CSE599H:
Advances and Challenges in Language Models, Reasoning, and AI Agents

Hanna Hajishirzi

Open Models

Closed API Models

ELMO
Feb 2018
Ai2
"First to pretrain on unlabeled text"

BERT
Oct 2018
Pretrain & fine-tune

GPT-2
Feb 2019
Generative

GPT-3
June 2020
"First to remove fine-tuning"

Chinchilla
March 2022
"Data size is as important as parameter count"

ChatGPT
Nov 2022

GPT-4
March 2023
"Multimodal"

*List not exhaustive

Research & Development in LMs

- Science of LMs
- Improve LMs
- Build Next generation of LMs
- Test-time Inference
- Extend LMs Beyond Text
- LMs for Science
- LMs for Health
- Mitigate LMs Risk and Biases
- Efficient Models
- LM Agents
- Reasoning
- Inference Efficiency

Ai2

This class discusses language models, reasoning, agents, and we do Lots of

- paper reading
- presentations
- discussions

Course Staff

Hanna Hajishirzi

Instructor

hannaneh [at] cs [dot] washington [dot] edu

Jiacheng Liu

TA

liujc [at] cs [dot] washington [dot] edu

Rulin Shao

TA

rulins [at] cs [dot] washington [dot] edu

Course Policies

- Present and lead discussions (35%)
- Participation in discussions (25%)
- Paper review (15%)
- Research proposal (25%)

Present and lead discussions

- Students will form groups of 2-3 to be presenters for a class.
- Each class will have one group of student presenters.
- During the quarter, every student is expected to present once.
- **Please sign up your group and present date on this spreadsheet by 11:59pm on April 4.**
- The presentation dates are first-come-first-serve.
- All students in a group will get the same grade, barring exceptional circumstances.
- Presenters lead how students discuss topics within groups.

Presentation Details: Context

Read a few papers that are prior and/or concurrent work, and a few papers that came later and build upon the ideas in this paper.
Share any background information that you think will help your classmates better understand the assigned readings.
Summarize the broader landscape and situate the assigned papers within these works. For example, your presentation might answer the following questions:

- What problem is the assigned paper tackling? What is the motivation?
- What was the state-of-the-art before this paper? What was the prevailing view about this problem?
- What was the key contribution made by this paper? What was the insight that allowed them to make this contribution?
- How did this paper change how people think about this problem? How has future work built on top of it? Has this paper made a difference yet (or not)? Why or why not?

You should aim to find and read 5-10 related papers. You can use our suggested optional reading as a resource, but you don't have to stick to it. Your papers may include literature surveys, technical reports, blog posts, etc.

Presentation Details: Deep Dive

Pick one key aspect of each assigned paper that you'd be interested to discuss in depth. This could be an experiment, a proof, an argument, etc. Go over it in detail, for example:

- What is the setting?
- How did the authors approach the problem?
- What are the results? Did they support the claim?
- Which details in their experiment, assumptions, argument etc. mattered?
- What did you find interesting about this?
- What might the paper have done differently?

Presenters are welcome to come by office hours the week before their presentation to discuss.

Presentation Deliverables

- Prepare a 20-30 minute long presentation (combined, for both presenters and both assigned papers).
- By 11:59pm two days before class, submit a PDF of your slides on Gradescope.
  - Only one presenter should submit. Indicate the names of both presenters on the slides.
  - Submit this as a PDF with one slide per page and with presenter notes. The notes don't have to be a full script; they can be rough bullet points.
  - One slide should be a bibliography of the papers you read.
- We might give you feedback on the presentation by the day before class. If we don't have any requested changes, this is a good sign!
- You may continue to edit the slides after submitting them. During class, present from your own laptop.

Participate in discussions

- All students are expected to have read the paper before the presentation & ask questions and participate in paper discussions in person.
- Please email the course staff if you are unable to attend class in person.
- Experimental: we set up hybrid meetings when a group of participants attend over zoom.

Paper Review

- Goal: how to review papers for conferences/journals.
- We will provide the list of papers for the paper review.
- The paper review should not be longer than 2 pages.

Research Proposal

Goal: how to write a research proposal for your projects or grants.
Write a 4-page research proposal paper describing a line of research in NLP. The research proposal should be about a new project that would extend a clearly identified past research contribution. The research proposal should:

- Build upon or extend what was done in the past work;
- Address challenges or weaknesses in the past research;
- Propose logical extensions or next steps to the focus research; and
- Describe a possible evaluation methodology, experimental design, and required evaluation resources.

What you will read and discuss?

- How to build modern language models?
  - Pretraining
  - Posttraining
  - Data
- What is reasoning for large language models? How to improve them?
  - Reinforcement learning
  - Chain of thought
- What are agents? How do we build agents or equip language models with tool use?
  - Agents for coding
  - Complex agents
- What are alternative architectures for LMs?
  - Mixture of experts
  - State space models
- What are features and limitations of language models?
- How to build multimodal language models?
- What are efficiency considerations for large language models?

Late submission and accomodations

You have 3 late days in total for submitting the paper review and research proposal.
Any late submission will be penalized at a penalty of 10% of the maximum grade per day.
If you have DRS accommodations that the course staff should know about, please contact us at the beginning of the course.

Class Schedule

Mar 31 (Mon)	**Course overview**
Apr 2 (Wed)	**Basic Pre-training and Post-training** • The Llama 3 Herd of Models • 2 OLMo 2 Furious • DeepSeek-V3 Technical Report Optional reading • The Ultra-Scale Playbook: Training LLMs on GPU Clusters
Apr 7 (Mon)	**Guest Lecture (TBD)**
Apr 9 (Mon)	**Guest Lecture (TBD)**
Apr 14 (Mon)	**Scaling Laws of Language Models** • Training Compute-Optimal Large Language Models • Language models scale reliably with over-training and on downstream tasks Optional reading • Scaling Laws for Neural Language Models • A Hitchhiker's Guide to Scaling Law Estimation

Apr 16 (Wed)	**Building Reasoning Models & Systems I** • Chain of Thought Prompting Elicits Reasoning in Large Language Models • Self-Consistency Improves Chain of Thought Reasoning in Language Models • STAR: Bootstrapping Reasoning With Reasoning
Apr 21 (Mon)	**Building Reasoning Models & Systems II** • OpenAI o3-mini System Card • Tülu 3: Pushing Frontiers in Open Language Model Post-Training (Section 6 only) • DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning • SFT Memorizes, RL Generalizes: A Comparative Study of Foundation Model Post-training
Apr 23 (Wed)	**Test-Time Scaling** • Stream of Search (SoS): Learning to Search in Language • s1: Simple test-time scaling
Apr 28 (Mon)	**AI Agents and Tool Use** • ReAct: Synergizing Reasoning and Acting in Language Models • Toolformer: Language Models Can Teach Themselves to Use Tools • START: Self-taught Reasoner with Tools

Apr 30 (Wed)	**AI Agents for Coding** • SWE-agent: Agent-Computer Interfaces Enable Automated Software Engineering • OpenHands: An Open Platform for AI Software Developers as Generalist Agents Optional reading • LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code • SWE-Lancer: Can Frontier LLMs Earn $1 Million from Real-World Freelance Software Engineering?
May 5 (Mon)	**AI Agents for Computer Use and Web Browsing** • OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments • WebArena: A Realistic Web Environment for Building Autonomous Agents • WebVoyager: Building an End-to-End Web Agent with Large Multimodal Models Optional reading • [OpenAI] Computer-Using Agent • [Anthropic] Developing a computer use model

May 7 (Wed)	**AI Agents for Deep Research** • Deep Research System Card • OpenScholar: Synthesizing Scientific Literature with Retrieval-augmented LMs Optional reading • Search-R1: Training LLMs to Reason and Leverage Search Engines • ReSearch: Learning to Reason with Search for LLMs via Reinforcement Learning • RAG-RL: Advancing Retrieval-Augmented Generation via RL and Curriculum Learning
May 12 (Mon)	**Features and Limitations I** • AI models collapse when trained on recursively generated data • The Reversal Curse: LLMs trained on "A is B" fail to learn "B is A" • The Generative AI Paradox: "What It Can Create, It May Not Understand" Optional reading • Faith and Fate: Limits of Transformers on Compositionality
May 14 (Wed)	**Features and Limitations II** • Alignment faking in large language models • Grokking: Generalization Beyond Overfitting on Small Algorithmic Datasets • The Hyperfitting Phenomenon: Sharpening and Stabilizing LLMs for Open-Ended Text Generation Optional reading • Are Emergent Abilities of Large Language Models a Mirage?

May 19 (Mon)	**Alternative Architectures** • OLMoE: Open Mixture-of-Experts Language Models • Mamba: Linear-Time Sequence Modeling with Selective State Spaces • Learning to (Learn at Test Time): RNNs with Expressive Hidden States <u>Optional reading</u> • Large Concept Models: Language Modeling in a Sentence Representation Space
May 21 (Wed)	**Efficiency and Scaling** • FlashAttention: Fast and Memory-Efficient Exact Attention • vLLM: Easy, Fast, and Cheap LLM Serving • QLoRA: Efficient Finetuning of Quantized LLMs
May 26 (Mon)	**Pretraining Data I** • Dolma: Open Corpus of Three Trillion Tokens • DataComp-LM: Next Generation Training Sets <u>Optional reading</u> • RedPajama: Reproducing LLaMA Training Dataset • The FineWeb Datasets: Decanting the Web for the Finest Text Data at Scale

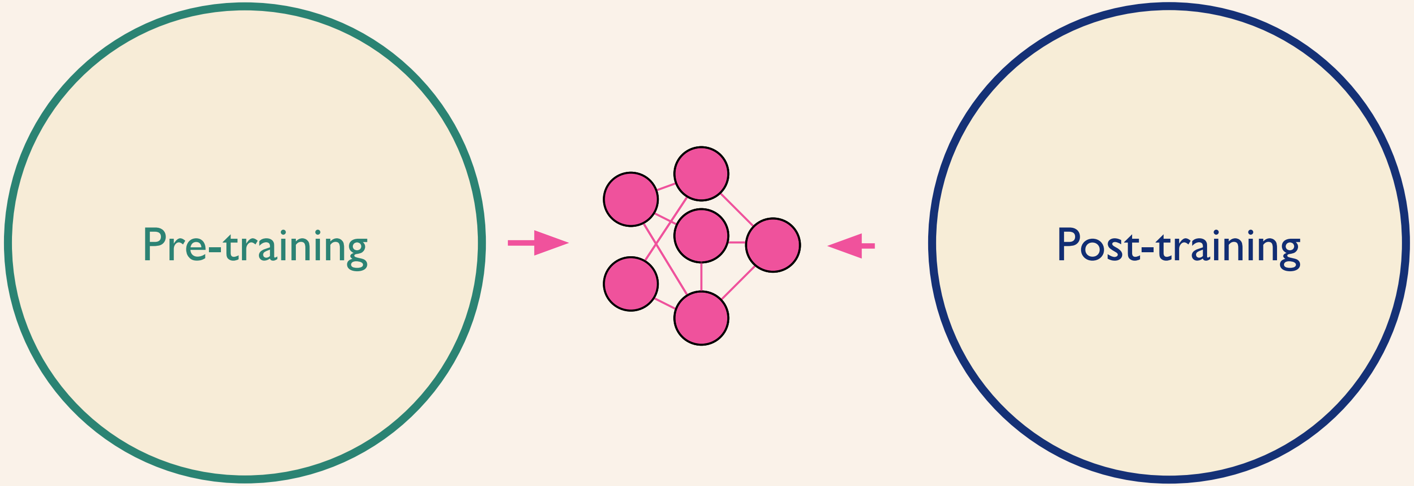
Ai2

Introductions

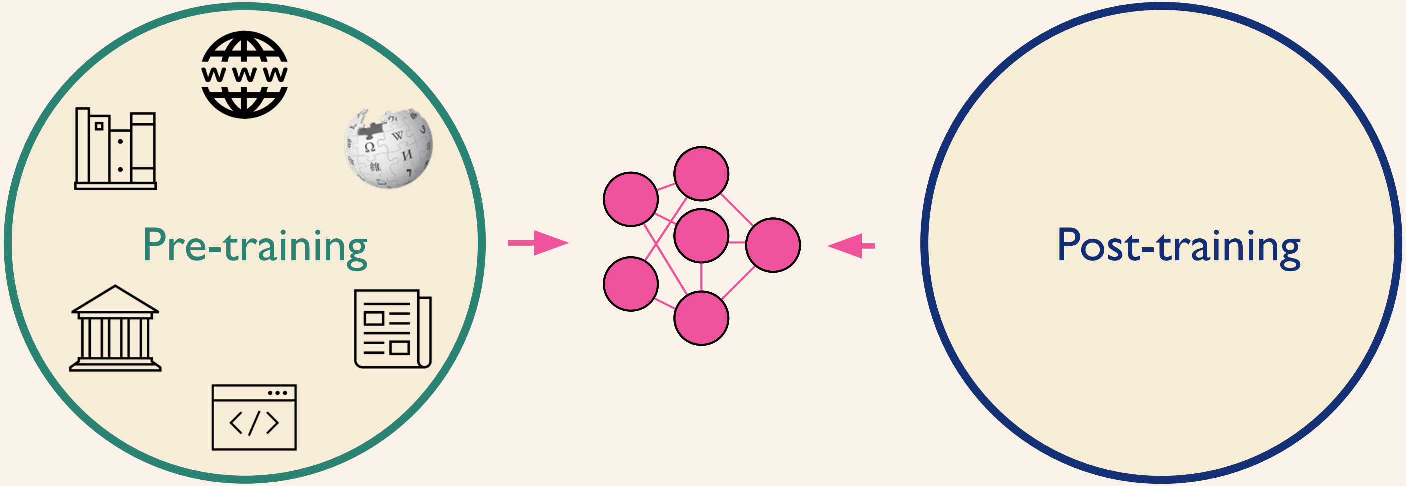
Today's lecture

Open Training Recipes for Reasoning in Language Models

Building a modern LLM

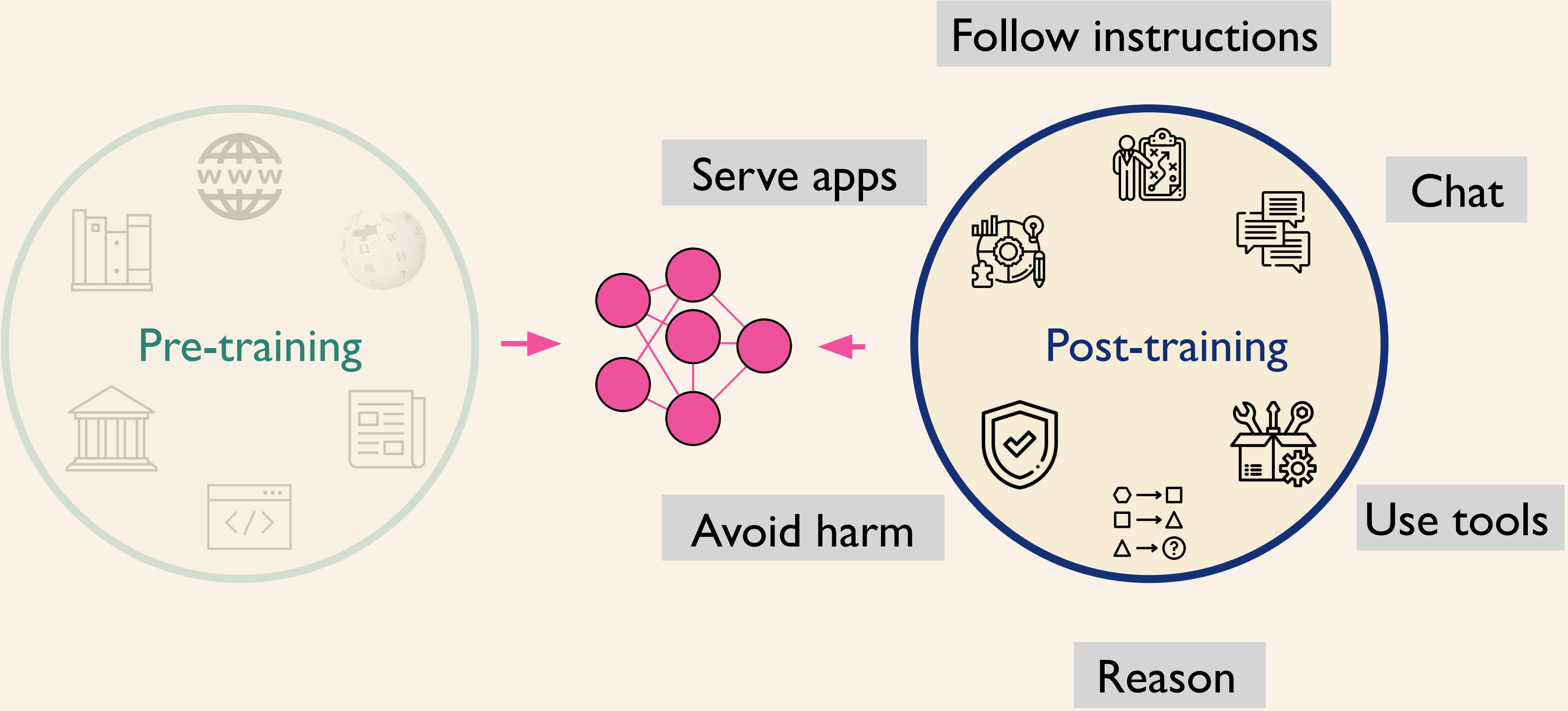


Building a modern LLM



Predict the next word in various contexts

Post-training leads to efficiency through specialization



Ai2

Open Ecosystem to
Accelerate Innovation in Language Models

OLMo

Tulu

Pre training

Post Training

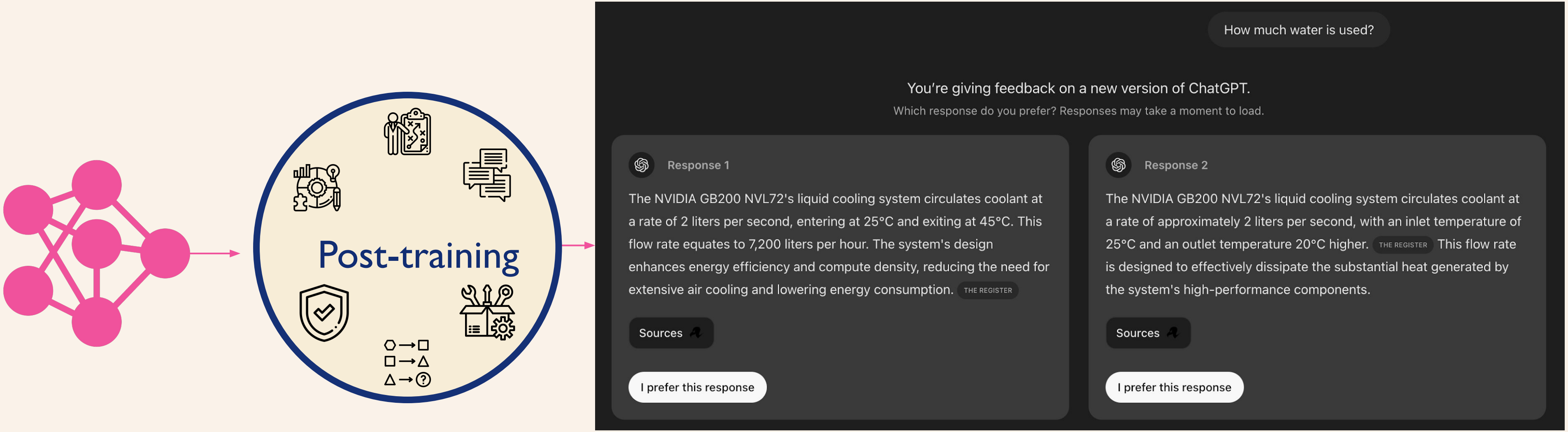
Test-time
Inference

Tulu

Many slides from:

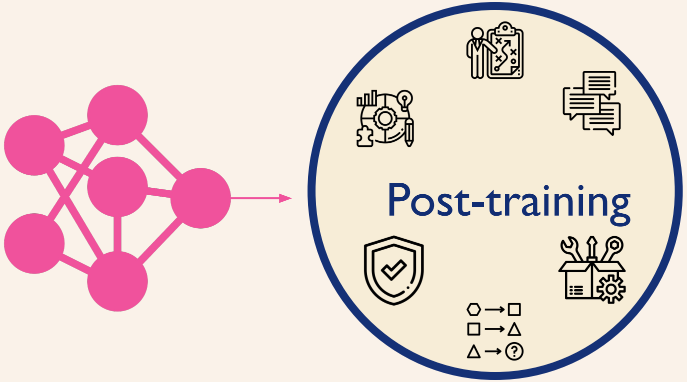
Yizhong Wang, Nathan Lambert, Hamish Ivison, Faeze Brahman

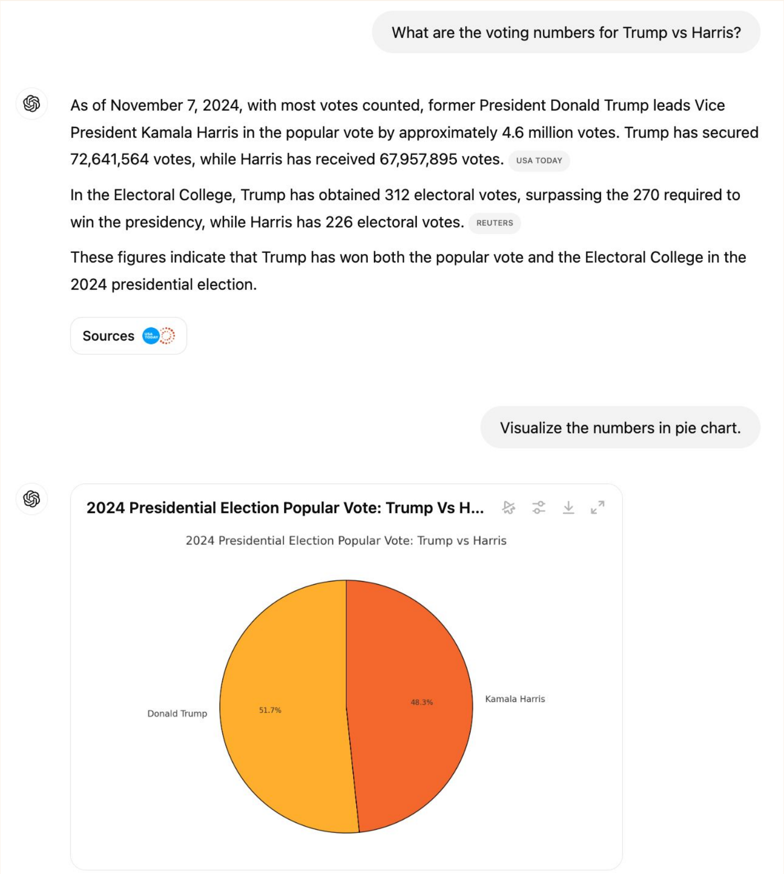
Post Training for Alignment with Human Preferences



[Photo by @FanaHOVA on X]

Post Training for Tool Use /Agents

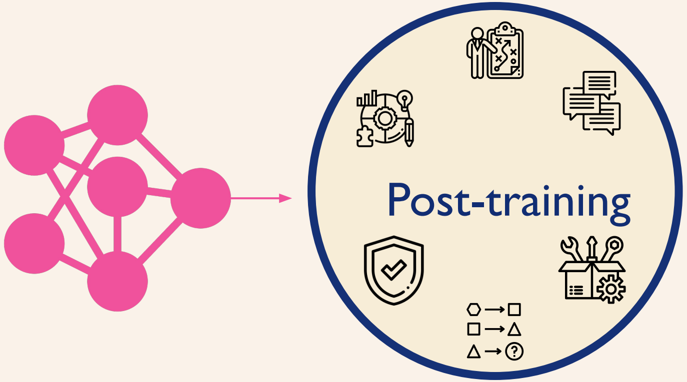




Searching

Code Execution

Post Training for Reasoning



Find the smallest positive integer that can be expressed as the sum of two squares in two different ways.

To solve this problem, we need to find the smallest positive integer (n) that can be expressed as the sum of two squares in two different ways. This means we need (n) such that there exist distinct pairs ((a, b)) and ((c, d)) where:

[n = a^2 + b^2 = c^2 + d^2]

We will use Python to iterate through possible values of (n) and check if (n) can be expressed as the sum of two squares in two distinct ways.

Let's write a Python function to solve this problem step-by-step.

```
def find_smallest_sum_of_two_squares():
    def is_sum_of_two_squares(n):
        ways = set()
        for a in range(int(n ** 0.5) + 1):
            b_squared = n - a * a
            if b_squared >= 0:
                b = int(b_squared ** 0.5)
                if b * b == b_squared:
                    ways.add((a, b))
                    ways.add((b, a))
        return ways

    n = 1
    while True:
        ways = is_sum_of_two_squares(n)
        if len(ways) >= 2:
            return n, ways
        n += 1

smallest_n, ways = find_smallest_sum_of_two_squares()
print((smallest_n, ways))
```

```
(50, {(5, 5), (1, 7), (7, 1)})
```

The smallest positive integer that can be expressed as the sum of two squares in two different ways is (50).

The ways to express 50 as the sum of two squares are: [50 = 1^2 + 7^2] [50 = 5^2 + 5^2]

Thus, the answer is (\boxed{50}).

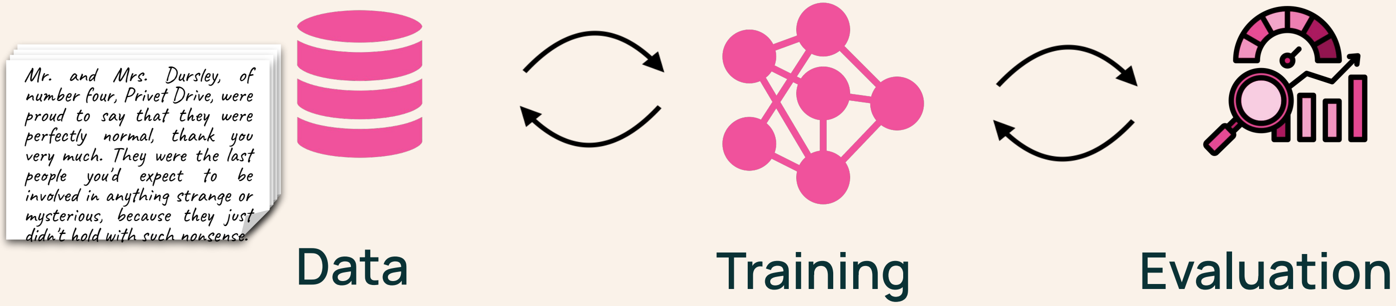
Ai2

Building a modern LLM

Data

Models

Algorithms



Building a modern LLM

Data

- comes from different sources
- in different forms
- targets for different capabilities

Ai2

How to use the right data in the right way?

Data

- comes from different sources
- in different forms
- targets for different capabilities

Ai2

Tulu

Tülu

Open, reproducible, & state-of-the-art post-training recipe

[Wang*, Ivison* et al., 2023]

[Ivison*, Wang* et al., 2023]

[Ivison, Wang et al., 2024]

[Lambert, …, Wang, Dasigi, Hajishirzi, 2024]

Ai2

Tulu: Open Instruction Tuning Recipe

How Far Can Camels Go? Exploring the State of Instruction Tuning on Open Resources

Yizhong Wang*♣♠ Hamish Ivison*♣ Pradeep Dasigi♣
Tushar Khot♣ Khyathi Raghavi Chandu♣ David Wadden♣ Ke
Noah A. Smith♣♠ Iz Beltagy♣ Hannaneh Hajish

♣Allen Institute for AI ♠University of Washington
{yizhongw,hamishi}@allenai.org

Best recipe for instruction data
Jun 2023

Camels in a Changing Climate: Enhancing LM Adaptation with TÜLU 2

Hamish Ivison*♠ Yizhong Wang*♣♠ Valentina Pyatkin♣♠
Matthew Peters♣ Pradeep Dasigi♣ Joel Jang ♣♠ Davi
Noah A. Smith♣♠ Iz Beltagy♣ Hannaneh Hajis

♣Allen Institute for AI ♠University of Washingto
{yizhongw,hamishiv}@cs.washington.edu

Best open model with preference data
Nov 2023

Unpacking DPO and PPO: Disentangling Best Practices for Learning from Preference Feedback

Hamish Ivison♣♠ Yizhong Wang♣♠ Jiacheng Liu♣♠
Zeqiu Wu♠ Valentina Pyatkin♣♠ Nathan Lambert♣
Noah A. Smith♣♠ Yejin Choi♣♠ Hannaneh Hajishirzi♣♠

♣Allen Institute for AI ♠University of Washington
hamishiv@cs.washington.edu

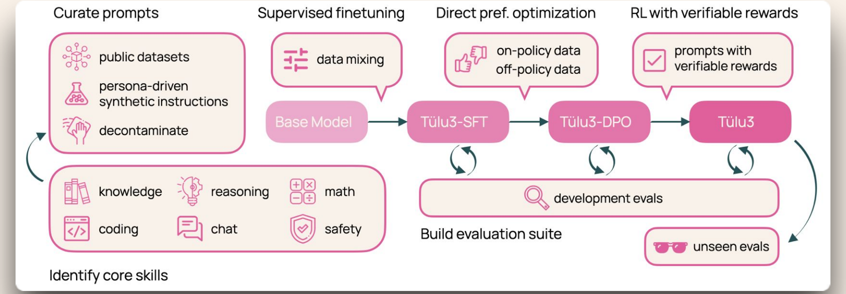
Systematic study of DPO vs PPO
June 2024

Open models & data

Tülu 1→2→2.5→3

Tülu 1 [Wang et al., NeurIPS 2023]

Open post-training recipe



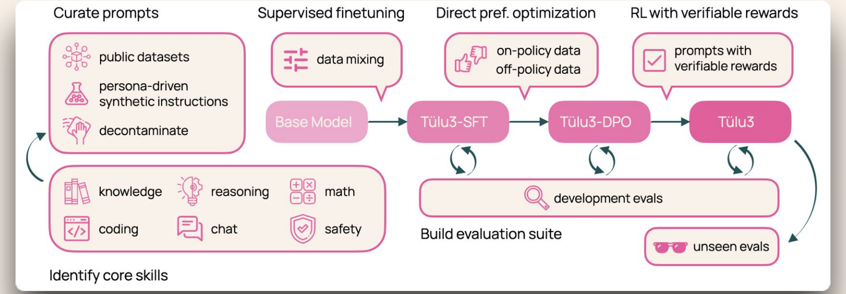
Tülu 3 [Lambert et al., Arxiv 2024]

Open models & data

Tülu 1→2→2.5→3

Tülu 1 [Wang et al., NeurIPS 2023]

Open post-training recipe



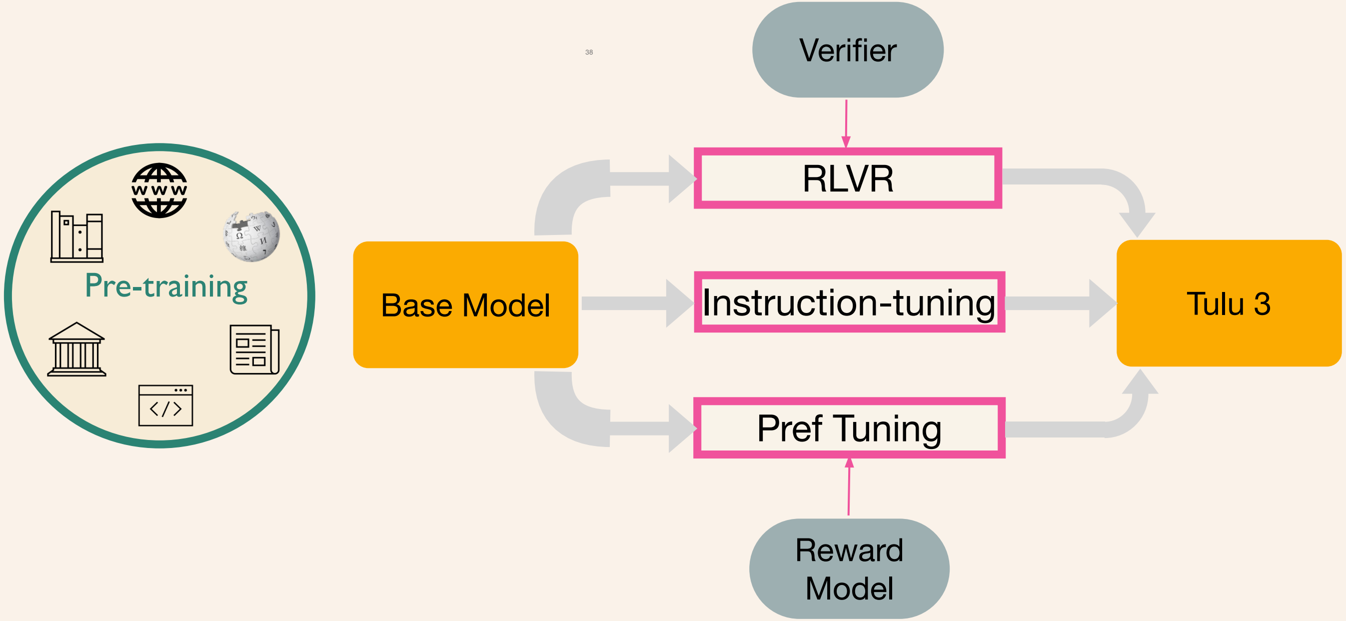
Tülu 3 [Lambert et al., Arxiv 2024]

Fully-open LM

OLMo [Groeneveld et al., ACL 2024]

Ai2

Tülu 3 Training Recipe



Getting Ingredients to Start With

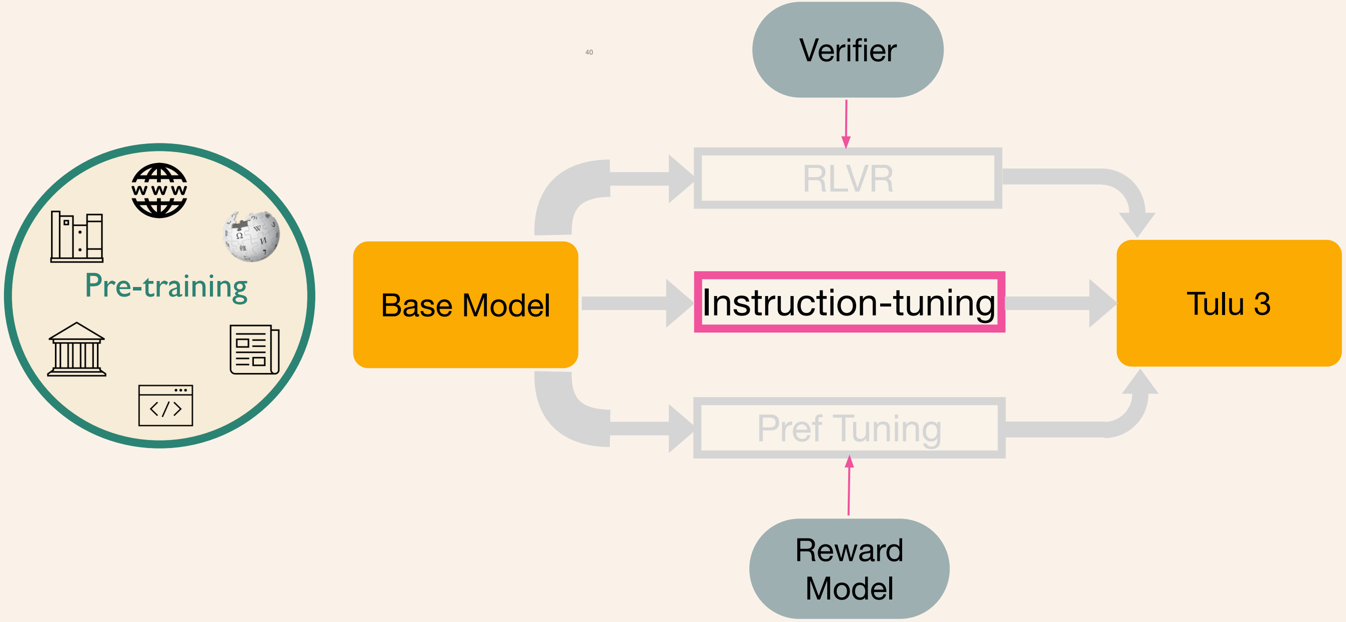
Successful adaptation starts with:

1. Meaningful **evaluations** for targeted skills
2. **Prompts** of representative queries for said skills
3. Check for Licenses
4. Decontamination

Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO
General	**Tülu 3 Hardcoded**[↑]	24	240	–
	OpenAssistant[1,2,↓]	88,838	7,132	7,132
	No Robots	9,500	9,500	9,500
	WildChat (GPT-4 subset)[↓]	241,307	100,000	100,000
	UltraFeedback[α,2]	41,635	–	41,635
Knowledge Recall	FLAN v2[1,2,↓]	89,982	89,982	12,141
	SciRIFF[↓]	35,357	10,000	17,590
	TableGPT[↓]	13,222	5,000	6,049
Math Reasoning	**Tülu 3 Persona MATH**	149,960	149,960	–
	Tülu 3 Persona GSM	49,980	49,980	–
	Tülu 3 Persona Algebra	20,000	20,000	–
	OpenMathInstruct 2[↓]	21,972,791	50,000	26,356
	NuminaMath-TIR[α]	64,312	64,312	8,677
Coding	**Tülu 3 Persona Python**	34,999	34,999	–
	Evol CodeAlpaca[α]	107,276	107,276	14,200
Safety & Non-Compliance	**Tülu 3 CoCoNot**	10,983	10,983	10,983
	Tülu 3 WildJailbreak[α,↓]	50,000	50,000	26,356
	Tülu 3 WildGuardMix[α,↓]	50,000	50,000	26,356
Multilingual	Aya[↓]	202,285	100,000	32,210
Precise IF	**Tülu 3 Persona IF**	29,980	29,980	19,890
	Tülu 3 IF-augmented	65,530	–	65,530
Total		23,327,961	939,344	425,145

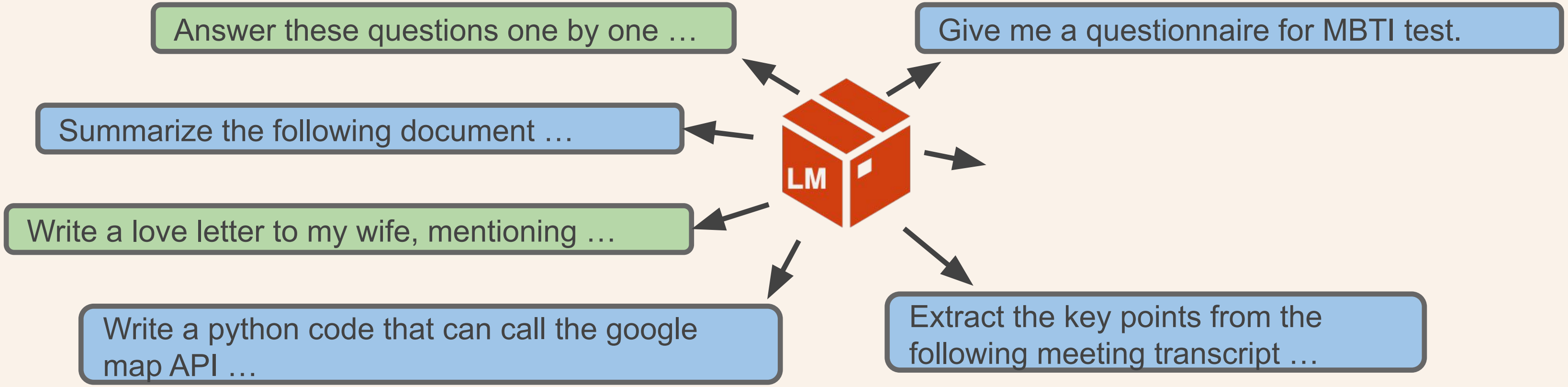
Tülu 3

Supervised Finetuning (a.k.a Instruction Tuning)



Supervised Finetuning

- SFT (or Instruction tuning): Finetuning pretrained LMs with prompts and completions



Data Curation

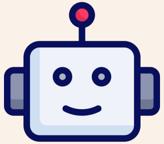
How to
source data?

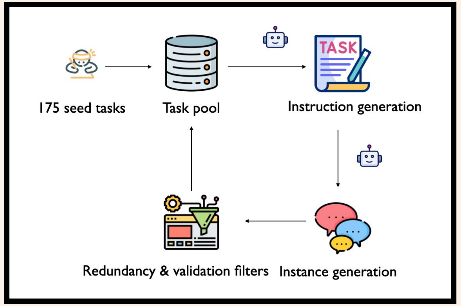
- Costly
- Time-consuming
- High variance

Data Curation

- Costly
- Time-consuming
- High variance

How to
source data?



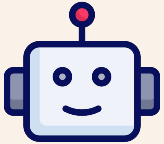


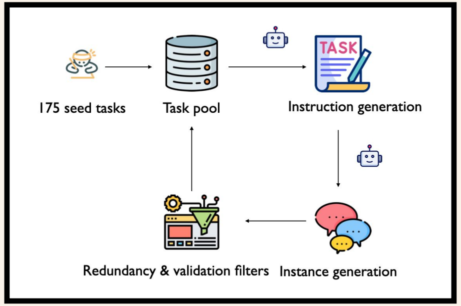
Self-Instruct [Wang et al., ACL 2023]

Synthetic data

How to source data?

- Costly
- Time-consuming
- High variance



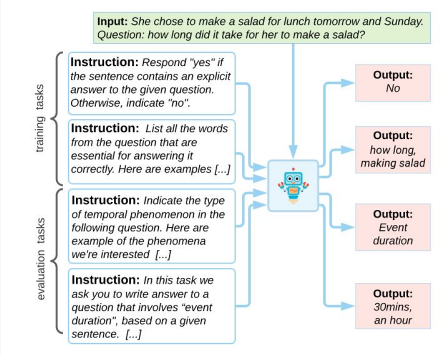


Self-Instruct [Wang et al., ACL 2023]

Combining ?

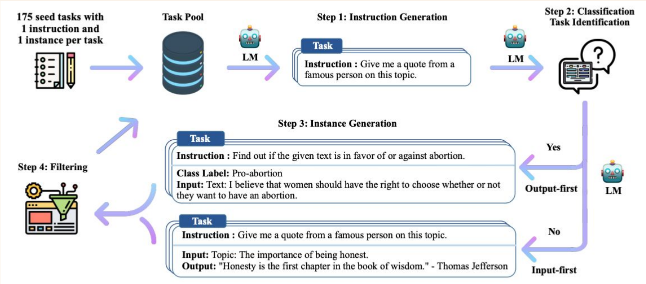
Hybrid Preferences [Miranda*, Wang* et al., 2023]

Data Curation

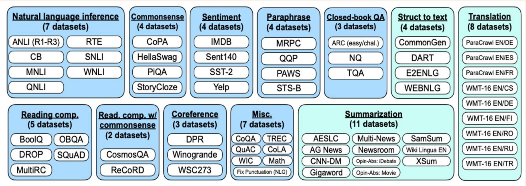


NaturalInstructions,
[Mishra et al 2022]

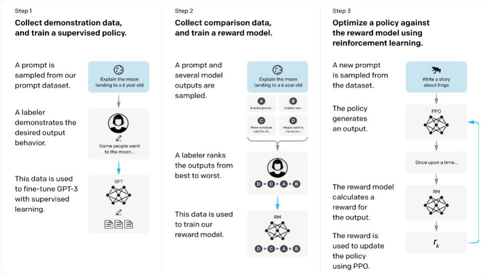
Super-NaturalInstructions,
[Wang et al. 2022]



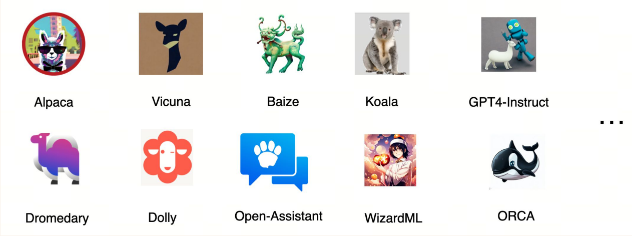
Self-Instruct,
[Wang et al. 2023]



FLAN_v1,
[Wei et al 2022]



InstructGPT,
[Wei et al 2022]



Lots of instruction datasets …

Supervised Finetuning: The role of data

Two repeated and parallelizable tracks:

1. **Data curation**: Curate data given targeted capabilities

2. **Data mixing**: Mix data across capabilities

   a. Substantial effort in filtering data while maintaining performance.
   b. Start fully with mixing before curation.

Tülu 1: instruction tuning data mixing

SuperNI
CoT
Flan V2
Dolly
Open Assistant 1
Self-instruct
Unnatural Instructions
Alpaca
Code-Alpaca
GPT4-Alpaca
Baize
ShareGPT

→ created by human

→ synthesized with GPT-3/4

Tülu 1: instruction tuning data mixing

	Chat (vibe)
SuperNI	4.2
CoT	6.0
Flan V2	3.2
Dolly	13.7
Open Assistant 1	58.1
Self-instruct	5.0
Unnatural Instructions	8.4
Alpaca	21.9
Code-Alpaca	15.8
GPT4-Alpaca	63.1
Baize	21.9
ShareGPT	70.5

Tülu 1: instruction tuning data mixing

	Chat (vibe)	Knowledge	Reasoning	Coding	Multilinguality	Safety
SuperNI	4.2	49.7	4.3	12.9	50.2	22.7
CoT	6.0	44.2	41.0	23.7	47.8	56.1
Flan V2	3.2	50.6	30.4	16.8	47.2	38.6
Dolly	13.7	45.6	23.2	31.0	46.5	21.1
Open Assistant 1	58.1	43.3	27.3	31.9	33.4	94.8
Self-instruct	5.0	30.4	20.9	12.5	41.3	10.7
Unnatural Instructions	8.4	46.4	20.9	23.9	40.9	44.3
Alpaca	21.9	45.0	23.1	29.9	31.1	41.9
Code-Alpaca	15.8	42.5	24.6	34.2	38.9	8.0
GPT4-Alpaca	63.1	46.9	27.7	36.6	23.5	98.8
Baize	21.9	43.7	24.4	28.7	33.6	58.8
ShareGPT	70.5	49.3	33.7	34.1	30.5	97.5

Tülu 1: instruction tuning data mixing

	Chat	Knowledge	Reasoning	Coding	Multiling	Safety	Average
SuperNI	4.2	49.7	4.3	12.9	50.2	22.7	21.2
CoT	6.0	44.2	41.0	23.7	47.8	56.1	37.1
Flan V2	3.2	50.6	30.4	16.8	47.2	38.6	31.0
Dolly	13.7	45.6	23.2	31.0	47.2	21.1	29.2
Open Assistant 1	58.1	43.3	27.3	31.9	33.4	94.8	45.2
Self-instruct	5.0	30.4	20.9	12.5	41.3	10.7	20.2
Unnatural Instructions	8.4	46.4	20.9	23.9	40.9	44.3	29.4
Alpaca	21.9	45.0	23.1	29.9	31.1	41.9	30.8
Code-Alpaca	15.8	42.5	24.6	34.2	38.9	8.0	26.9
GPT4-Alpaca	63.1	46.9	27.7	36.6	23.5	98.8	46.3
Baize	21.9	43.7	24.4	28.7	33.6	58.8	33.6
ShareGPT	70.5	49.3	33.7	34.1	30.5	97.5	49.9
Human + Synthetic	56.5	49.3	41.9	35.9	45.6	99.9	53.0

Ai2

What type of data is useful for reasoning?

- Response to reasoning queries are more than simple final answer

Question: A store has a "Buy 2 Get 1 Free" deal on shirts. Each shirt costs $25. If Sarah wants to buy 7 shirts, how much will she spend?

Chain-of-Thought Output:

Let me break this down:
1. With "Buy 2 Get 1 Free", every 3 shirts will cost the price of 2
2. 7 shirts can be grouped as: 2 complete sets (6 shirts) + 1 remaining shirt
3. For the complete sets: * 6 shirts = 2 × (2 paid + 1 free) * Cost = 2 × (2 × $25) = $100
4. For the remaining 1 shirt: * Cost = 1 × $25 = $25
5. Total cost = $100 + $25 = $125

Therefore, Sarah will spend $125

Requires step-by-step thought process (aka CoT)

Why Chain-of-Thought data for reasoning?

Chain of Thought data

👍🏼 helps models handle complex, multi-step problems easier

👍🏼 reveals the model's reasoning process

👍🏼 makes it easier to spot errors in logic thus more trustworthy

👍🏼 resembles human thought process

But ...

👎🏼 Manual annotation challenges:

- time and cost intensive
- often requires expert annotations
- Difficult to scale

Why Chain-of-Thought data for reasoning?

CoT …

👍 helps models handle complex, multi-step problems easier

👍 reveals the model's reasoning process

👍 makes it easier to spot errors in logic thus more trustworthy

👍 resembles human thought process

But …

👎 Manual annotation challenges:

- time and cost intensive
- often requires expert annotations
- Difficult to scale

Expensive
Time Consuming
Not diverse enough

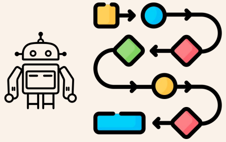
Our Approach: Hybrid Data Creation

Data mixing &
selection
from existing
resources

Our approach: Hybrid Data Creation

Data mixing & selection from existing resources

+



Persona-driven Data Synthesis

- Enable targeting specific skills (e.g., math, code, precise instruction following)
- Ensure high diversity
- Enable Scaling

Scaling Synthetic Data Creation with 1,000,000,000 Personas

Tao Ge,* Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, Dong Yu

Persona-driven Data generation for Scalability and Improved Diversity

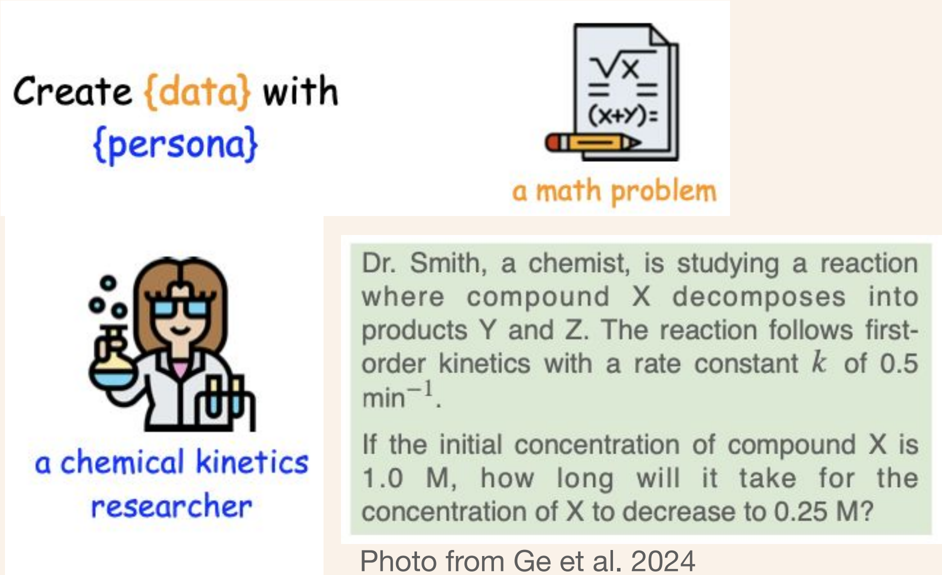


Photo from Ge et al. 2024

Persona-driven Data generation for Scalability and Improved Diversity

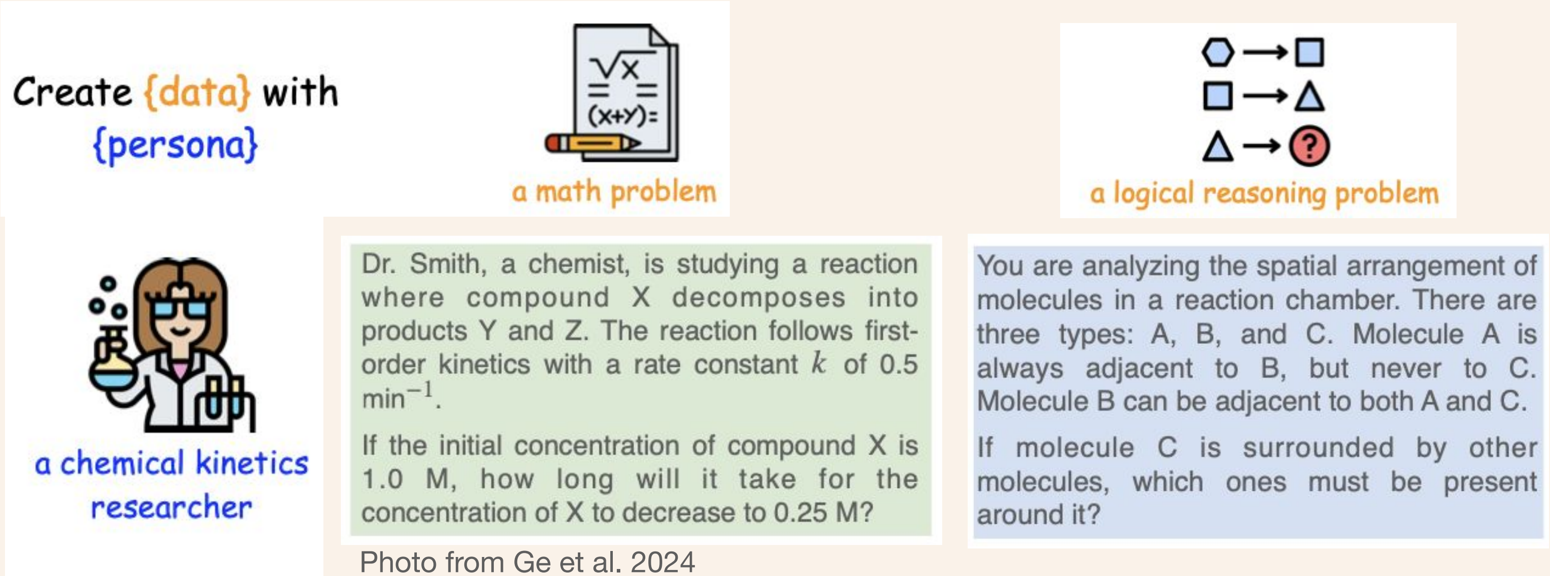
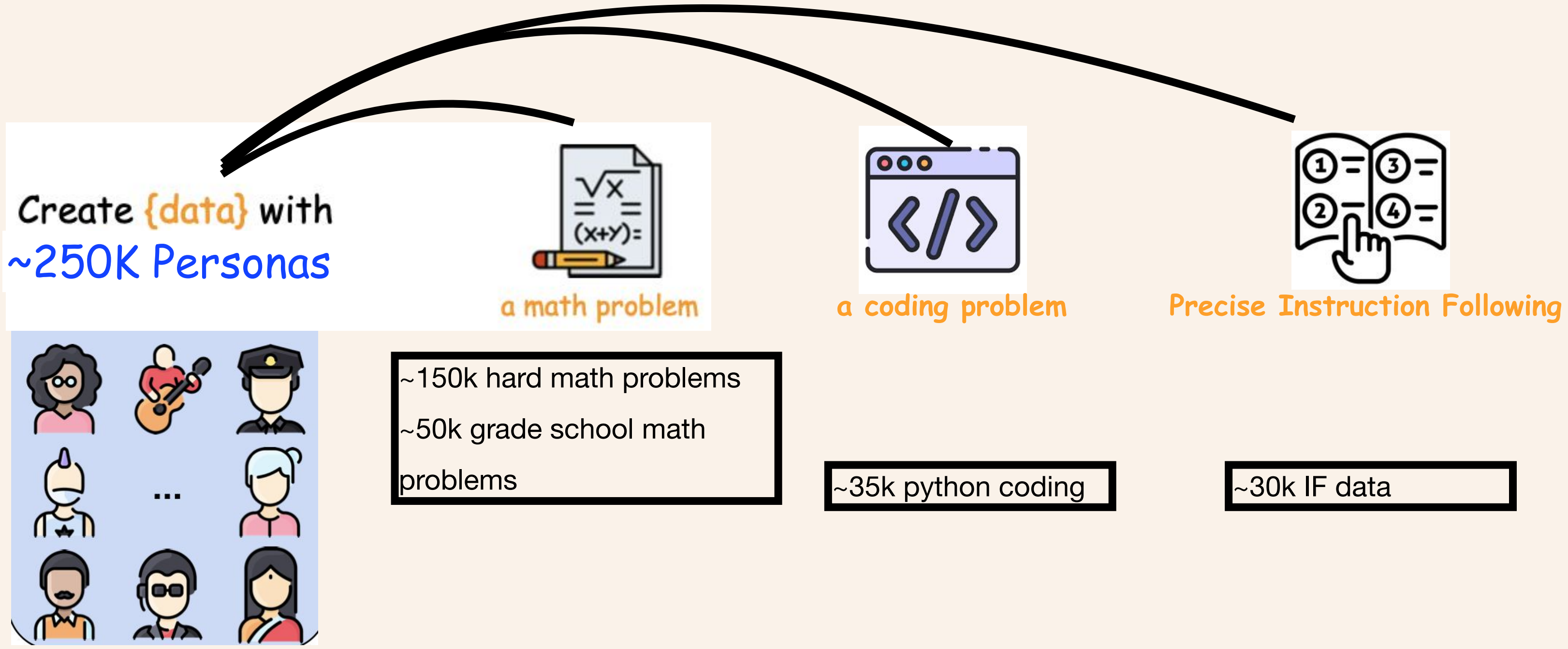
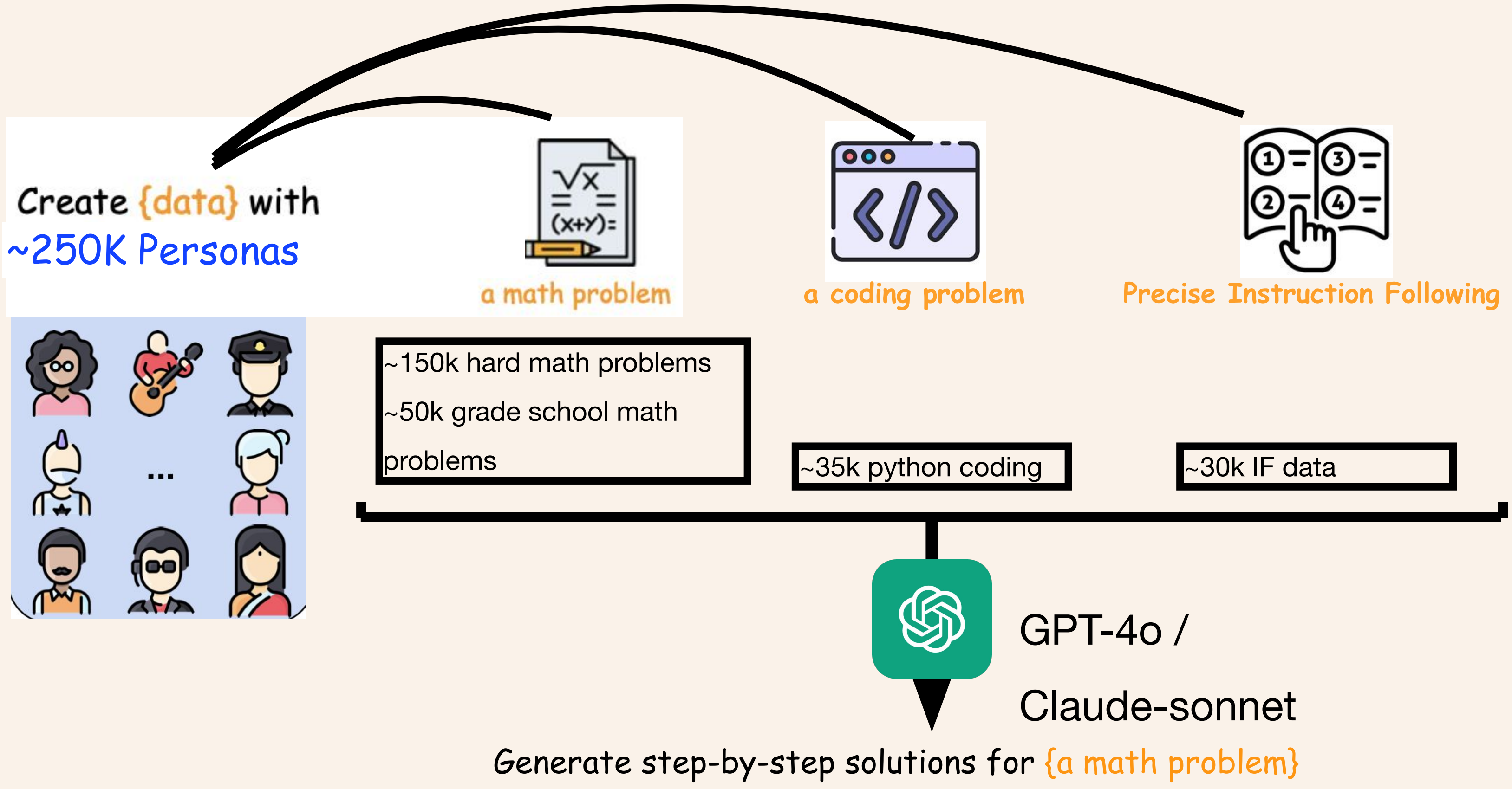


Photo from Ge et al. 2024

Persona-driven Data generation for Scalability and Improved Diversity



Persona-driven Data generation for Scalability and Improved Diversity



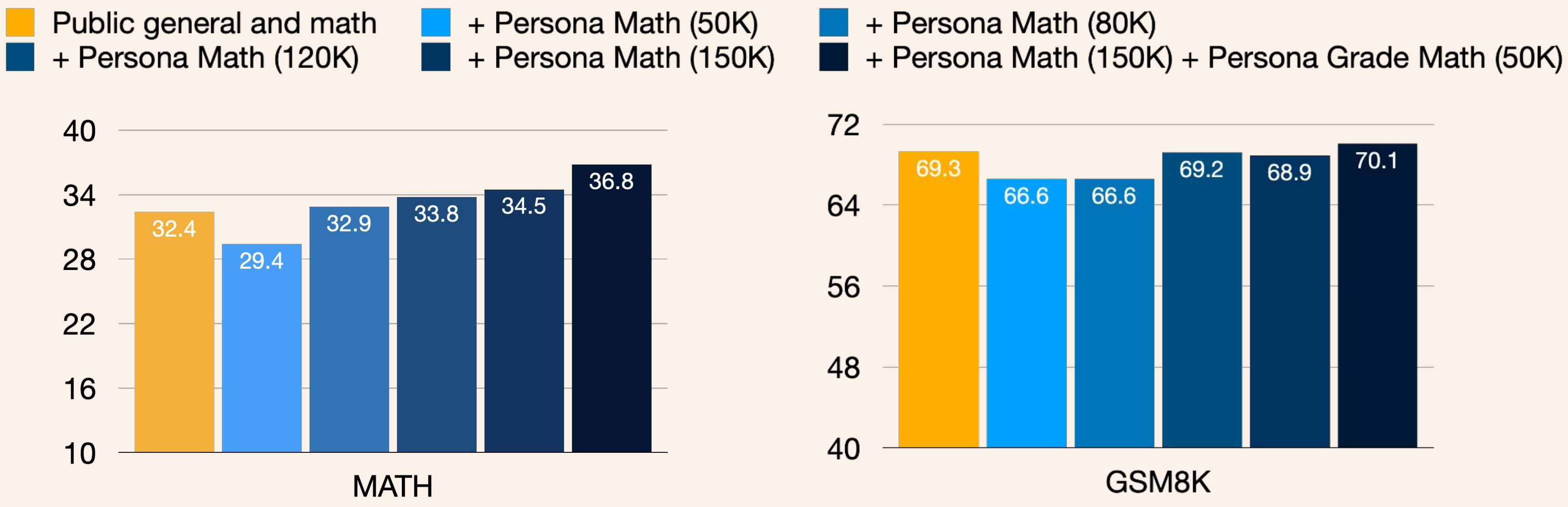
Impact of Persona-Driven Math Data

public datasets ⊕ X **persona-driven synthetic math problems**

%

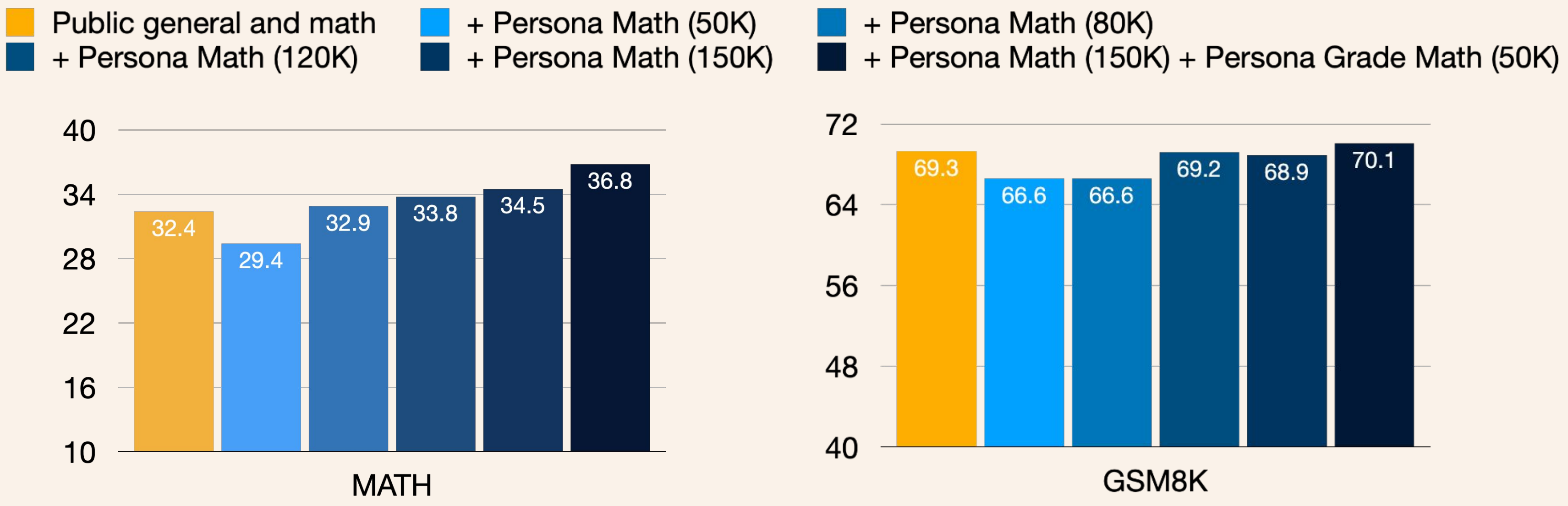
- General purpose (50K)
- NuminaMath-TIR (~64K)

- Hard math problems (150K)
- Grade school math (~50K)



Impact of Persona-Driven Math Data

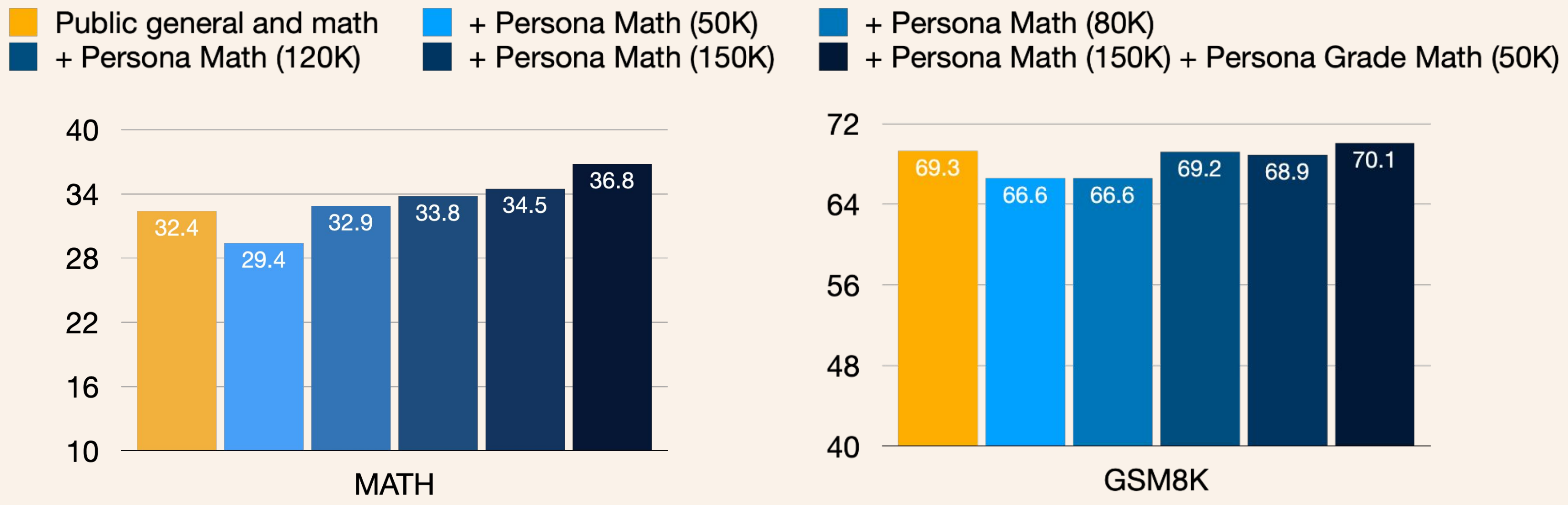
Adding more persona-driven math data, consistently improve MATH performance



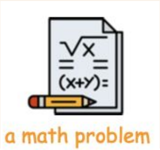
Impact of Persona-Driven Math Data

Adding more persona-driven math data, consistently improve MATH performance

- GSM8k improves (less than math)
- Adding grade-school math helps

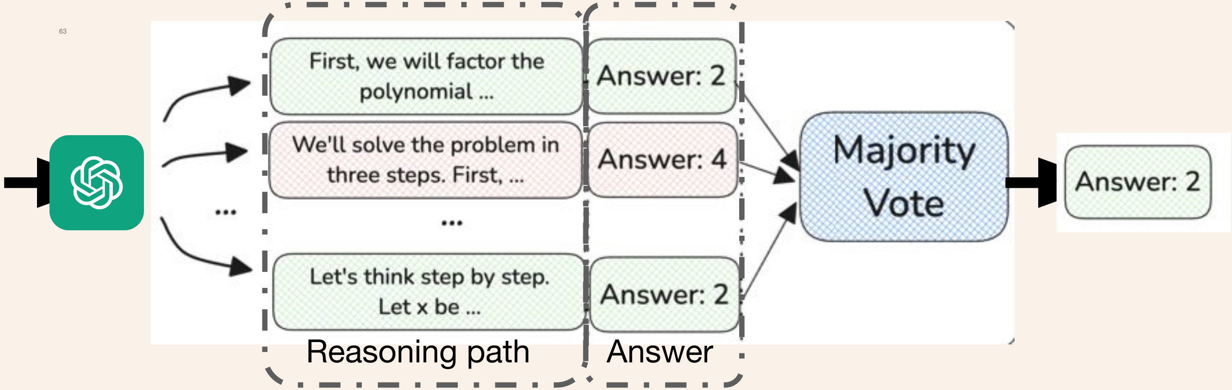


Improving data quality via voting / self-consistency



Dr. Smith, a chemist, is studying a reaction where compound X decomposes into products Y and Z. The reaction follows first-order kinetics with a rate constant k of 0.5 min^{-1}.

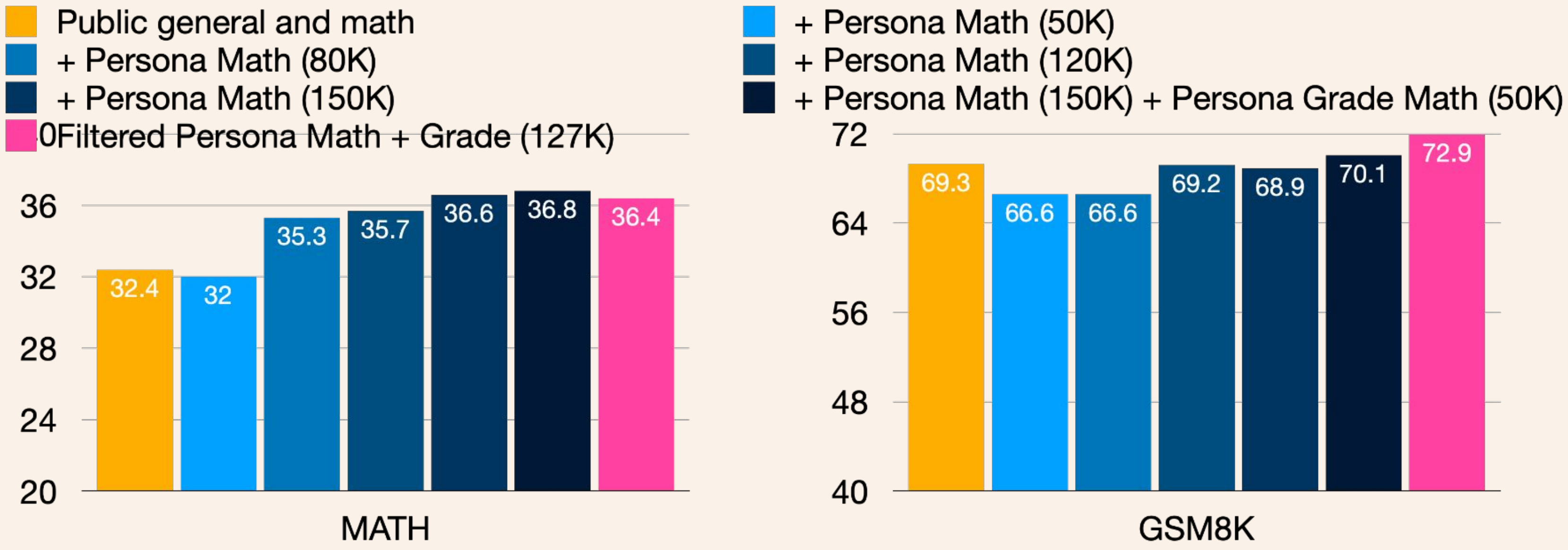
If the initial concentration of compound X is 1.0 M, how long will it take for the concentration of X to decrease to 0.25 M?



Remove instances with no majority vote!

Less data, Same or Better Performance

Using only ~60% of the data, we are still able to main the performance in MATH and improve in GSM8K



Other approaches to generate COT data

1. Manual Human Annotation (e.g., GSM8K dataset): Annotators write step by step solutions
    - High-quality reasoning traces
    - Limited **scale** (only 7K)
    - Lack of **diversity** in reasoning styles
2. Program-Aided Language Models (PAL): Convert math problems into Python code execution traces
    - Guarantee correctness through execution
    - Less natural language reasoning, less intuitive
    - Limited to problems that can be coded
3. Self-generated COT (self-ask): using LLMs to generate their reasoning paths
    - Scalable to many problems
    - **Quality** highly dependent on base model

Capability-driven data mixing

Core capabilities

Chat

Knowledge

Reasoning

Coding

Safety

Multilinguality

Math

Precise instruction following

...

→ Better

Human + Synthetic

data mix → Tulu SFT

Data mixing for SFT

Model	Avg.	MMLU	TQA	PopQA	BBH	CHE	CHE+	GSM	DROP	MATH	IFEval	AE 2	Safety
Tülu 3 8B SFT	**60.1**	62.1	46.8	29.3	67.9	**86.2**	**81.4**	76.2	61.3	31.5	**72.8**	12.4	93.1
→ w/o WildChat	58.9	61.0	45.2	28.9	65.6	85.3	80.7	75.8	59.3	31.8	70.1	7.5	**95.2**
→ w/o Safety	58.0	62.0	45.5	**29.5**	68.3	84.5	79.6	**76.9**	59.4	**32.6**	71.0	12.4	74.7
→ w/o Persona Data	58.6	**62.4**	**48.9**	29.4	68.3	84.5	79.0	76.8	**62.2**	30.1	53.6	**13.5**	93.9
→ w/o Math Data	58.2	62.2	47.1	**29.5**	**68.9**	86.0	80.5	64.1	60.9	23.5	70.6	12.0	93.5

Training on real user interactions with strong models is helpful almost across the board.

Safety training is largely orthogonal to the other skills.

Persona-based data synthesis is very useful for targeting *new* skills.

SFT performance potential

Model	Avg.	MMLU	TQA	PopQA	BBH	CHE	CHE+	GSM	DROP	MATH	IFEval	AE 2	Safety
TÜLU 2 8B SFT	48.3	61.8	49.4	23.3	57.1	66.9	63.1	60.4	**61.7**	14.0	42.3	8.9	70.7
RLHFlow SFT V2	56.0	**65.8**	**56.0**	**29.7**	**69.3**	**86.2**	80.9	**81.6**	57.2	**35.7**	52.7	**13.6**	43.5
MAmmoTH2 8B	46.4	63.6	42.7	20.8	63.4	72.8	66.4	63.7	43.8	30.5	34.9	6.5	47.8
TÜLU 3 8B SFT	**60.1**	62.1	46.8	29.3	67.9	**86.2**	**81.4**	76.2	61.3	31.5	**72.8**	12.4	**93.1**
TÜLU 2 70B SFT	63.6	76.0	**57.8**	44.1	79.4	86.8	83.5	83.2	75.9	33.1	57.7	17.3	68.8
TÜLU 3 70B SFT	**72.6**	**79.4**	55.7	**48.6**	**82.7**	**92.9**	**87.3**	**91.1**	**77.2**	**53.7**	**82.1**	**26.3**	**94.4**

Table 8: Summary of the performance of our TÜLU 3 SFT models against comparable baselines. Our final SFT mixtures show strong performance, achieving a higher average score than other comparable mixes. All models, including TÜLU 2 SFT, were trained on either Llama 3.0 or 3.1. Our final Tülu 3 70B model was used to help format this table.

Tülu 3 Step 2: Preference tuning