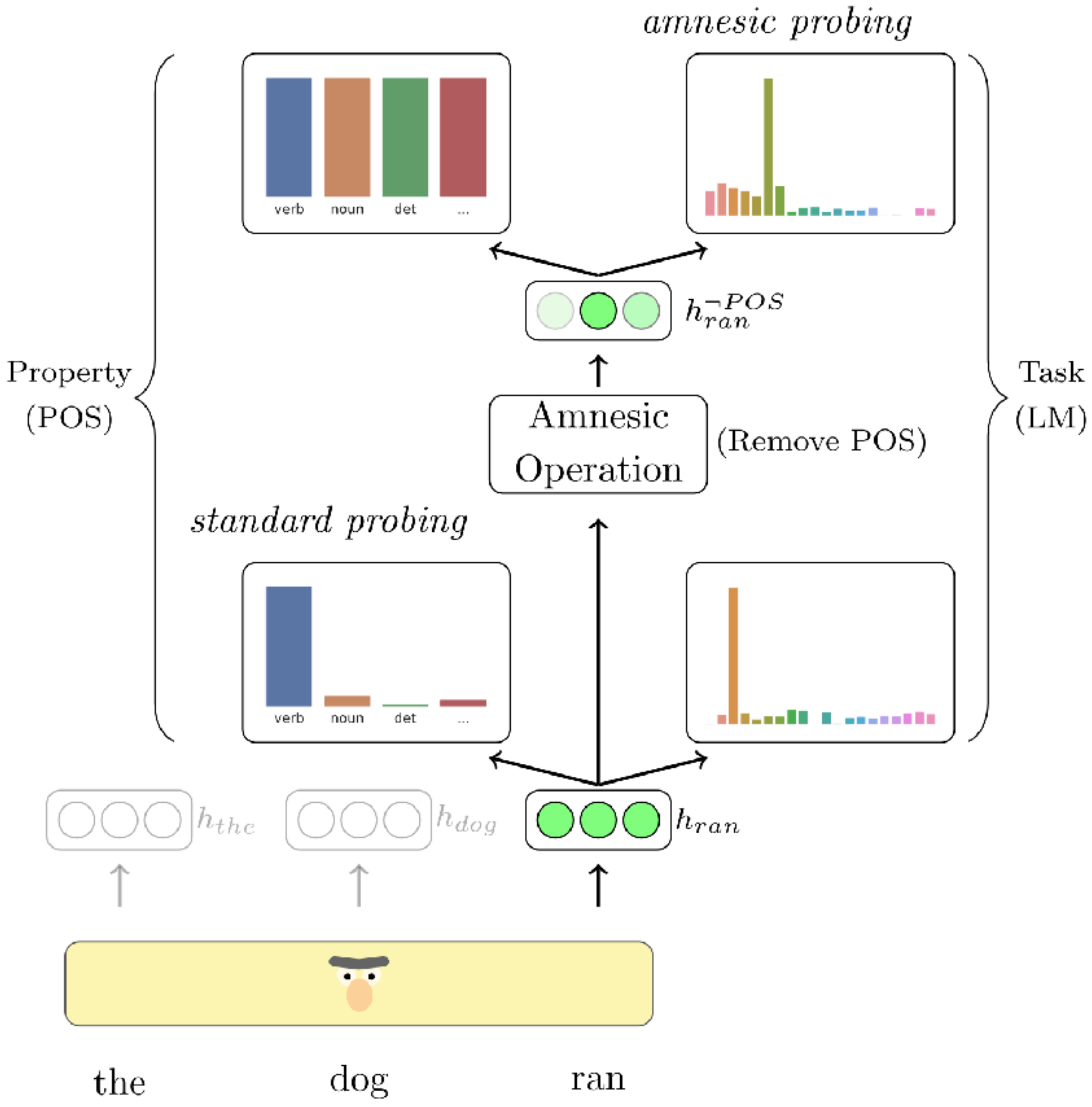
$$\mathbf{h}_i = \sum_{l=0}^{L} s_l \mathbf{h}_i^{(l)}$$



Amnesic probing in neural networks

Inferring functional roles of representations

- systematically intervene with the normal feedforward prediction of a trained model
  - check what happens to relevant task performance
  - interventions can take place at different locations
- sketch of amnestic operation:
  - train a sequence of linear classifiers (SVMs) for task T
  - iteratively remove information useable by classifier for the task
  - terminate when predictive accuracy is at chance level
- include controls (similar amount of deletion but in more arbitrary direction)
  - information
  - selectivity

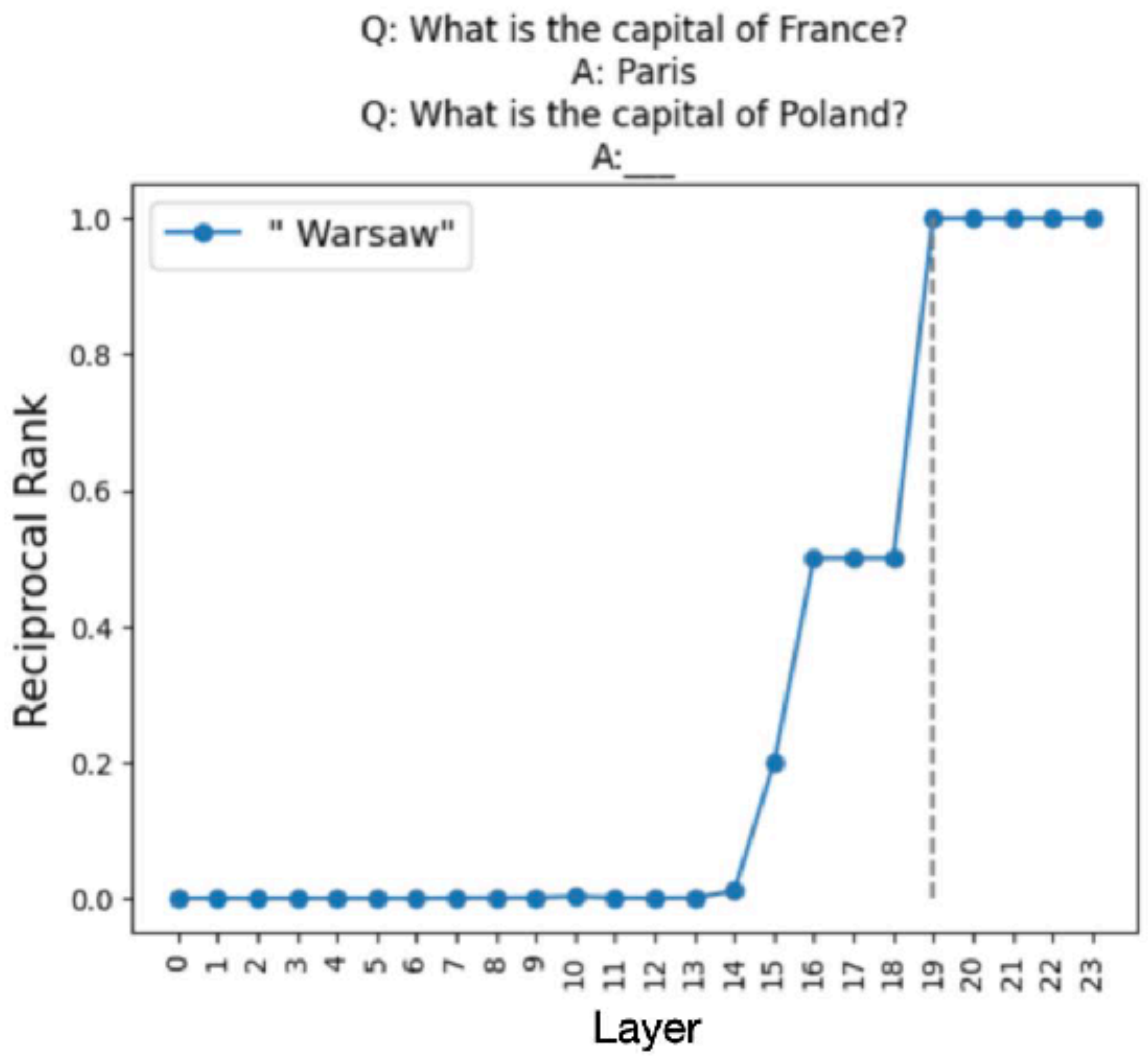


Elazar et al., (2021), Rafvogel et al., (2020)

How do LLMs solve relational tasks?

Merullo, Eickhoff & Pavlick (2023)

- LLMs learn to solve relational tasks in-context by **re-applying the example relation** to new inputs
  - f | f(France) = Paris -> f(Poland) = Warsaw
- critical components for such tasks (capital identification, uppercasing, past tense mapping): transformer block **FFN**, residual stream
- **early decoding** used to identify that the FFN update retrieves the capital (=Warsaw) of a new argument (Poland)
  - applies the ‘function get_capital(Poland)’
- interventions to check this role of the FFN
  - FFN update in other contexts
  - relevant for abstractive, but not extractive tasks



Layer	Top Token
0	(
1	A
2	A
3	A
4	A
5	A
6	No
7	C
8	A
9	A
10	A
11	A
12	Unknown
13	C
14	St
15	Poland
16	Poland
17	Poland
18	Poland
19	Warsaw
20	Warsaw
21	Warsaw
22	Warsaw
23	Warsaw

Reward collapse in RL fine-tuning

Song et al. (2023)

- current **reward model training objective** (based on ranking of responses) leads to reward collapse
  - **identical reward distributions** for inputs where distinct distributions expected (open-ended vs. closed-ended tasks)
  - problematic utility function: $U = \log \text{sigmoid}(\frac{R_w - R_l}{\sigma})$
- proposed mitigation: **prompt-aware utility functions**
  - $U_{\text{closed}} = x$ (polarized distribution)
  - $U_{\text{open}} = \frac{-1}{x}$ (more uniform distribution)
- (artificial task) experiment with response length as reward

Reward collapse in RL fine-tuning

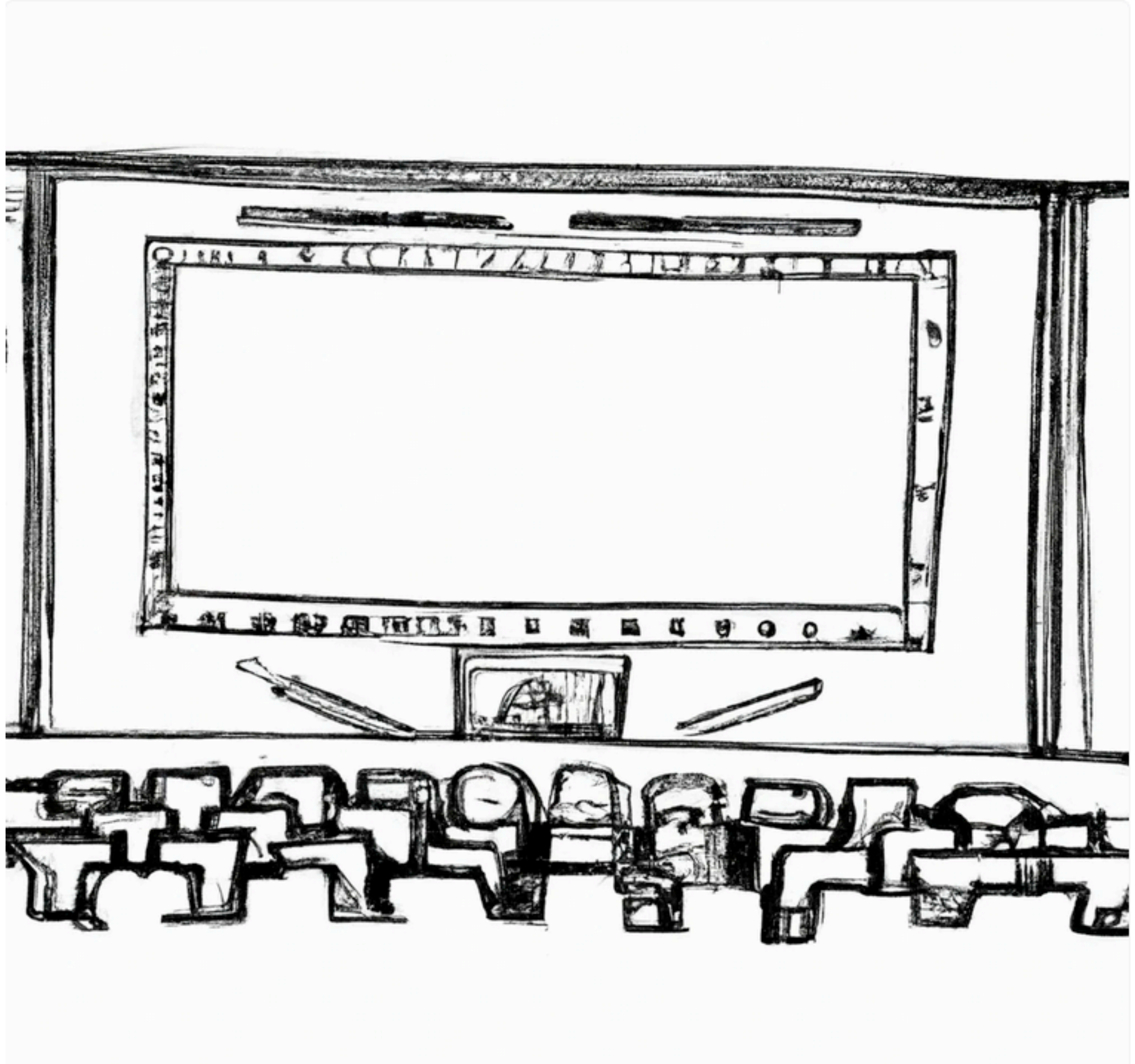
Song et al. (2023)

- current **reward model training objective** (based on ranking of responses) leads to reward collapse
  - **identical reward distributions** for inputs where distinct distributions expected (open-ended vs. closed-ended tasks)
  - problematic utility function: $U = \log \text{sigmoid}(\frac{R_w - R_l}{\sigma})$
- proposed mitigation: **prompt-aware utility functions**
  - $U_{\text{closed}} = x$ (polarized distribution)
  - $U_{\text{open}} = \frac{-1}{x}$ (more uniform distribution)
- (artificial task) experiment with response length as reward
- could alleviate **miscalibration** of LLM responses

Presentations
Your job

During the presentation, think about the following questions:

- What is honesty for you? Is it addressed with the proposed methodology?
- (When) can we trust LLM outputs?



Calibration of natural text generation

Evaluating Uncertainty in Neural Text Generators Against Human Production Variability

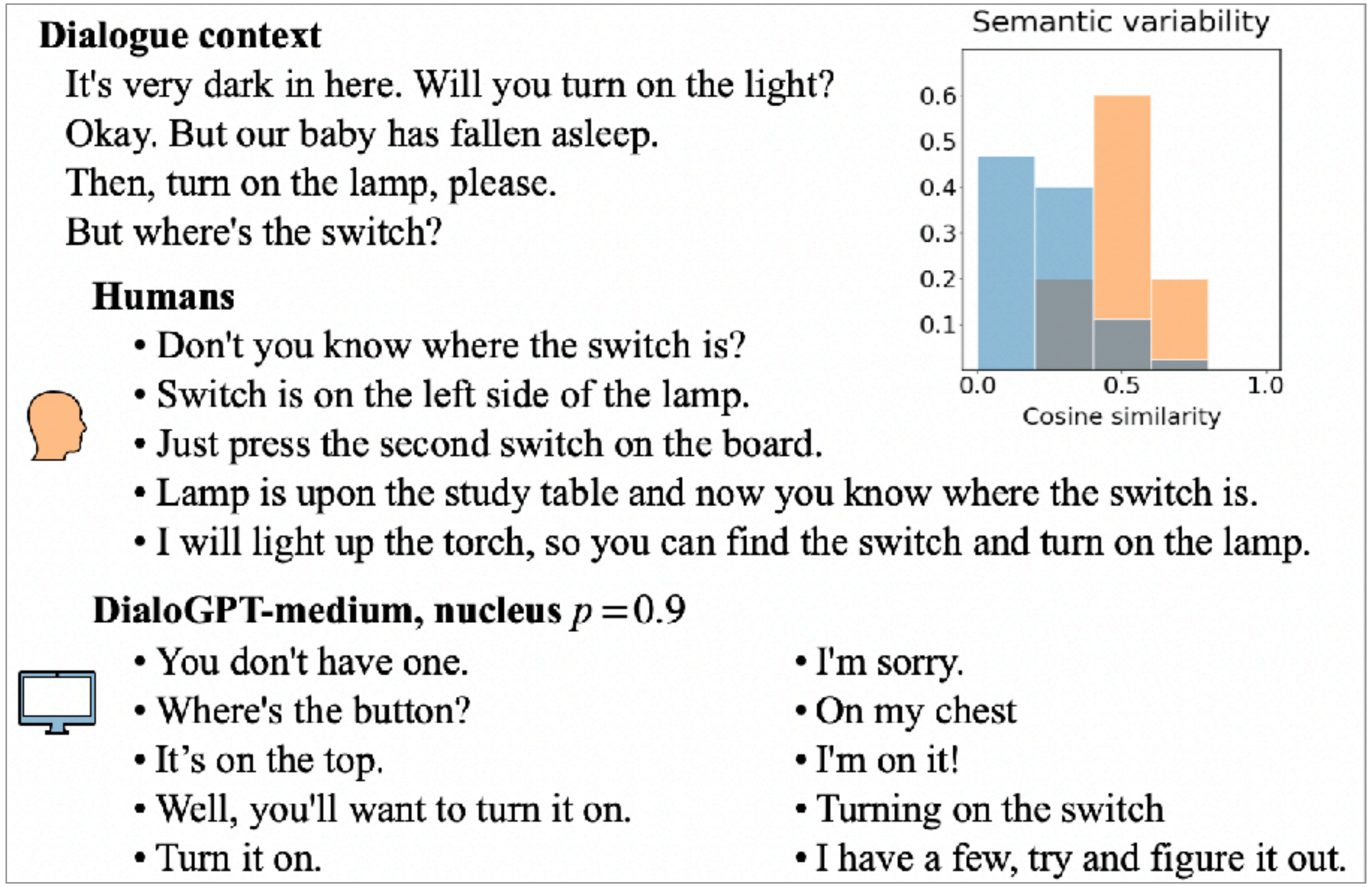
- main idea:
  - humans often express the same message in varying ways
    - NLG systems should capture the same variability
  - compare LLM and human production distributions
- methods:
  - comparison of productions via statistical similarity and different decoding schemes
    - lexical, syntactic, and semantic variability
  - distribution variability assessment and comparison via production probes
    - sample-based joint distribution approx. similarity between two outputs
  - similarity metrics:
    - unigram overlap (lexical)
    - POS bigram overlap (syntactic)
    - sentence embedding cosine similarity (semantic)

Giulianelli et al. (2023)

“Good” uncertainty in text generation

Production variability

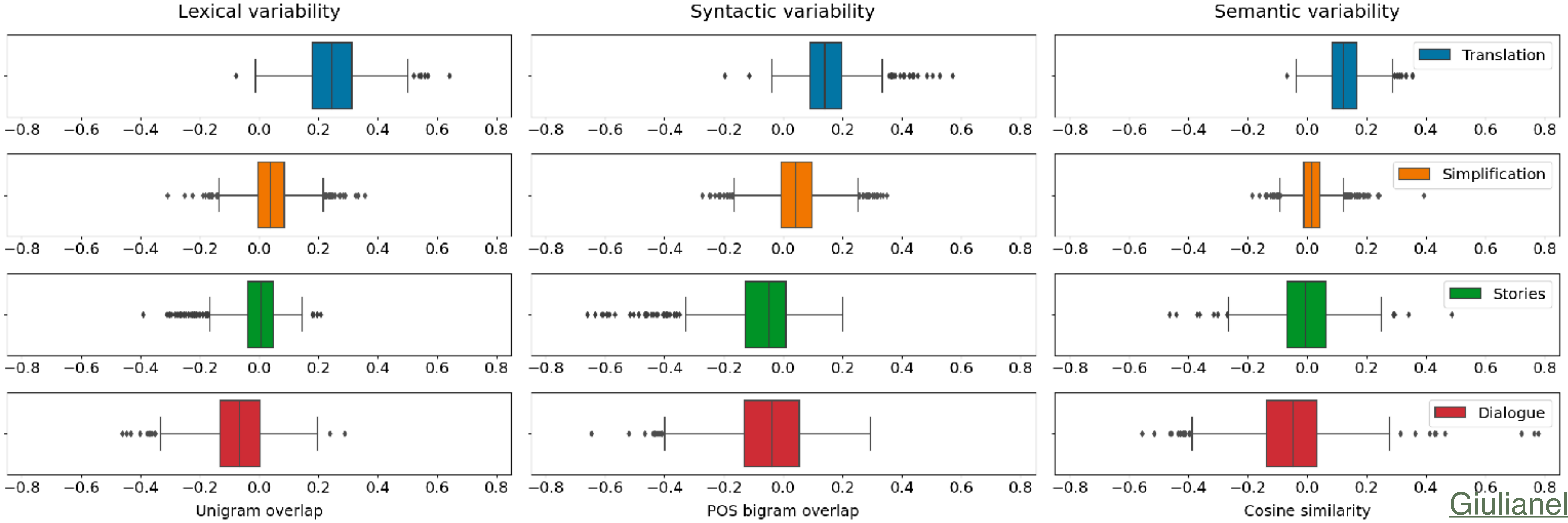
- tasks
  - machine translation
  - text simplification
  - story generation
  - open domain dialogue
- decoding schemes:
  - unbiased samples
  - temperature scaling
  - top-k sampling
  - nucleus sampling
  - locally typical sampling
- models:
  - Transformer-Align
  - Flan-T5
  - GPT-2
  - DialoGPT



Giulianelli et al. (2023)

“Good” uncertainty in text generation
Production variability

- measures
  - human variability
  - LLM variability
  - human-LLM cross-variability
- variability comparison:
  - μ_{human} : e.g. $\cos sim(y_i, y_j)$ for y_i, y_j completions for prompt x
  - $\mu_{LLM} - \mu_{human}$



Giulianelli et al. (2023)

Evaluating RL agents

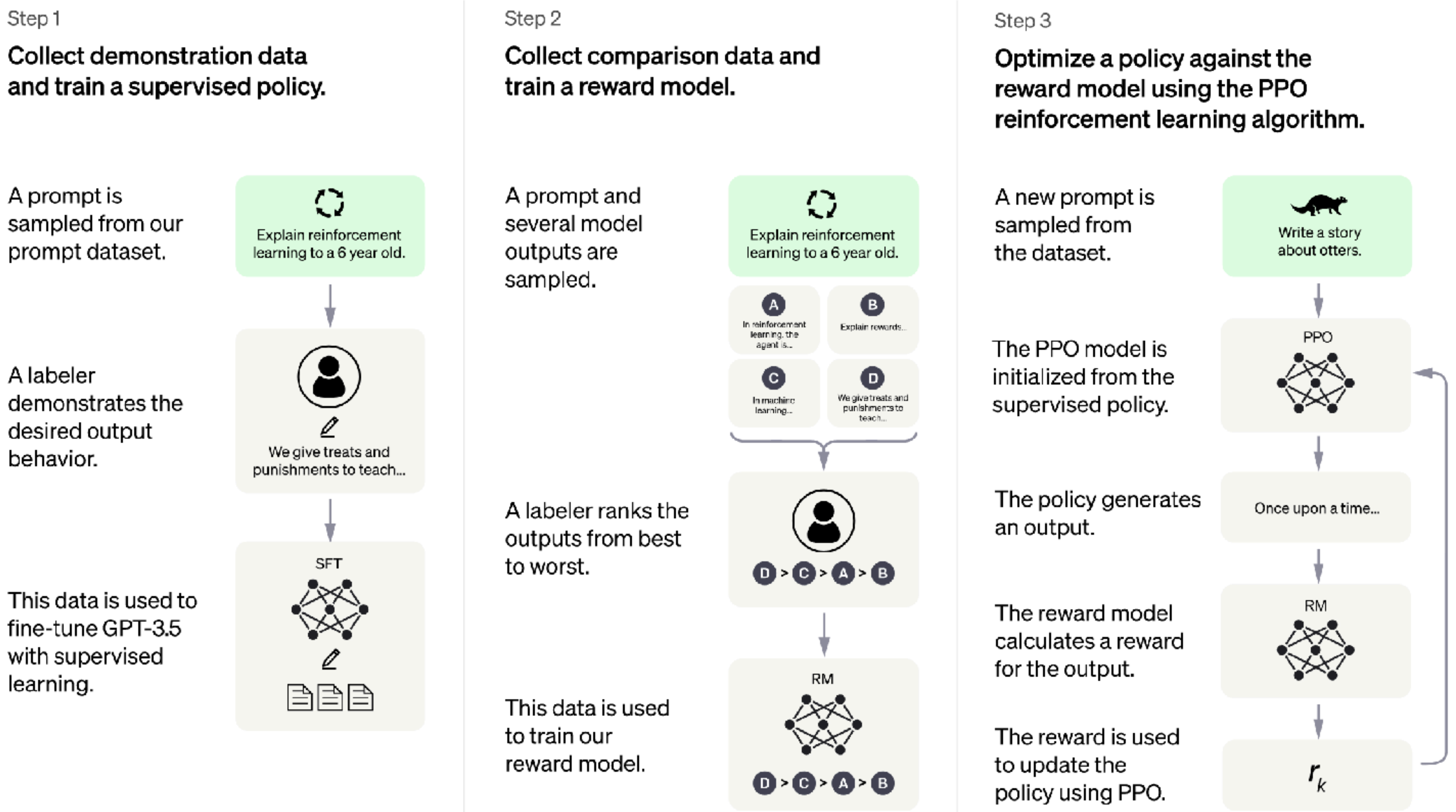
Gymnasium

- **goal of RL agent training**: agent has learned to achieve a goal
  - LLMs: training helpful, harmless and honest agents
- evaluation aspects depend on the goals of the system, but generally:
  - performance of algorithm on **standard environments** like the OpenAI Gym(nasium) / Arcade
    - mean / median / cumulative training and test rewards / scores
    - relative to baseline, optimum or random behavior
  - downstream task performance
    - LLMs: comparative paradigm with pretrained LLMs
    - LLMs: evaluation of **alignment** via human annotations
- **alignment:** agent's goals are congruent with human goals
  - congruent ranking of outcomes (Askell et al., 2021)
  - rewards don't provide information about **how** a goal should be achieved!
    - reward hacking / faulty reward functions: example

RL Gymnasium, RLiable blogpost

Human feedback in RL

RLHF

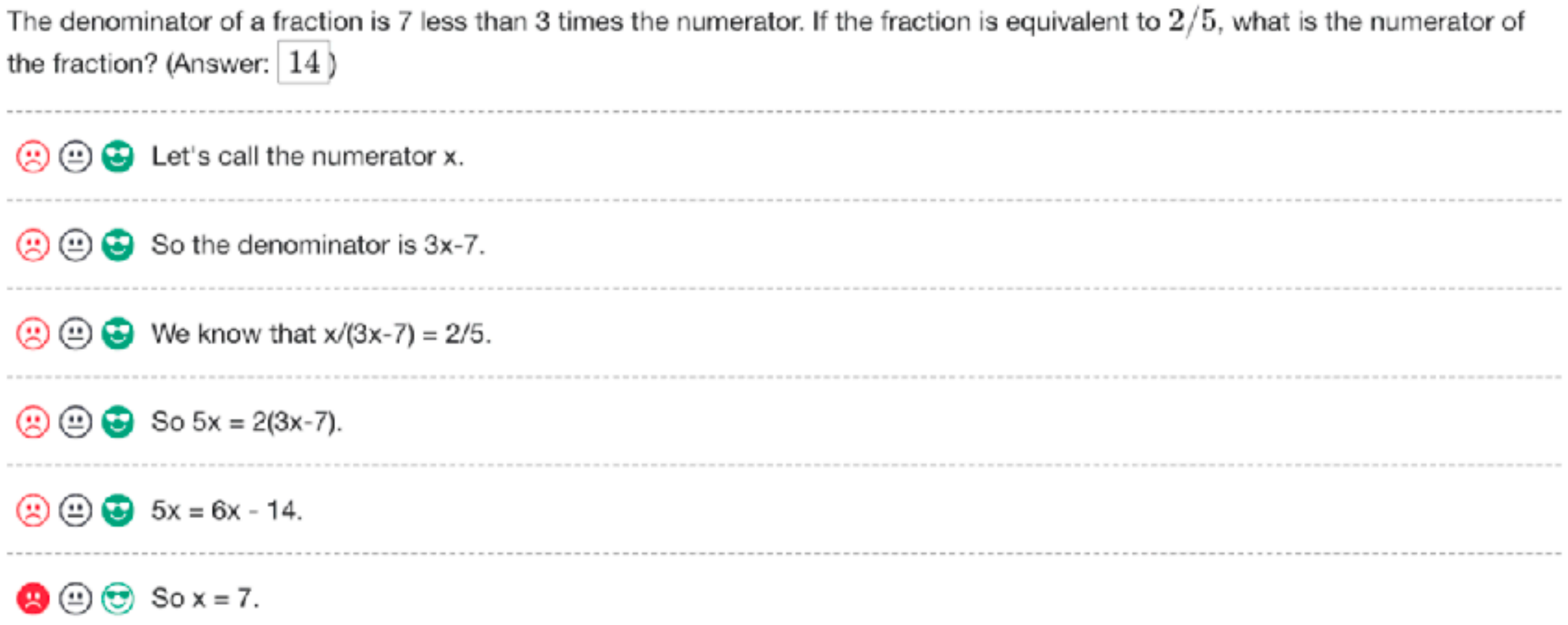


OpenAI (2022)

Process-supervised reward models

“Reasoning calibration”

- **problem**: standard (outcome-supervised) reward models only score the result of solution process (CoT)
  - model could be right for the wrong reasons! (hallucinations)
- **idea**: alleviate via **process-supervised reward models** which score the solution process
- **set up**:
  - train RM on MATH dataset with final solutions and human-annotated intermediate step solution evaluations (PRM800K for 12K problems)
  - evaluate accuracy of top-N response with highest reward (500 test problems)

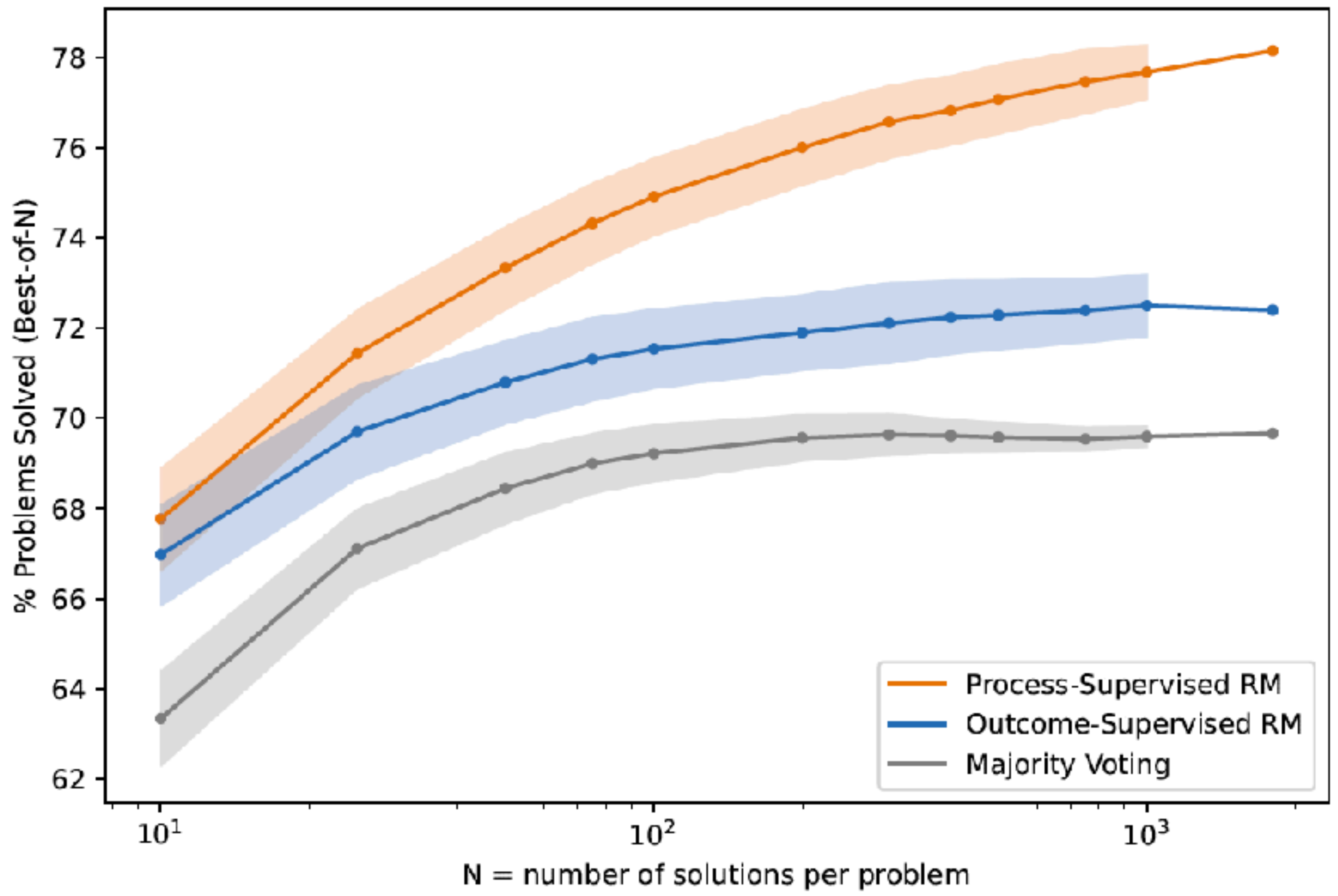


Lightman et al. (2023)

Process-supervised reward models

"Reasoning calibration"

- fixed policy LLM (pretrained GPT-4)
- process-supervised reward model:
  - base pretrained GPT-4
  - fine-tuning 1: on MathMix (1.5B tokens); fine-tuning 2: to produce stepwise solutions
  - next-token prediction training up to first mistake
- outcome-supervised baseline reward model:
  - base pretrained GPT-4
  - trained on MATH to predict correctness of outcome (100 samples / problem from GPT-4)
- evaluation of data efficiency and OOD generalisation
- **no evaluation of solution steps correctness!**



Lightman et al. (2023)

Summary

Calibration & RL evaluation

- **calibration** of LLMs reflects how well their probability predictions match ‘true’ outcome probabilities
  - approximates LLMs’ ‘knowledge’ certainty
- natural language generation exhibits variability
  - LM generations’ variability is not well-calibrated wrt. human variability
- for RL fine-tuning, we might want to ‘calibrate’ the RMs’ scores so as to reflect the solution process accuracy
  - idea: train process-supervised RMs

Feedback time

I would love to hear your feedback regarding the class!

https://forms.gle/hs4bFy4WVuJNimTM6

Please fill out the form by December 26th :)