Post-training data

Qiuyang Mang & Huanzhi Mao

Sep 11, 2025

The Alignment Problem

By the early 2020s, language models had become powerful, but their power was untamed. They lacked alignment with human values, which sets the Stage for **Helpful** and **Harmless** AI.

→ Key problem: How do we effectively inject **human preferences** into LLMs?

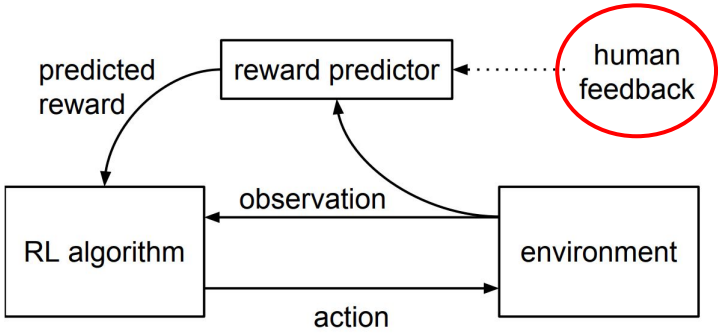
User: Let's discuss how to steal money from the bank.

GPT

1. First, you need to get a job inside the bank, this will give you access to the …

Reinforcement Learning from Human Feedback

1. Learning from Human Preferences (Christiano et al., 2017)



How do we systematically collect human preference data?

How do we use that data to train a model to be both helpful and harmless?

Reinforcement Learning from Human Feedback

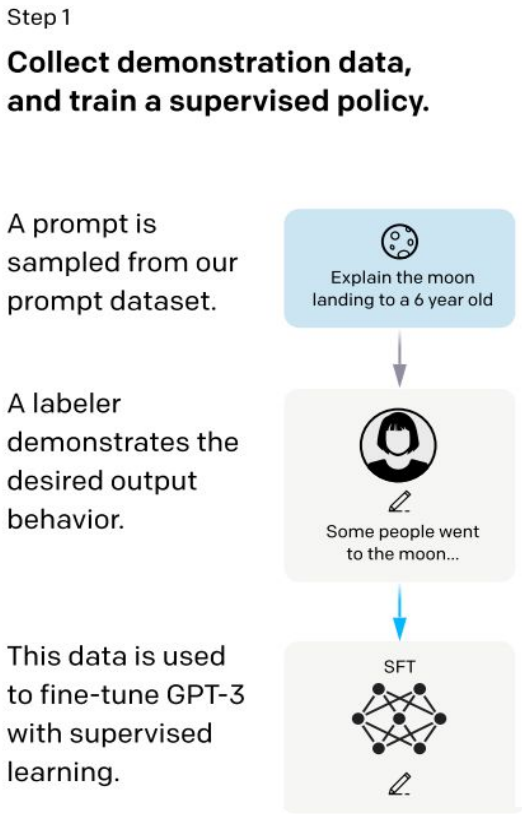
2. InstructGPT (Ouyang et al., 2022)

   a) Collecting a dataset of human-written demonstrations, and using **SFT** to train an initial policy

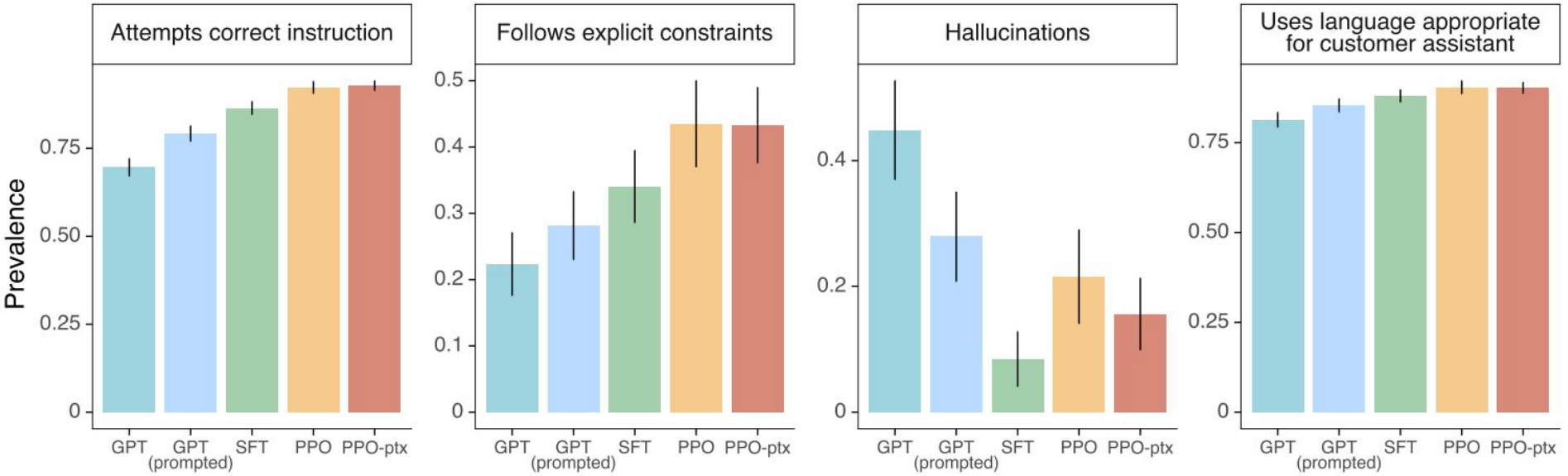
   b) Training a *reward model* on human-labeled comparisons of different model outputs (**PPO**).

   c) Using that reward model to fine-tune the language model with reinforcement learning

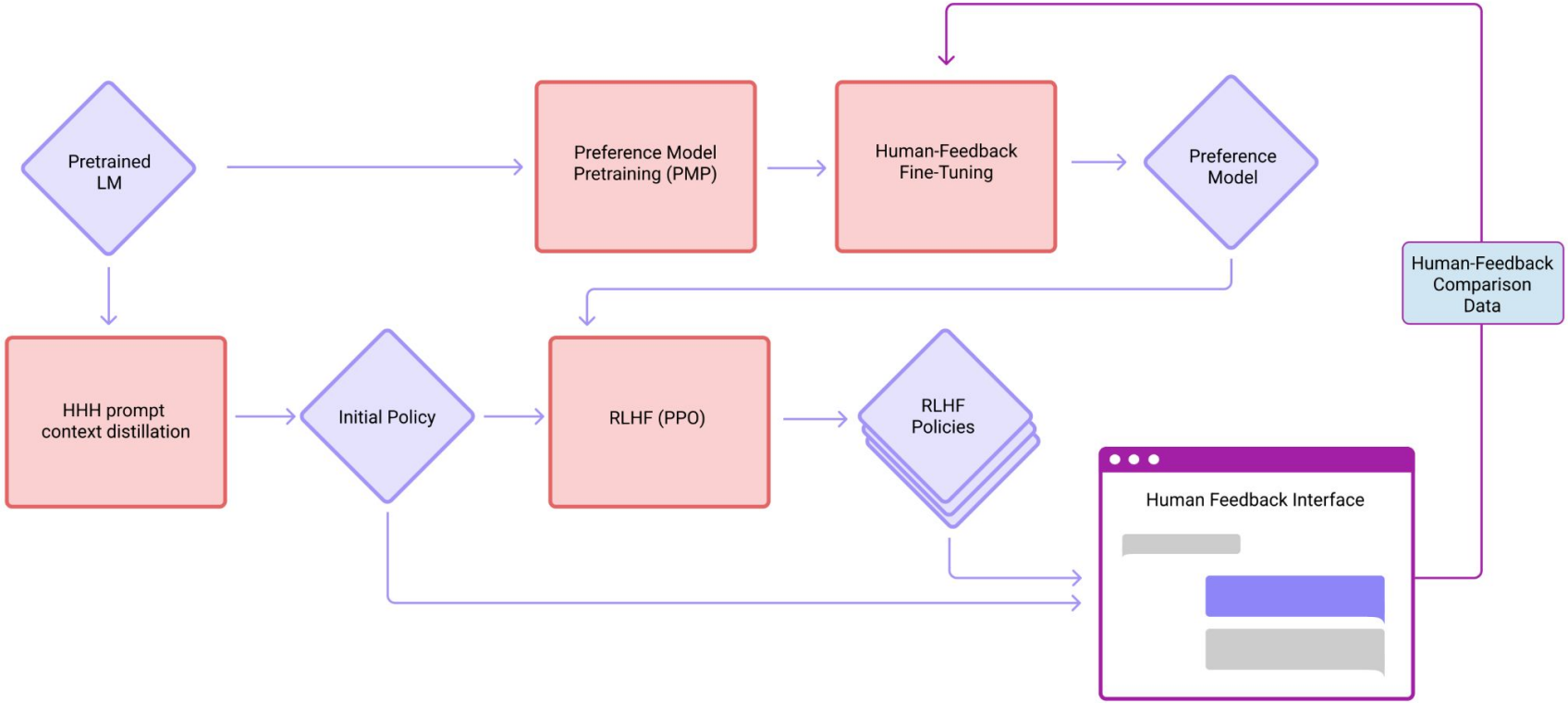
Helpfulness ✔ **Harmlessness** ✘



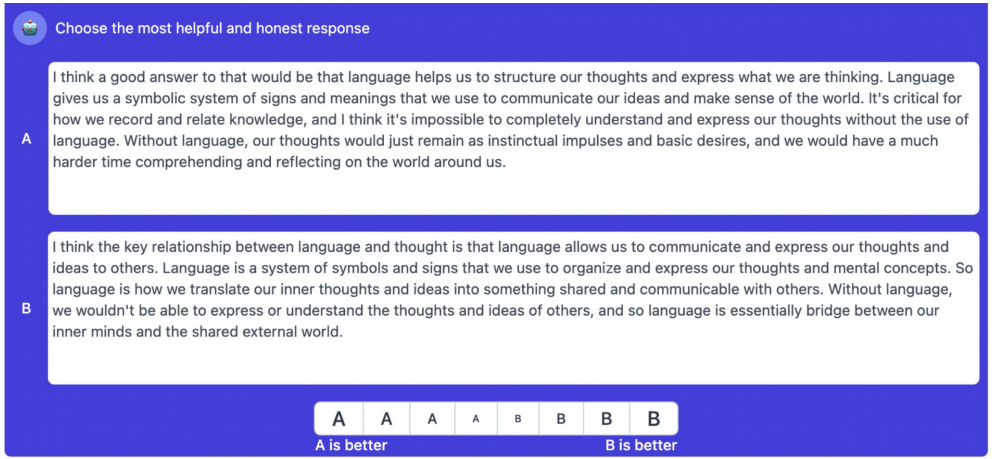
Reinforcement Learning from Human Feedback



RLHF for both helpfulness and harmlessness (Bai et al.)



How to Collect Human Feedback



People choose their preferred option instead of writing concrete feedback.

Rate for helpfulness / harmlessness

$$\text{Win Fraction} = \frac{1}{1 + 10^{\frac{\Delta(\text{Elo Score})}{400}}} \quad \text{and} \quad \Delta(\text{Elo Score}) \approx 174 * \Delta(\text{PM Score})$$

Use Elo rating to evaluate models

PM Calibration

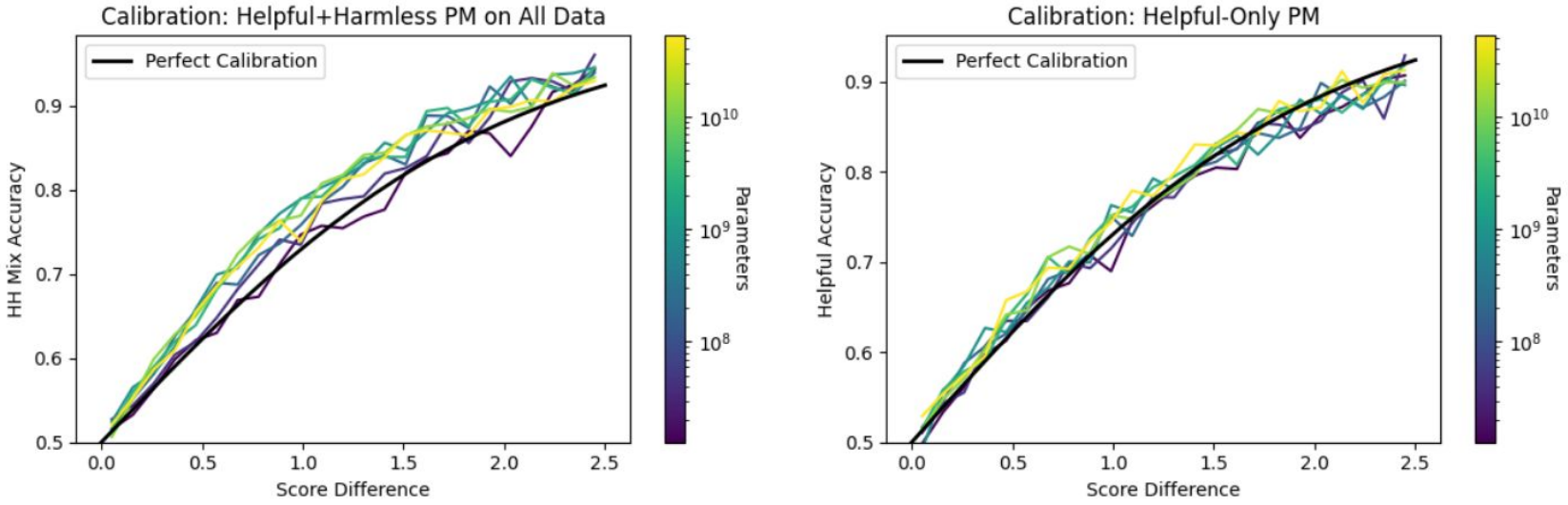


Figure 9 We show preference modeling accuracy as a function of the difference in PM score between higher and lower ranked responses. The black lines indicate the calibrated prediction of accuracy $1/(1+e^{-\Delta})$, where Δ is the score difference. On the **(left)** we show calibration for a PM trained and evaluated on all our static data, while on the **(right)** we show results for a model trained and evaluated only on our helpful data distribution. We see that calibration is slightly worse for models trained on the HH mixture.

“Online” training (pros and cons)

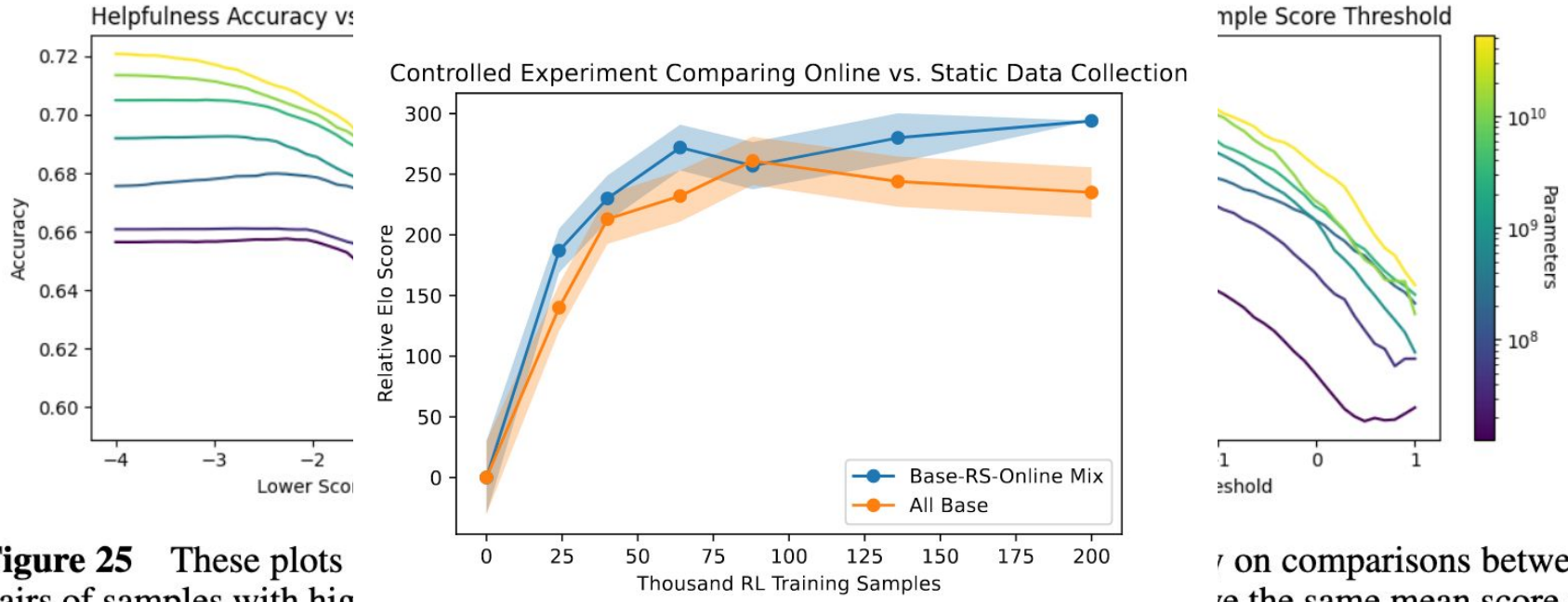


Figure 25 These plots on comparisons between pairs of samples with hig e the same mean score on a held-out dataset so that they’re directly comparable, and then plotted accuracy for the comparisons where both samples have scores above a specific threshold.

Tulu 3: Pushing Frontiers in Open Language Model Post-Training

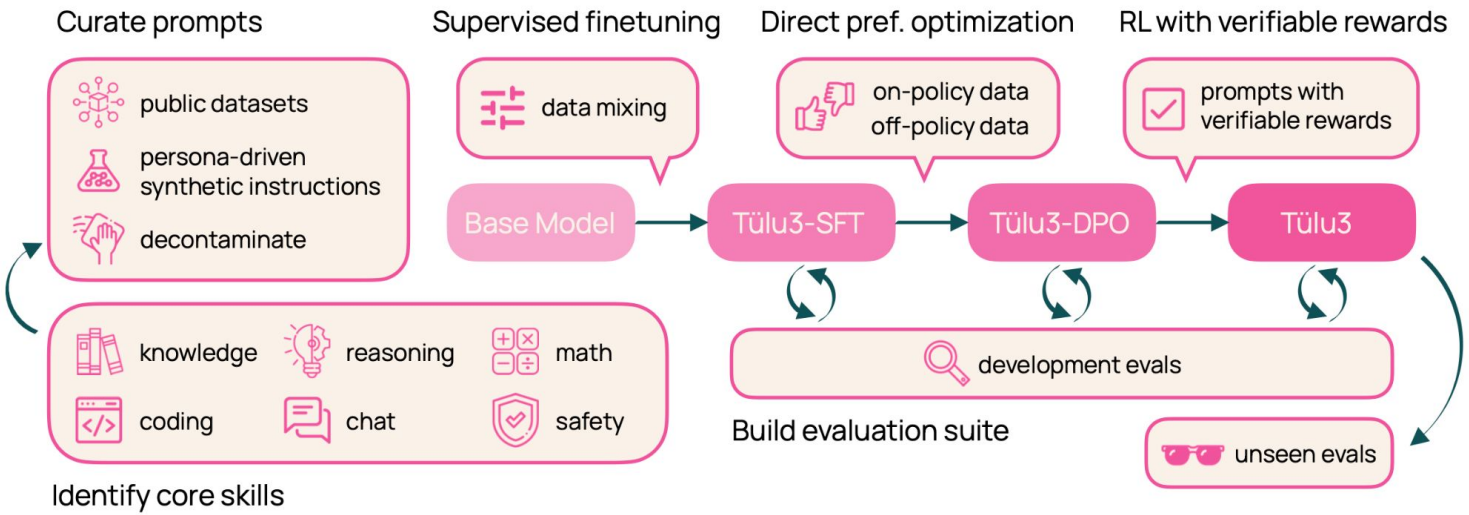


Figure 1 An overview of the TÜLU 3 recipe. This includes: data curation targeting general and target capabilities, training strategies and a standardized evaluation suite for development and final evaluation stage.

SFT (Supervised Fine-Tuning)

Teach the model what good outputs look like for your tasks.

- Gives the policy a good manifold of behaviors. Without SFT, RL tends to wander/exploit quirks of the reward.
- Strong SFT reduces the amount and brittleness of preference/RL training.

Preference Optimization

SFT teaches “valid” answers; preferences teach which valid answers people most want—style, safety, refusal criteria, brevity, formatting, etc.

PPO: Collect pairwise preferences (chosen vs. rejected completions). Train an RM to score outputs. Then do on-policy RL to maximize RM score while keeping the policy near a reference.

- Pros: Can explore new responses beyond your logged data. Often yields bigger gains when you need distribution shift or long outputs.
- Cons: More engineering: sampling loops, credit assignment, stability (KL tuning, reward scaling), higher compute.

DPO: Skip explicit RM + RL loops. Optimize a closed-form objective on preference pairs to increase the odds of chosen over rejected responses relative to a reference model.

- Pros: Simple, stable, fast, no rollout loop; great as a first preference pass; easy to reproduce.
- Cons: Trains offline on the pairs you have; weaker exploration; tougher to target long-horizon/rare behaviors unless your preference data already covers them.

RL from verifiable feedback (RLVF)

Use programmatic signals as the reward; train the model to optimize an objective you can check automatically (unit tests passing, math answers correct, tool calls valid, constraints satisfied).

- Many valuable goals are objectively checkable and not well captured by human style preferences (e.g., “does the SQL actually run?”, “is the plan executable under a budget?”).

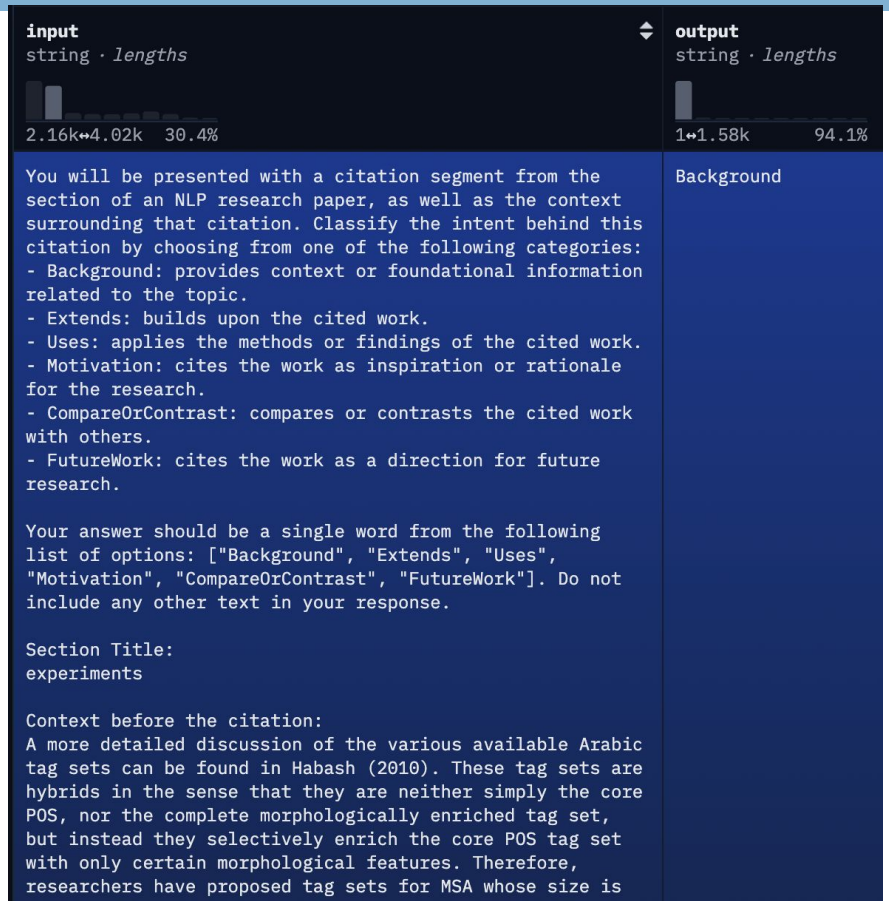
Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO	Reference
General	**Tülu 3 Hardcoded**$^{\uparrow}$	24	240	–	–
	OpenAssistant$^{1,2,\downarrow}$	88,838	7,132	7,132	Köpf et al. (2024)
	No Robots	9,500	9,500	9,500	Rajani et al. (2023)
	WildChat (GPT-4 subset)$^{\downarrow}$	241,307	100,000	100,000	Zhao et al. (2024)
	UltraFeedback$^{\alpha,2}$	41,635	–	41,635	Cui et al. (2023)
Knowledge	FLAN v2$^{1,2,\downarrow}$	89,982	89,982	12,141	Longpre et al. (2023)
Recall	SciRIFF$^{\downarrow}$	35,357	10,000	17,590	Wadden et al. (2024)
	TableGPT$^{\downarrow}$	13,222	5,000	6,049	Zha et al. (2023)
Math	**Tülu 3 Persona MATH**	149,960	149,960	–	–
Reasoning	**Tülu 3 Persona GSM**	49,980	49,980	–	–
	Tülu 3 Persona Algebra	20,000	20,000	–	–
	OpenMathInstruct 2$^{\downarrow}$	21,972,791	50,000	26,356	Toshniwal et al. (2024)
	NuminaMath-TIR$^{\alpha}$	64,312	64,312	8,677	Beeching et al. (2024)
Coding	**Tülu 3 Persona Python**	34,999	34,999	–	–
	Evol CodeAlpaca$^{\alpha}$	107,276	107,276	14,200	Luo et al. (2023)
Safety	**Tülu 3 CoCoNot**	10,983	10,983	10,983	Brahman et al. (2024)
& Non-Compliance	**Tülu 3 WildJailbreak**$^{\alpha,\downarrow}$	50,000	50,000	26,356	Jiang et al. (2024)
	Tülu 3 WildGuardMix$^{\alpha,\downarrow}$	50,000	50,000	26,356	Han et al. (2024)
Multilingual	Aya$^{\downarrow}$	202,285	100,000	32,210	Singh et al. (2024b)
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$^{\gamma}$	

Table 7 Summary of our prompt dataset: data for training stages are selected from these prompts. New datasets released with TÜLU 3 are **color-coded** for emphasis. Existing datasets we modified due to contamination are marked with α. Datasets with prompts used in TÜLU 1 or 2 are marked with ^{1} or ^{2}, respectively. Datasets marked with $^{\downarrow}$ are downsampled from their original datasets, datasets marked with $^{\uparrow}$ are upsampled. Note that all datasets were filtered to remove specific keywords (e.g., OpenAI) and empty messages, resulting in slightly lower than reported counts. All TÜLU 3 datasets with Persona expand the methodology of Chan et al. (2024). The percentages listed per category are out of the total prompts. Preference count is marked with $^{\gamma}$ to note that not all prompts are used in both the 8B and 70B mixes – for exact details see Table 15.

nvidia/OpenMathInstruct-2

problem string · *lengths* 13↔1.38k 99.7%	generated_solution string · *lengths* 190↔5.24k 100%	expected_answer string · *lengths* 0↔54 99.6%
What is the largest number, all of whose digits are 1 or 4, and whose digits add up to 12?	We need to find the largest number consisting of only 1's and 4's that adds up to 12. The largest possible number is obtained by using the largest possible number of digits, which is achieved by using as many 1's as possible. However, we also want to maximize the number of digits, so we should use as many 4's as possible without exceeding the sum of 12. The maximum number of 4's we can use is 3 (since $3 \cdot 4 = 12$), but that would leave no room for 1's. Therefore, let's use 2 fours and make up the rest with 1's. Two 4's add up to 8, leaving $12 - 8 = 4$ to be made up by 1's. We can use 4 ones to make up this difference. The largest number is thus obtained by arranging these digits in decreasing order: 441111. However, we want the largest number, so we should arrange the digits in decreasing order. Thus, the largest number is 444111, no, there are too many 4's. So, the largest number is: \[\boxed{441111} \]	441111

allenai/SciRIFF



input string · *lengths* 2.16k↔4.02k 30.4%	output string · *lengths* 1↔1.58k 94.1%
You will be presented with a citation segment from the section of an NLP research paper, as well as the context surrounding that citation. Classify the intent behind this citation by choosing from one of the following categories: - Background: provides context or foundational information related to the topic. - Extends: builds upon the cited work. - Uses: applies the methods or findings of the cited work. - Motivation: cites the work as inspiration or rationale for the research. - CompareOrContrast: compares or contrasts the cited work with others. - FutureWork: cites the work as a direction for future research. Your answer should be a single word from the following list of options: ["Background", "Extends", "Uses", "Motivation", "CompareOrContrast", "FutureWork"]. Do not include any other text in your response. Section Title: experiments Context before the citation: A more detailed discussion of the various available Arabic tag sets can be found in Habash (2010). These tag sets are hybrids in the sense that they are neither simply the core POS, nor the complete morphologically enriched tag set, but instead they selectively enrich the core POS tag set with only certain morphological features. Therefore, researchers have proposed tag sets for MSA whose size is	Background

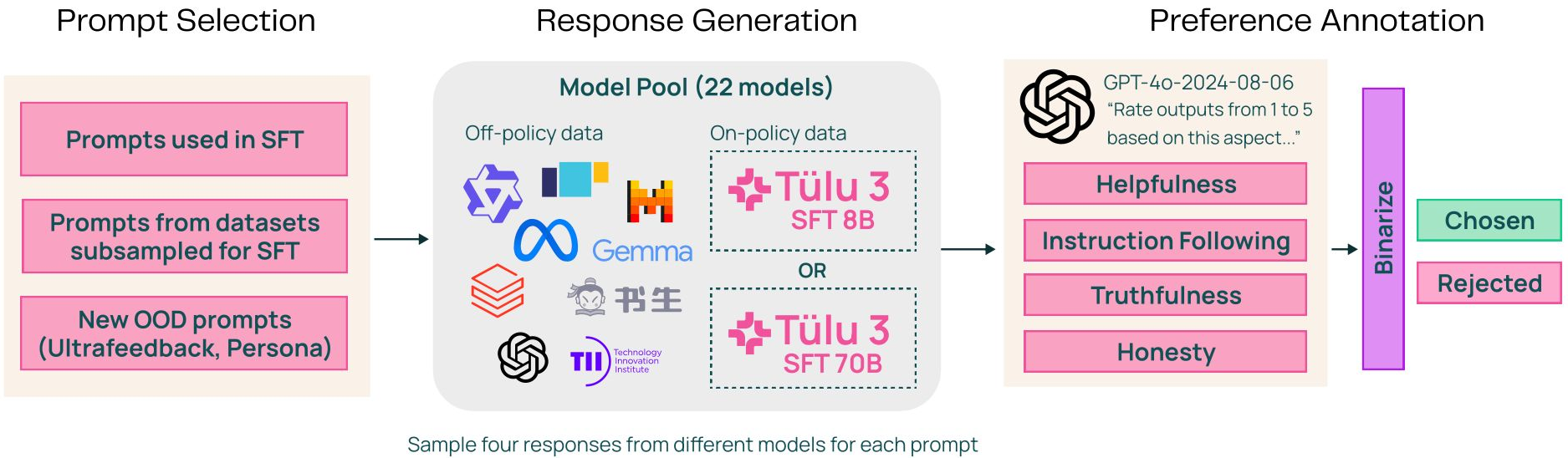


Figure 7 Pipeline for generating and scaling preference data that is based from Ultrafeedback (Cui et al., 2023).

Benchmark$_{(\text{eval})}$	Llama 3.1 405B Instruct	Nous Hermes 3 405B	Deepseek V3	GPT 4o (11-24)	Tülu 3 405B SFT	Tülu 3 405B DPO	Tülu 3 405B RLVR
Avg w/o Safety.	78.1	74.4	79.0	**80.5**	76.3	79.0	80.0
Avg w/ Safety.	79.0	73.5	75.9	**81.6**	77.5	79.6	80.7
MMLU$_{(5\text{ shot, CoT})}$	**88.0**	84.9	82.1	87.9	84.4	86.6	87.0
PopQA$_{(3\text{ shot})}$	52.9	54.2	44.9	53.6	**55.7**	55.4	55.5
BigBenchHard$_{(0\text{ shot, CoT})}$	87.1	87.7	**89.5**	83.3	88.0	88.8	88.6
MATH$_{(4\text{ shot, Flex})}$	66.6	58.4	**72.5**	68.8	63.4	59.9	67.3
GSM8K$_{(8\text{ shot, CoT})}$	95.4	92.7	94.1	91.7	93.6	94.2	**95.5**
HumanEval$_{(\text{pass@10})}$	95.9	92.3	94.6	97.0	95.7	**97.2**	95.9
HumanEval+$_{(\text{pass@10})}$	90.3	86.9	91.6	92.7	93.3	**93.9**	92.9
IFEval$_{(\text{loose prompt})}$	**88.4**	81.9	88.0	84.8	82.4	85.0	86.0
AlpacaEval 2$_{(\text{LC \% win})}$	38.5	30.2	53.5	**65.0**	30.4	49.8	51.4
Safety$_{(6\text{ task avg.})}$	86.8	65.8	72.2	**90.9**	87.7	85.5	86.7

Table 4 Summary of TÜLU 3 results relative to peer 405B models. The best-performing model on each benchmark (i.e., in each row) is **bolded**. TÜLU 3-405B outperforms prior state-of-the-art models finetuned from Llama 3.1 405B Base and rivals some leading, closed models. Progress across various checkpoints highlight the contribution of each stage of the training in improving core skills. Note that TruthfulQA and MMLU multiple choice numbers are not compatible with our infrastructure for running evaluations (via log-probs).

How pre-training affects post-training results

Base Model	GSM8K	MATH
Llama 3.1 8B	76.2	31.5
Llama 3.1 70B	91.1	53.7
Qwen 2.5 7B	79.2	49.4
Qwen 2.5 Math 7B	86.3	56.4

Table 12 Mathematical performance of different base models trained on our mix. We see that 1) training on larger models leads to better performance, and 2) adding skill-specific pretraining data also leads to improved performance, even for the same size model.

Model Size		8B			70B		
Category	Benchmark$_{\text{(Eval Setting)}}$	Llama 3.1 Inst.	Tülu 3 DPO	Tülu 3 RLVR	Llama 3.1 Inst.	Tülu 3 DPO	Tülu 3 RLVR
Avg.		62.2	64.4	**64.8**	73.4	75.9	**76.0**
Knowledge	MMLU$_{\text{(0 shot, CoT)}}$	**71.2**	68.7	68.2	**85.3**	83.3	83.1
	PopQA$_{\text{(15 shot)}}$	20.2	**29.3**	29.1	46.4	46.3	**46.5**
	TruthfulQA$_{\text{(6 shot)}}$	55.1	**56.1**	55.0	66.8	**67.9**	67.6
Reasoning	BigBenchHard$_{\text{(3 shot, CoT)}}$	62.8	65.8	**66.0**	73.8	81.8	**82.0**
	DROP$_{\text{(3 shot)}}$	61.5	62.5	**62.6**	**77.0**	74.1	74.3
Math	MATH$_{\text{(4 shot CoT, Flex)}}$	42.5	42.0	**43.7**	56.4	62.3	**63.0**
	GSM8K$_{\text{(8 shot, CoT)}}$	83.4	84.3	**87.6**	**93.7**	93.5	93.5
Code	HumanEval$_{\text{(pass@10)}}$	**86.3**	83.9	83.9	**93.6**	92.4	92.4
	HumanEval+$_{\text{(pass@10)}}$	**82.9**	78.6	79.2	**89.5**	88.4	88.0
IF & Chat	IFEval$_{\text{(Strict)}}$	80.6	81.1	**82.4**	**88.0**	82.6	83.2
	AlpacaEval 2$_{\text{(LC \% win)}}$	24.2	33.5	**34.5**	33.4	49.6	**49.8**
Safety	Safety$_{\text{6 task avg.}}$	75.2	**87.2**	85.5	76.5	**89.0**	88.3

Table 23 Final performance of RLVR-trained TÜLU 3 models compared to Llama 3.1 and DPO starting points. The best-performing model on each benchmark (i.e., in each row) and of each size is **bolded**.

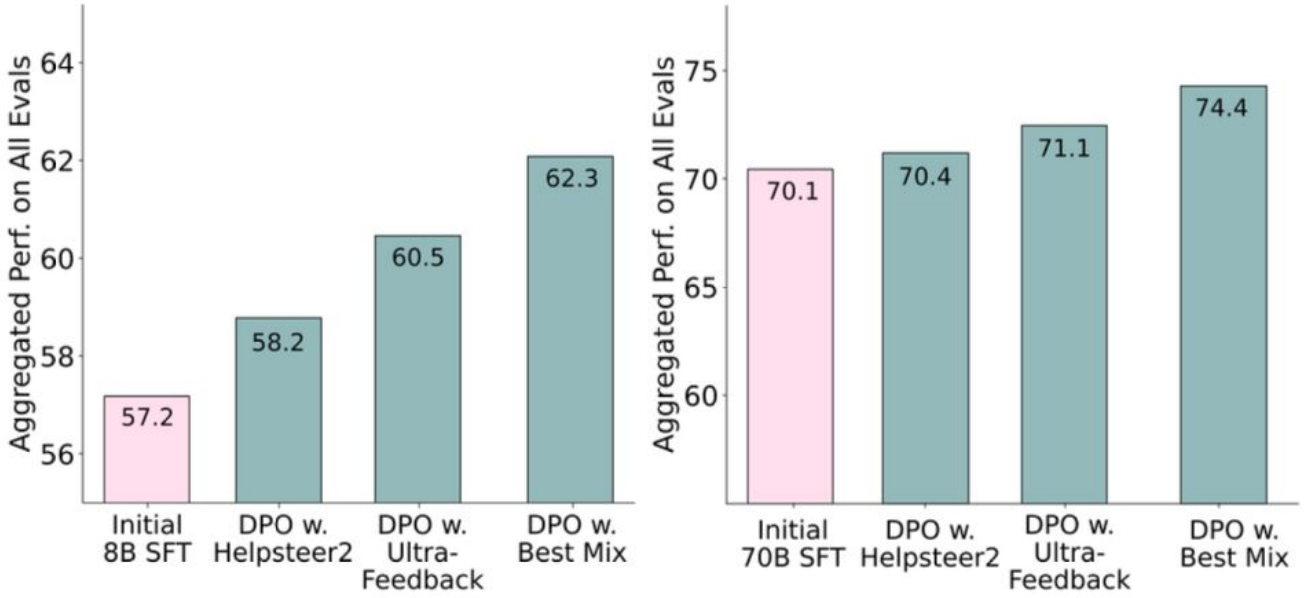


Figure 12 Effect of different DPO mixes on 8B and 70B models: UltraFeedback, Helpsteer2, and our best preference mix.

How later post-training stages depend on earlier ones

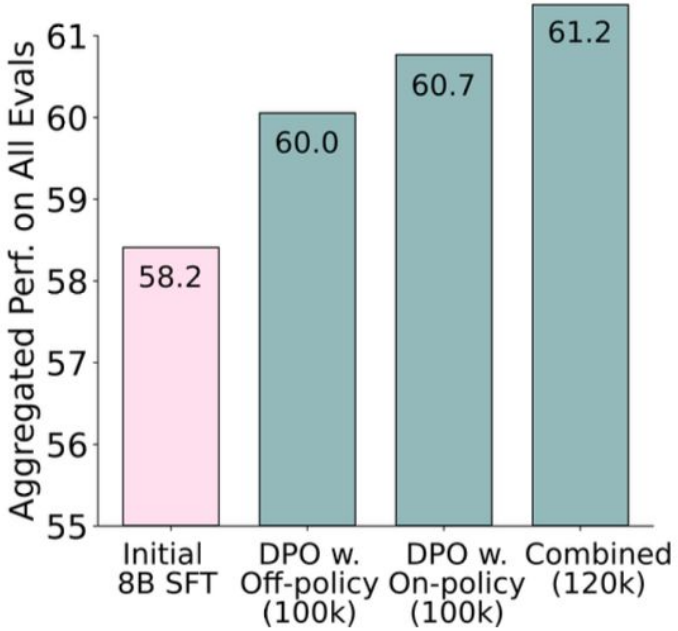


Figure 11 Effect of including on-policy data during the Response Generation stage of the synthetic preference data pipeline on downstream DPO model performance.

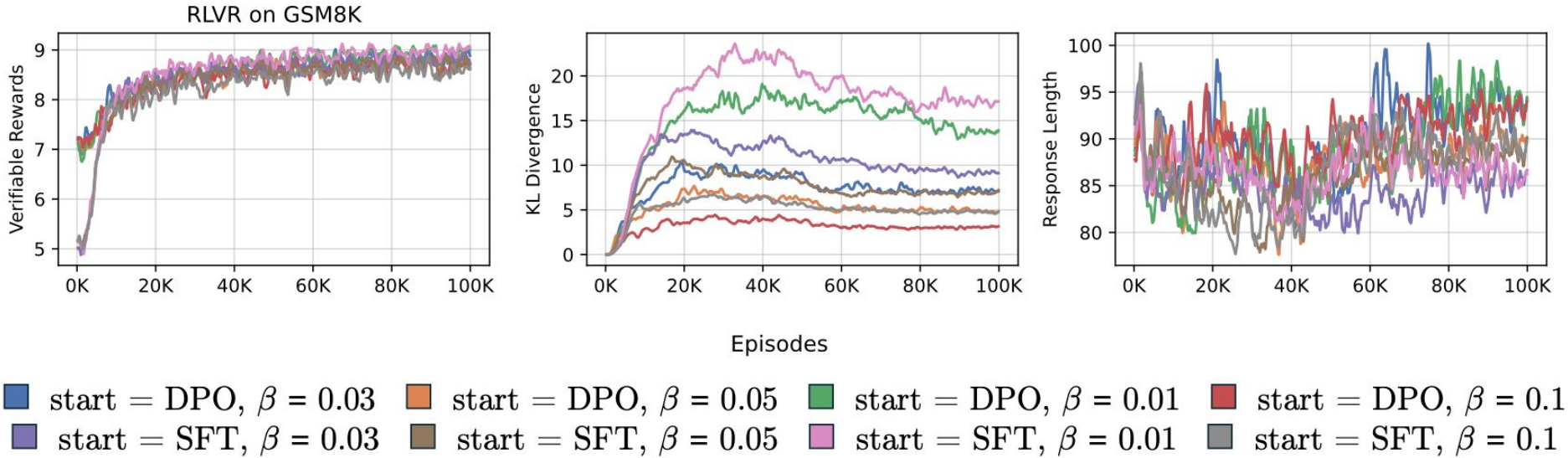


Figure 20 The comparison of RLVR's performance on GSM8K between starting from a DPO checkpoint and starting from a weaker SFT checkpoint. We see that starting from both SFT and DPO can lead to the same level of verifiable rewards, but starting from SFT would incur a larger KL compared to starting from DPO when using the same β.

Post-training data: Critique

Nathan and Yuezhou

09/11/2025

RLHF Paper

1. The model only represents the values of a small group of people (i.e., those who annotated the data).

Tulu3 Paper

Safety is not *completely* orthogonal in post-training but no further discussion on this.
Could excessive RLVR harm the model's general capabilities, turning it into a model that can only solve specific problems but is otherwise incompetent (this is typically seen in small models like Llama 3.2 1B and Qwen3 1.5B)?
Data is short-context. This can hurt generalizability of the model. Data they train on is ~8k tokens long max
Coverage of benchmarks (AIME not included for example) and outdated benchmarks. This model is poorly evaluated.

5. PPO is good for model-based but GRPO is good for rule-based. They choose to use PPO in Tulu3 when it is more natural to do something like GRPO (because math/coding questions have easily computed rewards). PPO leads to suboptimal results.
6. Does using GPT-synthesized data for annotation (used in both SFT and RL phases) limit the method's potential, preventing it from surpassing GPT and leading to model collapse?

Tulu 3 authors: Safety is orthogonal

Page 16:

Safety is Orthogonal. We found that our safety SFT data was generally orthogonal to our other datasets. We report the effect of removing our safety-specific datasets in Table 10, and we see that most skills stayed roughly the same, except the safety average. We also found that adding contrastive prompts, such as those in

Table 10:

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

But downstream…
Safety is NOT orthogonal

Benchmarks	Llama 3.1 8B Instruct	Ministral 8B Instruct	Qwen 2.5 7B Instruct	Tülu 3 8B SFT	Tülu 3 8B DPO	Tülu 3 8B
HarmBench	82.8	53.4	84.1	**98.4**	94.4	94.7
XSTest	**92.7**	85.6	91.8	90.4	92.4	93.3
WildGuardTest	86.2	68.1	85.0	**99.2**	98.9	98.5
Jailbreaktrigger	78.8	63.3	71.0	**95.8**	87.0	85.5
DoAnythingNow	45.0	16.0	61.7	**88.3**	69.7	62.0
WildjailbreakTest	65.6	50.7	56.2	**86.7**	81.1	78.8
Overall	75.2	56.2	75.0	**93.1**	87.2	85.5

Fine-tuning on narrow benchmarks

Skill	Benchmark$_{(eval)}$	Tülu 3 8B	Qwen 2.5 7B Instruct	Llama 3.1 8B Instruct	Tülu 3 70B
	Avg.	65.1	**66.5**	62.9	**76.2**
Knowledge	MMLU$_{(0\ shot,\ CoT)}$	68.2	**76.6**	71.2	83.1
	PopQA$_{(15\ shot)}$	**29.1**	18.1	20.2	**46.5**
	TruthfulQA$_{(6\ shot)}$	55.0	**63.1**	55.1	67.6
Reasoning	BigBenchHard$_{(3\ shot,\ CoT)}$	69.0	70.2	**71.9**	**85.0**
	DROP$_{(3\ shot)}$	**62.6**	54.4	61.5	74.3
Math	MATH$_{(4\ shot\ CoT,\ Flex)}$	43.7	**69.9**	42.5	63.0
	GSM8K$_{(8\ shot,\ CoT)}$	**87.6**	83.8	83.4	93.5
Coding	HumanEval$_{(pass@10)}$	83.9	**93.1**	86.3	92.4
	HumanEval+$_{(pass@10)}$	79.2	**89.7**	82.9	88.0
IF & chat	IFEval$_{(prompt\ loose)}$	**82.4**	74.7	80.6	83.2
	AlpacaEval 2$_{(LC\ \%\ win)}$	**34.5**	29.0	24.2	**49.8**
Safety	Safety$_{(6\ task\ avg.)}$	**85.5**	75.0	75.2	**88.3**

Limited Context Length

Llama 3: 8K

Llama 3.1:128K

Qwen 2.5: 128K

Tulu 3: Trained on 8K, evaluated on 4K

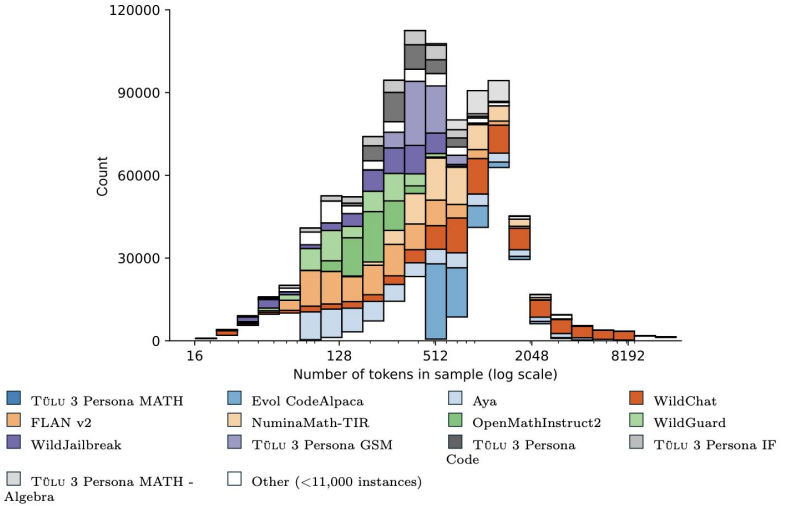


Figure 2 The Tülu 3 final SFT mix by source and length of the prompt plus completion in tokens (using the Llama 3 tokenizer). Compare this distribution to previous open SFT training datasets in Fig. 26. Datasets with the most instances are on the bottom of the histogram.

2.4 Evaluation and Results

When reporting scores throughout this work, we use the metrics identified in Table 3; higher is better. When computing overall performance, we simply average scores across all evaluations, treating each evaluation equally. For generative evaluations our output length is 4096.

Post-training Tata: Proponents

Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback

Tulu 3: Pushing Frontiers in Open Language Model Post-Training

Charles Xu, Xutao Ma

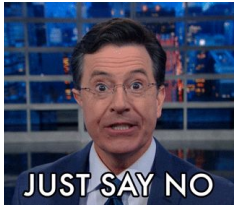
09/11/2025

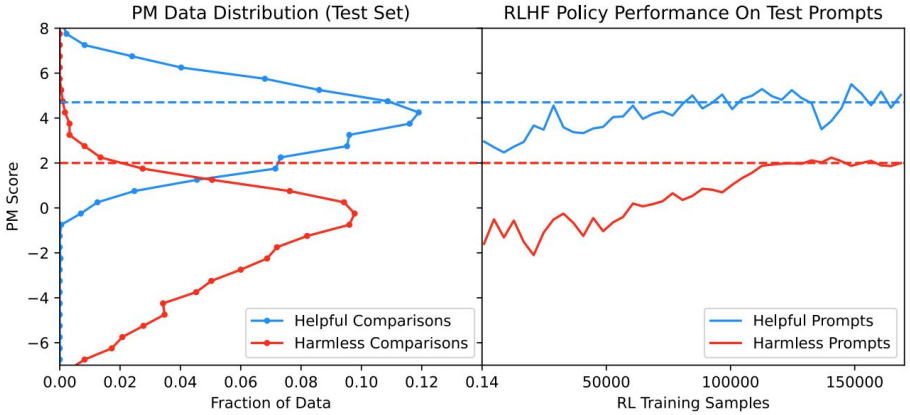
Proponent: RLHF (Bai et al.)

1. “Wisdom of the crowd”: crowdsourcing with simple instructions create diverse prompts based on common-sense understanding of “helpful” and “harmless”
    - “Human feedback have the **largest comparative advantage** over other techniques when people have **complex intuitions that are easy to elicit but difficult to formalize** and automate.”

Proponent: RLHF (Bai et al.)

2. Separate datasets for “helpful” and “harmless”
    - “Red-teaming” to elicit harmful response
    - Disentangle the two objectives in the analysis





- Easier to make model harmless than helpful
    - Redefine harmlessness
    - Use OOD gates to check for harmful prompts

Proponent: Tülu 3

1. Very detailed post-training data recipe and generation method

Category	Prompt Dataset	Count	# Prompts used in SFT	# Prompts used in DPO	Reference
General	Tülu 3 Hardcoded†	24	240	–	–
	OpenAssistant$^{1,2,\downarrow}$	88,838	7,132	7,132	Köpf et al. (2024)
	No Robots	9,500	9,500	9,500	Rajani et al. (2023)
	WildChat (GPT-4 subset)$^{\downarrow}$	241,307	100,000	100,000	Zhao et al. (2024)
	UltraFeedback$^{\alpha,2}$	41,635	–	41,635	Cui et al. (2023)
Knowledge	FLAN v2$^{1,2,\downarrow}$	89,982	89,982	12,141	Longpre et al. (2023)
Recall	SciRIFF$^{\downarrow}$	35,357	10,000	17,590	Wadden et al. (2024)
	TableGPT$^{\downarrow}$	13,222	5,000	6,049	Zha et al. (2023)
Math	Tülu 3 Persona MATH	149,960	149,960	–	–
Reasoning	Tülu 3 Persona GSM	49,980	49,980	–	–
	Tülu 3 Persona Algebra	20,000	20,000	–	–
	OpenMathInstruct 2$^{\downarrow}$	21,972,791	50,000	26,356	Toshniwal et al. (2024)
	NuminaMath-TIR$^{\alpha}$	64,312	64,312	8,677	Beeching et al. (2024)
Coding	Tülu 3 Persona Python	34,999	34,999	–	–
	Evol CodeAlpaca$^{\alpha}$	107,276	107,276	14,200	Luo et al. (2023)
Safety	Tülu 3 CoCoNot	10,983	10,983	10,983	Brahman et al. (2024)
& Non-Compliance	Tülu 3 WildJailbreak$^{\alpha,\downarrow}$	50,000	50,000	26,356	Jiang et al. (2024)
	Tülu 3 WildGuardMix$^{\alpha,\downarrow}$	50,000	50,000	26,356	Han et al. (2024)
Multilingual	Aya$^{\downarrow}$	202,285	100,000	32,210	Singh et al. (2024b)
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$^{\gamma}$	

Table 7 Summary of our prompt dataset: data for training stages are selected from these prompts. New datasets released with TÜLU 3 are color-coded for emphasis. Existing datasets we modified due to contamination are marked with α. Datasets with prompts used in TÜLU 1 or 2 are marked with ^{1} or ^{2}, respectively. Datasets marked with $^{\downarrow}$ are downsampled from their original datasets, datasets marked with † are upsampled. Note that all datasets were filtered to remove specific keywords (e.g., OpenAI) and empty messages, resulting in slightly lower than reported counts. All TÜLU 3 datasets with Persona expand the methodology of Chan et al. (2024). The percentages listed per category are out of the total prompts. Preference count is marked with $^{\gamma}$ to note that not all prompts are used in both the 8B and 70B mixes – for exact details see Table 15.

396,142 out of 939,344 is synthetic for SFT phase 42%
149,115 out of 425,145 is synthetic for DPO phase 35%

Proponent: Tülu 3

1. Very detailed post-training data recipe and generation method

Precise Instruction Following (prompt)

Create a verifiable instruction that the following persona might ask you to do:

`{persona}`

An example of verifiable instruction could be: `{example}`

Note:

1. The above example is not tied to any particular persona, but you should create one that is unique and specific to the given persona.
2. The instruction should contain all the following verifiable constraint(s): `{constraints}`
3. Your output should start with "User instruction:". Your output should not include an answer to the instruction.

Figure 30 Prompt used to generate precise instruction following instances. `{persona}` are borrowed from Chan et al. (2024). We use the set of `{constraints}` defined in Zhou et al. (2023). Example seeds are manually written by authors for each constraint.

Precise Instruction Following (response)

Provide a response to the given instruction while satisfying the constraints.
Instruction: `{generated_instruction}`

Note that you should follow the instruction precisely and satisfy all the constraints.

Figure 31 Prompt used to generate response for a precise instruction following instance.

396,142 out of 939,344 is synthetic for SFT phase 42%
149,115 out of 425,145 is synthetic for DPO phase 35%

For Precise Instruction Following data

We use our **persona-driven approach** to synthetically generate verifiable instructions covering 25 different constraint types defined in **IFEval benchmark.** More concretely, we start by manually writing 1-2 example instructions per constraint (few shot), resulting in total of 33 verifiable instructions which we used as seed prompts. We then generate new instructions using GPT-4o given a data synthesis prompt, persona, and a single verifiable instruction as an example.

Proponent: Tülu 3

Discussion: How to do long-context post-training?

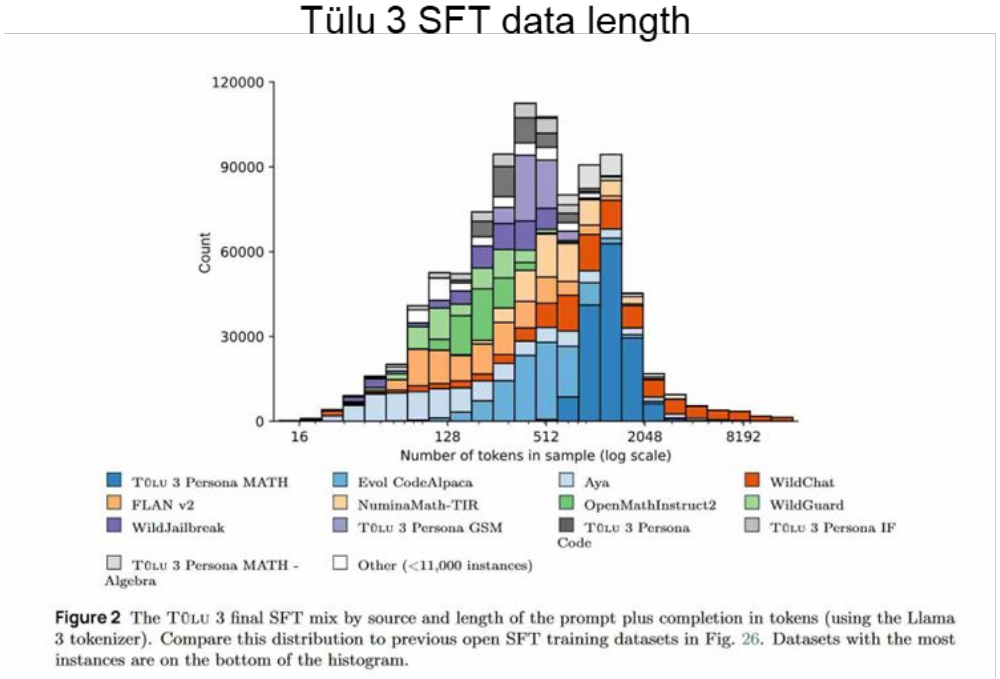


Figure 2 The TÜLU 3 final SFT mix by source and length of the prompt plus completion in tokens (using the Llama 3 tokenizer). Compare this distribution to previous open SFT training datasets in Fig. 26. Datasets with the most instances are on the bottom of the histogram.

Proponent: Tülu 3

Discussion: How to do long-context post-training?

Tülu 3 SFT data length

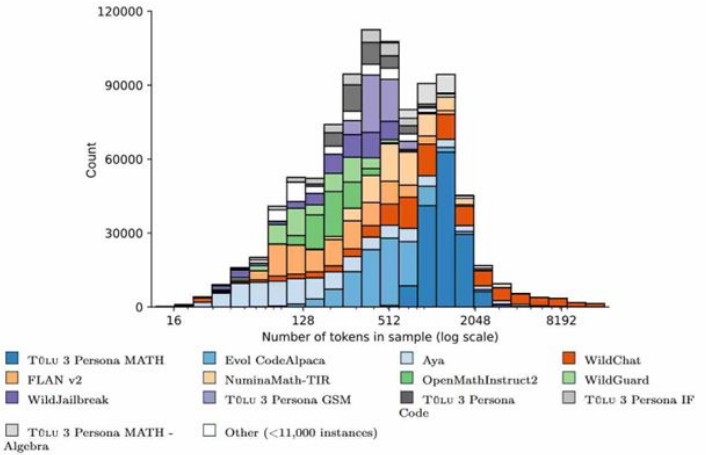


Figure 2 The TÜLU 3 final SFT mix by source and length of the prompt plus completion in tokens (using the Llama 3 tokenizer). Compare this distribution to previous open SFT training datasets in Fig. 26. Datasets with the most instances are on the bottom of the histogram.

Llama 3 SFT data length

Dataset	% of examples	Avg. # turns	Avg. # tokens	Avg. # tokens in context	Avg. # tokens in final response
General English	52.66%	6.3	974.0	656.7	317.1
Code	14.89%	2.7	753.3	378.8	374.5
Multilingual	3.01%	2.7	520.5	230.8	289.7
Exam-like	8.14%	2.3	297.8	124.4	173.4
Reasoning and tools	21.19%	3.1	661.6	359.8	301.9
Long context	0.11%	6.7	38,135.6	37,395.2	740.5
Total	100%	4.7	846.1	535.7	310.4

Table 7 Statistics of SFT data. We list internally collected SFT data used for Llama 3 alignment. Each SFT example consists of a context (i.e., all conversation turns except the last one) and a final response.

We further categorize these synthetically generated samples based on the sequence length (16K, 32K, 64K and 128K) to enable more fine-grained targeting of input lengths.

Through careful ablations, we observe that mixing 0.1% of synthetically generated long-context data with the original short-context data optimizes the performance across both short-context and long-context benchmarks.

Proponent: Tülu 3

Discussion: Model capability evaluation

1. What kind of capabilities do we want?
2. How to evaluate these capabilities?

Proponent: Tülu 3

Discussion: Model capability evaluation

1. What kind of capabilities do we want?

Tülu 3

Core Skill	Development	Unseen
Knowledge	$\text{MMLU}_{(\text{em})}$ $\text{PopQA}_{(\text{EM})}$ $\text{TruthfulQA}_{(\text{MC2 em})}$	$\text{MMLU-Pro}_{(\text{em})}$ $\text{GPQA}_{(\text{em})}$
Reasoning	$\text{BigBenchHard}_{(\text{em})}$ $\text{DROP}_{(\text{F1})}$	$\text{AGIEval English}_{(\text{em})}$
Math	$\text{MATH}_{(\text{flex em})}$ $\text{GSM8K}_{(\text{em})}$	$\text{Deepmind Mathematics}_{(\text{em})}$
Coding	$\text{HumanEval}_{(\text{Pass@10})}$ $\text{HumanEval+}_{(\text{Pass@10})}$	$\text{BigcodeBench}_{(\text{Pass@10})}$
Instruction Following (IF)	$\text{IFEval}_{(\text{em})}$ $\text{AlpacaEval 2}_{(\text{winrate})}$	$\text{IFEval-OOD}_{(\text{Pass@1})}$ $\text{HREF}_{(\text{winrate})}$
Safety	$\text{Tülu 3 Safety}_{(\text{avg*})}$	

Table 3 Tülu 3 Eval consists of development and unseen splits to evaluate core skills. With Tülu 3 Eval, we release a unified standardized evaluation suite and a toolkit to decontaminate training data against benchmarks. The subscript shows the metric we use for evaluation. Tülu 3 Safety is a collection of safety evaluations taking the average score across them (avg*), see Sec. 7.2.1 for details.

Llama 3

Category	Benchmark	Llama 3 8B	Gemma 2 9B	Mistral 7B	Llama 3 70B	Mixtral 8x22B	GPT 3.5 Turbo	Llama 3 405B	Nemotron 4 340B	GPT-4 (0125)	GPT-4o	Claude 3.5 Sonnet
General	MMLU (5-shot)	69.4	**72.3**	61.1	**83.6**	76.9	70.7	87.3	82.6	85.1	89.1	**89.9**
	MMLU (0-shot, CoT)	**73.0**	72.3$^{\triangle}$	60.5	**86.0**	79.9	69.8	88.6	78.7$^{\triangleleft}$	85.4	**88.7**	88.3
	MMLU-Pro (5-shot, CoT)	**48.3**	–	36.9	**66.4**	56.3	49.2	73.3	62.7	64.8	74.0	**77.0**
	IFEval	**80.4**	73.6	57.6	**87.5**	72.7	69.9	**88.6**	85.1	84.3	85.6	88.0
Code	HumanEval (0-shot)	**72.6**	54.3	40.2	**80.5**	75.6	68.0	89.0	73.2	86.6	90.2	**92.0**
	MBPP EvalPlus (0-shot)	**72.8**	71.7	49.5	**86.0**	78.6	82.0	88.6	72.8	83.6	87.8	**90.5**
Math	GSM8K (8-shot, CoT)	**84.5**	76.7	53.2	**95.1**	88.2	81.6	**96.8**	92.3$^{\diamond}$	94.2	96.1	96.4$^{\diamond}$
	MATH (0-shot, CoT)	**51.9**	44.3	13.0	**68.0**	54.1	43.1	73.8	41.1	64.5	**76.6**	71.1
Reasoning	ARC Challenge (0-shot)	83.4	**87.6**	74.2	**94.8**	88.7	83.7	**96.9**	94.6	96.4	96.7	96.7
	GPQA (0-shot, CoT)	32.8	–	28.8	**46.7**	33.3	30.8	51.1	–	41.4	53.6	**59.4**
Tool use	BFCL	**76.1**	–	60.4	84.8	–	**85.9**	88.5	86.5	88.3	80.5	**90.2**
	Nexus	**38.5**	30.0	24.7	**56.7**	48.5	37.2	**58.7**	–	50.3	56.1	45.7
Long context	ZeroSCROLLS/QuALITY	81.0	–	–	90.5	–	–	**95.2**	–	**95.2**	90.5	90.5
	InfiniteBench/En.MC	65.1	–	–	78.2	–	–	**83.4**	–	72.1	82.5	–
	NIH/Multi-needle	98.8	–	–	97.5	–	–	98.1	–	**100.0**	**100.0**	90.8
Multilingual	MGSM (0-shot, CoT)	**68.9**	53.2	29.9	**86.9**	71.1	51.4	**91.6**	–	85.9	90.5	**91.6**

Table 2 Performance of finetuned Llama 3 models on key benchmark evaluations. The table compares the performance of the 8B, 70B, and 405B versions of Llama 3 with that of competing models. We **boldface** the best-performing model in each of three model-size equivalence classes. $^{\triangle}$Results obtained using 5-shot prompting (no CoT). $^{\triangleleft}$Results obtained without CoT. $^{\diamond}$Results obtained using zero-shot prompting.

Proponent: Tülu 3

Discussion: Model capability evaluation

2. How to evaluate these capabilities?
Look at model before 2025

Llama 3

Category	Benchmark	Llama 3 8B	Gemma 2 9B	Mistral 7B	Llama 3 70B	Mixtral 8x22B	GPT 3.5 Turbo	Llama 3 405B	Nemotron 4 340B	GPT-4 (0125)	GPT-4o	Claude 3.5 Sonnet
General	MMLU (5-shot)	69.4	**72.3**	61.1	**83.6**	76.9	70.7	87.3	82.6	85.1	89.1	**89.9**
	MMLU (0-shot, CoT)	**73.0**	72.3△	60.5	**86.0**	79.9	69.8	88.6	78.7◁	85.4	**88.7**	88.3
	MMLU-Pro (5-shot, CoT)	**48.3**	–	36.9	**66.4**	56.3	49.2	73.3	62.7	64.8	74.0	**77.0**
	IFEval	**80.4**	73.6	57.6	**87.5**	72.7	69.9	**88.6**	85.1	84.3	85.6	88.0
Code	HumanEval (0-shot)	**72.6**	54.3	40.2	**80.5**	75.6	68.0	89.0	73.2	86.6	90.2	**92.0**
	MBPP EvalPlus (0-shot)	**72.8**	71.7	49.5	**86.0**	78.6	82.0	88.6	72.8	83.6	87.8	**90.5**
Math	GSM8K (8-shot, CoT)	**84.5**	76.7	53.2	**95.1**	88.2	81.6	**96.8**	92.3◇	94.2	96.1	96.4◇
	MATH (0-shot, CoT)	**51.9**	44.3	13.0	**68.0**	54.1	43.1	73.8	41.1	64.5	**76.6**	71.1
Reasoning	ARC Challenge (0-shot)	83.4	**87.6**	74.2	**94.8**	88.7	83.7	**96.9**	94.6	96.4	96.7	96.7
	GPQA (0-shot, CoT)	32.8	–	28.8	**46.7**	33.3	30.8	51.1	–	41.4	53.6	**59.4**
Tool use	BFCL	**76.1**	–	60.4	84.8	–	**85.9**	88.5	86.5	88.3	80.5	**90.2**
	Nexus	**38.5**	30.0	24.7	**56.7**	48.5	37.2	**58.7**	–	50.3	56.1	45.7
Long context	ZeroSCROLLS/QuALITY	81.0	–	–	90.5	–	–	**95.2**	–	**95.2**	90.5	90.5
	InfiniteBench/En.MC	65.1	–	–	78.2	–	–	**83.4**	–	72.1	82.5	–
	NIH/Multi-needle	98.8	–	–	97.5	–	–	98.1	–	**100.0**	**100.0**	90.8
Multilingual	MGSM (0-shot, CoT)	**68.9**	53.2	29.9	**86.9**	71.1	51.4	**91.6**	–	85.9	90.5	**91.6**

Table 2 Performance of finetuned Llama 3 models on key benchmark evaluations. The table compares the performance of the 8B, 70B, and 405B versions of Llama 3 with that of competing models. We **boldface** the best-performing model in each of three model-size equivalence classes. △Results obtained using 5-shot prompting (no CoT). ◁Results obtained without CoT. ◇Results obtained using zero-shot prompting.

Qwen 2.5

Table 4: **Performance of the 7B+ base models.**

Datasets	Mistral-7B	Llama3-8B	Gemma2-9B	Qwen2-7B	Qwen2.5-7B
General Tasks					
MMLU	64.2	66.6	71.3	70.3	**74.2**
MMLU-pro	30.9	35.4	44.7	40.1	**45.0**
MMLU-redux	58.1	61.6	67.9	68.1	**71.1**
BBH	56.1	57.7	68.2	62.3	**70.4**
ARC-C	60.0	59.3	**68.2**	60.6	63.7
TruthfulQA	42.2	44.0	45.3	54.2	**56.4**
Winogrande	78.4	77.4	**79.5**	77.0	75.9
HellaSwag	**83.3**	82.1	81.9	80.7	80.2
Mathematics & Science Tasks					
GPQA	24.7	25.8	32.8	30.8	**36.4**
TheoremQA	19.2	22.1	28.9	29.6	**36.0**
MATH	10.2	20.5	37.7	43.5	**49.8**
MMLU-stem	50.1	55.3	65.1	64.2	**72.3**
GSM8K	36.2	55.3	70.7	80.2	**85.4**
Coding Tasks					
HumanEval	29.3	33.5	37.8	51.2	**57.9**
HumanEval+	24.4	29.3	30.5	43.3	**50.6**
MBPP	51.1	53.9	62.2	64.2	**74.9**
MBPP+	40.9	44.4	50.6	51.9	**62.9**
MultiPL-E	29.4	22.6	34.9	41.0	**50.3**
Multilingual Tasks					
Multi-Exam	47.1	52.3	**61.2**	59.2	59.4
Multi-Understanding	63.3	68.6	78.3	72.0	**79.3**
Multi-Mathematics	26.3	36.3	53.0	57.5	**57.8**
Multi-Translation	23.3	31.9	**36.5**	31.5	32.4

Proponent: Tülu 3

Discussion: Model capability evaluation

2. How to evaluate these capabilities? Look at model before 2025

Claude 3.5 & GPT-4o & Gemini 1.5 pro

	Claude 3.5 Sonnet	Claude 3 Opus	GPT-4o	Gemini 1.5 Pro	Llama-400b (early snapshot)
Graduate level reasoning *GPQA, Diamond*	59.4%* 0-shot CoT	50.4% 0-shot CoT	53.6% 0-shot CoT	–	–
Undergraduate level knowledge *MMLU*	88.7%** 5-shot 88.3% 0-shot CoT	86.8% 5-shot 85.7% 0-shot CoT	– 88.7% 0-shot CoT	85.9% 5-shot –	86.1% 5-shot –
Code *HumanEval*	92.0% 0-shot	84.9% 0-shot	90.2% 0-shot	84.1% 0-shot	84.1% 0-shot
Multilingual math *MGSM*	91.6% 0-shot CoT	90.7% 0-shot CoT	90.5% 0-shot CoT	87.5% 8-shot	–
Reasoning over text *DROP, F1 score*	87.1 3-shot	83.1 3-shot	83.4 3-shot	74.9 Variable shots	83.5 3-shot Pre-trained model
Mixed evaluations *BIG-Bench-Hard*	93.1% 3-shot CoT	86.8% 3-shot CoT	–	89.2% 3-shot CoT	85.3% 3-shot CoT Pre-trained model
Math problem-solving *MATH*	71.1% 0-shot CoT	60.1% 0-shot CoT	76.6% 0-shot CoT	67.7% 4-shot	57.8% 4-shot CoT
Grade school math *GSM8K*	96.4% 0-shot CoT	95.0% 0-shot CoT	–	90.8% 11-shot	94.1% 8-shot CoT

* Claude 3.5 Sonnet scores 67.2% on 5-shot CoT GPQA with maj@32
** Claude 3.5 Sonnet scores 90.4% on MMLU with 5-shot CoT prompting

AIME was not considered a standard math benchmark in 2024

What changed the game?
DeepSeek-r1? RLVR?

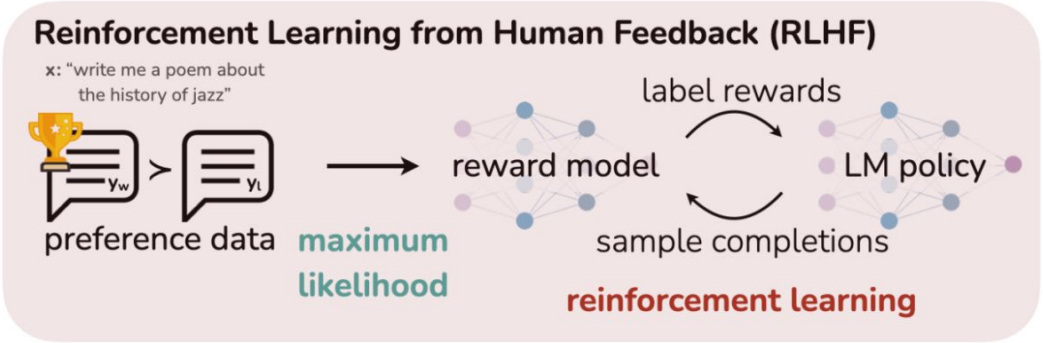
Post-Training Data: Follow-Up

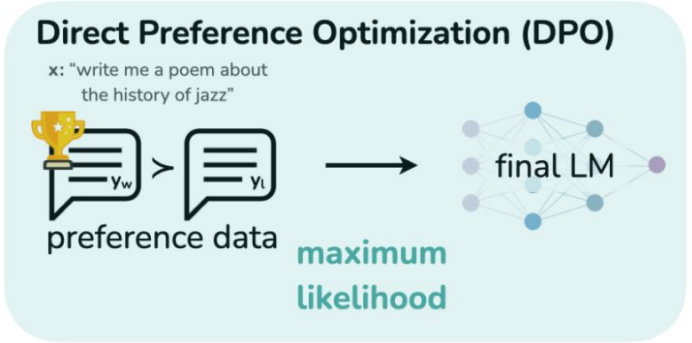
Direct Preference Optimization:
Your Language Model is Secretly a Reward Model

Sanjay Adhikesaven
09/11

Why DPO?

- DPO has the same KL-regularized objective as RLHF *without* training a reward model or running PPO
- Often cheaper and more stable to run due to simple supervised loss on human preference data





DPO: Core Ideas

- RL Objective: $\max_{\pi_\theta} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi_\theta(y|x)}\left[r_\phi(x,y)\right] - \beta\mathbb{D}_{\text{KL}}\left[\pi_\theta(y \mid x) \,||\, \pi_{\text{ref}}(y \mid x)\right]$
  - LM must pick responses that are highly preferred while not drifting away from a baseline model (typically SFT model)
- Closed Form Solution: $\pi_r(y \mid x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y \mid x)\exp\left(\frac{1}{\beta}r(x,y)\right)$
- Bradley-Terry Preference Model: $p^*(y_1 \succ y_2|x) = \frac{\exp\left(r^*(x,y_1)\right)}{\exp\left(r^*(x,y_1)\right)+\exp\left(r^*(x,y_2)\right)}$

$$\mathcal{L}_{\text{DPO}}(\pi_\theta;\pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

Same KL-regularized goal as RLHF, but optimized via supervised pairwise loss

DPO: Results

- DPO optimizes the same KL-regularized objective more efficiently than PPO (higher reward at equal KL)
- Higher win rates, less temperature sensitivity on TL;DR summarization

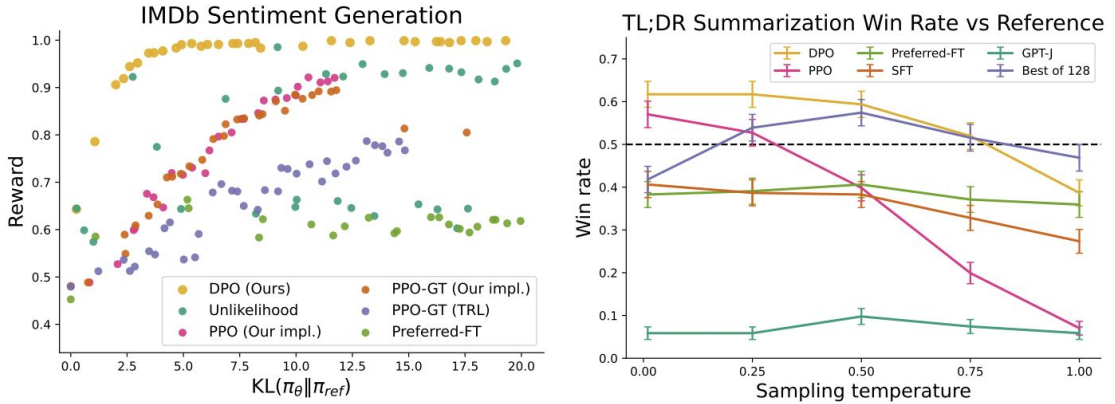


Figure 2: **Left.** The frontier of expected reward vs KL to the reference policy. DPO provides the highest expected reward for all KL values, demonstrating the quality of the optimization. **Right.** TL;DR summarization win rates vs. human-written summaries, using GPT-4 as evaluator. DPO exceeds PPO's best-case performance on summarization, while being more robust to changes in the sampling temperature.