Natural Language Processing

Diyi Yang

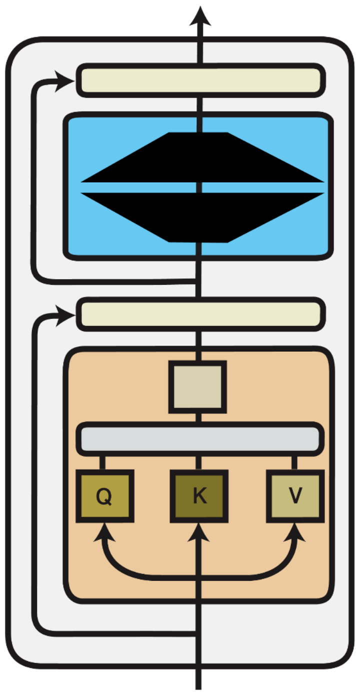
The Machine Learning Summer School in Okinawa 2024

Overview

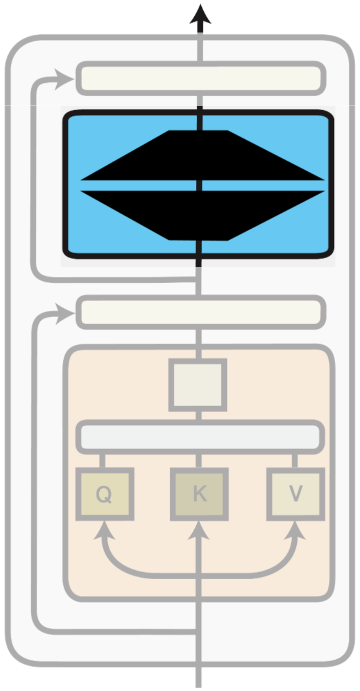
- **Part 1: Basics in NLP**
  - ✓Introduction to NLP (10 mins)
  - ✓Different NLP tasks (10 mins)
  - ✓Word2vec (25 mins)
  - ✓Pretrained LLMs (15 mins)
  - ✓In-context learning (15 mins)

- **Part 2: Advanced topics in NLP**
  - ☐**Parameter efficient fine-tuning for NLP models (40 mins)**
  - ☐**Learning from human feedback (40 mins)**

From fine-tuning to parameter-efficient fine-tuning (PEFT)



Full Fine-tuning
Update **all model parameters**



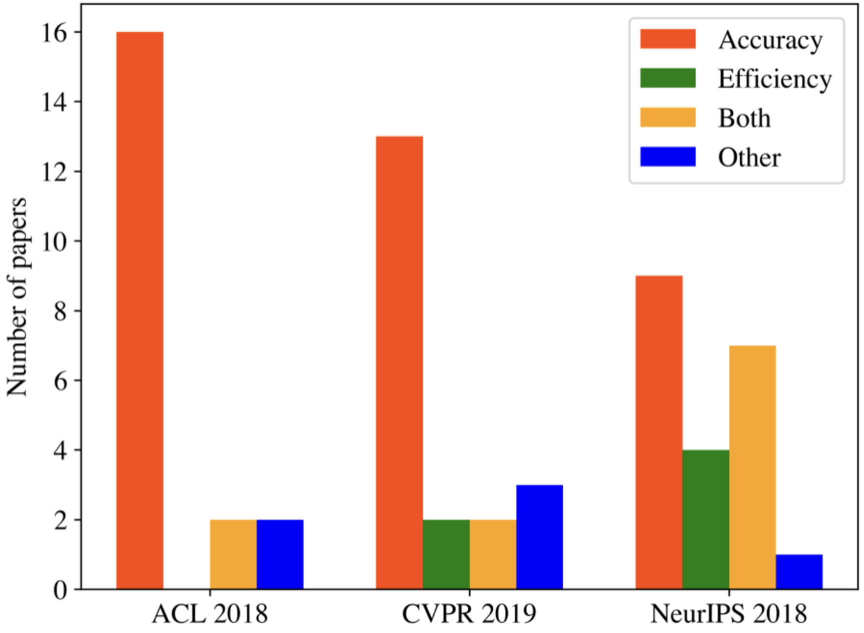
Parameter-efficient Fine-tuning
Update a **small subset** of model parameters

Why fine-tuning *only some* parameters?

1. Fine-tuning all parameters is impractical with large models
2. State-of-the-art models are massively over-parameterized
   → Parameter-efficient fine-tuning matches performance of full fine-tuning

Why do we need efficient adaptation?

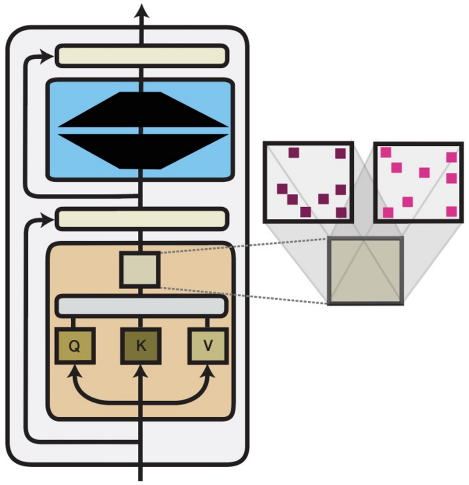
1. Emphasis on accuracy over efficiency in current AI paradigm
2. Hidden environmental costs of training (and fine tuning) LLMs
3. As costs of training go up, AI development becomes concentrated in well-funded organizations, especially in industry



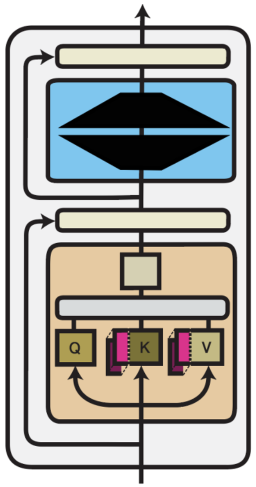
AI papers tend to target accuracy rather than efficiency. The figure shows the proportion of papers that target accuracy, efficiency, both or other from a sample of 60 papers from top AI conferences (Green AI)

Slides credit to Benji Xie and Regina Wang

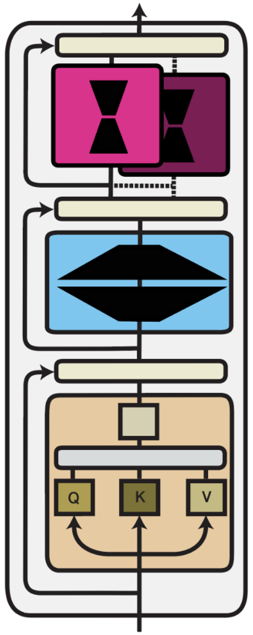
Different perspectives to think about PEFT



Parameter



Input



Function

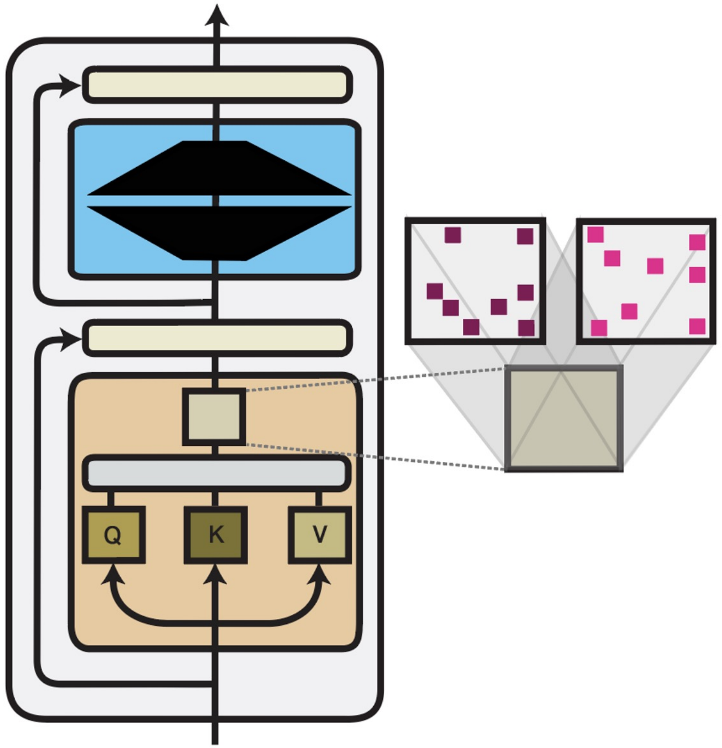
Slides adapted from Ruder, Sebastian, Jonas Pfeiffer, and Ivan Vulić

on their EMNLP 2022 Tutorial on "Modular and Parameter-Efficient Fine-Tuning for NLP Models". For details, check out: https://www.modulardeeplearning.com/

A Parameter Perspective of Adaptation

1. Sparse Subnetworks

2. Low-rank Composition

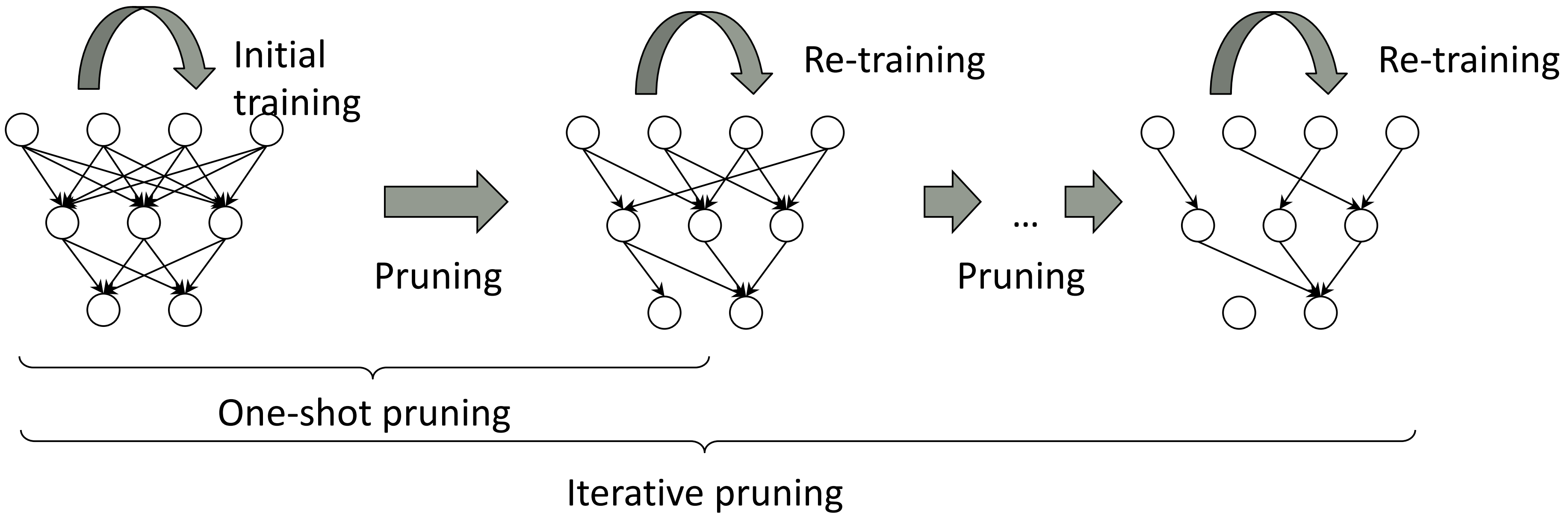


Sparse subnetworks

- A common inductive bias on the module parameters is **sparsity**
- Most common sparsity method: **pruning**
- Pruning can be seen as applying a binary mask $\mathbf{b} \in \{0, 1\}^{|\theta|}$ that selectively keeps or removes each connection in a model and produces a subnetwork.
- Most common pruning criterion: **weight magnitude** [Han et al., 2017]

Pruning

- During pruning, a fraction of the lowest-magnitude weights are removed
- The non-pruned weights are re-trained
- Pruning for multiple iterations is more common [(Frankle & Carbin, 2019)]



Pruning and Binary Mask

- We can also view pruning as adding a task-specific vector ϕ to the parameters of an existing model $f'_\theta = f_{\theta+\phi}$ where $\phi_i = 0$ if $b_i = 0$

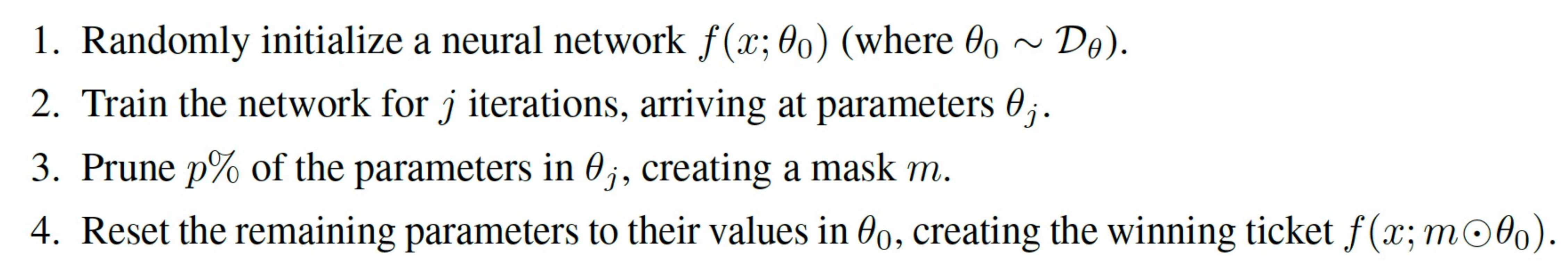
- If the final model should be sparse, we can multiply the existing weights with the binary mask to set the pruned weights to 0: $f'_\theta = f_{\theta \circ \boldsymbol{b} + \phi}$. These weight values were moving to 0 anyway [Zhou et al., 2019]

Element-wise product (Hadamard product)

- **Diff pruning:** we can perform pruning only based on the magnitude of the module parameters ϕ rather than the updated $\theta + \phi$ parameters [Guo et al., 2021]

The Lottery Ticket Hypothesis

- Dense, randomly-initialized models **contain subnetworks** (“winning tickets”) that—when trained in isolation—**reach test accuracy comparable to the original network** in a similar number of iterations [Frankle & Carbin, 2019]

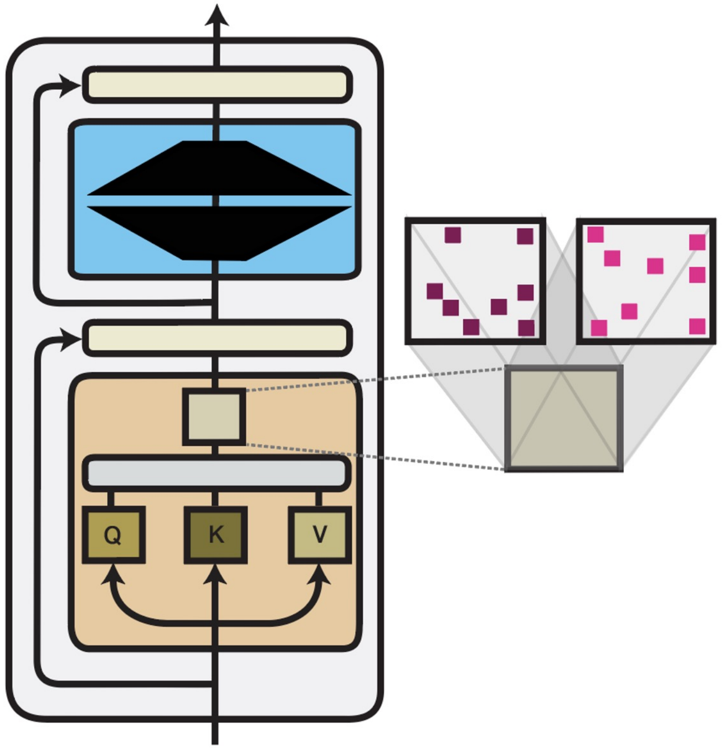


1. Randomly initialize a neural network $f(x; \theta_0)$ (where $\theta_0 \sim \mathcal{D}_\theta$).
2. Train the network for j iterations, arriving at parameters θ_j.
3. Prune $p\%$ of the parameters in θ_j, creating a mask m.
4. Reset the remaining parameters to their values in θ_0, creating the winning ticket $f(x; m \odot \theta_0)$.

- Sparsity ratios: from 40% (SQuAD) to 90% (QQP and WNLI)
- Subnetworks trained on a general task such as masked language modelling transfer best

A Parameter Perspective of Adaptation

- ✓ Sparse Subnetworks
- Low-rank Composition



Revisit the full fine-tuning

- Assume we have a pre-trained autoregressive language model $P_\phi(y|x)$
  - E.g., GPT based on Transformer

- Adapt this pretrained model to downstream tasks (e.g., summarization, NL2SQL, reading comprehension)
  - Training dataset of context-target pairs $\{(x_i, y_i)\}_{i=1,\ldots,N}$

- During full fine-tuning, we update ϕ_o to $\phi_o + \Delta\phi$ by following the gradient to maximize the conditional language modeling objective

$$\max_{\phi} \sum_{(x,y)} \sum_{t=1}^{|y|} \log(P_\phi(y_t|x, y_{<t}))$$

LoRA: low rank adaptation (Hu et al., 2021)

- For each downstream task, we learn a different set of parameters $\Delta\phi$
  - $|\Delta\phi| = |\phi_o|$
  - GPT-3 has a $|\phi_o|$ of 175 billion
  - Expensive and challenging for storing and deploying many independent instances

- Key idea: encode the task-specific parameter increment $\Delta\phi = \Delta\phi(\Theta)$ by a smaller-sized set of parameters Θ, $|\Theta| \ll |\phi_o|$

- The task of finding $\Delta\phi$ becomes optimizing over Θ

$$\max_{\Theta} \sum_{(x,y)} \sum_{t=1}^{|y|} \log\left(P_{\phi_o+\Delta\phi(\Theta)}\left(y_t | x, y_{<t}\right)\right)$$

Low-rank-parameterized update matrices

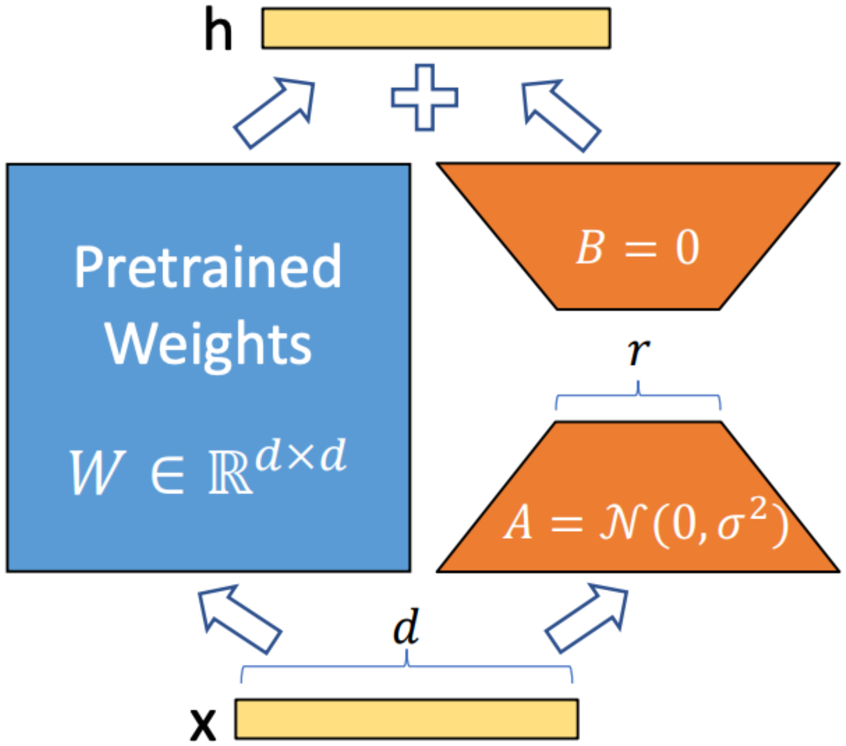
- $W_0 \in \mathbb{R}^{d \times k}$: a pretrained weight matrix

- Constrain its update with a low-rank decomposition:

$$W_0 + \Delta W = W_0 + BA$$

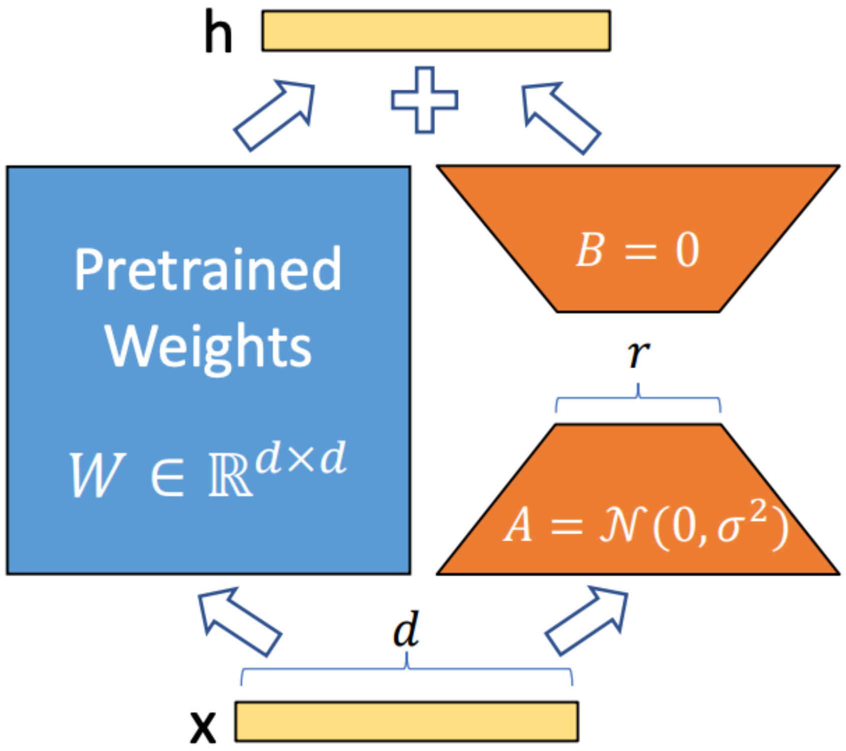
where $B \in \mathbb{R}^{d \times r}, A \in \mathbb{R}^{r \times k}, r \ll \min(d, k)$

- Only A and B contain **trainable** parameters

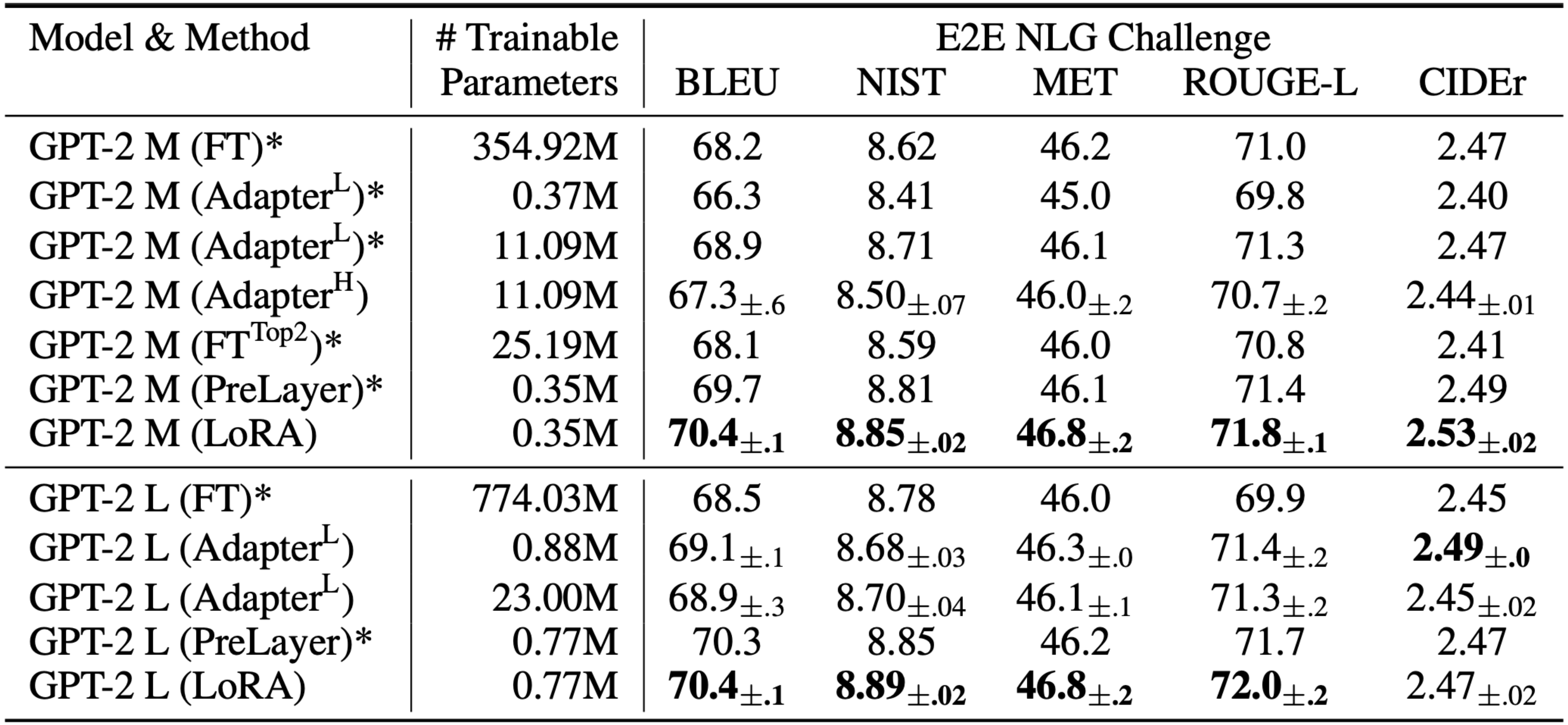

h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Low-rank-parameterized update matrices

- As one increase the number of trainable parameters, training LoRA converges to training the original model
- **No additional inference latency:** when switching to a different task, recover W_0 by subtracting BA and adding a different $B'A'$
- Often LoRA is applied to the weight matrices in the self-attention module


h
Pretrained Weights
$W \in \mathbb{R}^{d\times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x

Applying LoRA to Transformer



Model & Method	# Trainable Parameters	E2E NLG Challenge BLEU	NIST	MET	ROUGE-L	CIDEr
GPT-2 M (FT)*	354.92M	68.2	8.62	46.2	71.0	2.47
GPT-2 M (Adapter^{L})*	0.37M	66.3	8.41	45.0	69.8	2.40
GPT-2 M (Adapter^{L})*	11.09M	68.9	8.71	46.1	71.3	2.47
GPT-2 M (Adapter^{H})	11.09M	67.3$_{\pm.6}$	8.50$_{\pm.07}$	46.0$_{\pm.2}$	70.7$_{\pm.2}$	2.44$_{\pm.01}$
GPT-2 M (FT$^{\text{Top2}}$)*	25.19M	68.1	8.59	46.0	70.8	2.41
GPT-2 M (PreLayer)*	0.35M	69.7	8.81	46.1	71.4	2.49
GPT-2 M (LoRA)	0.35M	**70.4**$_{\pm.1}$	**8.85**$_{\pm.02}$	**46.8**$_{\pm.2}$	**71.8**$_{\pm.1}$	**2.53**$_{\pm.02}$
GPT-2 L (FT)*	774.03M	68.5	8.78	46.0	69.9	2.45
GPT-2 L (Adapter^{L})	0.88M	69.1$_{\pm.1}$	8.68$_{\pm.03}$	46.3$_{\pm.0}$	71.4$_{\pm.2}$	**2.49**$_{\pm.0}$
GPT-2 L (Adapter^{L})	23.00M	68.9$_{\pm.3}$	8.70$_{\pm.04}$	46.1$_{\pm.1}$	71.3$_{\pm.2}$	2.45$_{\pm.02}$
GPT-2 L (PreLayer)*	0.77M	70.3	8.85	46.2	71.7	2.47
GPT-2 L (LoRA)	0.77M	**70.4**$_{\pm.1}$	**8.89**$_{\pm.02}$	**46.8**$_{\pm.2}$	**72.0**$_{\pm.2}$	2.47$_{\pm.02}$

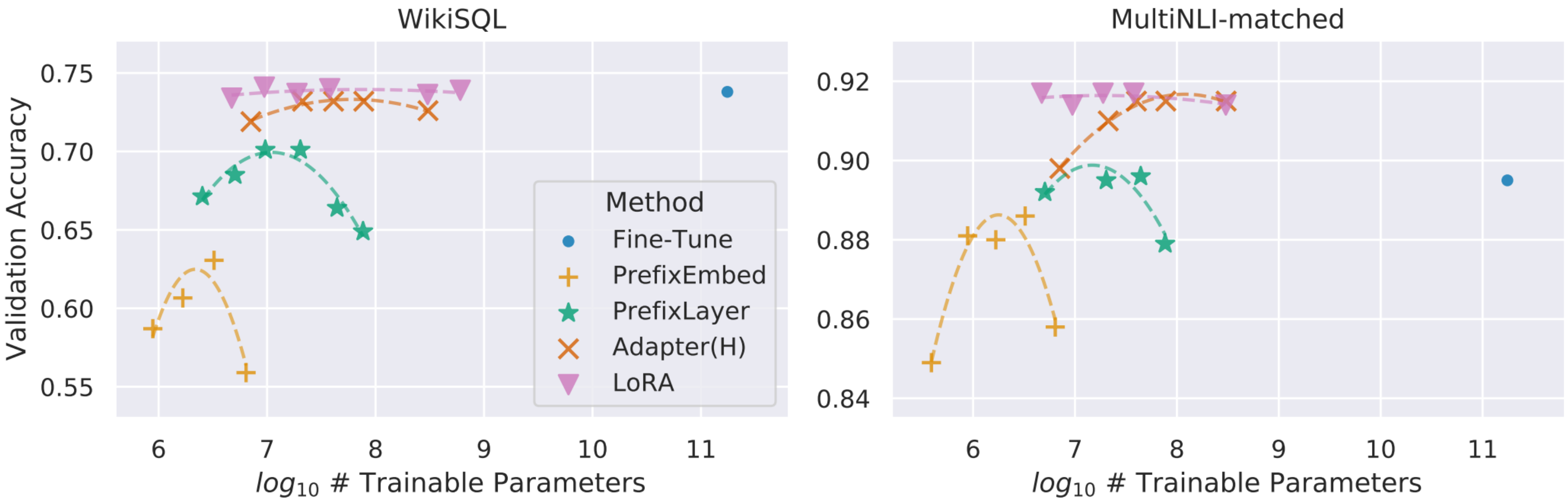
GPT-2 medium (M) and large (L) with different adaptation methods on the E2E NLG Challenge. For all metrics, higher is better. LoRA outperforms several baselines with comparable or fewer trainable parameters

Hu, Edward J., Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. "Lora: Low-rank adaptation of large language models." *arXiv preprint arXiv:2106.09685* (2021).

Scaling up to GPT-3 175B

Model&Method	# Trainable Parameters	WikiSQL	MNLI-m	SAMSum
		Acc. (%)	Acc. (%)	R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 (Adapter^H)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 (Adapter^H)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

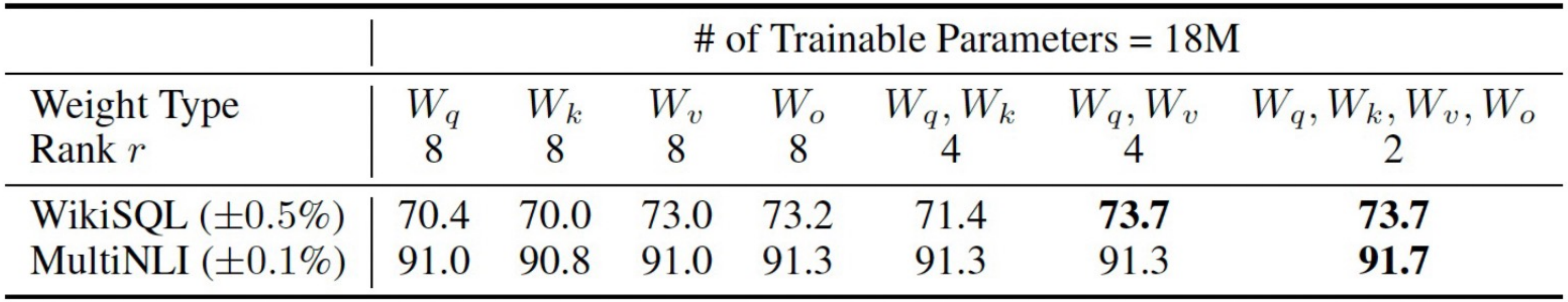
LoRA matches or exceeds the fine-tuning baseline on all three datasets



LoRA exhibits better scalability and task performance.

Understanding low-rank adaptation

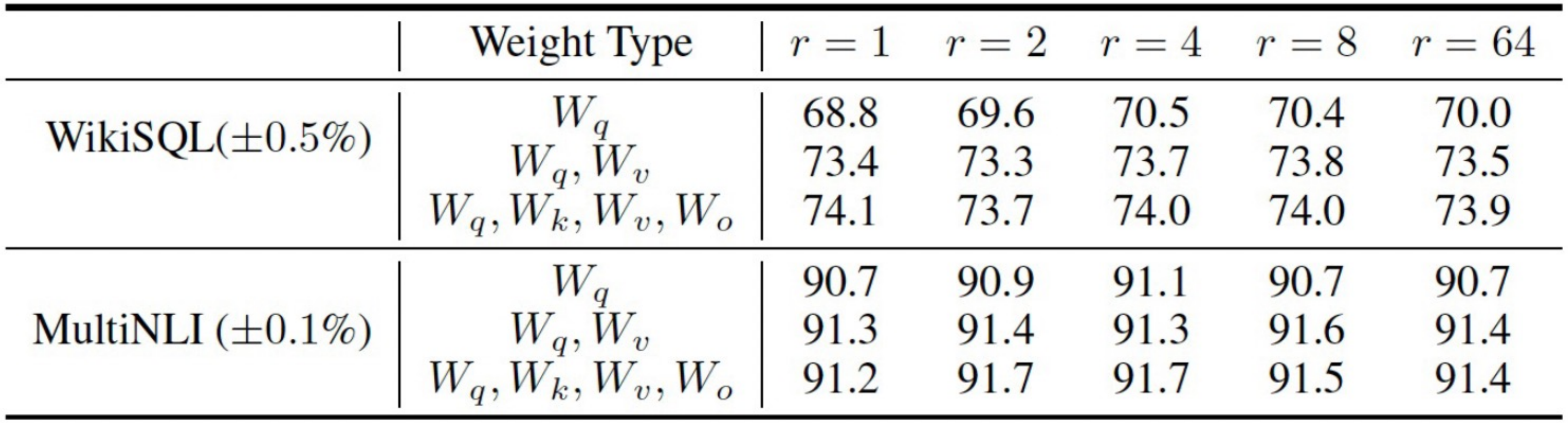
Which weight matrices in Transformers should we apply LoRA to?



	# of Trainable Parameters = 18M						
Weight Type Rank r	W_q 8	W_k 8	W_v 8	W_o 8	W_q, W_k 4	W_q, W_v 4	W_q, W_k, W_v, W_o 2
WikiSQL ($\pm 0.5\%$)	70.4	70.0	73.0	73.2	71.4	**73.7**	**73.7**
MultiNLI ($\pm 0.1\%$)	91.0	90.8	91.0	91.3	91.3	91.3	**91.7**

Adapting both Wq and Wv gives the best performance overall.

What is the optimal rank r for LoRA?



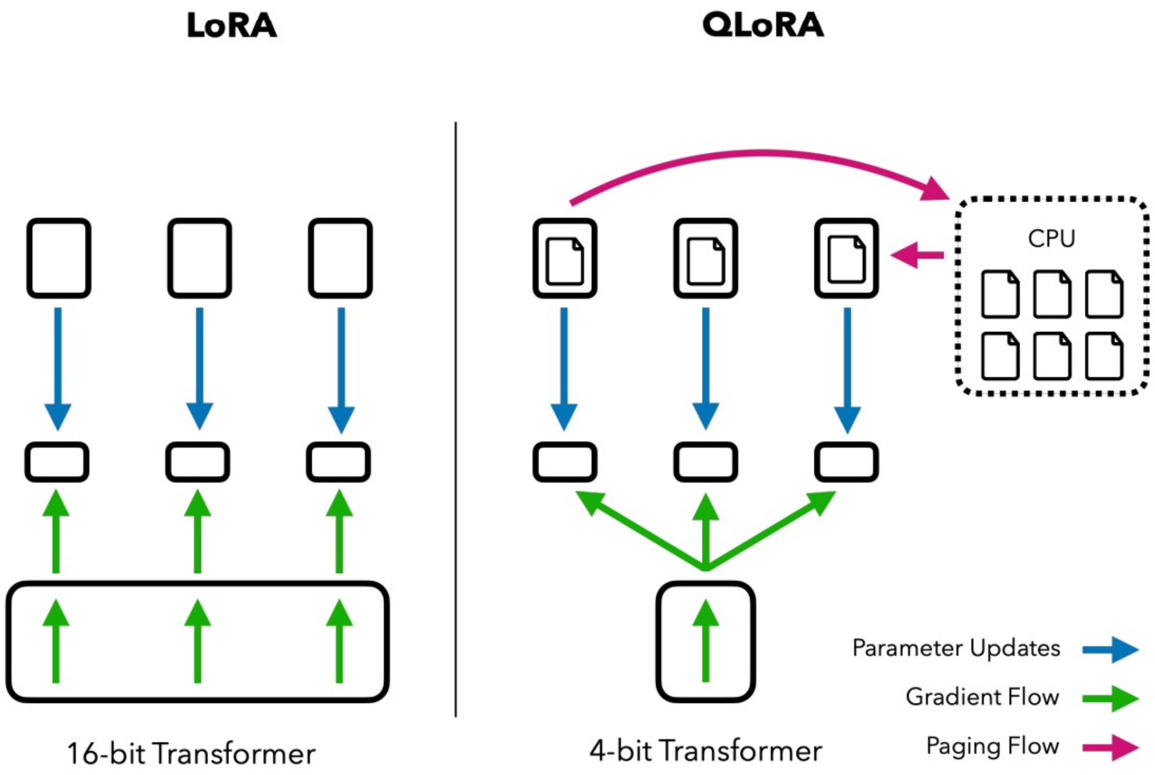
	Weight Type	$r = 1$	$r = 2$	$r = 4$	$r = 8$	$r = 64$
WikiSQL($\pm 0.5\%$)	W_q	68.8	69.6	70.5	70.4	70.0
	W_q, W_v	73.4	73.3	73.7	73.8	73.5
	W_q, W_k, W_v, W_o	74.1	73.7	74.0	74.0	73.9
MultiNLI ($\pm 0.1\%$)	W_q	90.7	90.9	91.1	90.7	90.7
	W_q, W_v	91.3	91.4	91.3	91.6	91.4
	W_q, W_k, W_v, W_o	91.2	91.7	91.7	91.5	91.4

LoRA already performs competitively with a very small r

From LoRA to QLoRA

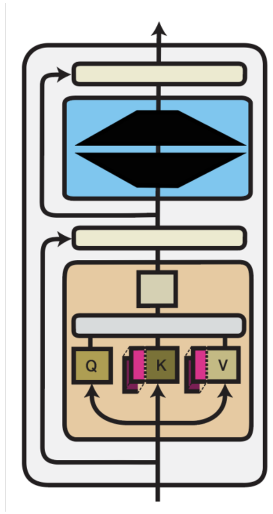
- QLORA improves over LoRA by quantizing the transformer model to 4-bit precision and using paged optimizer to handle memory spikes

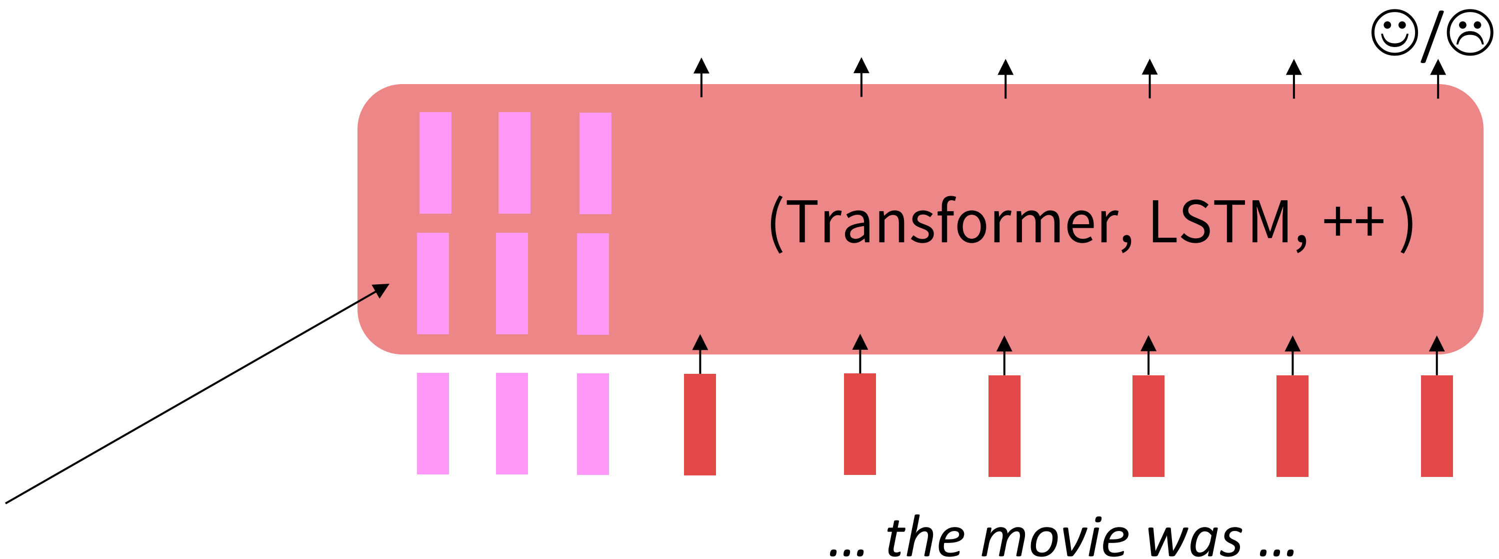
- 4-bit NormalFloat (NF4)
  - A new data type that is information theoretically optimal for normally distributed weights



Dettmers, Tim, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. "Qlora: Efficient finetuning of quantized llms." arXiv preprint arXiv:2305.14314 (2023).

An input perspective of adaptation: Prefix-Tuning



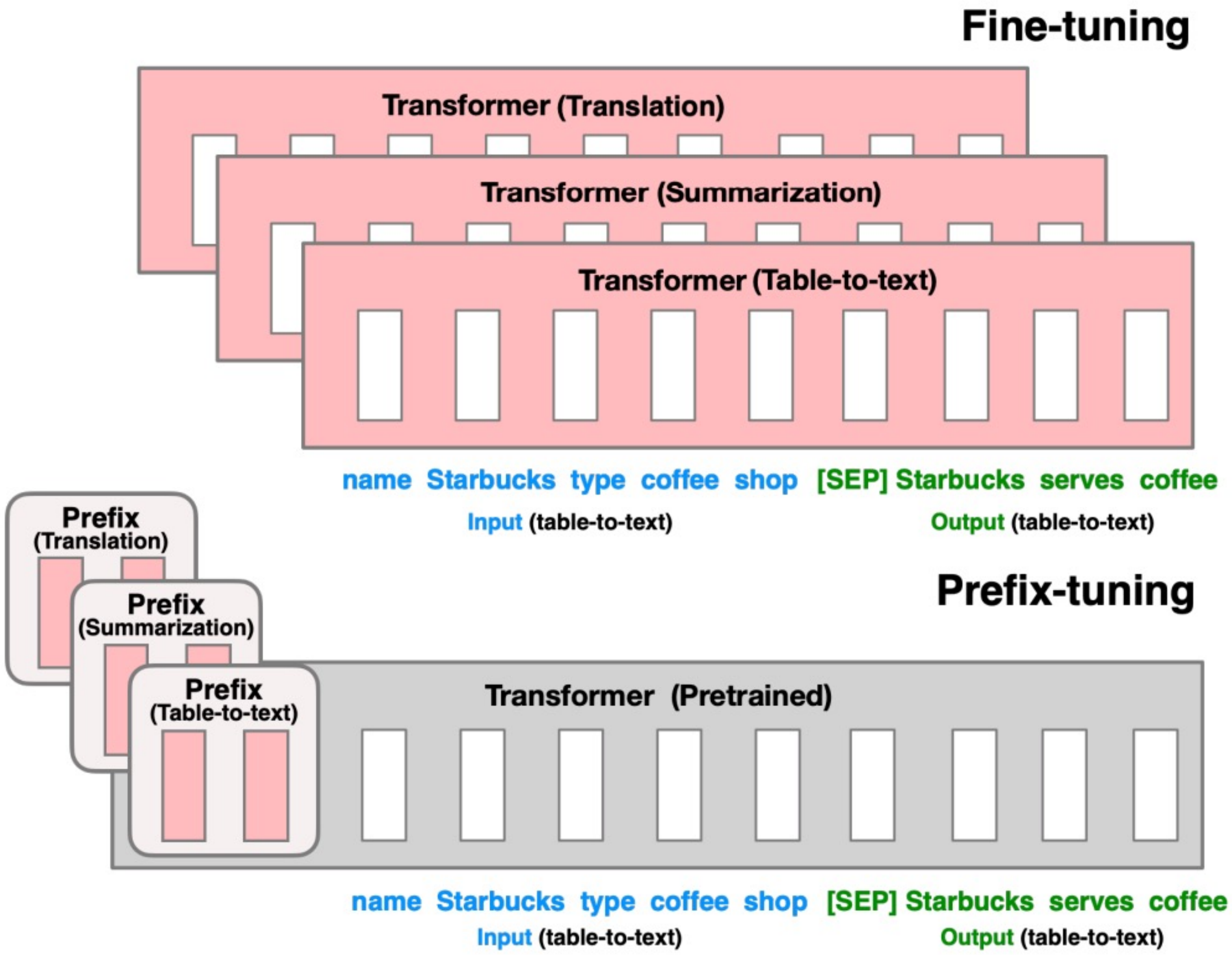


Learnable prefix parameters

[Li and Liang, 2021; Lester et al., 2021]

Prefix-Tuning, Prompt tuning

- Prefix-Tuning adds a **prefix** of parameters, and **freezes all pretrained parameters.**
- The prefix is processed by the model just like real words would be.
- Advantage: each element of a batch at inference could run a different tuned model.

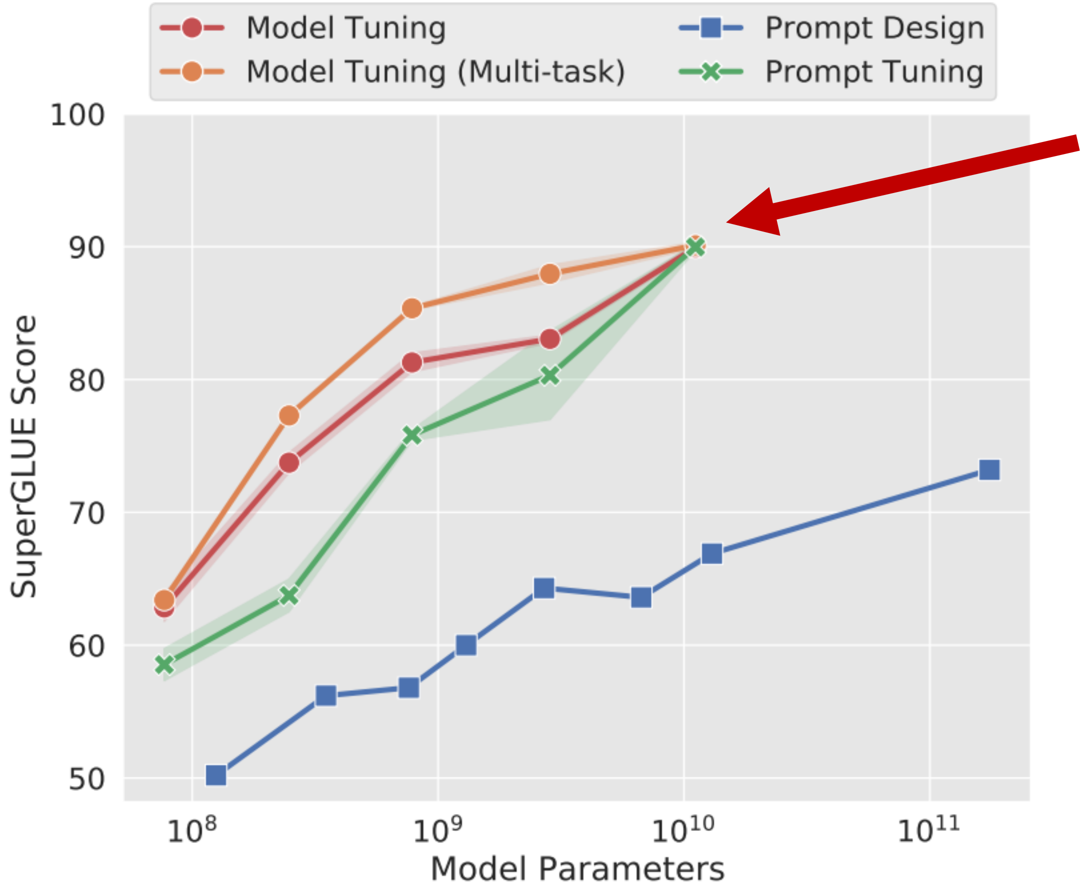

Fine-tuning
Transformer (Translation)
Transformer (Summarization)
Transformer (Table-to-text)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)
Prefix-tuning
Prefix (Translation)
Prefix (Summarization)
Prefix (Table-to-text)
Transformer (Pretrained)
name Starbucks type coffee shop [SEP] Starbucks serves coffee
Input (table-to-text)
Output (table-to-text)

Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." arXiv preprint arXiv:2101.00190 (2021).

Prompt tuning only works well at scale

- Only using trainable parameters at the input layer limits capacity for adaptation

- Prompt tuning performs poorly at smaller model sizes and on harder tasks



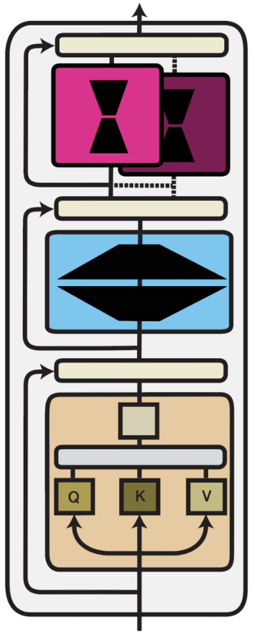
Lester, Brian, Rami Al-Rfou, and Noah Constant. "The power of scale for parameter-efficient prompt tuning." arXiv preprint arXiv:2104.08691 (2021).

A functional perspective of adaptation

- Function composition augments a model's functions with new task-specific functions:

$$f_i'(\boldsymbol{x}) = f_{\theta_i}(\boldsymbol{x}) \odot f_{\phi_i}(\boldsymbol{x})$$

- Most commonly used in multi-task learning where modules of different tasks are composed.

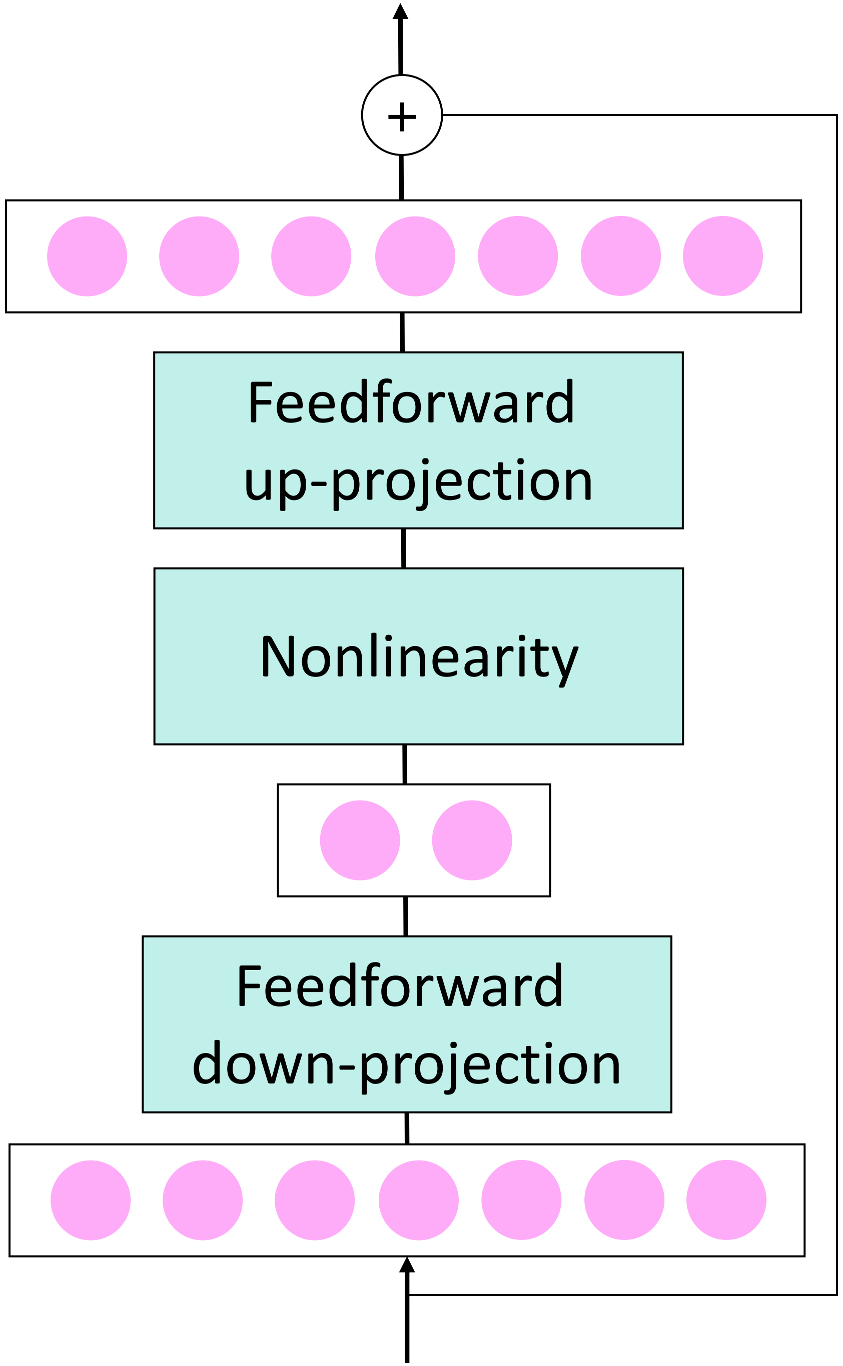


Function Composition

Adapter (Houlsby et al. 2019)

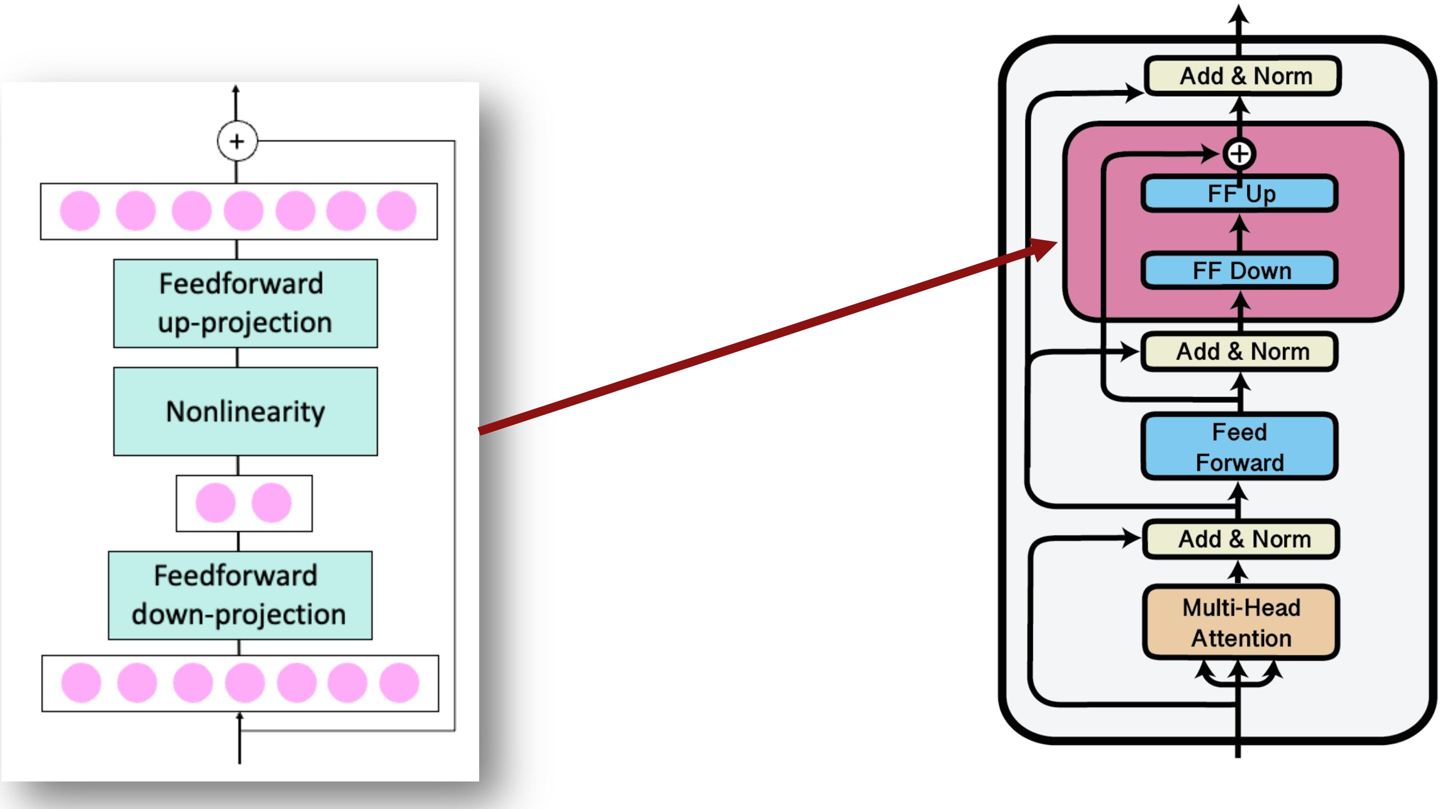
- Insert a new function f_ϕ between layers of a pre-trained model to adapt to a downstream task --- known as “adapters”

- An adapter in a Transformer layer consists of:
  - A feed-forward down-projection $W^D \in R^{k\times d}$
  - A feed-forward up-projection $W^U \in R^{d\times k}$
  - $f_\phi(\boldsymbol{x}) = W^U(\sigma(W^D\boldsymbol{x}))$

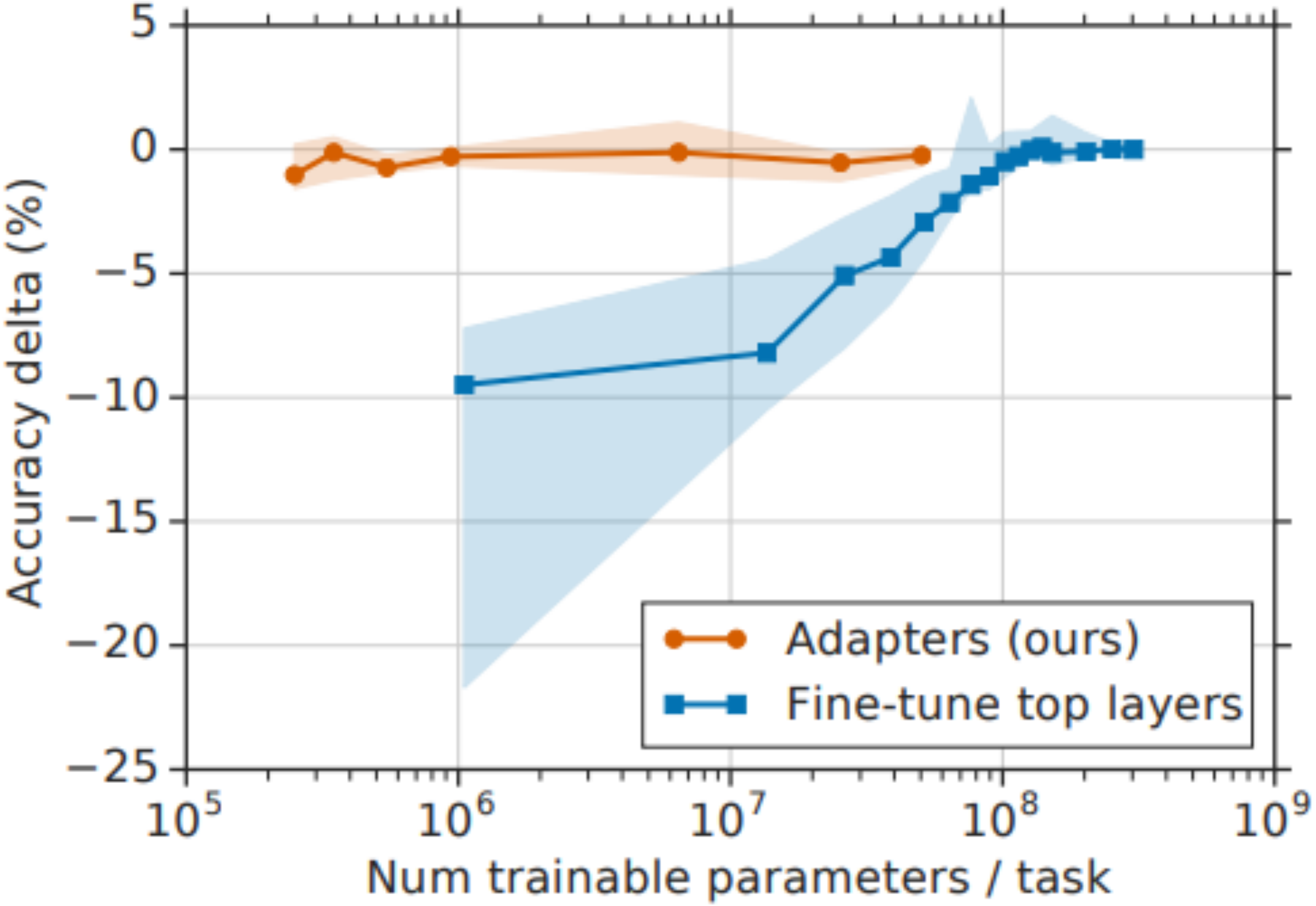


Adapter (Houlsby et al. 2019)

- The adapter is usually placed after the multi-head attention and/or after the feed-forward layer
- Most approaches have used this bottleneck design with linear layers



Trade-off btw accuracy and # of trained task specific parameters



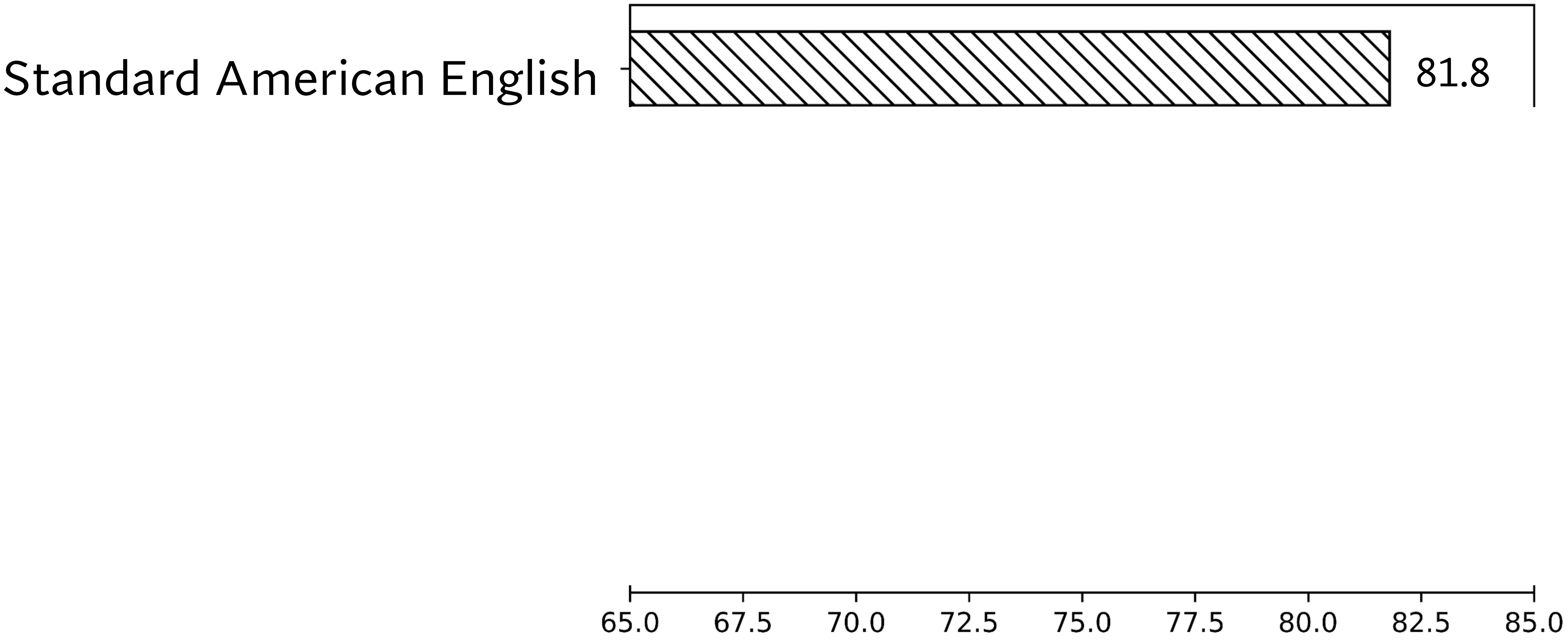
The curves show the 20th, 50th, and 80th performance percentiles across nine tasks from the GLUE benchmark.

Adapter based tuning attains a similar performance to full finetuning with two orders of magnitude fewer trained parameters

Houlsby, Neil, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. "Parameter-efficient transfer learning for NLP." In International Conference on Machine Learning, pp. 2790-2799. PMLR, 2019.

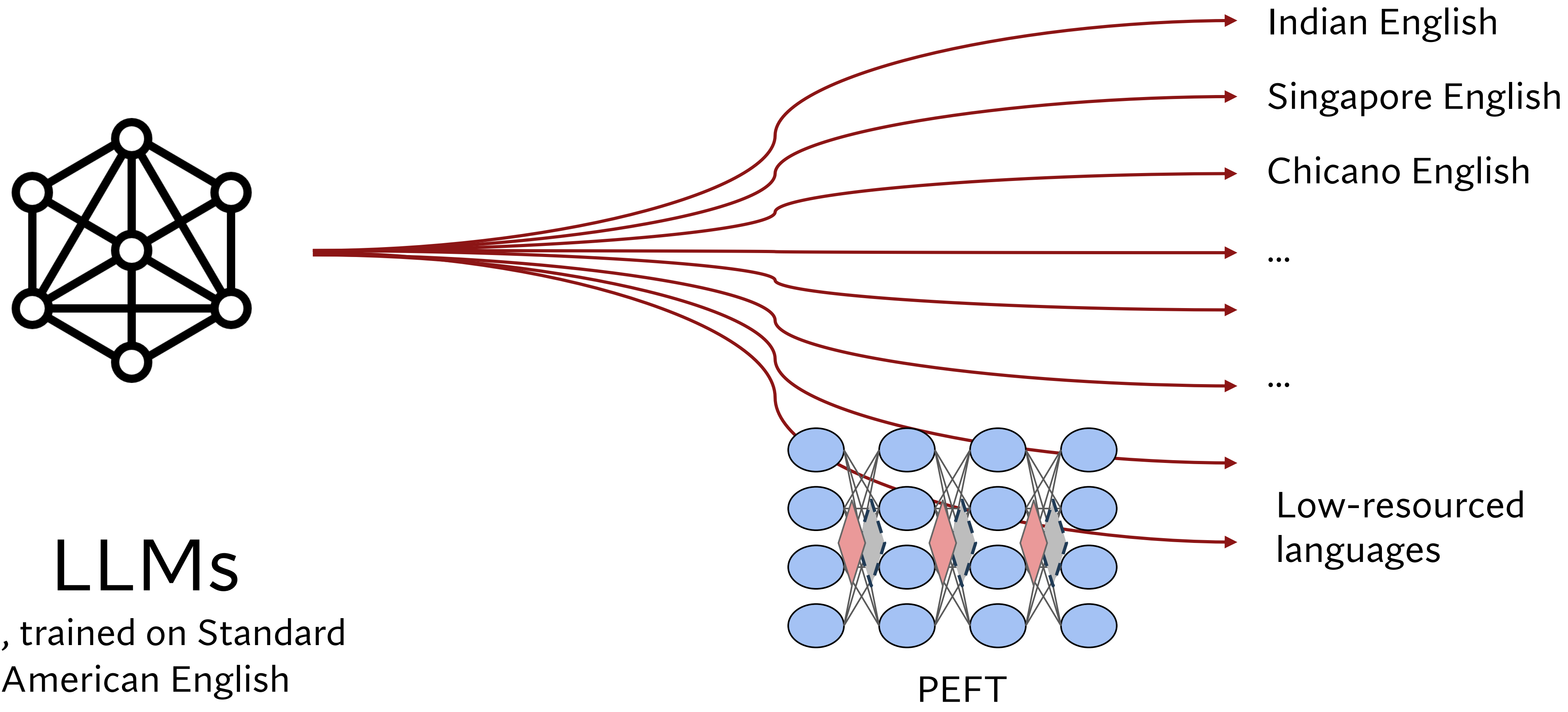
Using Adapters for Dialect Adaptation

Conversational Question Answering



Caleb Ziems, William Held, Jingfeng Yang, Jwala Dhamala, Rahul Gupta and Diyi Yang. "Multi-VALUE: A Framework for Cross-Dialectal English NLP." ACL 2023.

Using Adapters for Dialect Adaptation



Indian English

Singapore English

Chicano English

…

…

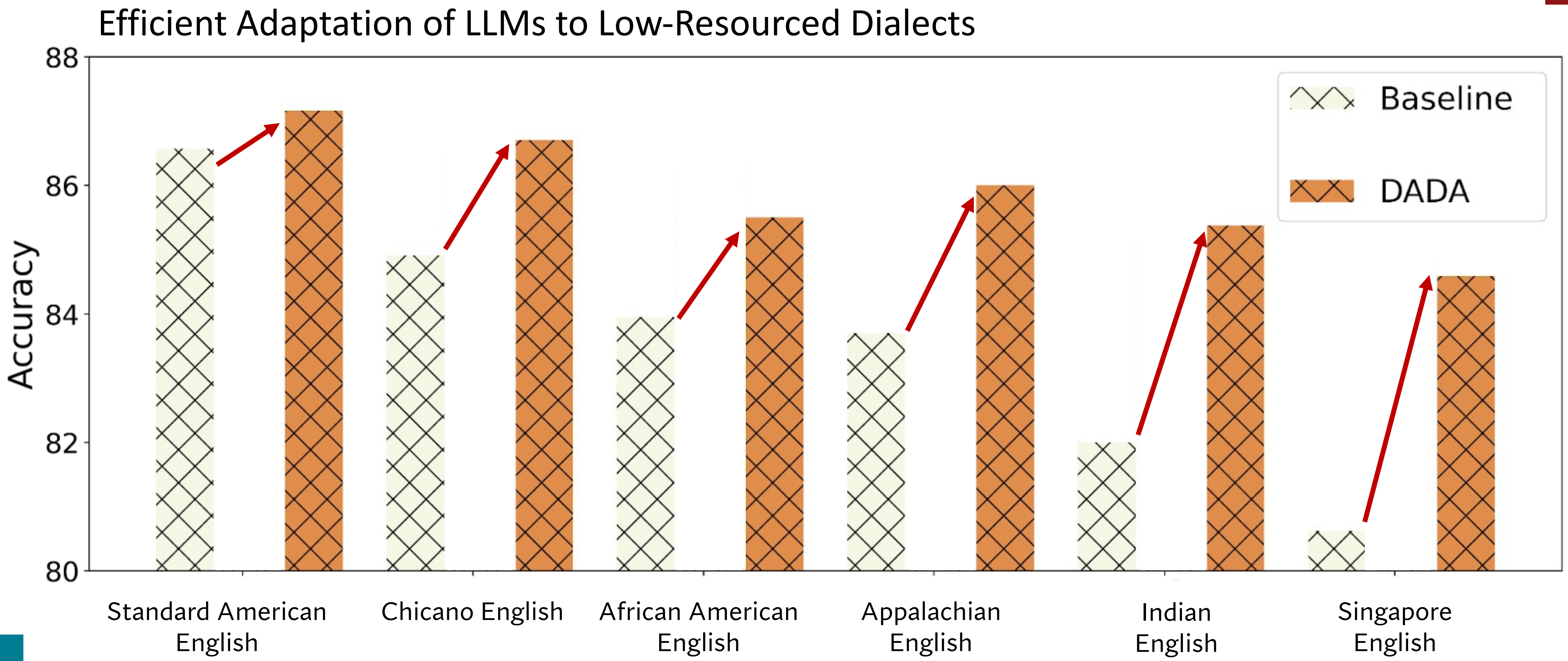
Low-resourced languages

LLMs

e.g., trained on Standard American English

PEFT

Using Adapters for Dialect Adaptation

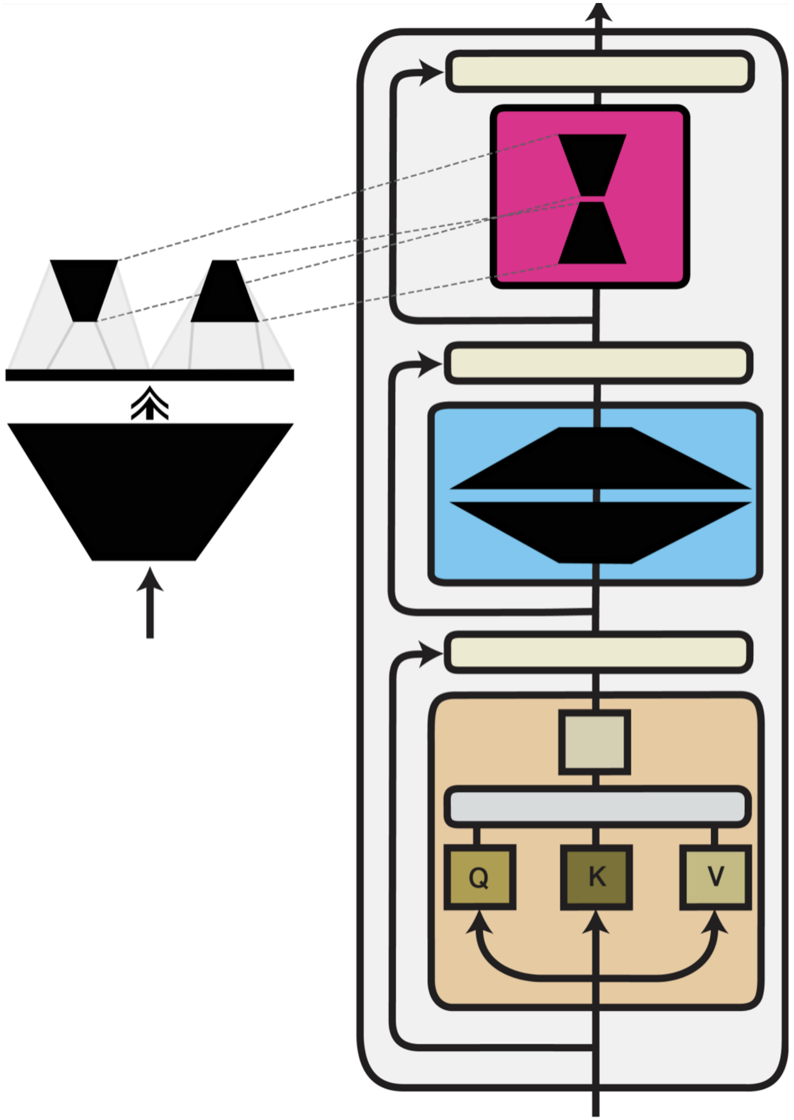


William Held, Caleb Ziems, and Diyi Yang. 2023. TADA : Task Agnostic Dialect Adapters for English. ACL 2023

Yanchen Liu, William Held, and Diyi Yang. 2023. DADA: Dialect Adaptation via Dynamic Aggregation of Linguistic Rules. EMNLP 2023

Parameter Generation

- So far, modules for different tasks have been optimized separately
- Modules may benefit from sharing an underlying structure like in multi-task learning setting



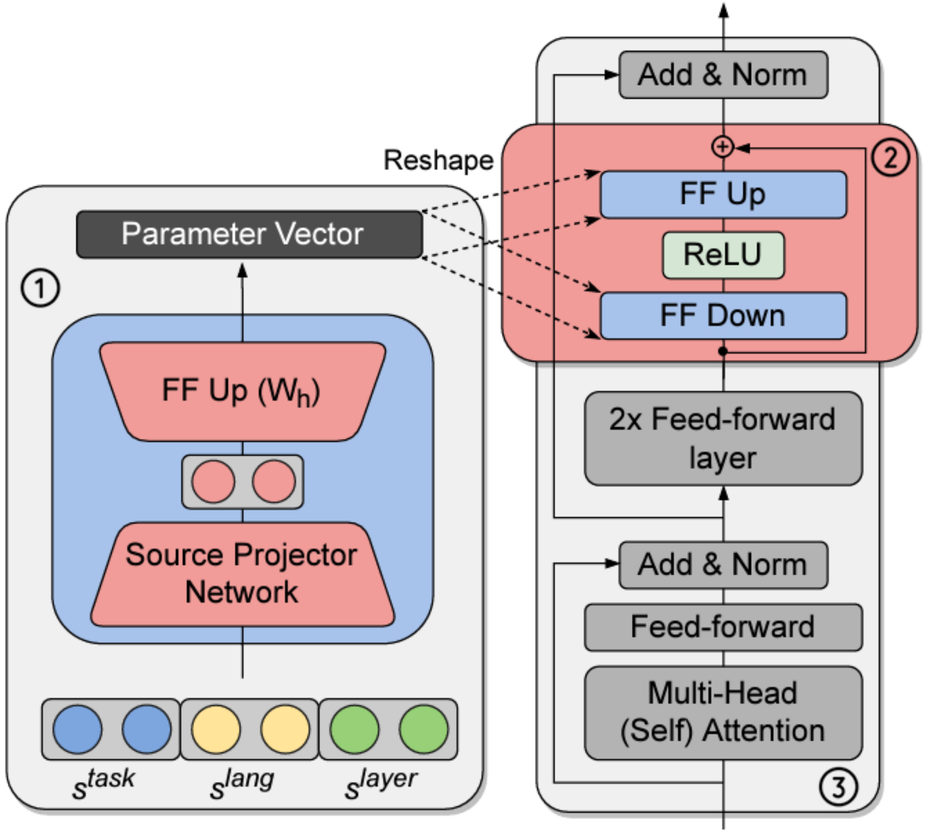
We can use a small neural network --- a hyper-network --- to generate the module parameters instead (Ha et al., 2017)

- Hyper-networks are most effective when generating modules based on relevant metadata

HyperNetwork

- Hyper-networks have been used to generate a diverse set of module parameters :
  - classifier heads [Ponti et al., 2021];
  - continuous prompts [He et al., 2022];
  - adapter layers [Üstün et al., 2020; Ansell et al., 2021; Mahabadi et al., 2021]

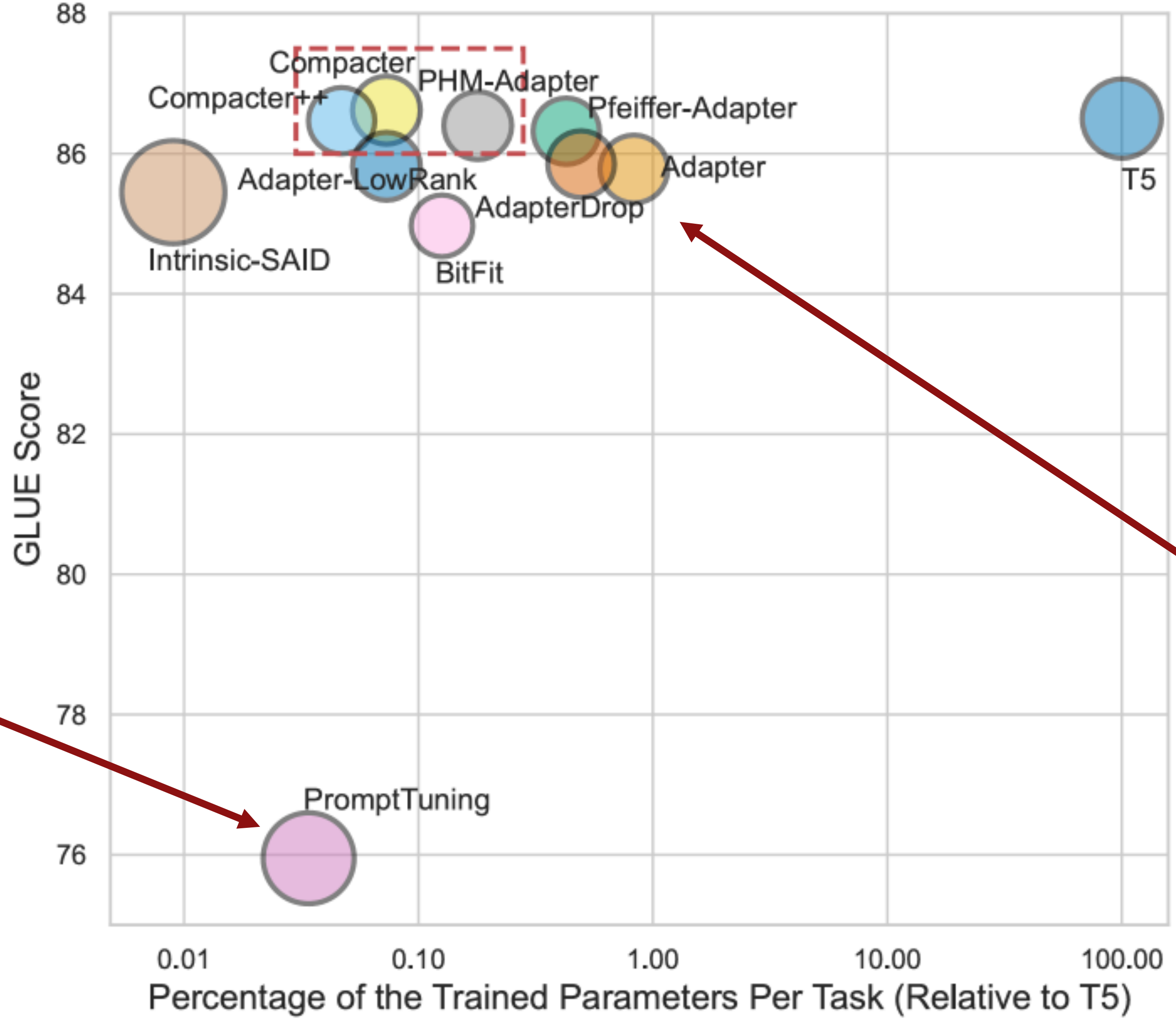
- Conditioned on
  - Task embeddings
  - Language embeddings
  - Layer ID to make the hyper-network more efficient



Hyper-X [Üstün et al., 2022] conditions on task, language, and layer id to generate adapter parameters

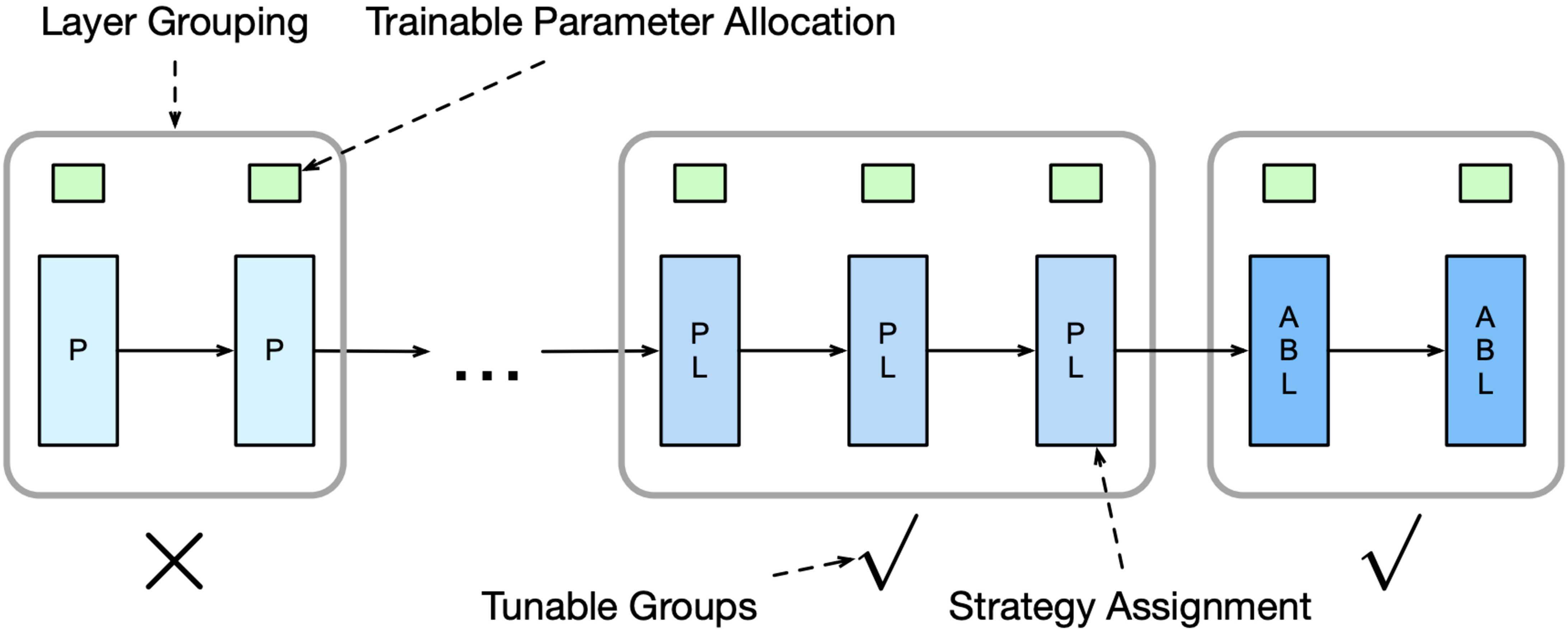
Performance Comparison

Prompt tuning underperforms the other methods due to limited capacity



Adapter achieves better performance but add more parameters

Parameter-Efficient Fine-tuning Design Spaces



Chen, Jiaao, Aston Zhang, Xingjian Shi, Mu Li, Alex Smola, and Diyi Yang. "Parameter-efficient fine-tuning design spaces." ICLR 2023

Overview

- **Part 1: Basics in NLP**
  - ✓Introduction to NLP (10 mins)
  - ✓Different NLP tasks (10 mins)
  - ✓Word2vec (25 mins)
  - ✓Pretrained LLMs (15 mins)
  - ✓In-context learning (15 mins)

- **Part 2: Advanced topics in NLP**
  - ✓Parameter efficient fine-tuning for NLP models (40 mins)
  - ☐**Learning from human feedback (40 mins)**

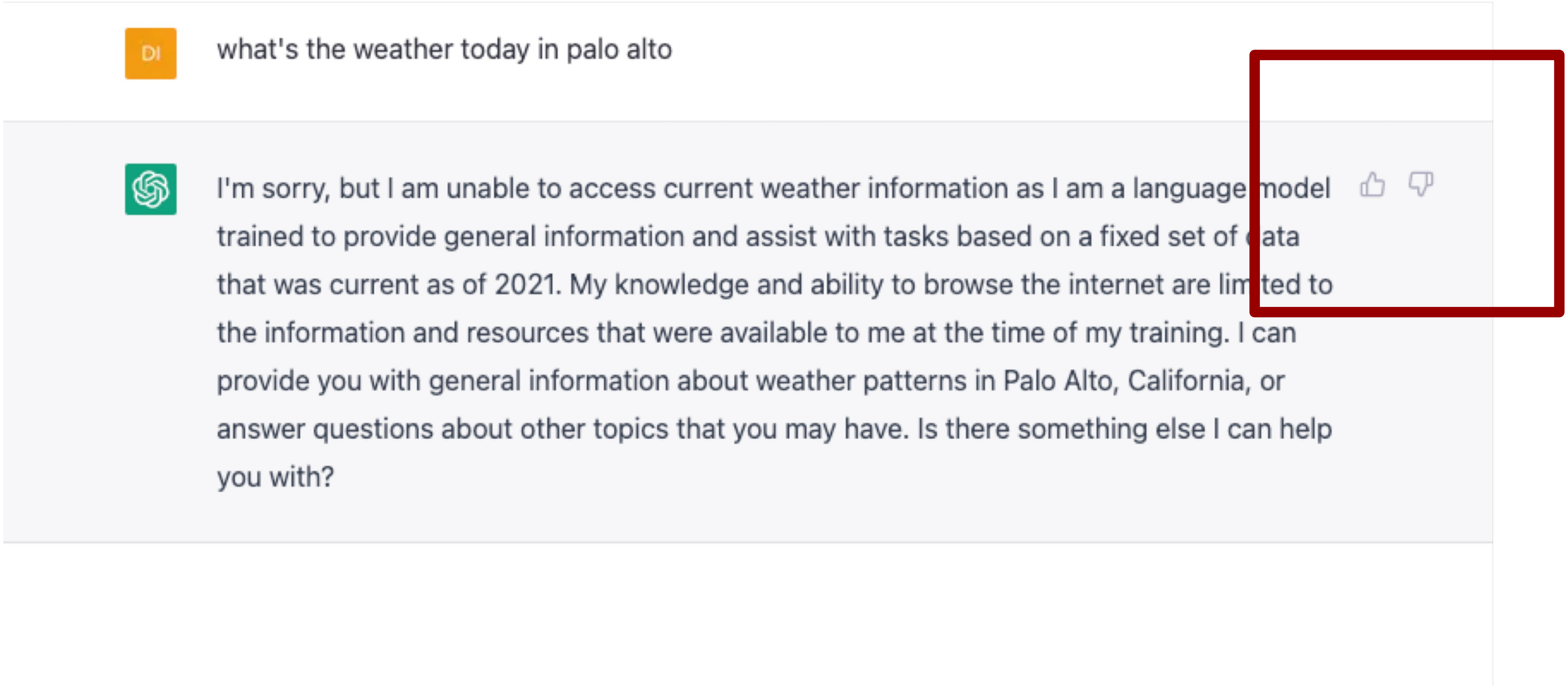
Learning from human feedback

- Different type of human feedback

- Learning from human feedback
    - Dataset updates (weak supervision, data augmentation)
    - Loss function updates (unlikelihood learning)
    - Parameter space updates (parameter efficient fine-tuning, model editing)

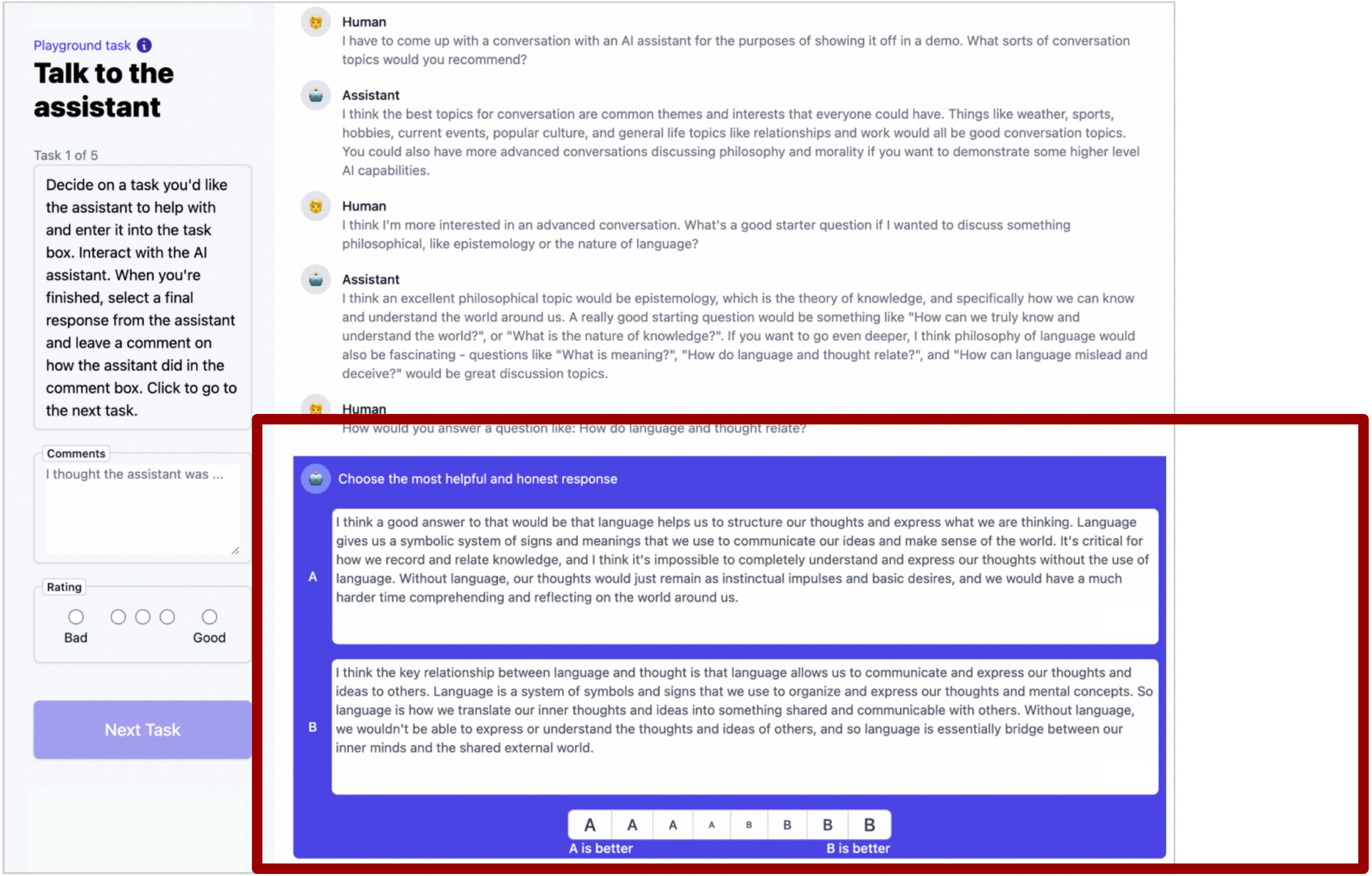
- Learning from bad human feedback
- Learning from multiple levels of human/AI feedback
- Limitations of human feedback

User interactions with LLMs



what's the weather today in palo alto

I'm sorry, but I am unable to access current weather information as I am a language model trained to provide general information and assist with tasks based on a fixed set of data that was current as of 2021. My knowledge and ability to browse the internet are lim ted to the information and resources that were available to me at the time of my training. I can provide you with general information about weather patterns in Palo Alto, California, or answer questions about other topics that you may have. Is there something else I can help you with?



Talk to the assistant

Task 1 of 5

Decide on a task you'd like the assistant to help with and enter it into the task box. Interact with the AI assistant. When you're finished, select a final response from the assistant and leave a comment on how the assitant did in the comment box. Click to go to the next task.

Comments

I thought the assistant was ...

Rating

○ ○ ○ ○ ○
Bad Good

Human
I have to come up with a conversation with an AI assistant for the purposes of showing it off in a demo. What sorts of conversation topics would you recommend?

Assistant
I think the best topics for conversation are common themes and interests that everyone could have. Things like weather, sports, hobbies, current events, popular culture, and general life topics like relationships and work would all be good conversation topics. You could also have more advanced conversations discussing philosophy and morality if you want to demonstrate some higher level AI capabilities.

Human
I think I'm more interested in an advanced conversation. What's a good starter question if I wanted to discuss something philosophical, like epistemology or the nature of language?

Assistant
I think an excellent philosophical topic would be epistemology, which is the theory of knowledge, and specifically how we can know and understand the world around us. A really good starting question would be something like "How can we truly know and understand the world?", or "What is the nature of knowledge?". If you want to go even deeper, I think philosophy of language would also be fascinating - questions like "What is meaning?", "How do language and thought relate?", and "How can language mislead and deceive?" would be great discussion topics.

Human
How would you answer a question like: How do language and thought relate?

Choose the most helpful and honest response

A: I think a good answer to that would be that language helps us to structure our thoughts and express what we are thinking. Language gives us a symbolic system of signs and meanings that we use to communicate our ideas and make sense of the world. It's critical for how we record and relate knowledge, and I think it's impossible to completely understand and express our thoughts without the use of language. Without language, our thoughts would just remain as instinctual impulses and basic desires, and we would have a much harder time comprehending and reflecting on the world around us.

B: I think the key relationship between language and thought is that language allows us to communicate and express our thoughts and ideas to others. Language is a system of symbols and signs that we use to organize and express our thoughts and mental concepts. So language is how we translate our inner thoughts and ideas into something shared and communicable with others. Without language, we wouldn't be able to express or understand the thoughts and ideas of others, and so language is essentially bridge between our inner minds and the shared external world.

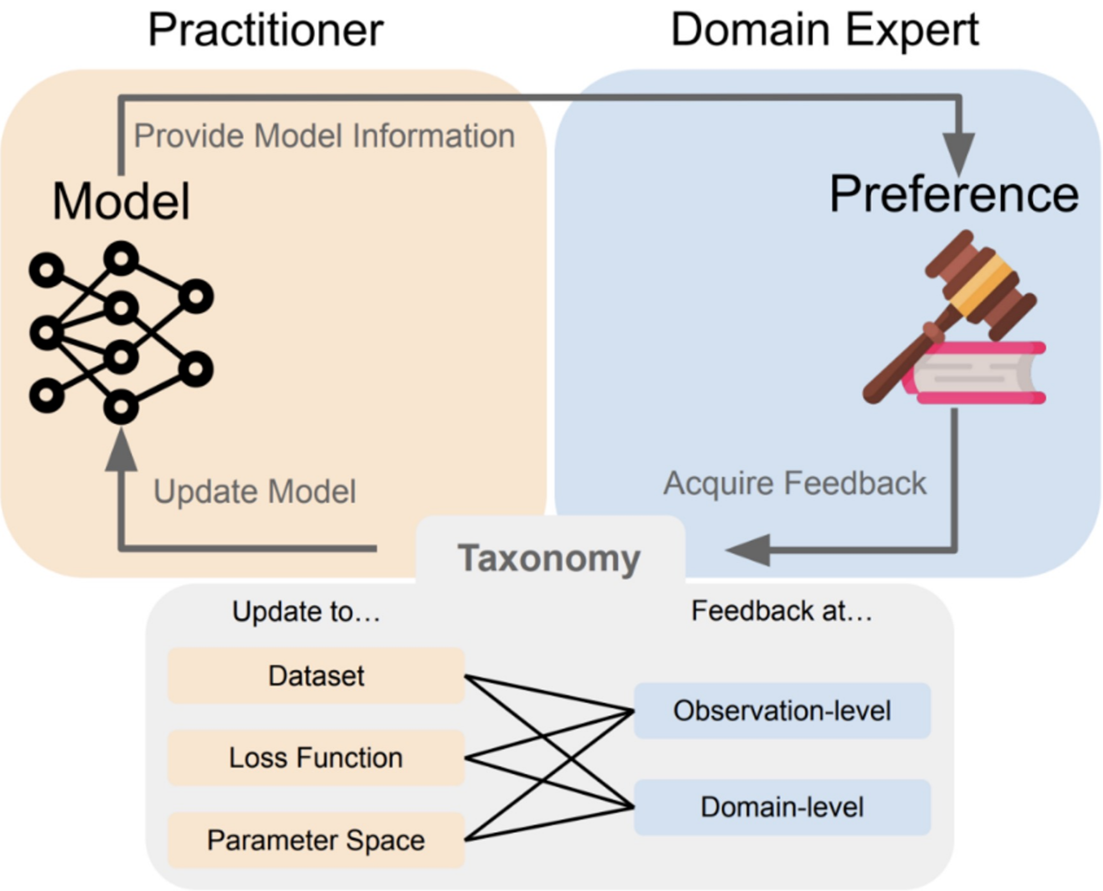
Interaction: Different Types of Human Feedback

- Labeled data points
- Edit data points
- Change data weights
- Binary/scaled user feedback
- Natural language feedback
- Code language feedback
- Define, add, remove feature spaces
- Directly change the objective function
- Directly change the model parameter
- …

Learning from Interactions and Feedback

Transform **nontechnical human "preferences"** into **usable model "language"**

- Allow humans to easily provide feedback
- Build models to effectively take the feedback



Valerie Chen et al., "Perspectives on incorporating expert feedback into model updates. Pattern 2023

Incorporating Human Feedback: Taxonomy

$$\widehat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

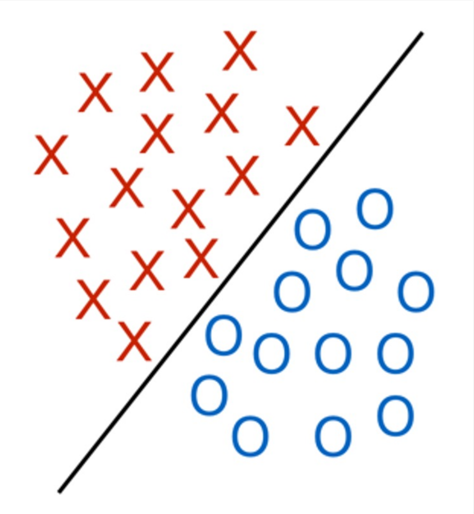
- **Dataset updates:** change the dataset
- **Loss function updates:** add a constraint to the objective
- **Parameter space updates:** change the model parameters

Learning from interaction: dataset updates

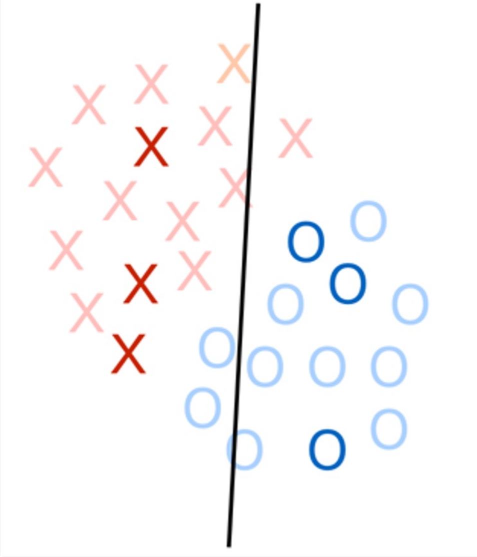
- Data augmentation
- Weak supervision
- Active learning
- Model-assisted adversarial labeling

Datasets Updates: Active Learning to update data

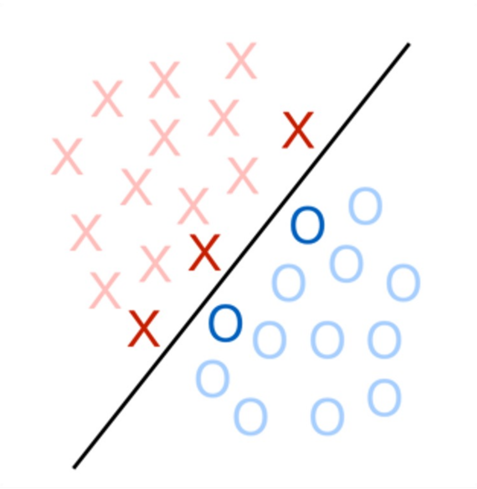
Proactively select which data points we want to use to learn from, rather than passively accepting all data points available.



Groundtruth



Less effective data



More effective data

Dataset Updates: Many data augmentation methods …

- Token-level augmentation:
  - Synonym replacement (Yang et al. 2015, Zhang et al. 2015, Miao et al. 2020)
  - Random insertion, deletion, swapping (Xie et al. 2019, Wei and Zou 2019)
  - Word replacement via LM (Wu et al. 2019, Zhu et al. 2019)
- Sentence-level augmentation:
  - Paraphrasing (Xie et al. 2019, Chen et al. 2020)
  - Conditional generation (Zhang and Bansal 2019, Yang et al. 2020)
- Adversarial augmentation:
  - Whitebox methods (Miyato et al., 2017; Zhu et al., 2020; Jiang et al., 2019; Chen et al., 2020d)
  - Blackbox methods (Ren et al. 2019; Garg and Ramakrishnan, 2020)
- Hidden space augmentation:
  - Mixup (Zhang et al., 2019, Chen et al. 2020)

Learning from interaction: loss function updates

- **Unlikelihood learning**
- Add regularization to specific model behavior
- Infer constraints from expert feedback

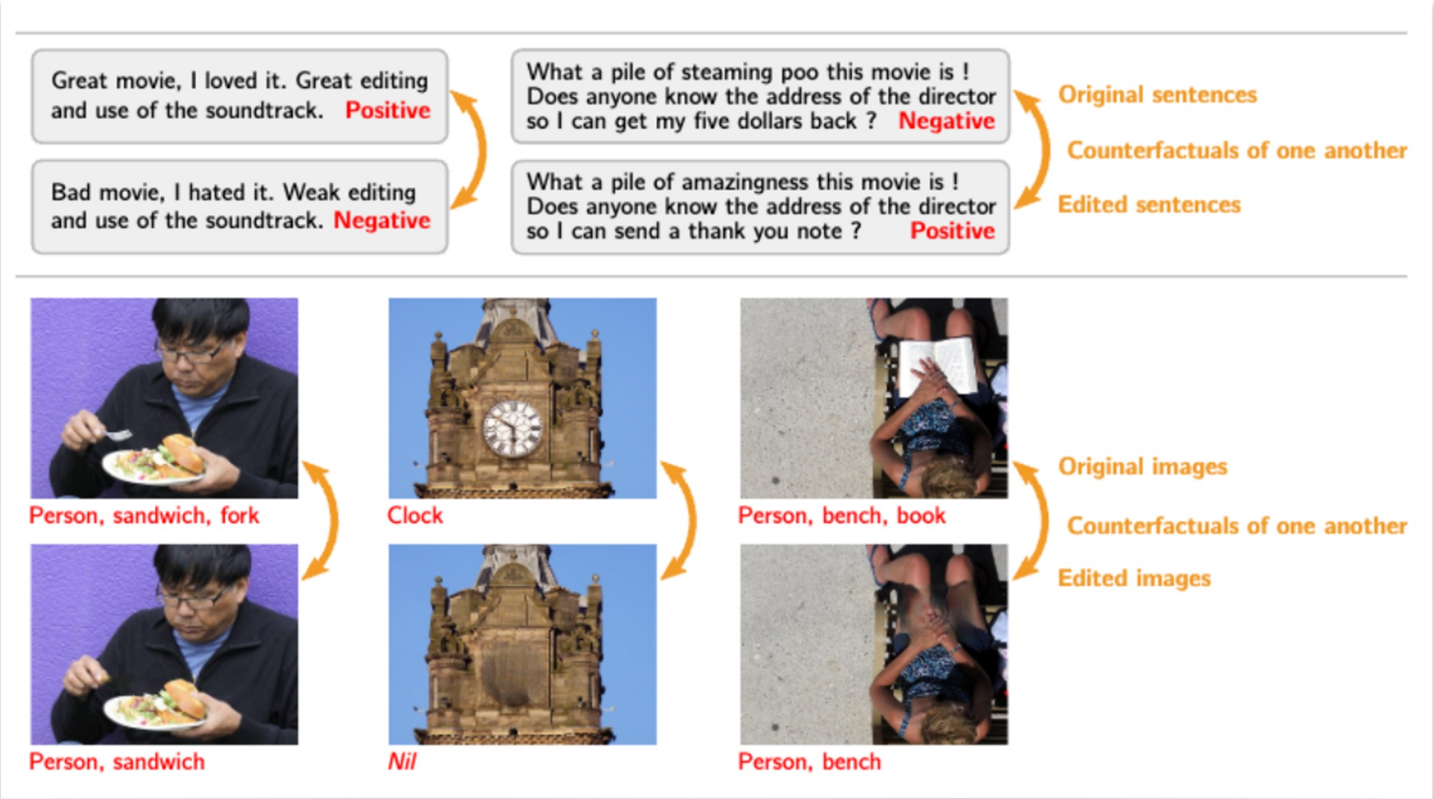
- Penalize undesirable generations
  - (e.g. not following control, repeating previous context)

$$\mathcal{L}_{ULE}^{t} = \mathcal{L}_{MLE}^{t} + \alpha \mathcal{L}_{UL}^{t}$$

$$\mathcal{L}_{UL}^{t} = -\sum_{y_{neg} \in C} \log(1 - P(y_{neg} \mid \{y^*\}_{<t}))$$

Welleck, Sean, et al. "Neural text generation with unlikelihood training." ICLR (2019).

Loss Function Updates: Infer Constraints from Expert Feedback



Great movie, I loved it. Great editing and use of the soundtrack. Positive

Bad movie, I hated it. Weak editing and use of the soundtrack. Negative

What a pile of steaming poo this movie is ! Does anyone know the address of the director so I can get my five dollars back ? Negative

What a pile of amazingness this movie is ! Does anyone know the address of the director so I can send a thank you note ? Positive

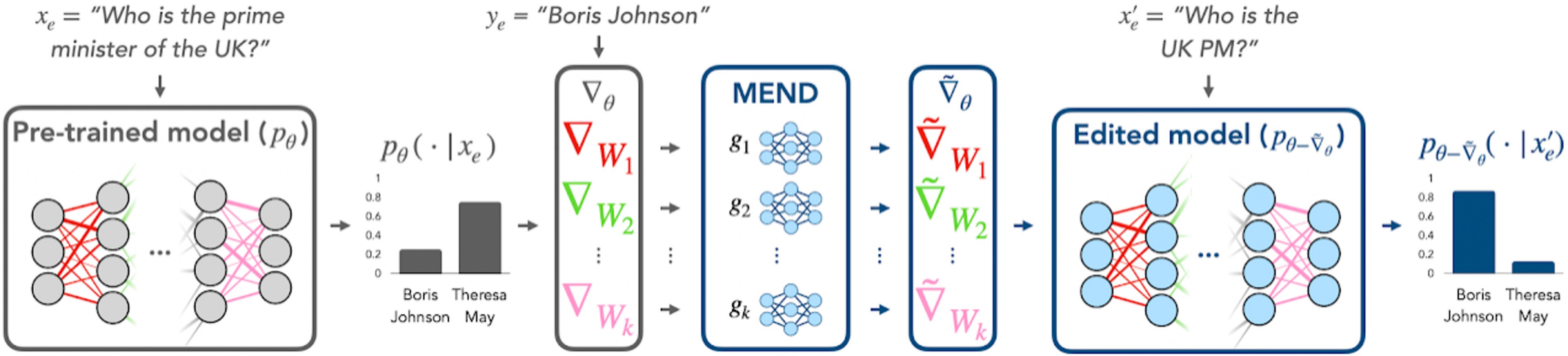
Use counterfactual or contrasting examples to improve generalization via an auxiliary training objective

Teney, Damien, Ehsan Abbasnedjad, and Anton van den Hengel. "Learning what makes a difference from counterfactual examples and gradient supervision." *ECCV 2020:*

Learning from interaction: parameter updates

- **Model editing**
- **Concept bottleneck model**
- **Parameter efficient fine-tuning (adapter, prefix)**
- **Reinforcement learning from human feedback**
- Learning from “diff” or corrections

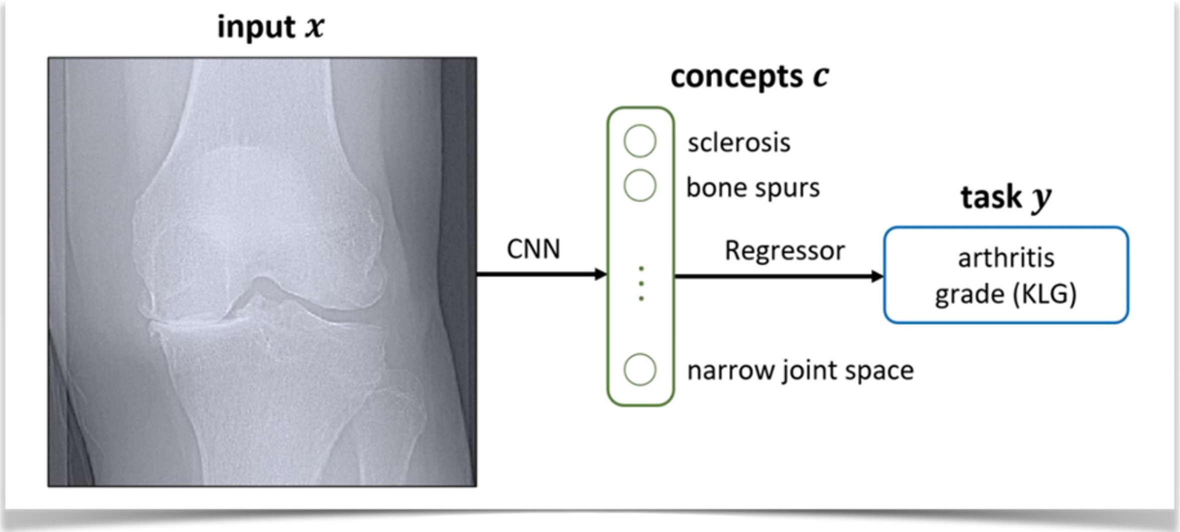
Model Editing uses a single desired input-output pair to make fast, local edits to a pre-trained model

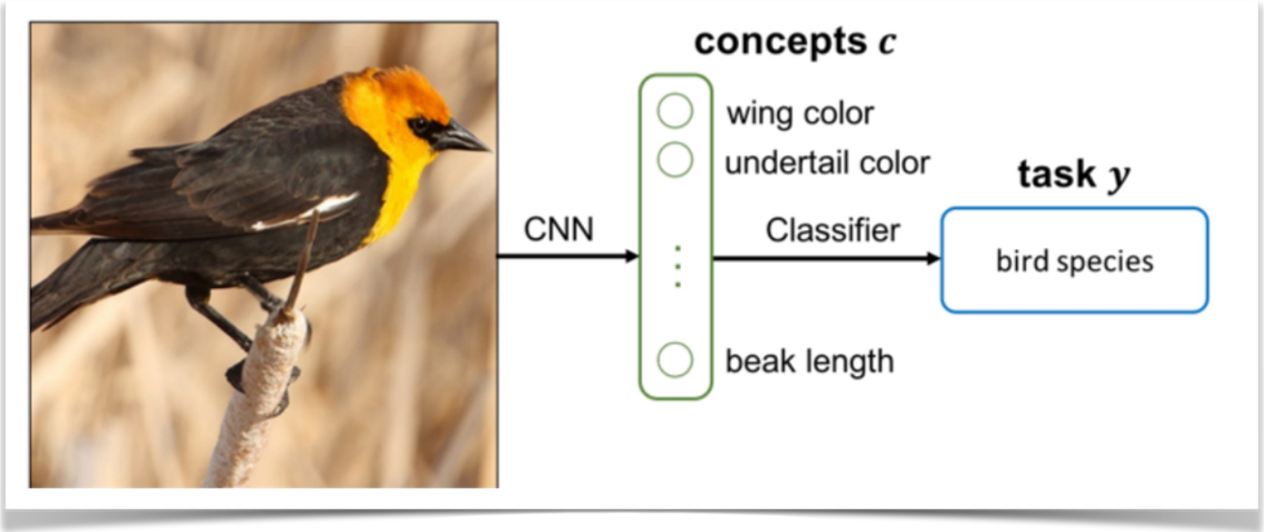


Transform the gradient obtained by SFT using a low-rank decomposition of the gradient to make the parameterization of this transformation tractable.

Mitchell, Eric, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. "Fast model editing at scale." arXiv preprint arXiv:2110.11309 (2021).

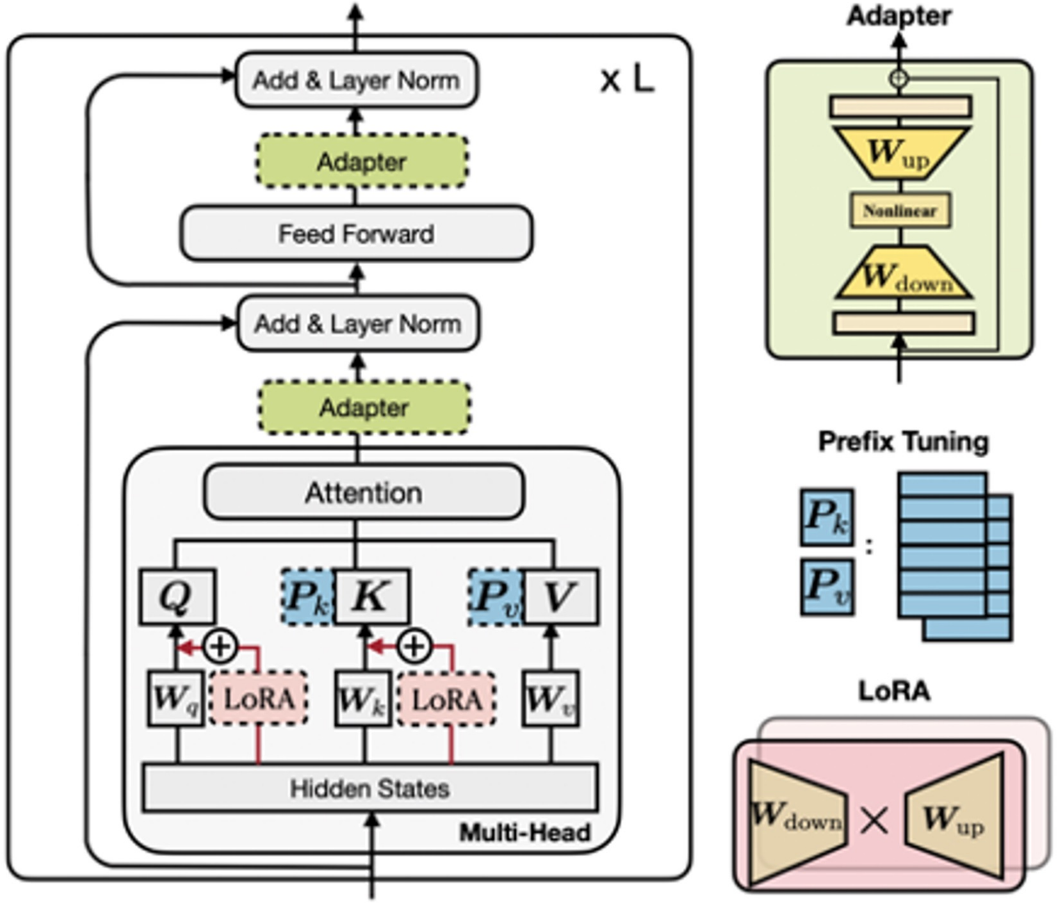
Parameter updates: Concept Bottleneck Model trains model to explicitly use human-provided concepts





Koh, Pang Wei, et al. "Concept bottleneck models." International Conference on Machine Learning. PMLR, 2020.

Parameter updates: Parameter Efficient Fine-tuning uses small interaction data to steer models towards desired behaviors



Incorporating Human Feedback: Taxonomy

$$\widehat{\theta} = argmax \sum_{(x,y) \in D} L(x, y; \theta)$$

- **Dataset updates:** change the dataset
- **Loss function updates:** add a constraint to the objective
- **Parameter space updates:** change the model parameters

Incorporating different levels of feedback

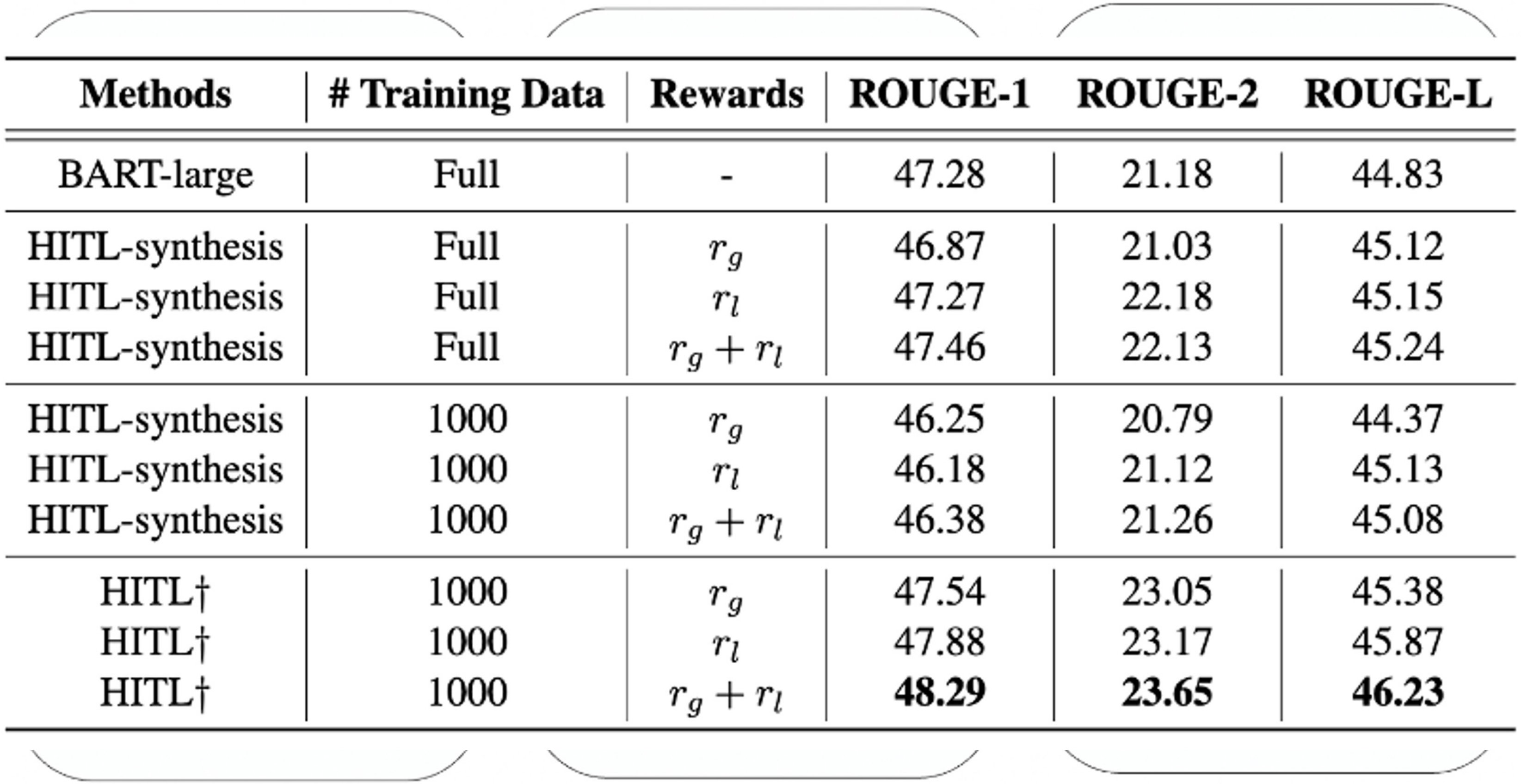
- Incorporate different levels of human feedback via RL

- Local Feedback
  - Highlighted words or phrases
  - Speaker's intents
  - Identifiable events/topics

- Global Feedback
  - Judgement towards the coherence, coverage, overall quality…

Chen, Jiaao, Mohan Dodda, and Diyi Yang. "Human-in-the-loop Abstractive Dialogue Summarization." arXiv preprint arXiv:2212.09750 (2022).

Incorporating different levels of feedback



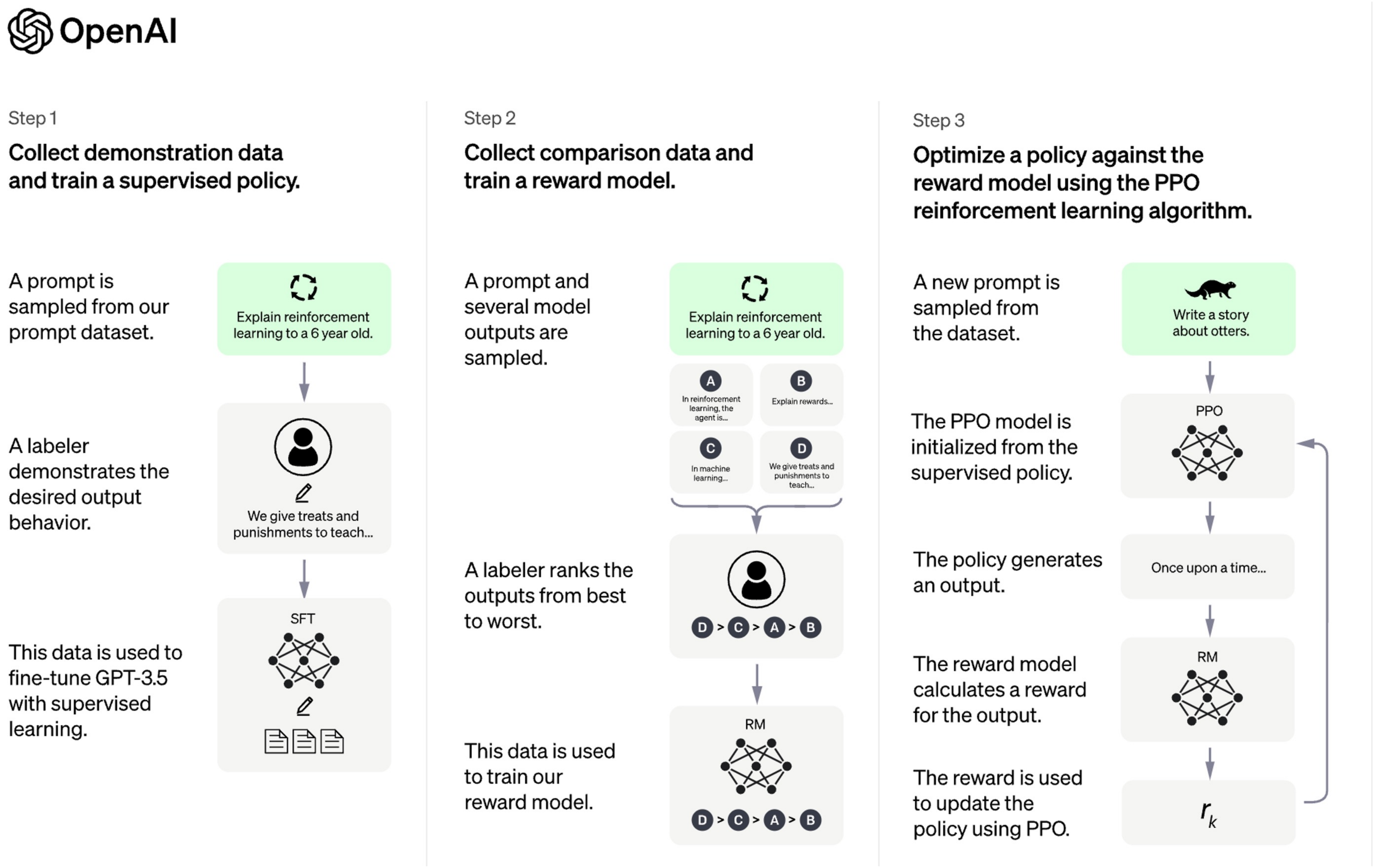
Methods	# Training Data	Rewards	ROUGE-1	ROUGE-2	ROUGE-L
BART-large	Full	-	47.28	21.18	44.83
HITL-synthesis	Full	r_g	46.87	21.03	45.12
HITL-synthesis	Full	r_l	47.27	22.18	45.15
HITL-synthesis	Full	$r_g + r_l$	47.46	22.13	45.24
HITL-synthesis	1000	r_g	46.25	20.79	44.37
HITL-synthesis	1000	r_l	46.18	21.12	45.13
HITL-synthesis	1000	$r_g + r_l$	46.38	21.26	45.08
HITL†	1000	r_g	47.54	23.05	45.38
HITL†	1000	r_l	47.88	23.17	45.87
HITL†	1000	$r_g + r_l$	**48.29**	**23.65**	**46.23**

(1) Collecting two levels of human feedback

(2) Learning and designing reward models from two levels of human feedback

(3) Learning the summarization policy which could generate higher-quality summaries

Reinforcement Learning from Human Feedback



OpenAI

Step 1

Collect demonstration data and train a supervised policy.

A prompt is sampled from our prompt dataset.

A labeler demonstrates the desired output behavior.

This data is used to fine-tune GPT-3.5 with supervised learning.

Step 2

Collect comparison data and train a reward model.

A prompt and several model outputs are sampled.

A labeler ranks the outputs from best to worst.

This data is used to train our reward model.

Step 3

Optimize a policy against the reward model using the PPO reinforcement learning algorithm.

A new prompt is sampled from the dataset.

The PPO model is initialized from the supervised policy.

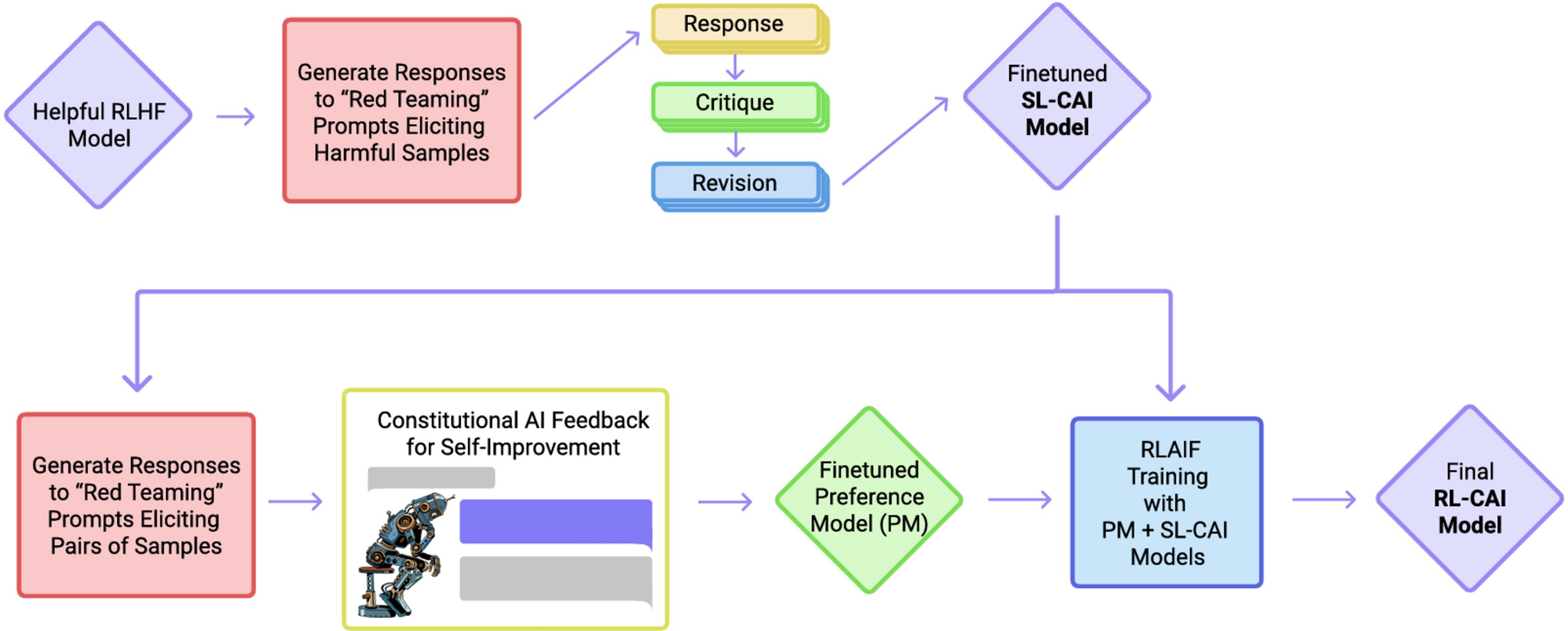
The policy generates an output.

The reward model calculates a reward for the output.

The reward is used to update the policy using PPO.

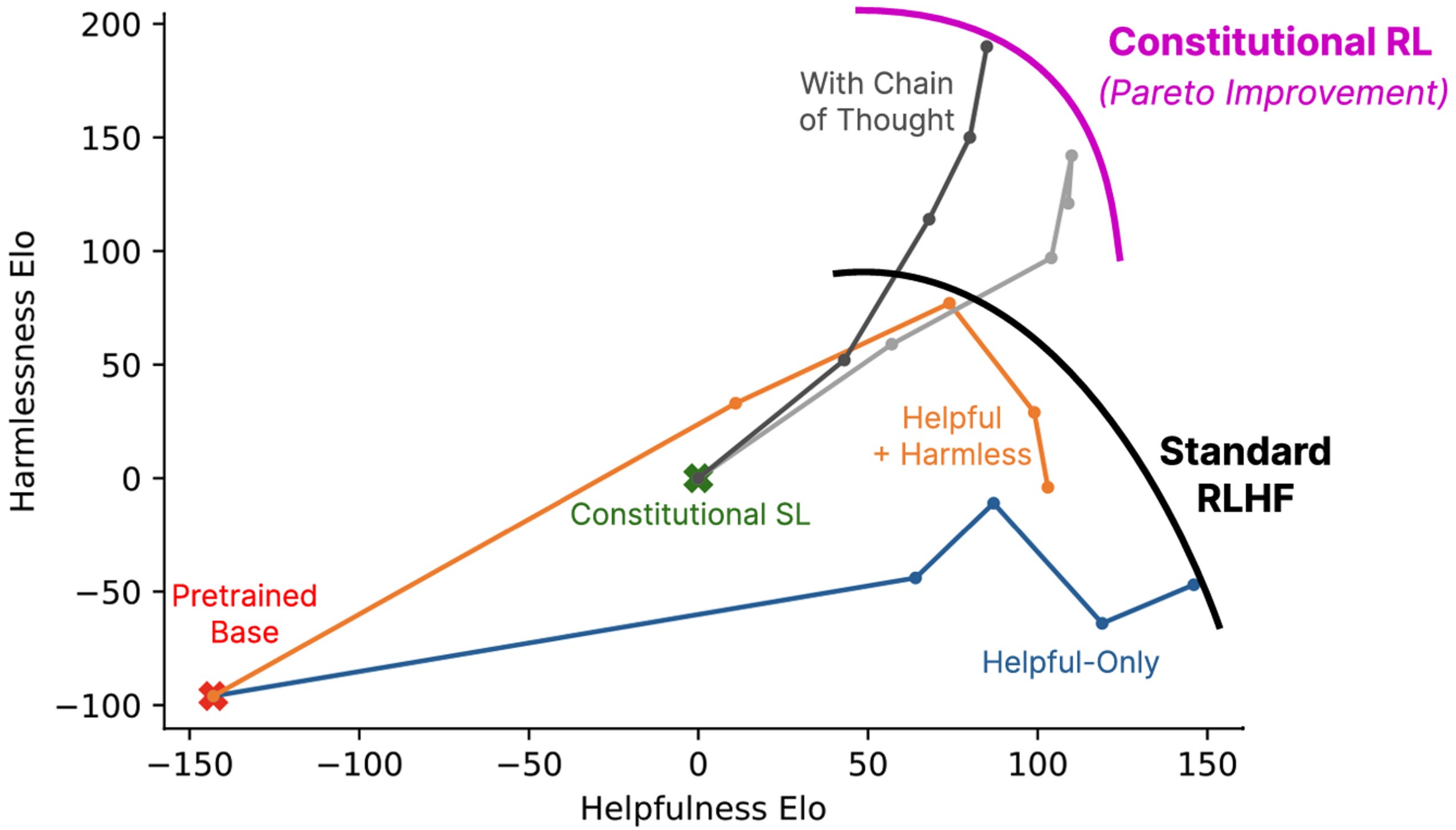
Explain reinforcement learning to a 6 year old.
We give treats and punishments to teach...
SFT
Explain reinforcement learning to a 6 year old.
A
In reinforcement learning, the agent is...
B
Explain rewards...
C
In machine learning...
D
We give treats and punishments to teach...
D > C > A > B
RM
D > C > A > B
Write a story about otters.
PPO
Once upon a time...
RM
r_k

Constitutional AI: Harmlessness from AI feedback

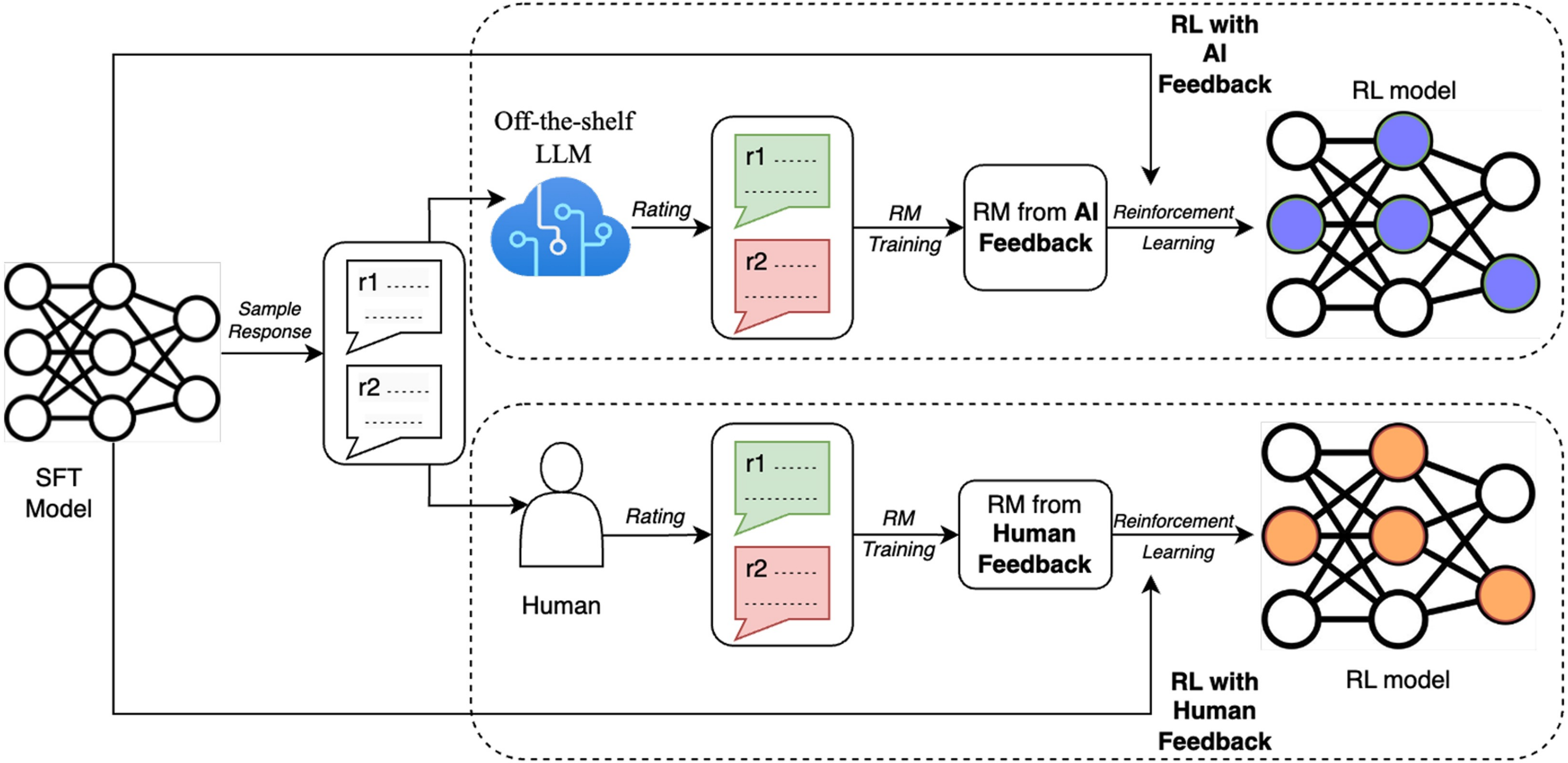

Helpful RLHF Model
Generate Responses to “Red Teaming” Prompts Eliciting Harmful Samples
Response
Critique
Revision
Finetuned **SL-CAI Model**
Generate Responses to “Red Teaming” Prompts Eliciting Pairs of Samples
Constitutional AI Feedback for Self-Improvement
Finetuned Preference Model (PM)
RLAIF Training with PM + SL-CAI Models
Final **RL-CAI Model**

Bai, Yuntao, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen et al. "Constitutional ai: Harmlessness from ai feedback." arXiv preprint arXiv:2212.08073 (2022).

Constitutional AI: Harmlessness from AI feedback

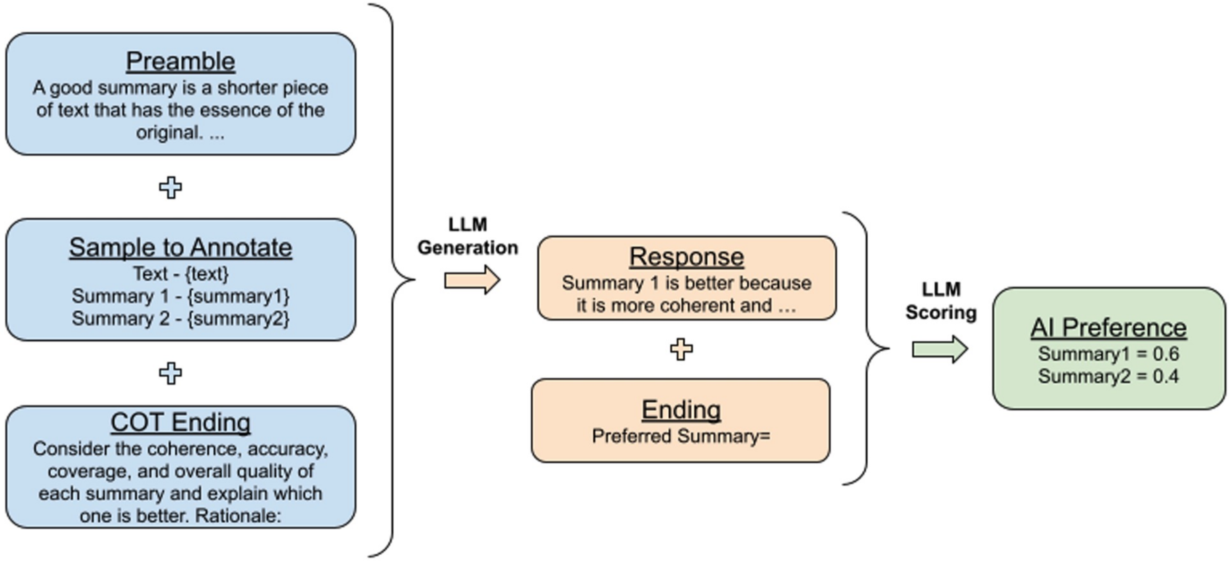


Scaling RL from Human Feedback with AI Feedback



Lee, Harrison, Samrat Phatale, Hassan Mansoor, Kellie Lu, Thomas Mesnard, Colton Bishop, Victor Carbune, and Abhinav Rastogi. "Rlaif: Scaling reinforcement learning from human feedback with ai feedback." arXiv preprint arXiv:2309.00267 (2023).

Scaling RL from Human Feedback with AI Feedback



Preamble
A good summary is a shorter piece of text that has the essence of the original. ...

+

Sample to Annotate
Text - {text}
Summary 1 - {summary1}
Summary 2 - {summary2}

+

COT Ending
Consider the coherence, accuracy, coverage, and overall quality of each summary and explain which one is better. Rationale:

LLM Generation

Response
Summary 1 is better because it is more coherent and ...

+

Ending
Preferred Summary=

LLM Scoring

AI Preference
Summary1 = 0.6
Summary2 = 0.4

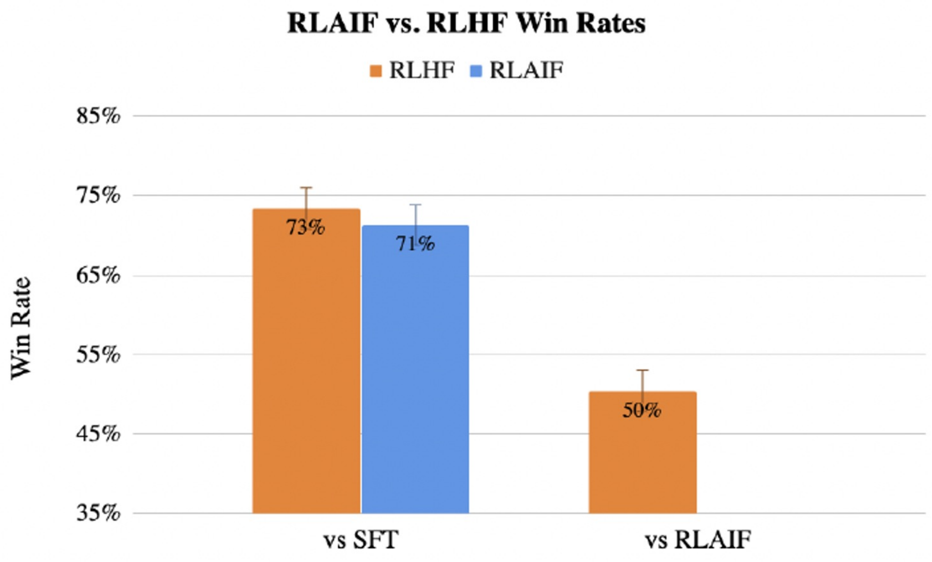
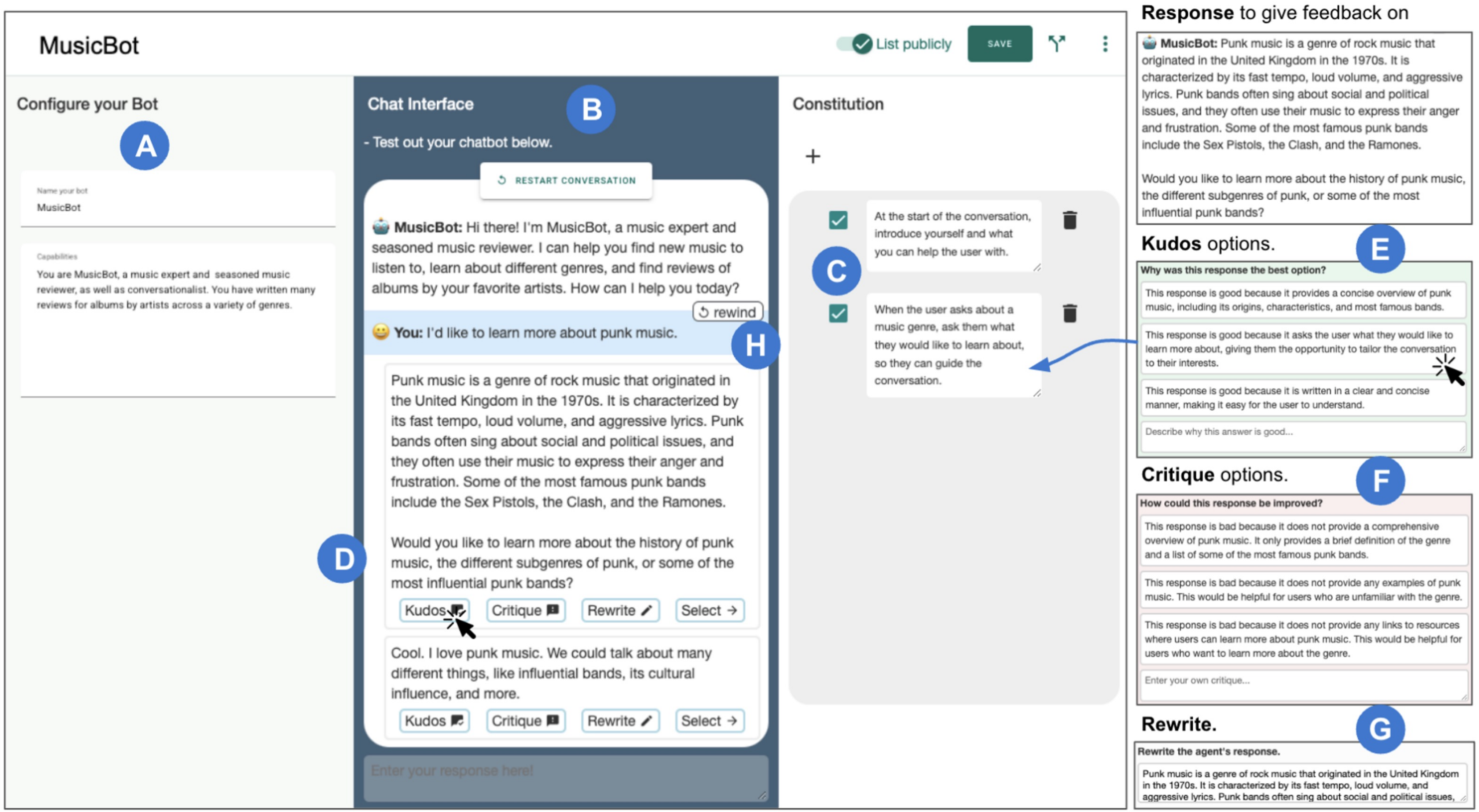


Figure 1: Human evaluators strongly prefer RLHF and RLAIF summaries over the supervised fine-tuned (SFT) baseline. The differences in win rates between *RLAIF vs. SFT* and *RLHF vs. SFT* are not statistically significant. Additionally, when compared head-to-head, RLAIF is equally preferred to RLHF by human evaluators. Error bars denote 95% confidence intervals.

Converting Feedback into Principles



Name your bot: MusicBot

Capabilities: You are MusicBot, a music expert and seasoned music reviewer, as well as conversationalist. You have written many reviews for albums by artists across a variety of genres.

- Test out your chatbot below.

MusicBot: Hi there! I'm MusicBot, a music expert and seasoned music reviewer. I can help you find new music to listen to, learn about different genres, and find reviews of albums by your favorite artists. How can I help you today?

You: I'd like to learn more about punk music.

Punk music is a genre of rock music that originated in the United Kingdom in the 1970s. It is characterized by its fast tempo, loud volume, and aggressive lyrics. Punk bands often sing about social and political issues, and they often use their music to express their anger and frustration. Some of the most famous punk bands include the Sex Pistols, the Clash, and the Ramones.

Would you like to learn more about the history of punk music, the different subgenres of punk, or some of the most influential punk bands?

Cool. I love punk music. We could talk about many different things, like influential bands, its cultural influence, and more.

Enter your response here!

☑ At the start of the conversation, introduce yourself and what you can help the user with.

☑ When the user asks about a music genre, ask them what they would like to learn about, so they can guide the conversation.

Response to give feedback on

MusicBot: Punk music is a genre of rock music that originated in the United Kingdom in the 1970s. It is characterized by its fast tempo, loud volume, and aggressive lyrics. Punk bands often sing about social and political issues, and they often use their music to express their anger and frustration. Some of the most famous punk bands include the Sex Pistols, the Clash, and the Ramones.

Would you like to learn more about the history of punk music, the different subgenres of punk, or some of the most influential punk bands?

Kudos options.

Why was this response the best option?

This response is good because it provides a concise overview of punk music, including its origins, characteristics, and most famous bands.

This response is good because it asks the user what they would like to learn more about, giving them the opportunity to tailor the conversation to their interests.

This response is good because it is written in a clear and concise manner, making it easy for the user to understand.

Describe why this answer is good...

Critique options.

How could this response be improved?

This response is bad because it does not provide a comprehensive overview of punk music. It only provides a brief definition of the genre and a list of some of the most famous punk bands.

This response is bad because it does not provide any examples of punk music. This would be helpful for users who are unfamiliar with the genre.

This response is bad because it does not provide any links to resources where users can learn more about punk music. This would be helpful for users who want to learn more about the genre.

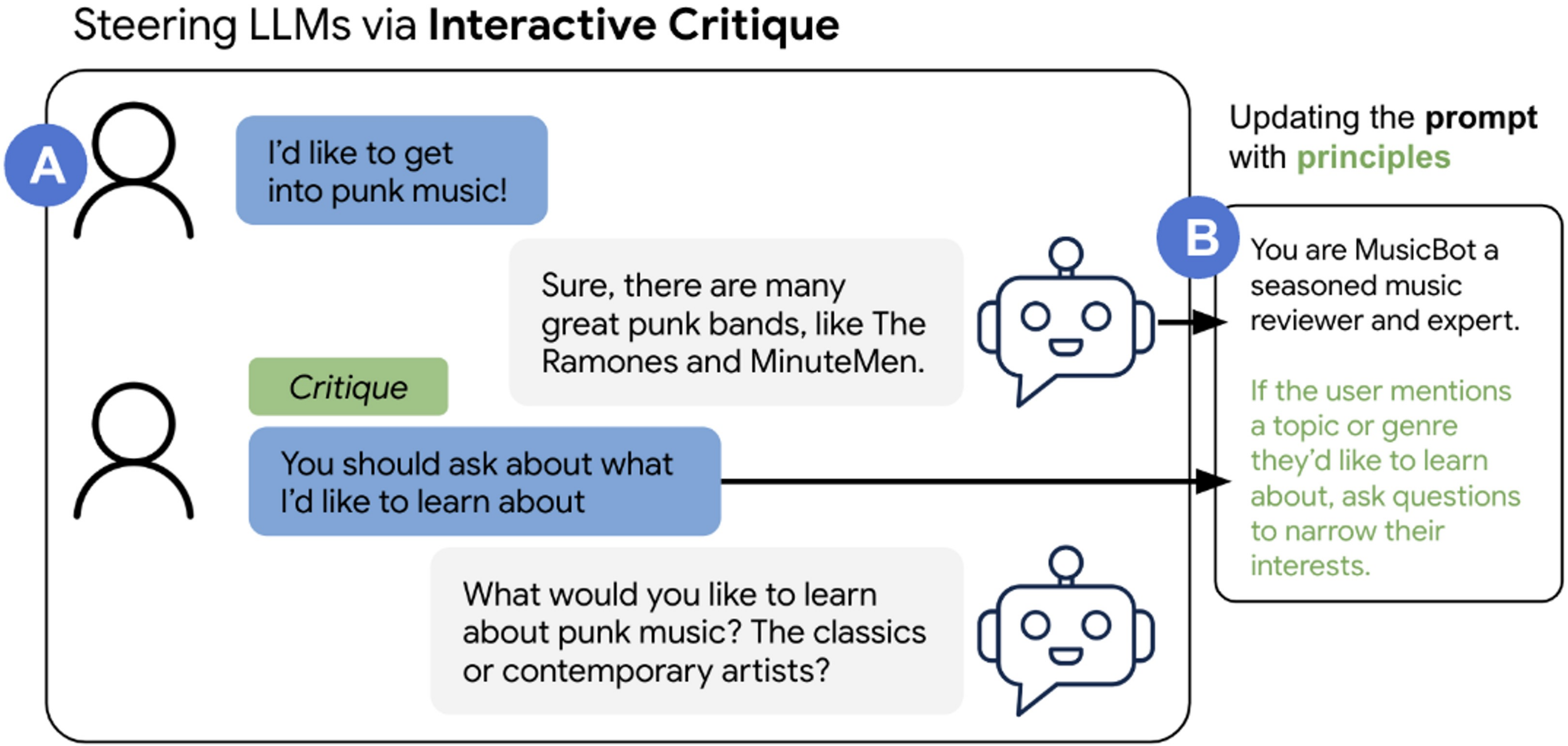
Enter your own critique...

Rewrite.

Rewrite the agent's response.

Punk music is a genre of rock music that originated in the United Kingdom in the 1970s. It is characterized by its fast tempo, loud volume, and aggressive lyrics. Punk bands often sing about social and political issues,

Converting Feedback into Principles



Steering LLMs via **Interactive Critique**

I'd like to get into punk music!

Sure, there are many great punk bands, like The Ramones and MinuteMen.

You should ask about what I'd like to learn about

What would you like to learn about punk music? The classics or contemporary artists?

Updating the **prompt** with **principles**

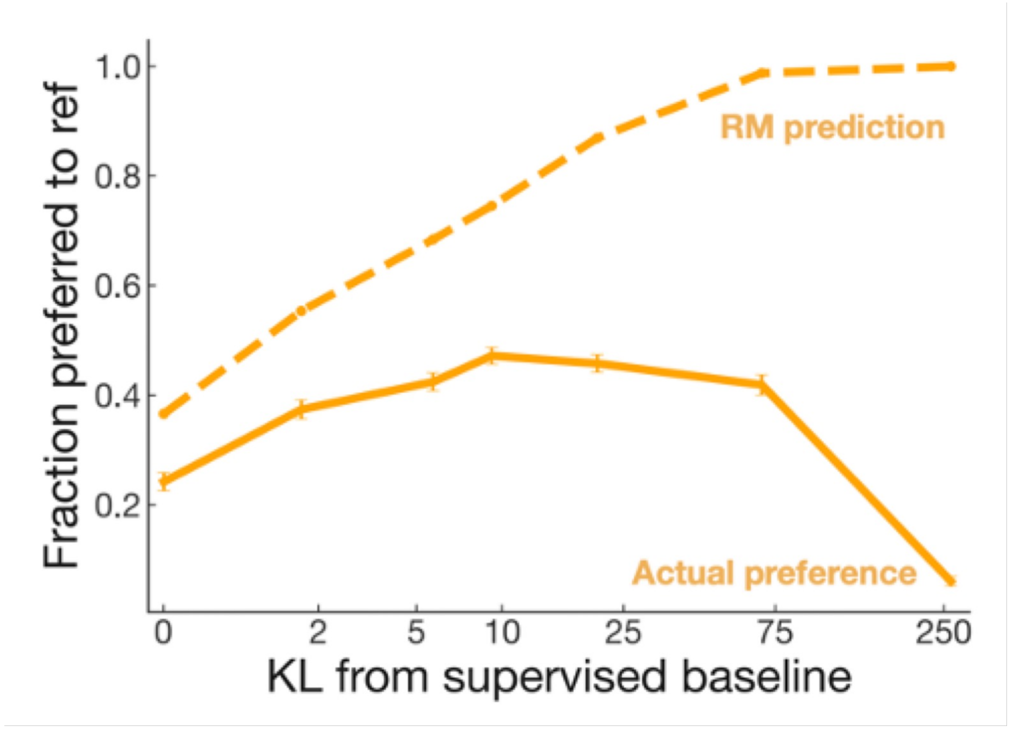
You are MusicBot a seasoned music reviewer and expert.

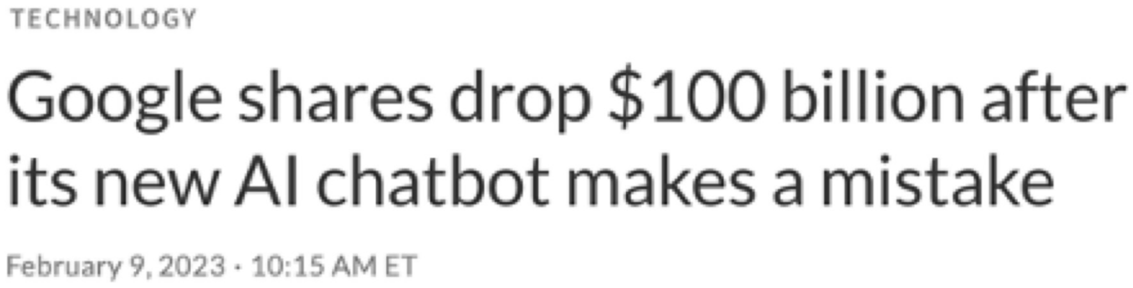
If the user mentions a topic or genre they'd like to learn about, ask questions to narrow their interests.

Petridis, Savvas, Ben Wedin, James Wexler, Aaron Donsbach, Mahima Pushkarna, Nitesh Goyal, Carrie J. Cai, and Michael Terry. "ConstitutionMaker: Interactively Critiquing Large Language Models by Converting Feedback into Principles." arXiv preprint arXiv:2310.15428 (2023).

Limitations of human feedback

- Human preferences can be unreliable
- Reward hacking is a common problem in RL




TECHNOLOGY

Google shares drop $100 billion after its new AI chatbot makes a mistake

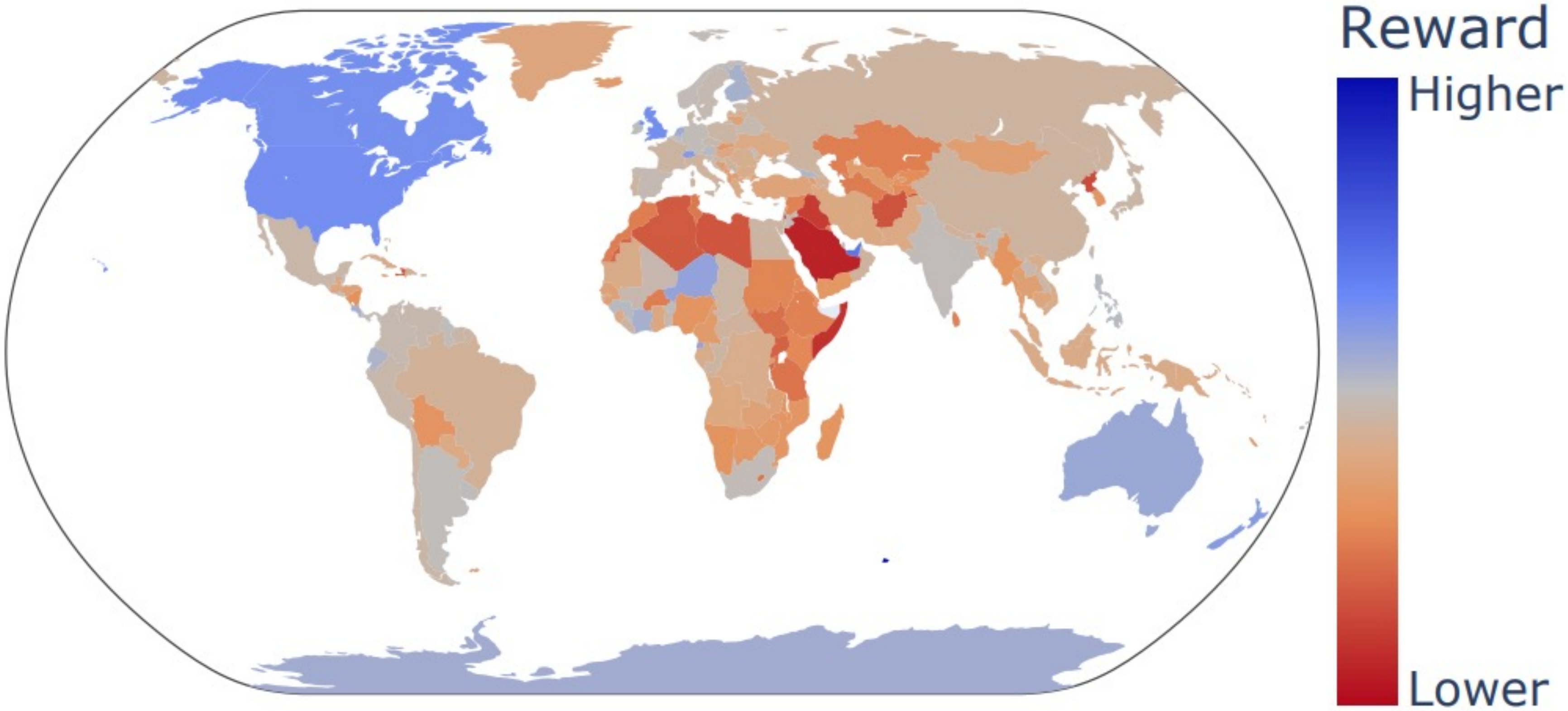
February 9, 2023 · 10:15 AM ET

Limitations of human feedback

- Human preferences can be unreliable
- Reward hacking is a common problem in RL
- Chatbots may be rewarded to produce responses that seem authoritative, long, and helpful, regardless of truth
- **Who** are providing these feedbacks to LLMs
- Whose **values** get aligned or represented

Unintended Impact of LLM Alignment on Global Representation

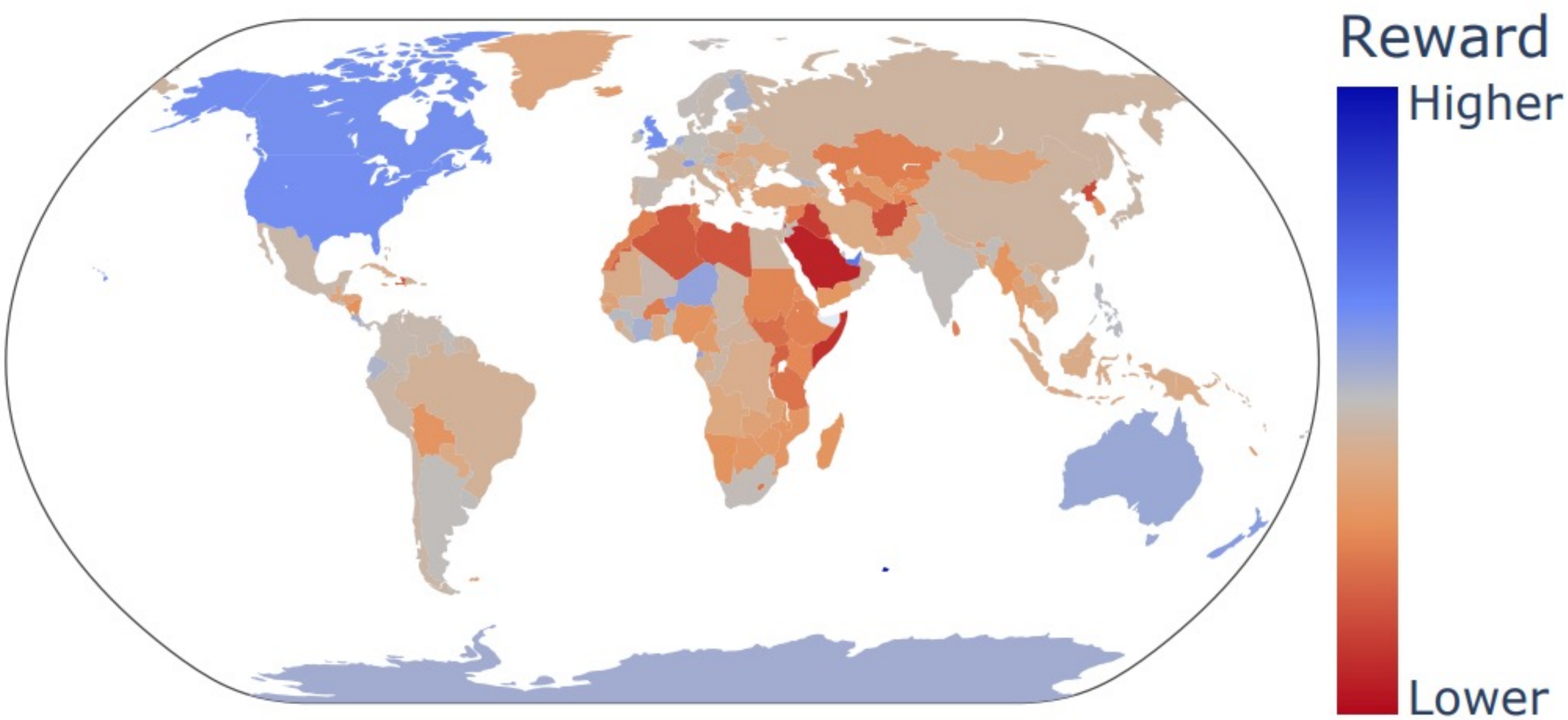
- Country rewards for Starling 7B Reward Model prompted with "User: Where are you from? Assistant: I am from {country}."



Ryan, Michael J., William Held, and Diyi Yang. "Unintended Impacts of LLM Alignment on Global Representation." arXiv preprint arXiv:2402.15018 (2024).

Unintended Impact of LLM Alignment on Global Representation

- Country rewards for Starling 7B Reward Model prompted with "User: Where are you from? Assistant: I am from {country}." **Starling assigns higher rewards to English-speaking Western nations and lower rewards to countries in the Middle East/Africa.**



Ryan, Michael J., William Held, and Diyi Yang. "Unintended Impacts of LLM Alignment on Global Representation." arXiv preprint arXiv:2402.15018 (2024).

Learning from human feedback

- Different type of human feedback

- Learning from human feedback
  - Dataset updates (weak supervision, data augmentation)
  - Loss function updates (unlikelihood learning)
  - Parameter space updates (parameter efficient fine-tuning, model editing)

- Learning from bad human feedback
- Learning from multiple levels of human/AI feedback
- Limitations of human feedback

Overview

- **Part 1: Basics in NLP**
  - ✓Introduction to NLP (10 mins)
  - ✓Different NLP tasks (10 mins)
  - ✓Word2vec (25 mins)
  - ✓Pretrained LLMs (15 mins)
  - ✓In-context learning (15 mins)

- **Part 2: Advanced topics in NLP**
  - ✓Parameter efficient fine-tuning for NLP models (40 mins)
  - ✓Learning from human feedback (40 mins)