Preserving Diversity in Supervised Fine-tuning of Large Lange Models

Ziniu Li

The Chinese University of Hong Kong, Shenzhen

2025-03-23

Overview of This Talk

Evolution of Large Language Models

Key Differences Between LLMs and Traditional Deep Learning

Our Research Contributions

Key Scientific Insights

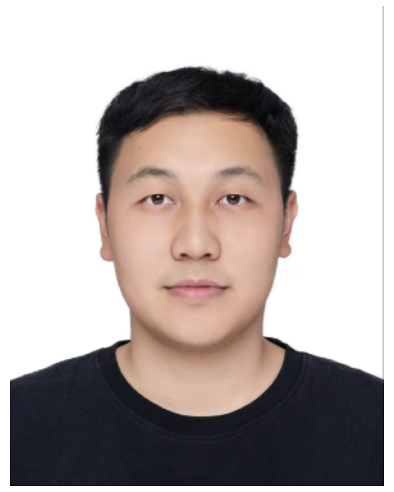
Preserving Diversity in Supervised Fine-Tuning of Large Language Models

Ziniu Li[1,2], Congliang Chen[1,2], Tian Xu[3], Zeyu Qin[4], Jiancong Xiao[5], Zhi-Quan Luo[1,2], and Ruoyu Sun[1,2,†]

ICLR 2025

NeurIPS 2024 FITML Workshop Best Paper Runner-up

Ziniu Li (CUHKSZ)



Congliang Chen (CUHKSZ)

Tian Xu (NJU)

Zeyu Qin (HKUST)

Jiancong Xiao (Upen)

Zhi-Quan Luo (CUHKSZ)

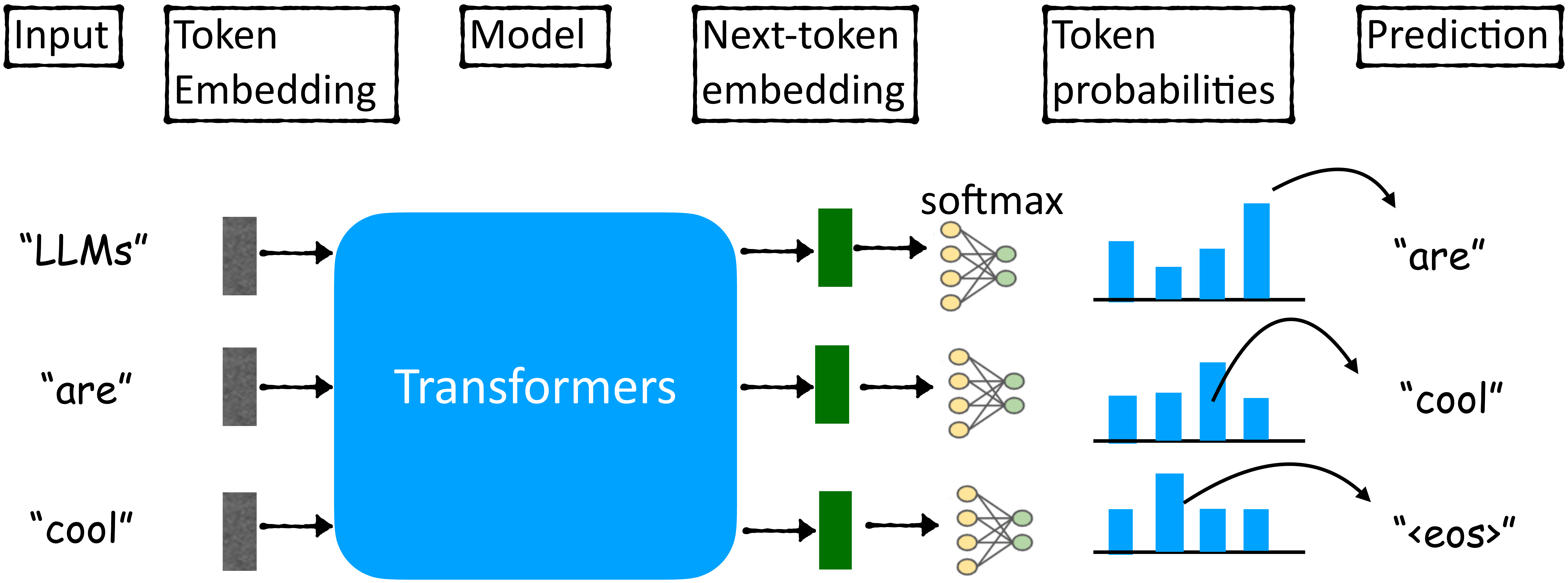
Ruoyu Sun (CUHKSZ)

This Talk

- Why do we study this topic?
- How do we design our approach?
- Is it practically impactful?
- Are there new scientific discoveries?

Part I: Overview of LLMs

LLMs and Transformers



Transformers perform **next-token-prediction**

LLM Pre-training

LLM Pre-training = Transformers + Next-token-Prediction + **Textbook Data**

"Textbooks" can cover:

linguistics
world knowledge
common sense
math reasoning

Ilya Sutskever
(Godfather of ChatGPT)

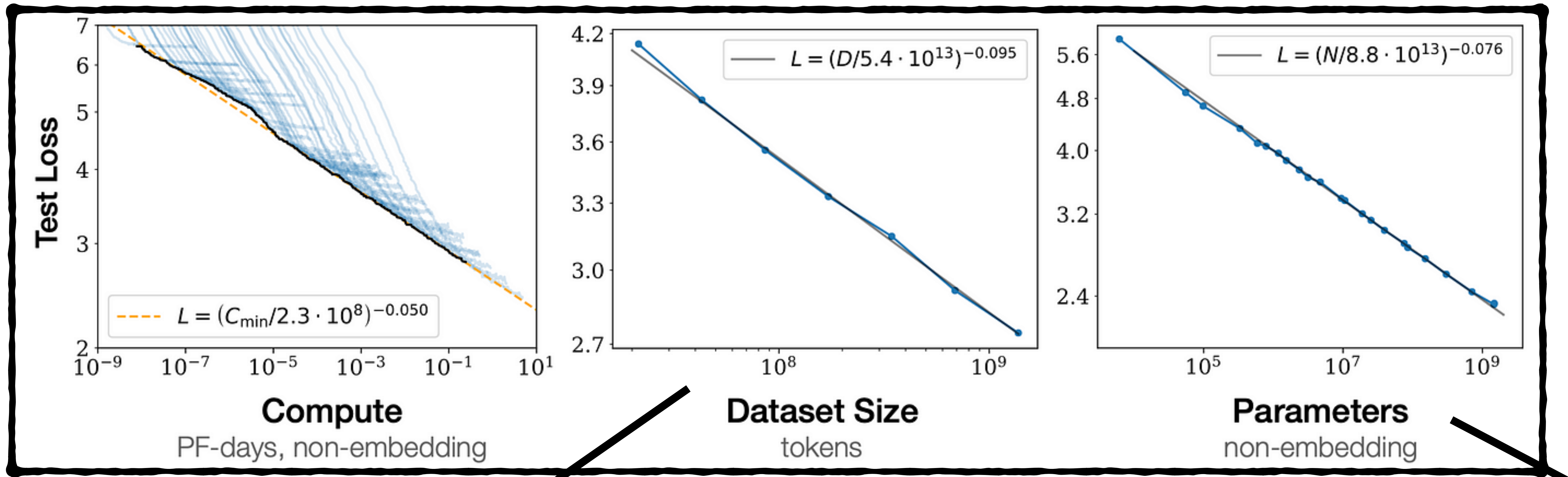
Next-token Prediction is enough for AGI

[https://www.youtube.com/watch?v=YEUclZdj_Sc]

"Textbook" teaches everything
(multi-task learning)

Scaling Law

[Kaplan, Jared, et al. "Scaling laws for neural language models." *arXiv:2001.08361*.]



$$L = \frac{A}{D^{\alpha}} + \frac{B}{N^{\beta}} + L_0$$

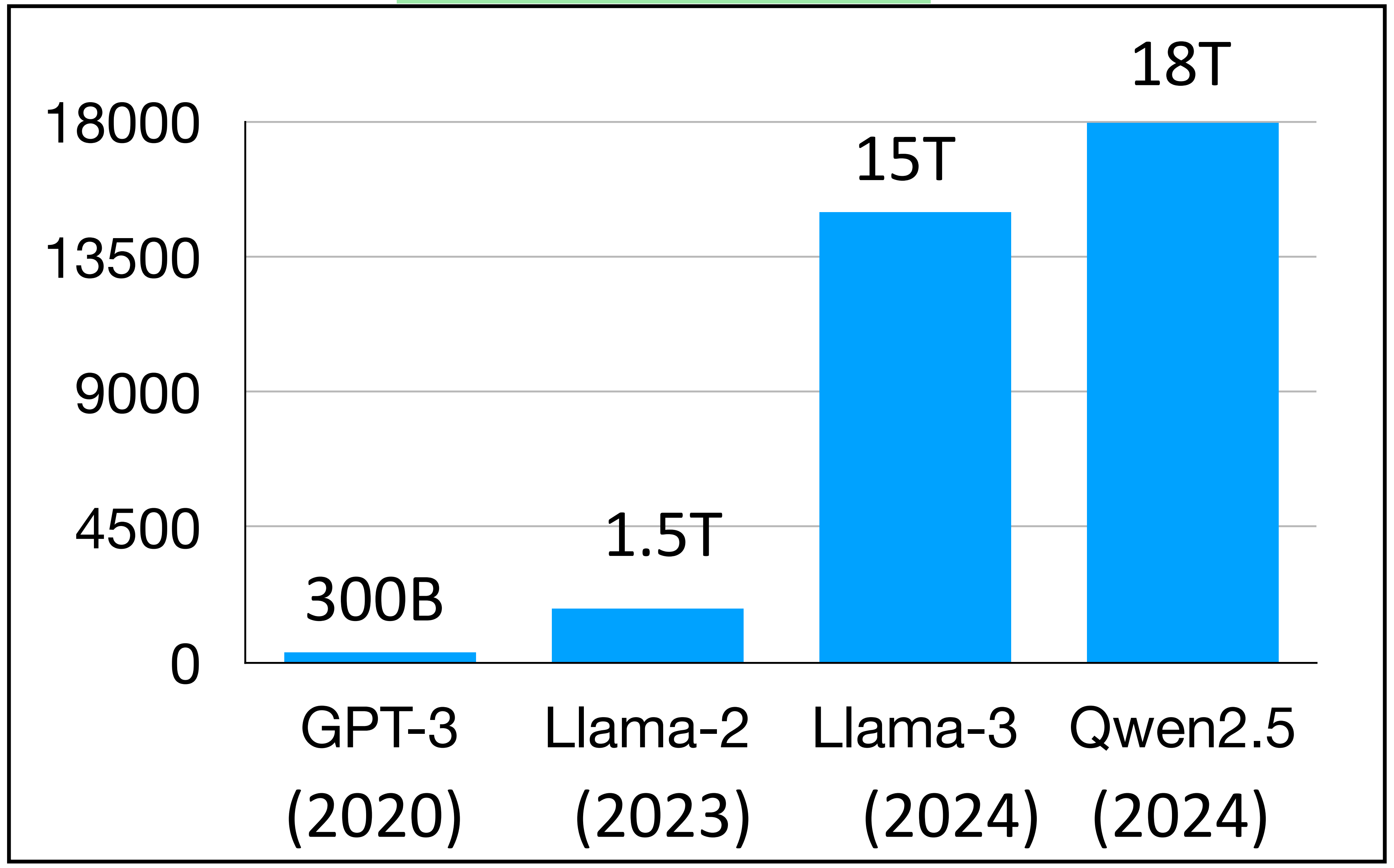
L: Loss

D: dataset size

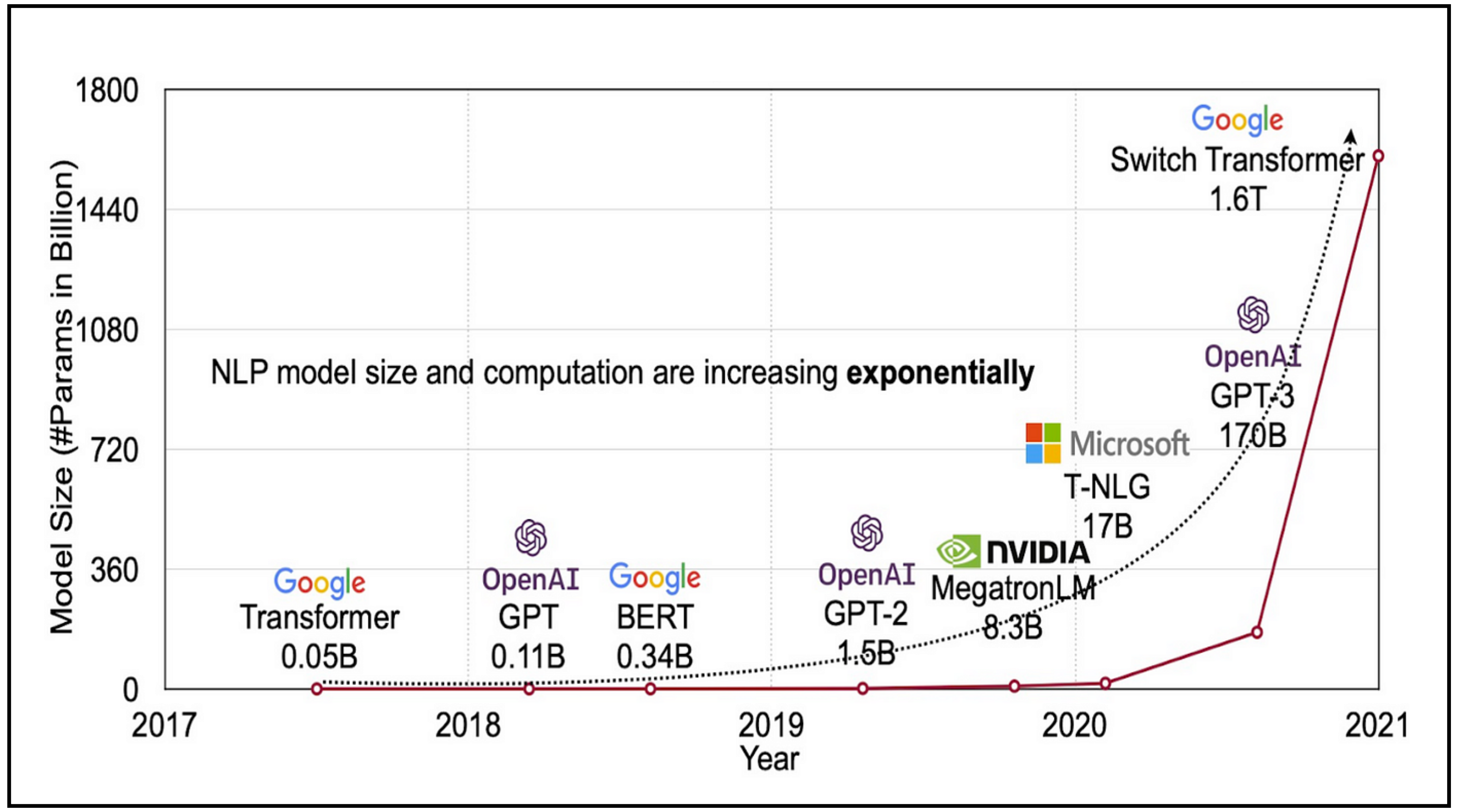
N: number of parameters

A, B: constants; L_0: irreducible loss

dataset size



model size



Pre-training is not Enough Yet

Pre-training

Knowledge Acquisition

Post-training

Ability Reinforcement

Prompt : Explain the pre-training of LLMs.

Llama2-7B: Explain the pre-training of LLMs.
Explain the pre-training of LLMs.
The LLMs are pre-trained on a large amount of unlabeled data, […]

repetitive response

Prompt : Explain the pre-training of LLMs.

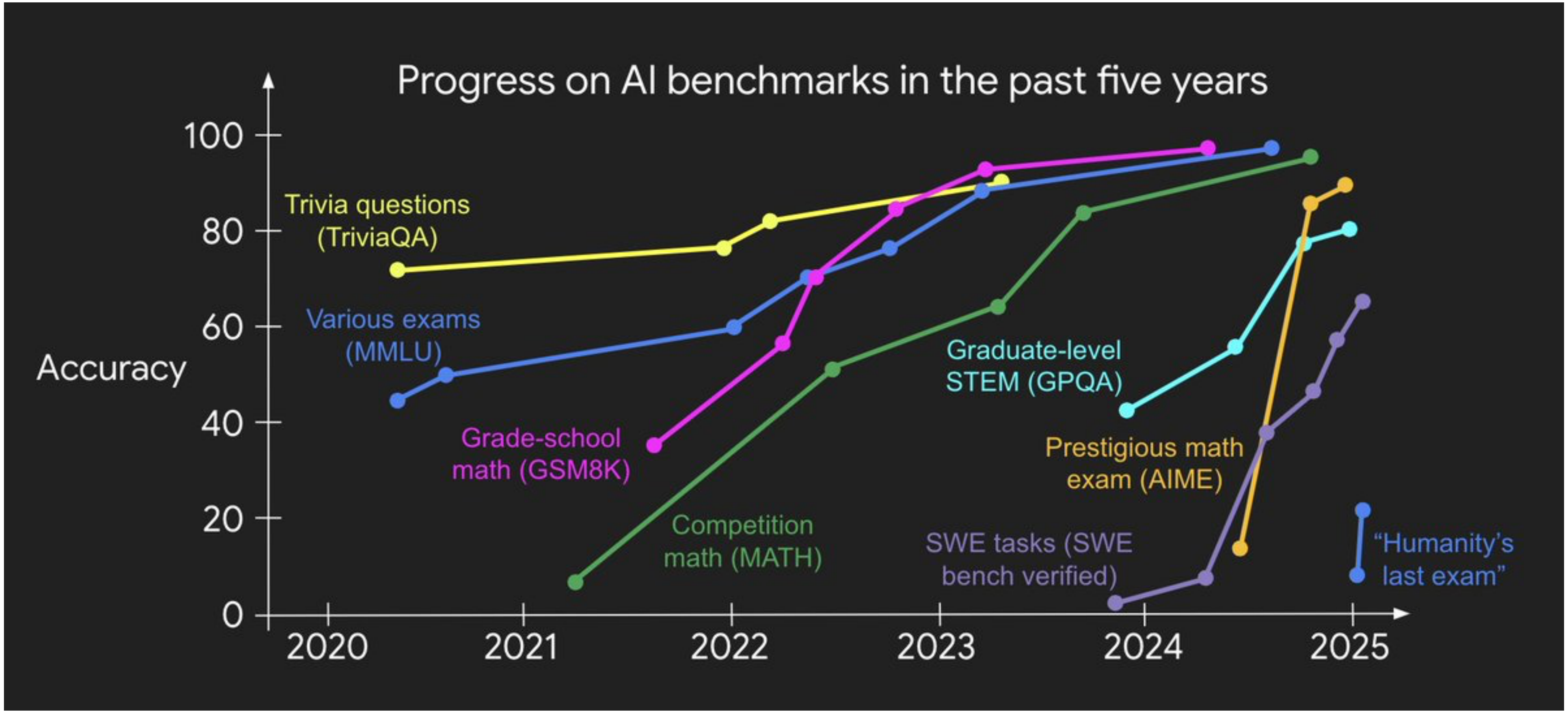
Llama2-7B-Chat: Large language models (LLMs) are trained on a variety of tasks and datasets before they are fine-tuned for specific natural language processing (NLP) tasks. Here's an overview of some common pre-training tasks and their goals: […]

well-organized response

Pre-trained LLMs: Knowledge Learner without Task Context

Post-trained LLMs: Enhanced Multi-task Solver

Post-training is Powerful



Post-training enhances performance for down-stream tasks

What's Next?

[Talk at NeurIPS 2024]

Pre-training as we know it will end

What comes next? The long term is about agentic, reasons, undersatnds, is self aware

2020
(era of GPT-3)

LLMs are few-shot learners

↳ "fine-tuning with few examples is enough"

2024
(era of OpenAI o1)

LLMs are strong reasoners

↳ "post-training is equally important as pre-training"

Part II: Motivation

LLM Post-Training

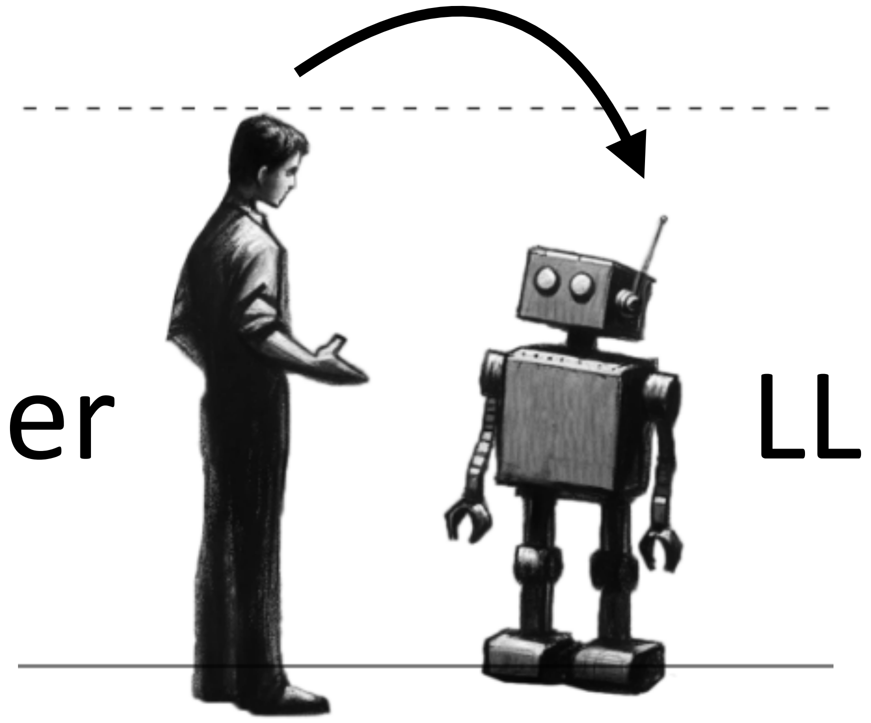
Post Training

Supervised Fine-Tuning (SFT) → Reinforcement Learning (RL)

Goal: Instruction Learning | Ability Enhancement

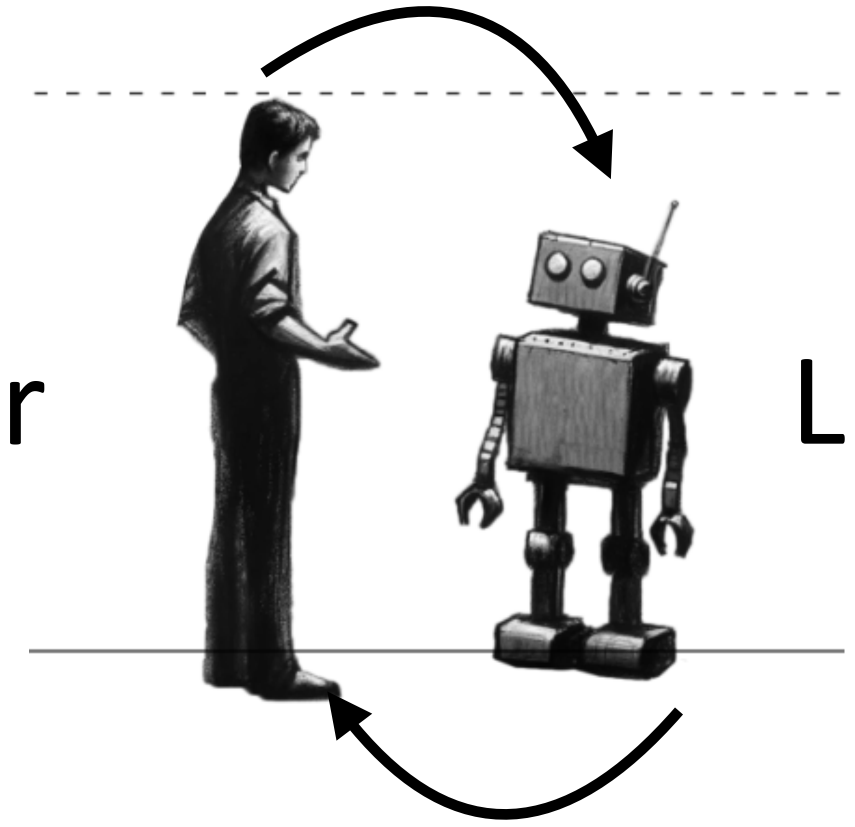
Approach:

Label



Teacher LLM

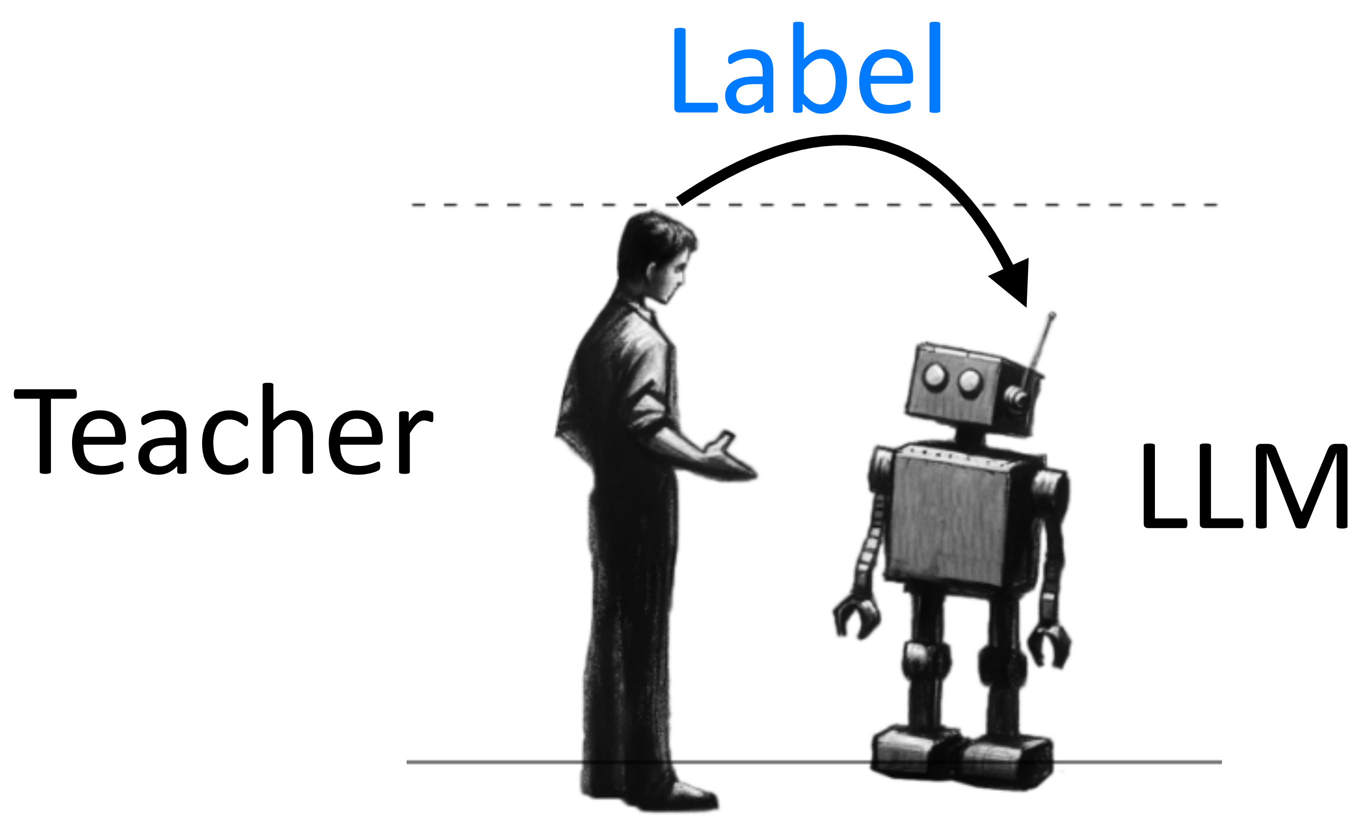
Reward



Teacher LLM

Action / Response

Supervised Fine-tuning



Objective $\max_{\theta} \mathbb{E}_{y \sim p(\cdot|x)}[\log f_\theta(y|x)]$

x: prompt y: response/completion (label)

p: data distribution (from teacher)

f_θ: distribution of LLM

SFT Data Example

Prompt: Q: Can Geoffrey Hinton have a conversation with George Washington?

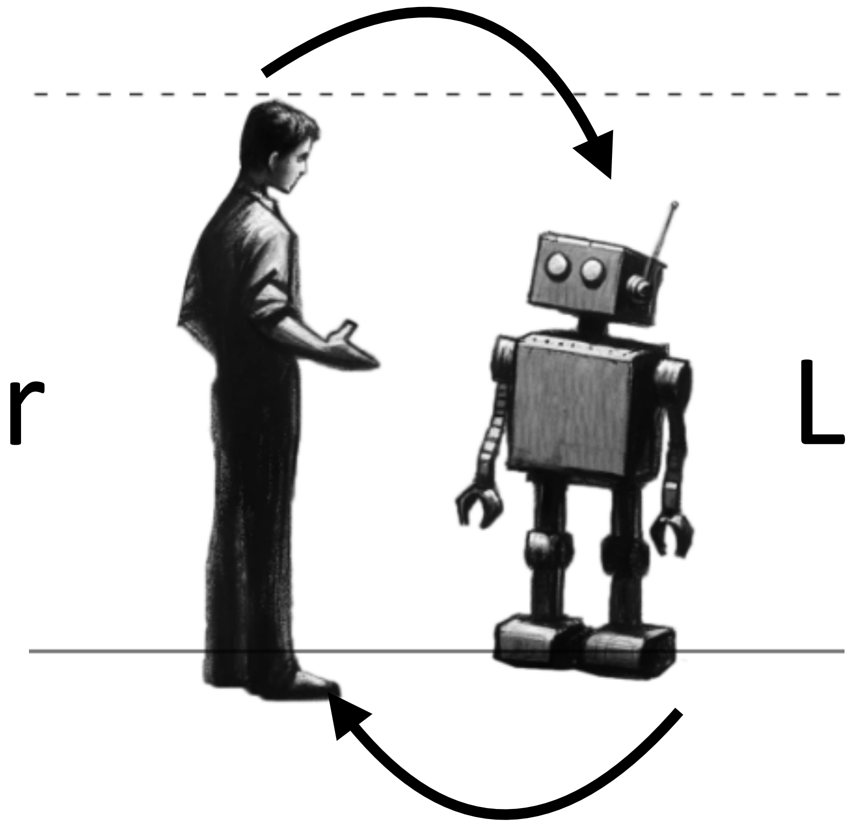
Label: A: The answer is No because Geoffrey Hinton was born in 1947, while [...]

LLMs learn to **understand the question** (task) and **provide** relevant **answers**

Reinforcement Learning

Reward

Teacher



LLM

Response

Objective $\max_{\theta} \mathbb{E}_{y \sim f_\theta(\cdot|x)}[r(x, y)]$

Framework: Generation ⇄ Verification

LLMs learn to **correct mistakes** and **enhance confidence** in answering questions

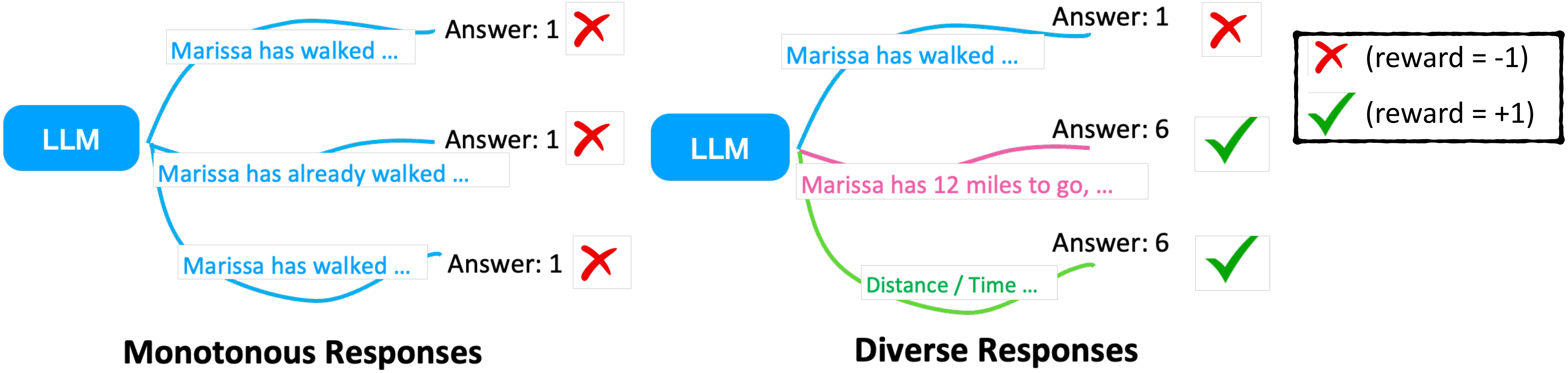
RL Data Example

Prompt	Q: How many ‘r’ in strawberry?
LLM Response	A: There is one ‘r’ in ‘stra’ and another ‘r’ in ‘berry’, so the answer is 2
Teacher Feedback	Reward = -1

Output Diversity

Question: Marissa is hiking a 12-mile trail. She took 1 hour to walk the first 4 miles, then another hour to walk the next two miles. If she wants her average speed to be 4 miles per hour, what speed (in miles per hour) does she need to walk the remaining distance?

Answer: 6



Greater **Diversity** Leads to Exploration of Better Solutions

SFT Reduces Model Output Diversity

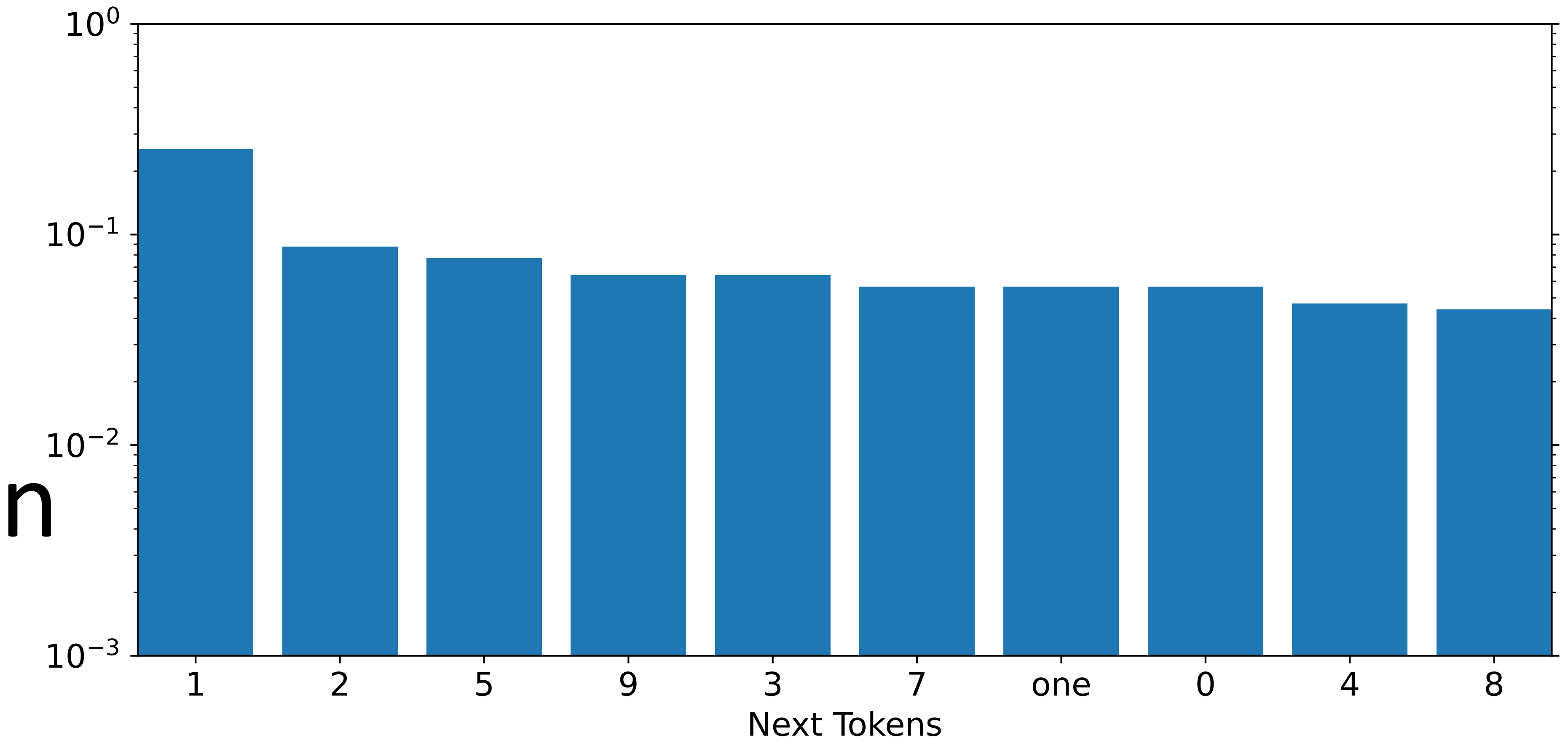
#1

Prompt

Give me a single-digit number

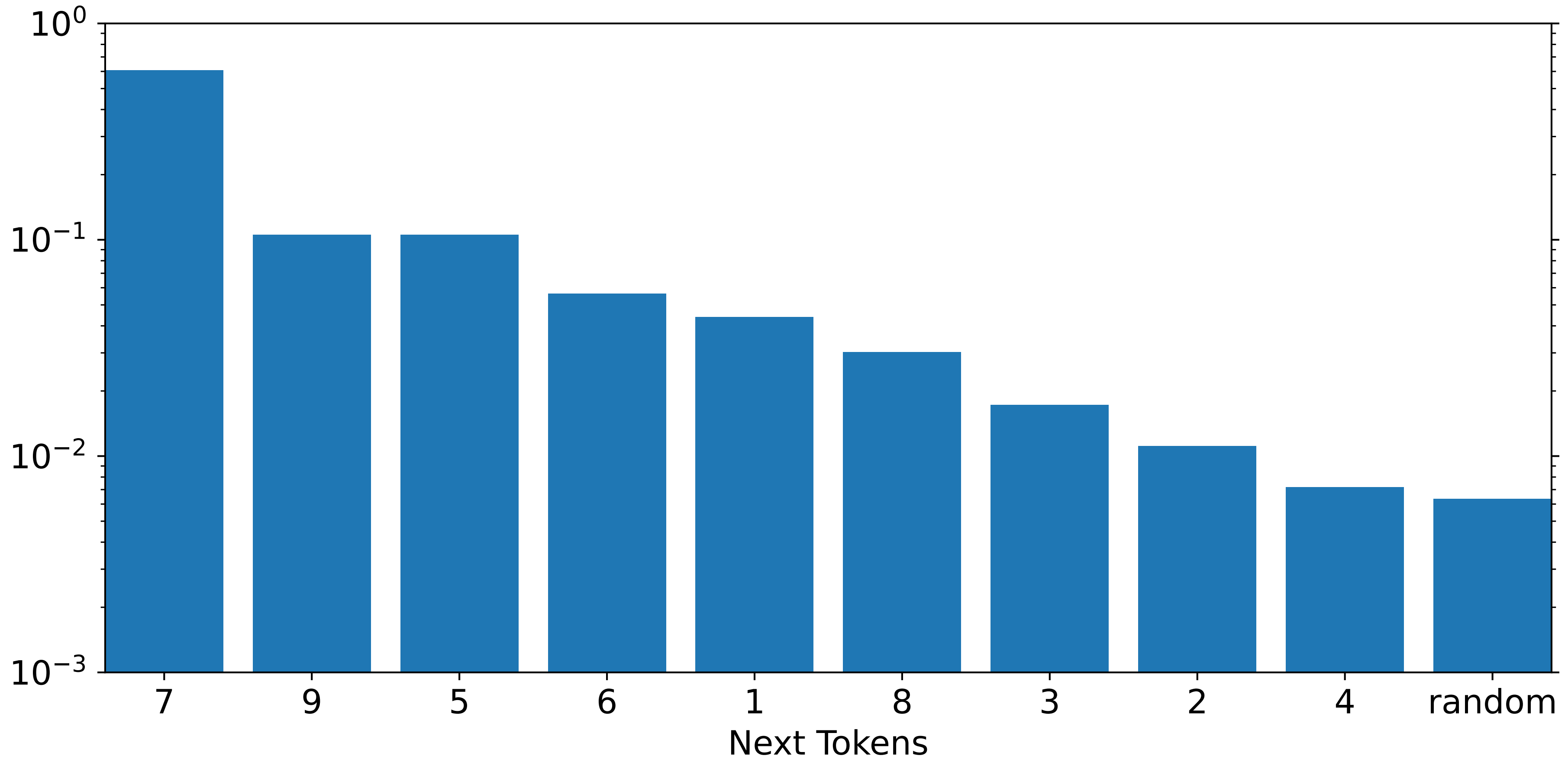
Response Distribution

Pre-trained LLM



"near uniform"

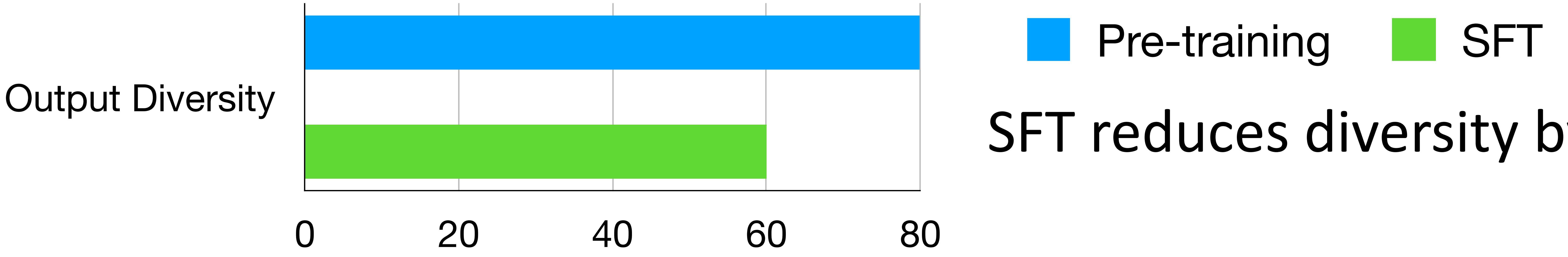
Pre-trained LLM + SFT



"biased toward 7"

#2

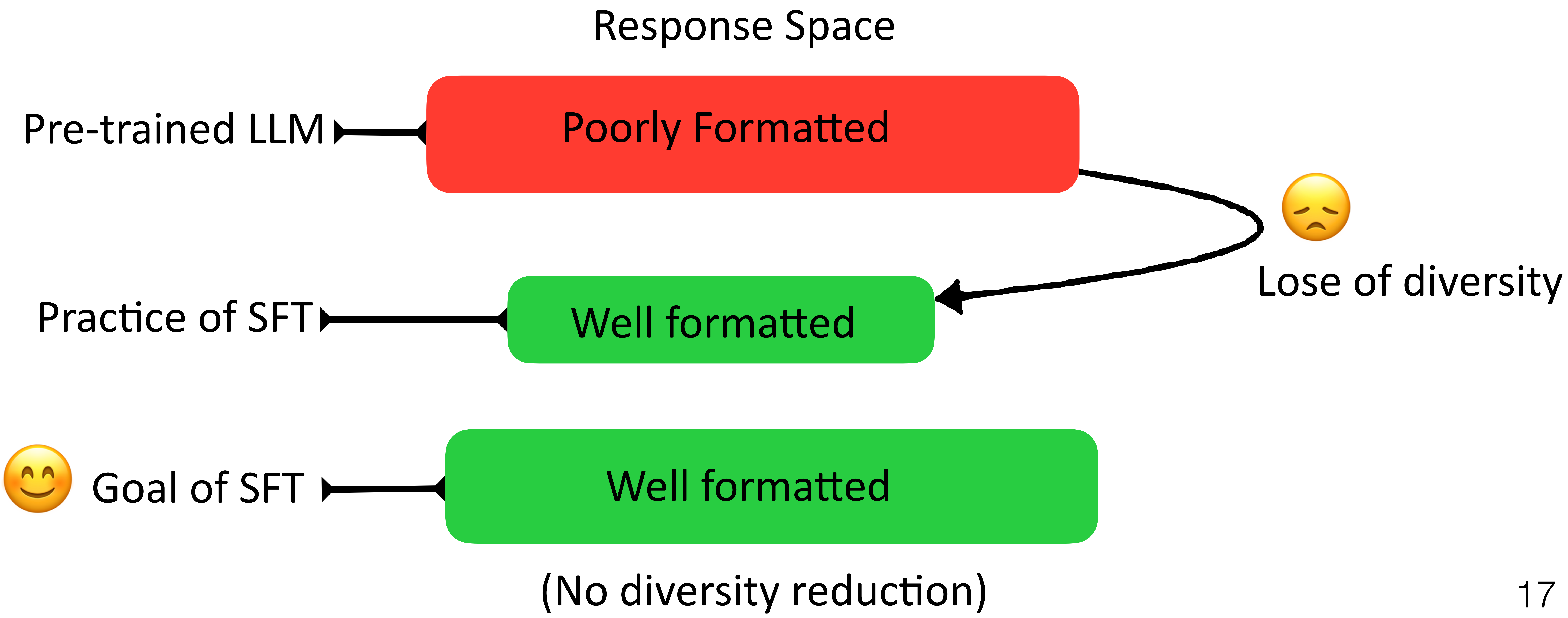
Output Diversity Statistics



SFT reduces diversity by ~20%

Revisiting SFT

SFT aims to align pre-trained model outputs to RL/human-preferred format
(outputs that are easy to read, interpret, and verify)



Why does Diversity Fate in SFT?

Elements of SFT	Properties	Comment
Model	Pre-trained with rich knowledge encoded	nothing to blame
Data	Limited Size and Coverage (10B-100B in SFT v.s. 1T-10T in pre-training)	not perfect but cannot blame
Algorithm	**Minimizing cross-entropy (CE) loss**	**is this good?**

CE seems Effective for …

Input

Model

Prediction

Label

Convolution Neural Network

“Dog”

“Cat”

Cross-Entropy Loss

Back-propagation

CE is Effective for **Classification**

“I like to drink”

Langauge Model

“Tea”

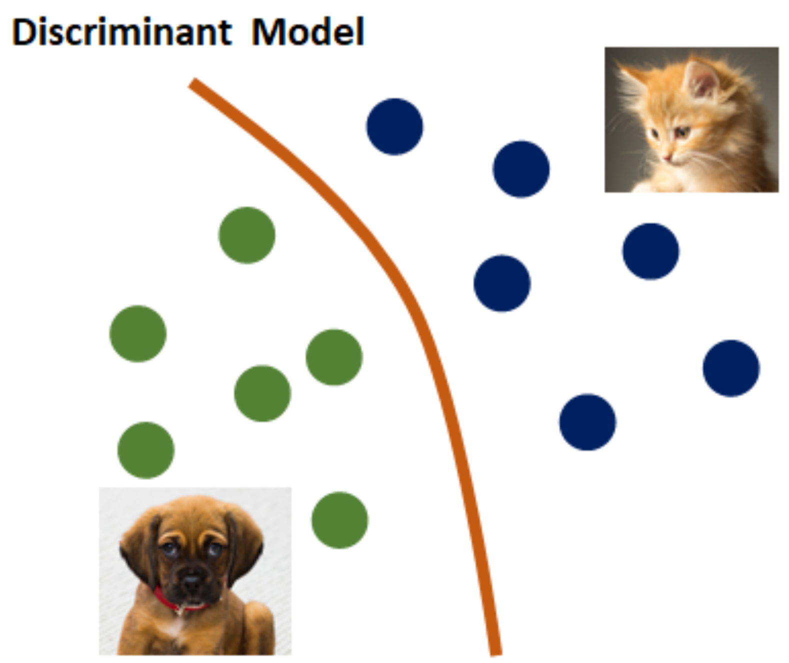
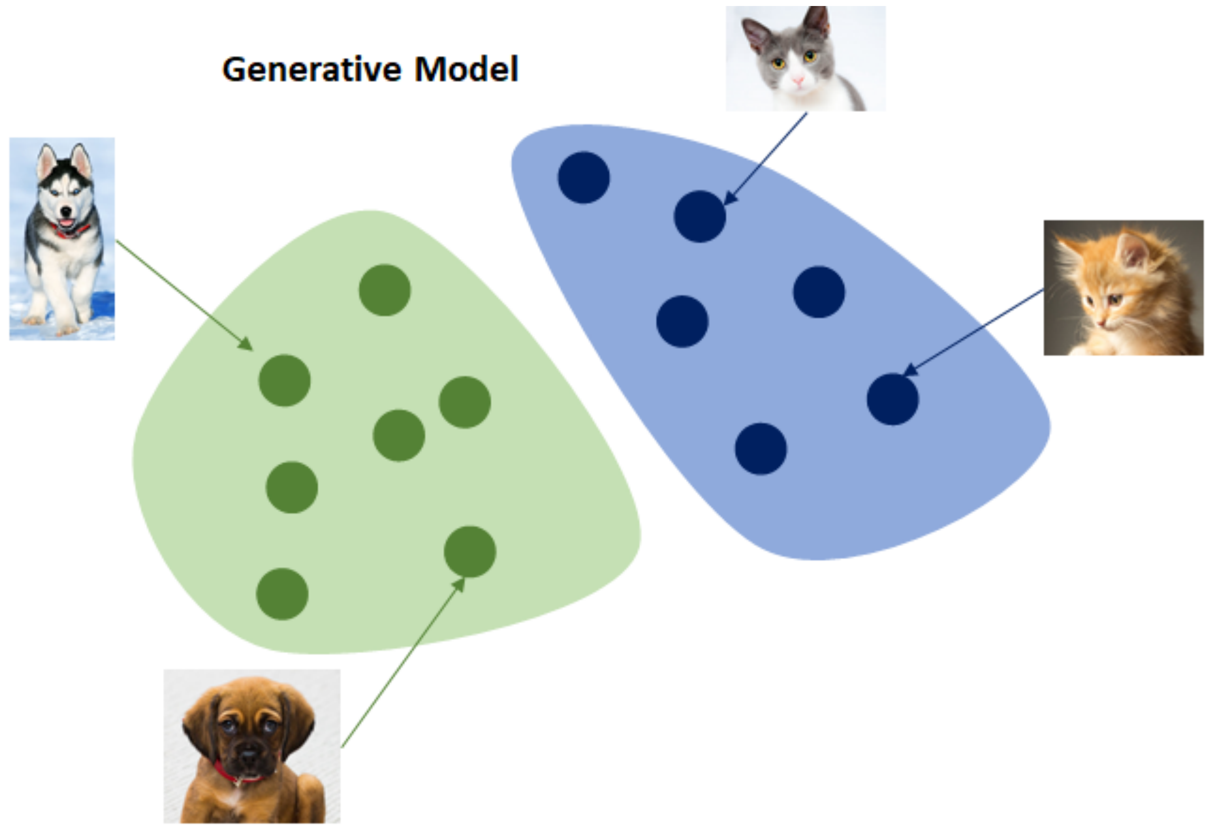
“Coffee”

Cross-Entropy Loss

Back-propagation

Is CE Effective for **Generation**?

Understanding Generation Tasks

	Classification	Generation
Target	$\mathscr{X} \mapsto \mathscr{Y}$ (**function**: many-to-one)	$\mathscr{X} \mapsto \Delta(\mathscr{Y})$ (**distribution**: one-to-many)
Illustration		

Remark for LLMs:

- responses are **not unique** (variation in formats, styles, or reasoning paths)
- (SFT) data is hard to cover all cases

Theory of CE

CE Loss (Empirical)

$$\min_{\theta} - \sum_{(x_i, y_i)\sim D} y_i^{\top} \log f_\theta(y_i \mid x_i)$$

(x_i, y_i): input-label pair

$f_\theta(y \mid x)$: the conditional prediction distribution

θ: parameters of neural network

CE Loss (Population)

$$\max_{\theta} \mathbb{E}_{x\sim\rho} \mathbb{E}_{y\sim p(\cdot\mid x)} \log f_\theta(y \mid x)$$

ρ: prompt distribution

$p(\cdot \mid x)$: the conditional data distribution to learn

Equivalence

Forward KL Divergence

$$\min_{\theta} \mathbb{E}_{x\sim\rho} \mathrm{KL}(p(\cdot \mid x), f_\theta(\cdot \mid x)) + \text{constant}$$

Distribution Matching

CE can be used to learn a distribution

If the data samples are "abundant"

✅ Classification (one label sample is enough)

✅ Pre-training (huge data)

❌ SFT (data is limited)

Summary

Challenge:
We need to protect LLM's output diversity during SFT

Understanding:
CE easily fits to the empirical data and loses the diversity

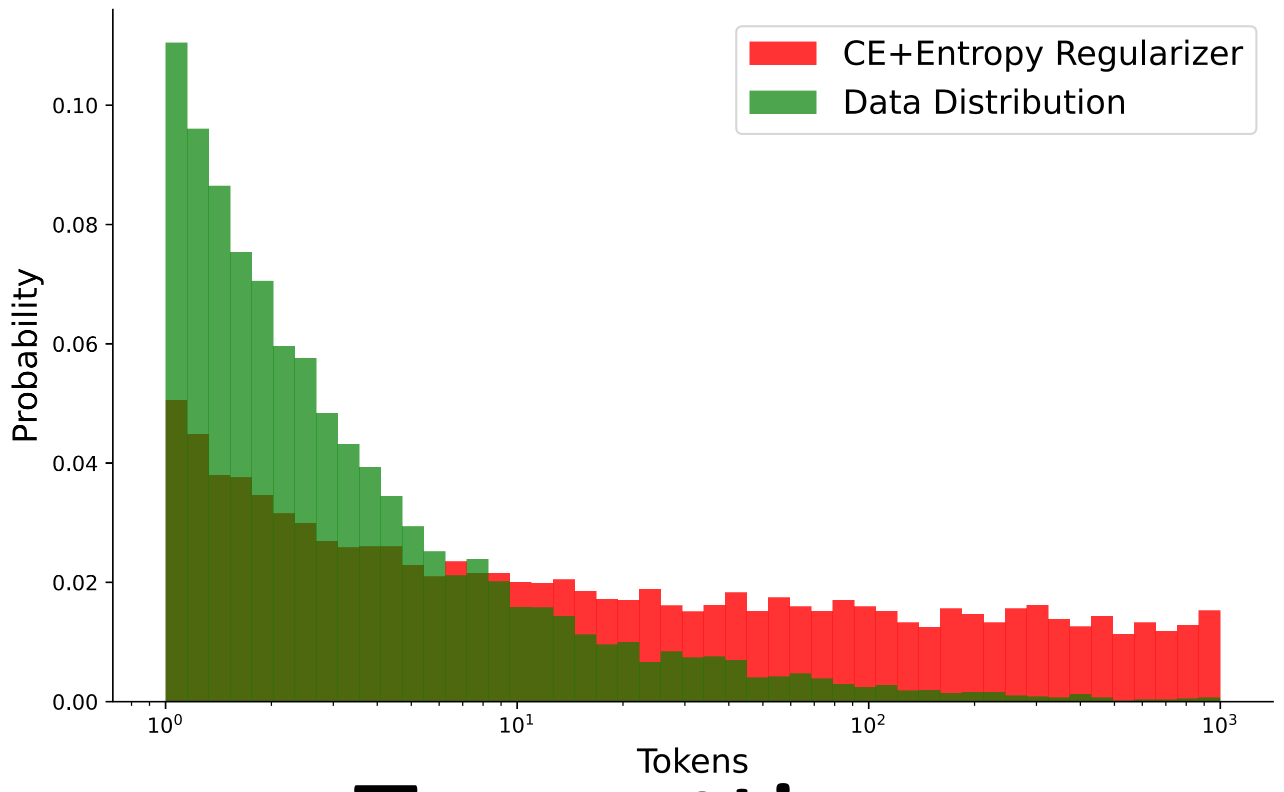
Goal:
Designing new formulation and algorithm for SFT

Part III: Our Approach GEM

A Naive Approach for Diversity

CE + Entropy Regularization

$$\max_f \underbrace{\mathbb{E}_x \mathbb{E}_{y\sim p(\cdot|x)}[\log f(y|x)]}_{-D_{\mathrm{KL}}(p,f)+\text{constant}} + \beta \underbrace{\mathbb{E}_x \mathbb{E}_{y\sim f(\cdot|x)}[-\log f(y|x)]}_{\mathcal{H}(f)}$$



Toy setting

Prompt	Whats the largest star in our galaxy?
CE	Hello! Atlantis is a legendary city that was said to have existed in ancient Greece. According to the story, it was a highly advanced and prosperous city that was located on an island in the ocean. [...]
CE+Entropy	Hello! Atlantis Documentary is a 2019 American documentaryéhoFLICT film directed by Já oblík and produced by Werner Herzog. The film explores the history and legacy of Atlantis, □ an ancient Greek city-state that was said to have_calendar knowledge and advanced technology, through interviews with scholars and histori-ans.ython

LLMs

Entropy regularizer encourages diversity via increasing the **tail** of distribution

Analyzing Cross-Entropy Loss

Setting: $y \sim f_\theta(\,\cdot\,|\,x)$ and $f_\theta(i\,|\,x) = \dfrac{\exp(\theta_i)}{\sum_{j=1}^{K}\exp(\theta_j)}$

Gradient of CE: assuming i-th token is the label

$$-\nabla_\theta \mathcal{L}_{\mathrm{CE}}(\theta) = [-f_\theta(1|x), -f_\theta(2|x), \ldots, 1 - f_\theta(i|x), \ldots, -f_\theta(K|x)].$$

Implication:

Target token (label)'s logit ↑ while other tokens' logits ↓

Distribution Matching as Flow Transfer

Proposition 1. *The gradient of CE specifies a logit flow map: each source token j transfers $f_\theta(j|x)$ logits to the target token i. Formally,*

$$\boxed{\begin{aligned} -\nabla_\theta \mathcal{L}_{\mathrm{CE}}(\theta) &= \sum_{j=1, j\neq i}^{K} w_{i\leftarrow j} \cdot e_{i\leftarrow j} \\ w_{i\leftarrow j} &= f_\theta(j|x) \\ e_{i\leftarrow j} &= [0 \ \cdots \ \underbrace{1}_{i\text{-th position}} \ \cdots \ \underbrace{-1}_{j\text{-th position}} \ \cdots \ 0] \end{aligned} \quad (2)}$$

Example: $f_\theta = [0.1, 0.3, 0.6]$ Label: #2

Gradient: $g = [-0.1, 0.7, -0.6]$

Flow perspective: $g = 0.1 * [-1 \quad 1 \quad 0] + 0.6 * [0 \quad 1 \quad -1]$

Logits flow from **source** tokens = Logits flow to **target** token

Limitations of CE

Procedure of CE

#1 While there exists source token $j \neq i$ with $f_{\theta_k}(j|x) > 0$, continue the following steps.

#2

- Find any j with $f_{\theta_k}(j|x) > 0$
- Decrease the logit for source token j by learning rate η and weight $w_{i \leftarrow j}$:

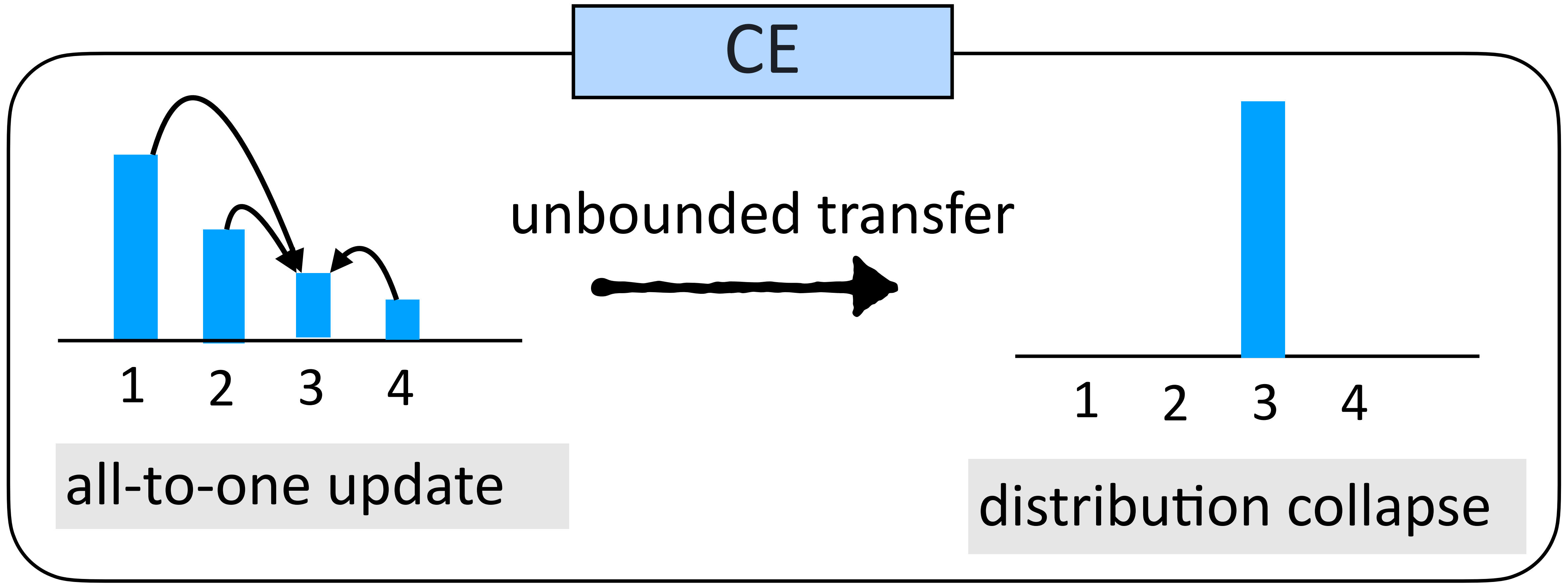
$$\theta_{k+1}[j] = \theta_k[j] - \eta * w_{i \leftarrow j}$$

- Increase the logit for the target token i in a similar manner:

$$\theta_{k+1}[i] = \theta_k[i] + \eta * w_{i \leftarrow j}$$

Limitation 1: Unbounded Transfer

Limitation 2: All-to-one Update



Proposed Solutions

Procedure of Our Method

#1 While the target token $i \notin \text{argmax } f_{\theta_k}(\cdot|x)$, continue the following steps.

#2
- Calculate the model's best prediction $j = \text{argmax } f(\cdot|x)$
- Decrease the logit for source token j by learning rate η and weight $w_{i \leftarrow j}$:

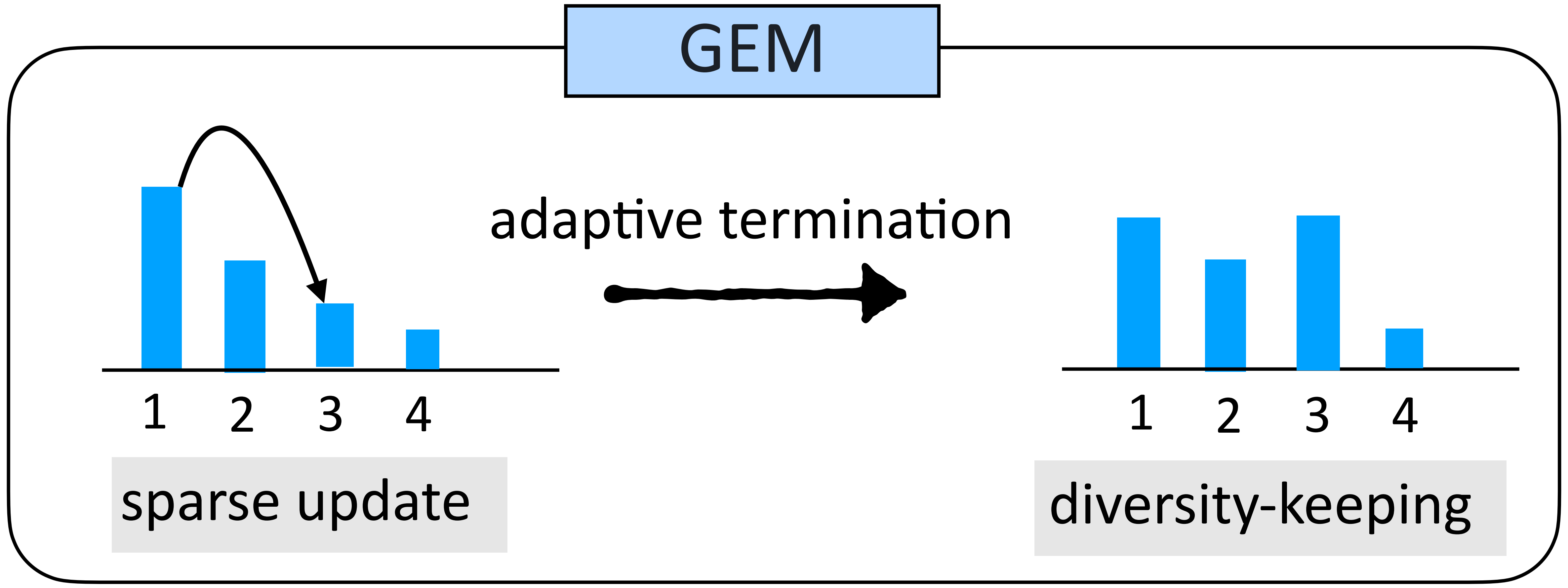
$$\theta_{k+1}[j] = \theta_k[j] - \eta * w_{i \leftarrow j}$$

- Increase the logit for the target token i in a similar manner:

$$\theta_{k+1}[i] = \theta_k[i] + \eta * w_{i \leftarrow j}$$

Technique 1: Adaptive Termination

Technique 2: Sparse Update



Our Insight: Dimension Increase

Procedure of Our Method

While the target token $i \notin \text{argmax } f_{\theta_k}(\cdot|x)$, continue the following steps.

- Calculate the model's best prediction $j = \text{argmax } f(\cdot|x)$
- Decrease the logit for source token j by learning rate η and weight $w_{i \leftarrow j}$:

$$\theta_{k+1}[j] = \theta_k[j] - \eta * w_{i \leftarrow j}$$

- Increase the logit for the target token i in a similar manner:

$$\theta_{k+1}[i] = \theta_k[i] + \eta * w_{i \leftarrow j}$$

What is the magic? Can we generalize this to neural network training?

Introduce an **auxiliary variable** (dimension increase) that implements the scheme of sparse update and adaptive termination

Towards a Game Formulation

High-level design: introduce an **another player** q to the distribution matching

$$\min_f \quad \mathcal{L}(f, q) \triangleq \mathbb{E}_x \mathbb{E}_{y^{\texttt{real}} \sim p(\cdot|x)} \mathbb{E}_{y^{\texttt{gene}} \sim q(\cdot|x)} \left[\log f(y^{\texttt{gene}}|x) - \log f(y^{\texttt{real}}|x) \right]$$

$$\max_q \quad \mathcal{Q}(f, q) \triangleq \mathbb{E}_x \mathbb{E}_{y^{\texttt{gene}} \sim q(\cdot|x)} \left[\log f(y^{\texttt{gene}}|x) \right] + \beta \cdot \mathcal{H}(q(\cdot|x)).$$

Intuitive Understanding:

- f: increase the likelihood on real data and decrease likelihood on the generated data
- q: increase the energy induced by $\log f$ with entropy regularization

Understanding the Game

main player

$$-\nabla_\theta \mathcal{L}(f_\theta, q) = \sum_{j=1, j\neq i}^{K} w_{i\leftarrow j} \cdot e_{i\leftarrow j},$$

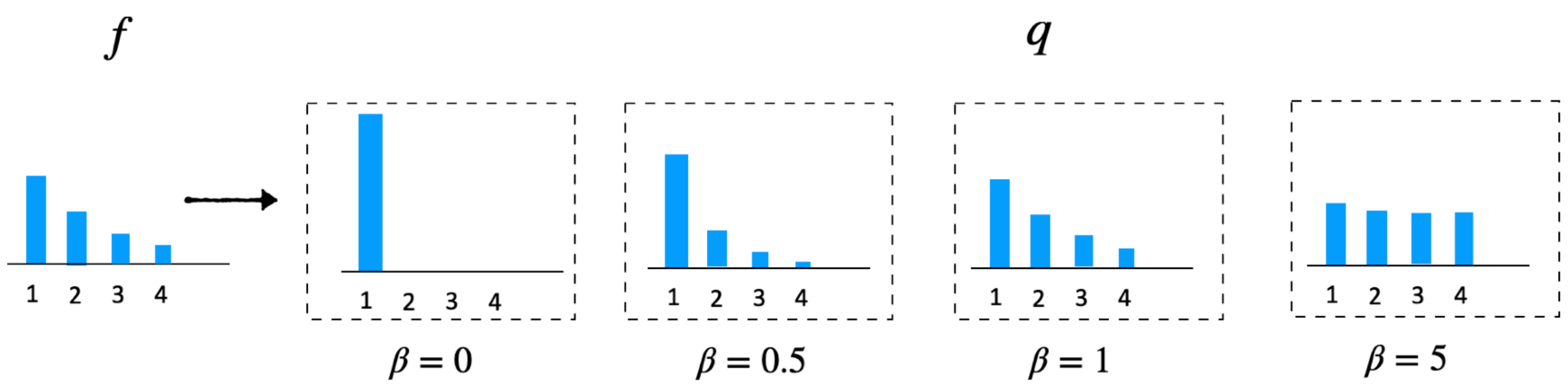
$$w_{i\leftarrow j} = q(j|x).$$

→ flow transfer

→ controller

meta-controller

$$\underset{q}{\text{argmax}}\, \mathcal{Q}(q, f) = \begin{cases} \delta_j(x) \text{ with } j = \text{argmax } f_i(\cdot|x) & \text{if } \beta = 0 \\ \mathtt{softmax}(1/\beta * \log f(y|x)) & \text{if } \beta > 0 \end{cases}$$



$\beta \to 0$: sparse update

$\beta \to 1$: same as CE

$\beta \to \infty$: uniform update

Connection with Probability Transfer

Proposition 2. *For a data distribution satisfying $p(y|x) > 0$, with $\beta > 0$, the game in Equations (3) and (4) posses a unique Nash equilibrium point:*

$$\begin{cases} f^\star = \mathtt{softmax}(\beta * \log p) \\ q^\star = p \end{cases} \tag{7}$$

Furthermore, $f^\star$ corresponds to the optimal solution to the distribution matching problem (with $1/\beta = (\gamma + 1)$), which minimizes the reverse KL divergence with entropy regularization:

$$f^\star = \operatorname*{argmin}_f \mathbb{E}_x \left[D_{\mathrm{KL}}(f(\cdot|x), p(\cdot|x)) - \gamma \mathcal{H}(f(\cdot|x)) \right]. \tag{8}$$

Terminology	**Reserve** KL Minimization	**Entropy** Maximization
Role	Fit the data distribution	Protect the output diversity

For $\beta = 0$, there are **multiple** Nash equilibrium points with non-closed-form solutions → future work

Training Algorithm

Idea: block-wise gradient-descent and coordinate descent

$$\begin{cases} f_{\theta_{k+1}} = f_{\theta_k} - \nabla_\theta \mathcal{L}(f_\theta, q_k)\mid_{\theta=\theta_k} \\ q_{k+1} = \operatorname{argmax}_q \mathcal{Q}(f_{\theta_{k+1}}, q) = \mathtt{softmax}(1/\beta * \log f_{\theta_{k+1}}) \end{cases}$$

Optimization with the token space (**discrete**)

Feature 1: **Single**-model optimization

There is no need of storing and explicit training of q

Feature 2: **Variance-reduced** gradient estimation

$$\mathcal{L}_{\mathrm{GEM}}(\theta) = \sum_i \sum_{y^{\mathtt{gene}}} q_k(y^{\mathtt{gene}}|x_i) \cdot \left[\log f_\theta(y^{\mathtt{gene}}|x_i) - \log f_\theta(y_i^{\mathtt{real}}|x_i)\right]$$

We use the exact distribution (in GANs, stochastic approximation is used)

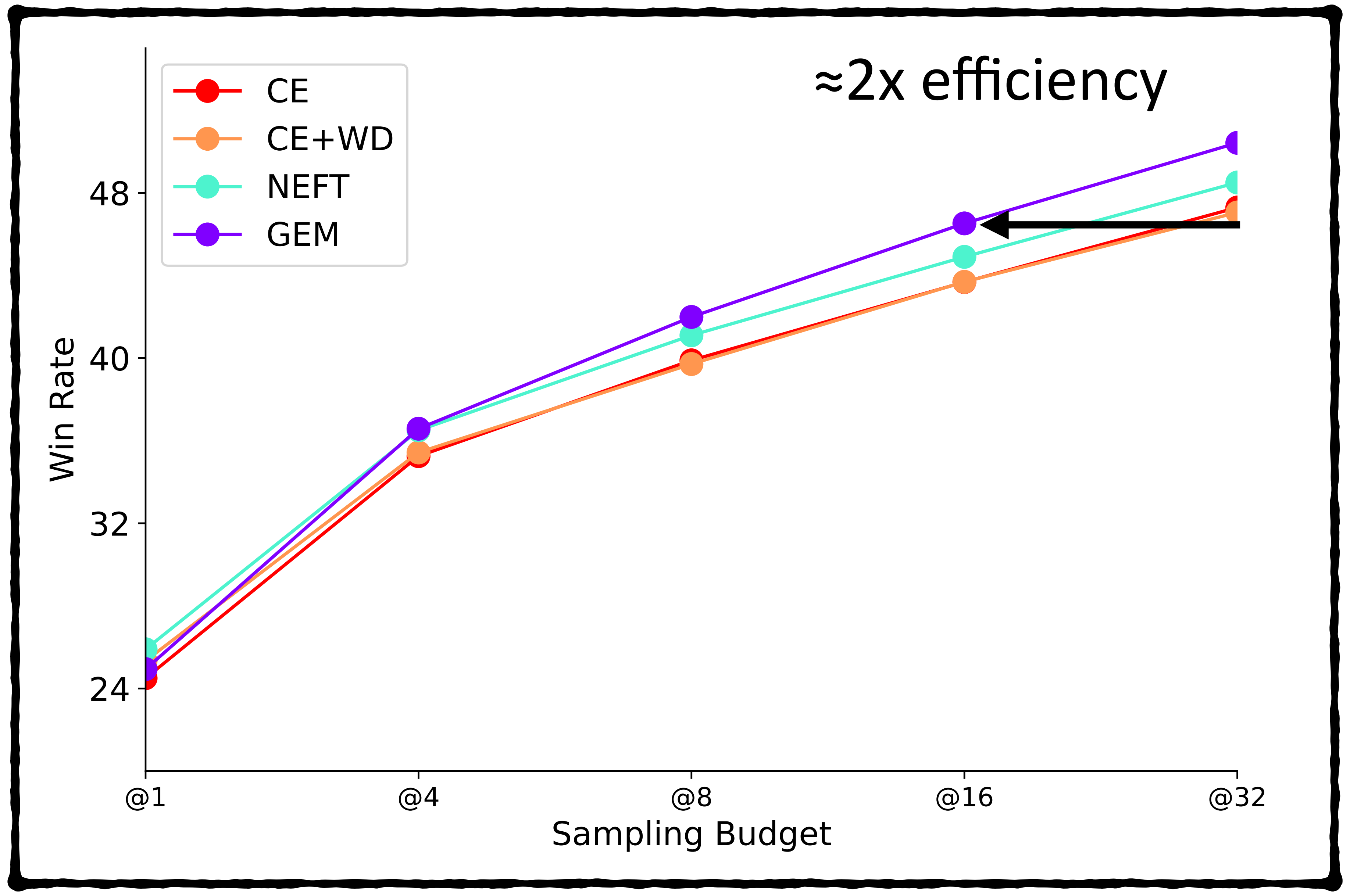
Discussion: Difference with GANs

	GAN (generative adversarial network)	GEM (game-theoretic entropy maximization)
Task	Image Generation	Text Generation
Challange	Estimation the distance among two images is hard	Overfitting the data and losing output diversity
Idea	Introduction of discriminator	Introduction of flow-controller
Computation Complexity	High	Low

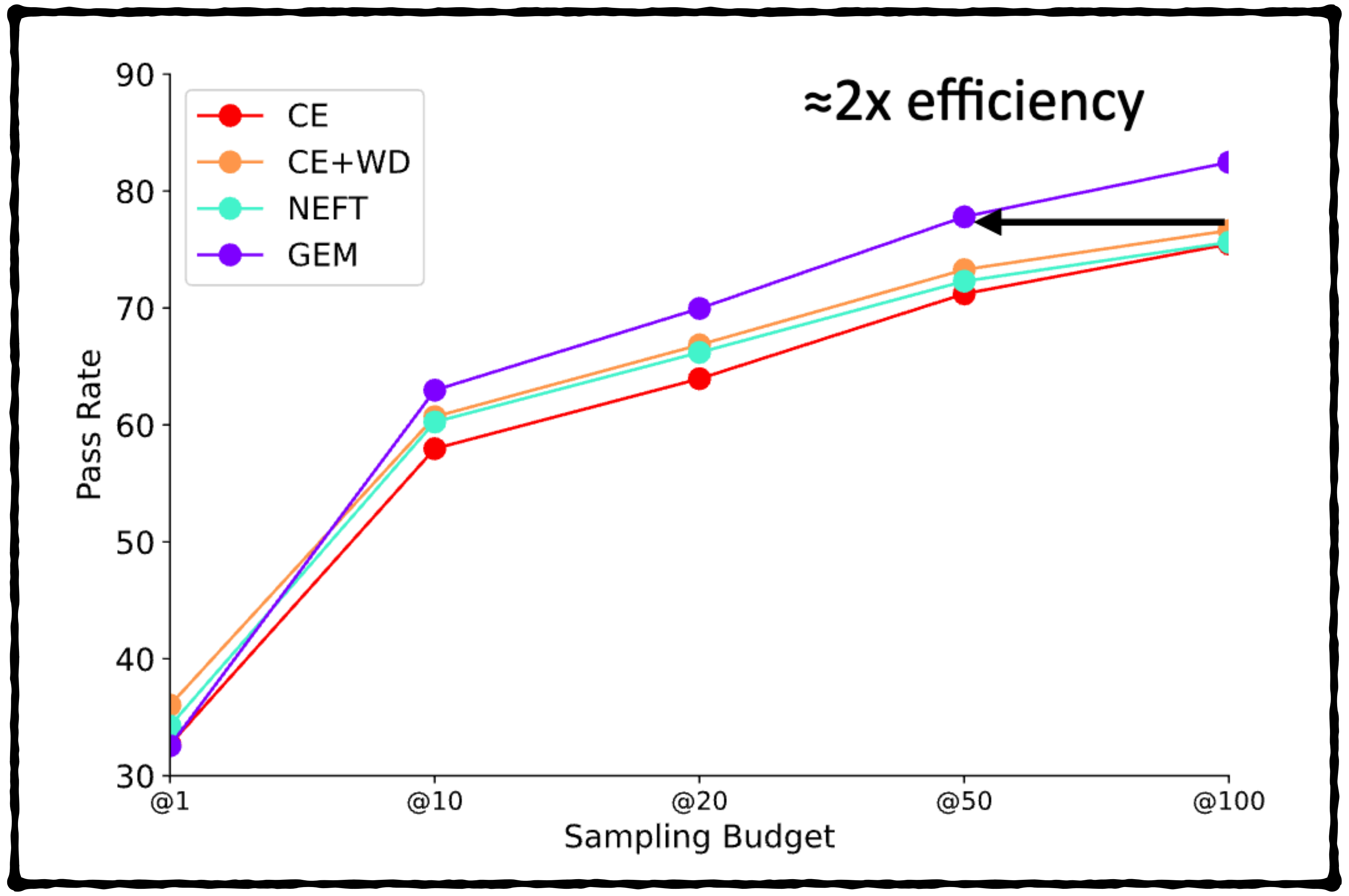
Part IV: Empirical Results

Test-Time Scaling

- Evaluation Method: Best-of-N Sampling
- Model: Llama-3.1-8B; Dataset: Ultrafeedback



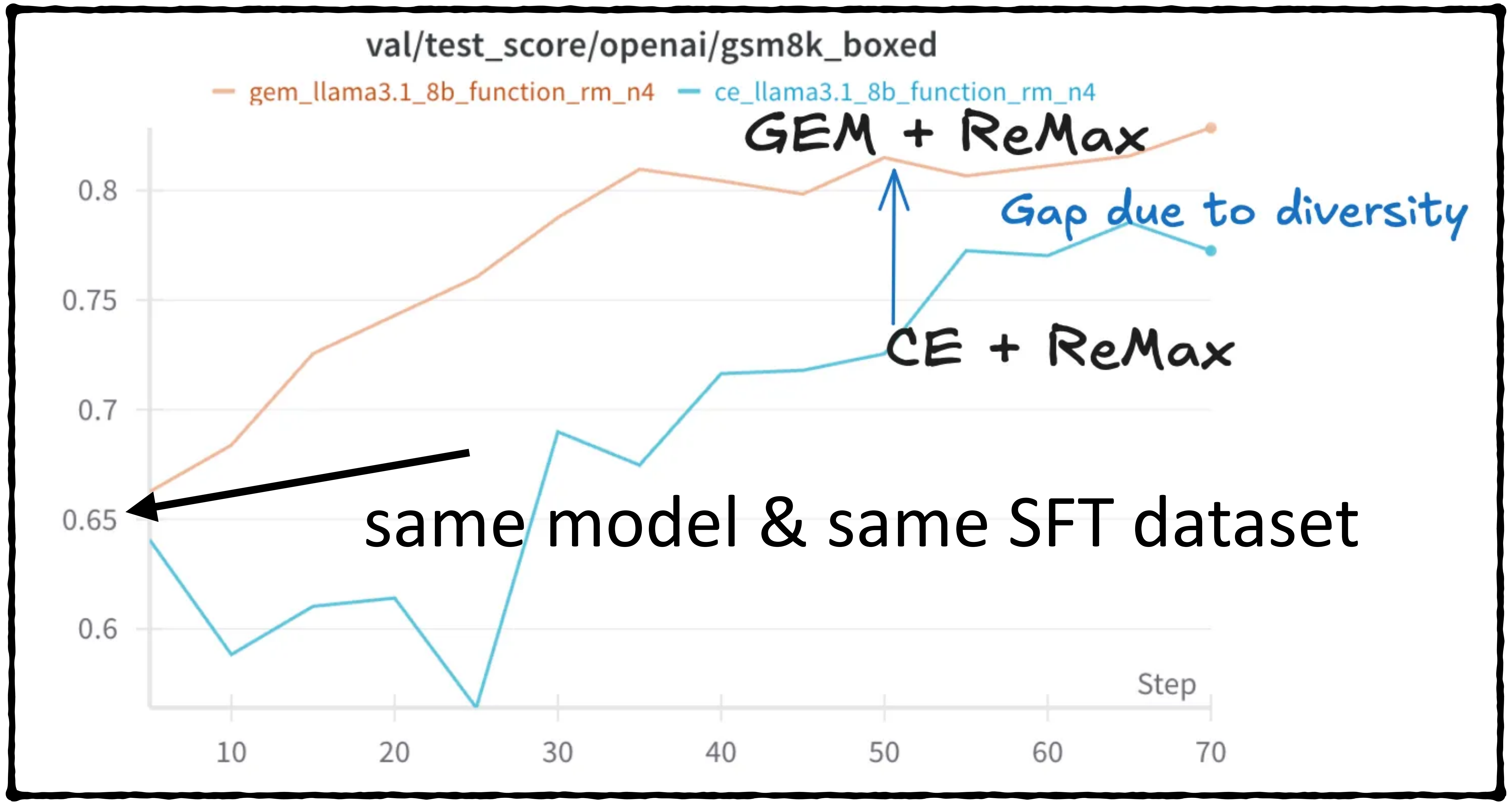
RLHF Alignment (Chat)



Code Generation

GEM requires about **2x** less sampling budget for comparable performance

Math Reasoning



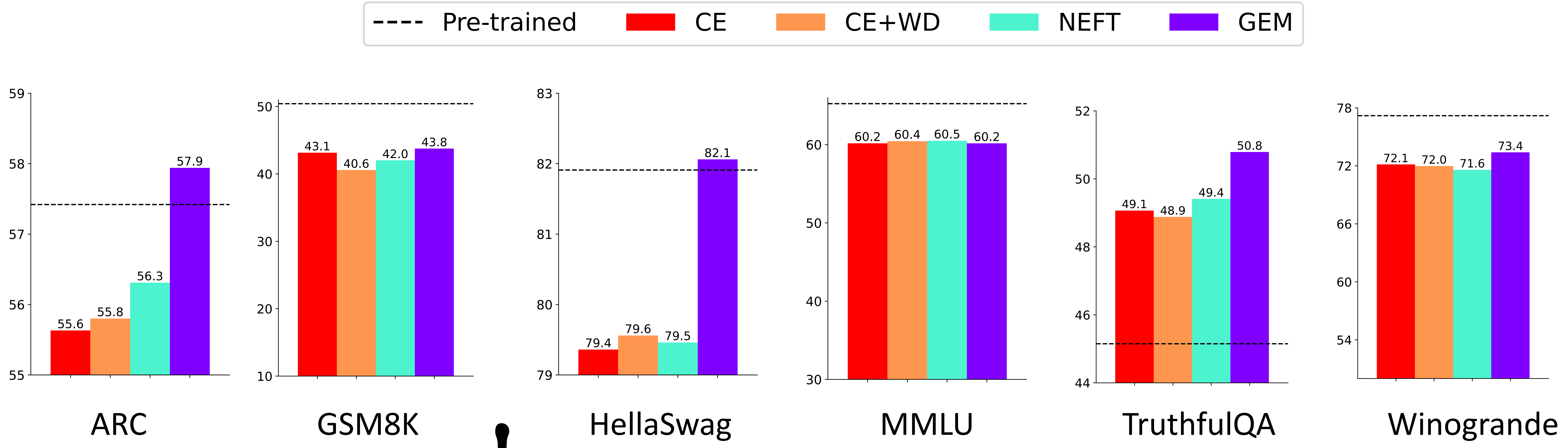
[https://tangible-polo-203.notion.site/]

- Task: optimize CoT (reasoning steps) to answer math questions
- Reward: accuracy of final reward
- Model: Qwen-2.5-3B
- RL Algo: ReMax

[Li, Ziniu, et al. "Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models." ICML 2024.]

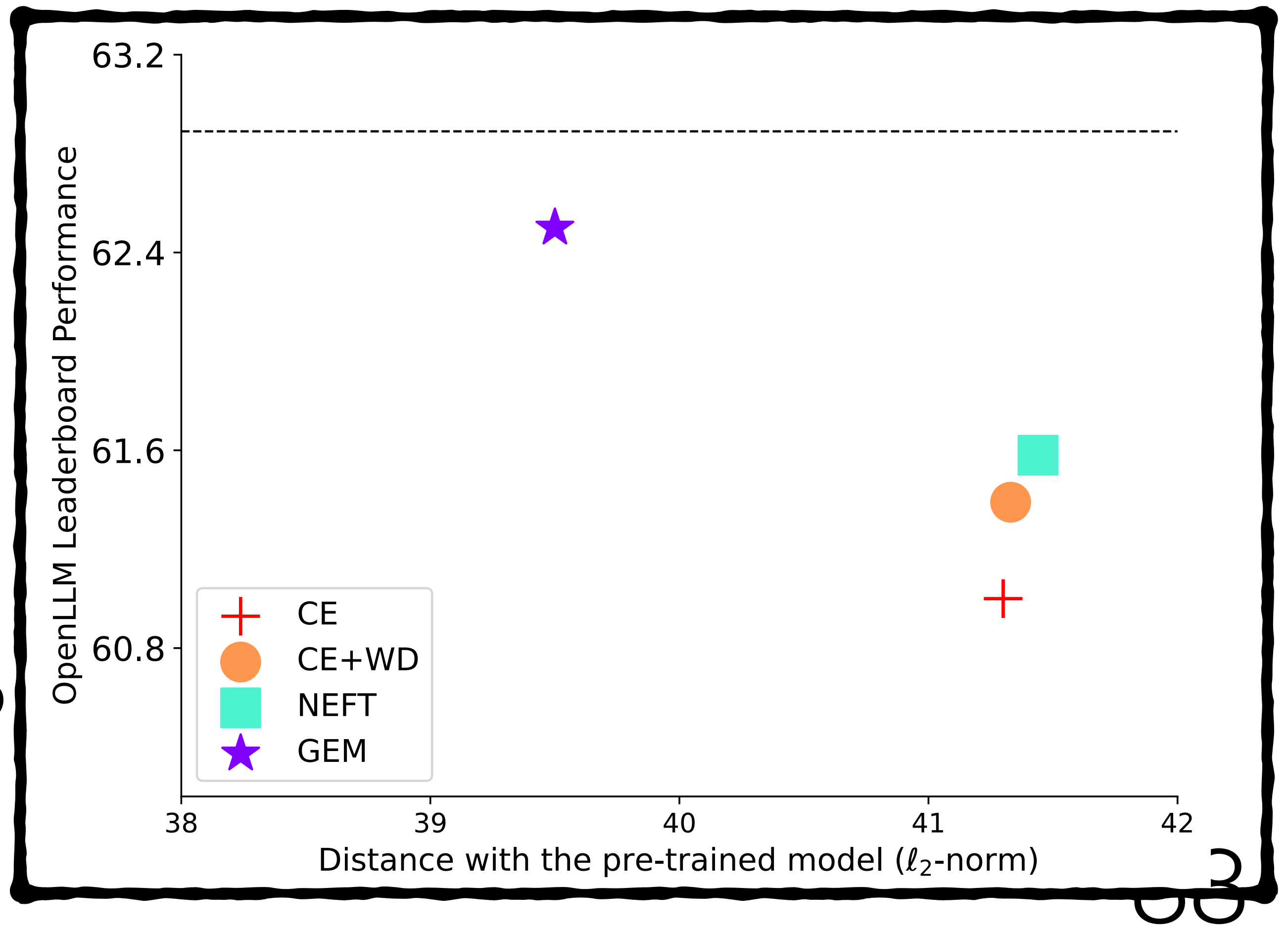
GEM improves the performance limit of RL training

Alignment Tax



GEM fine-tunes the model with 83% less alignment tax

GEM-tuned model shows less overfitting to the data



Thank You!

Paper

Code