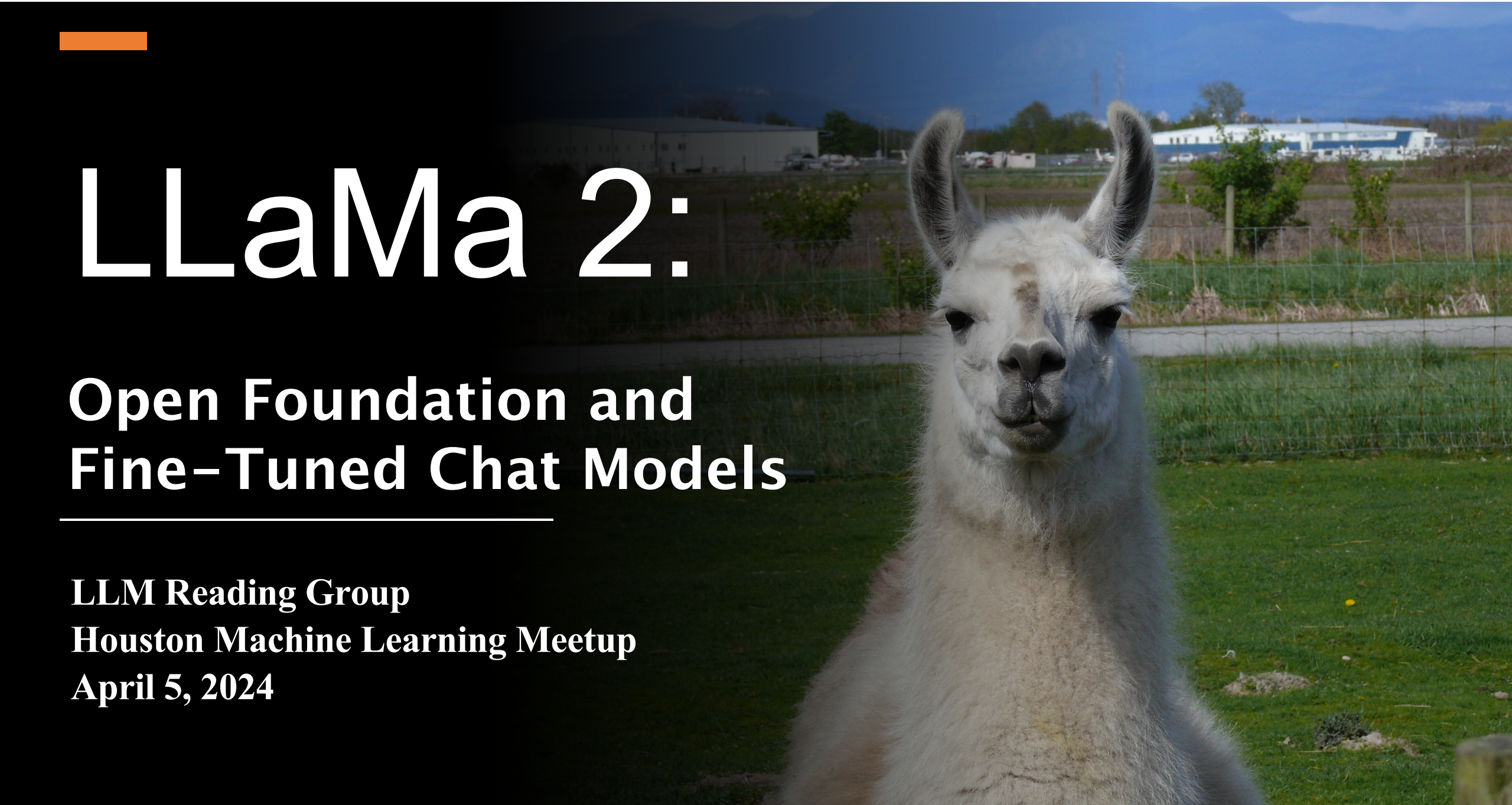
LLaMa 2:

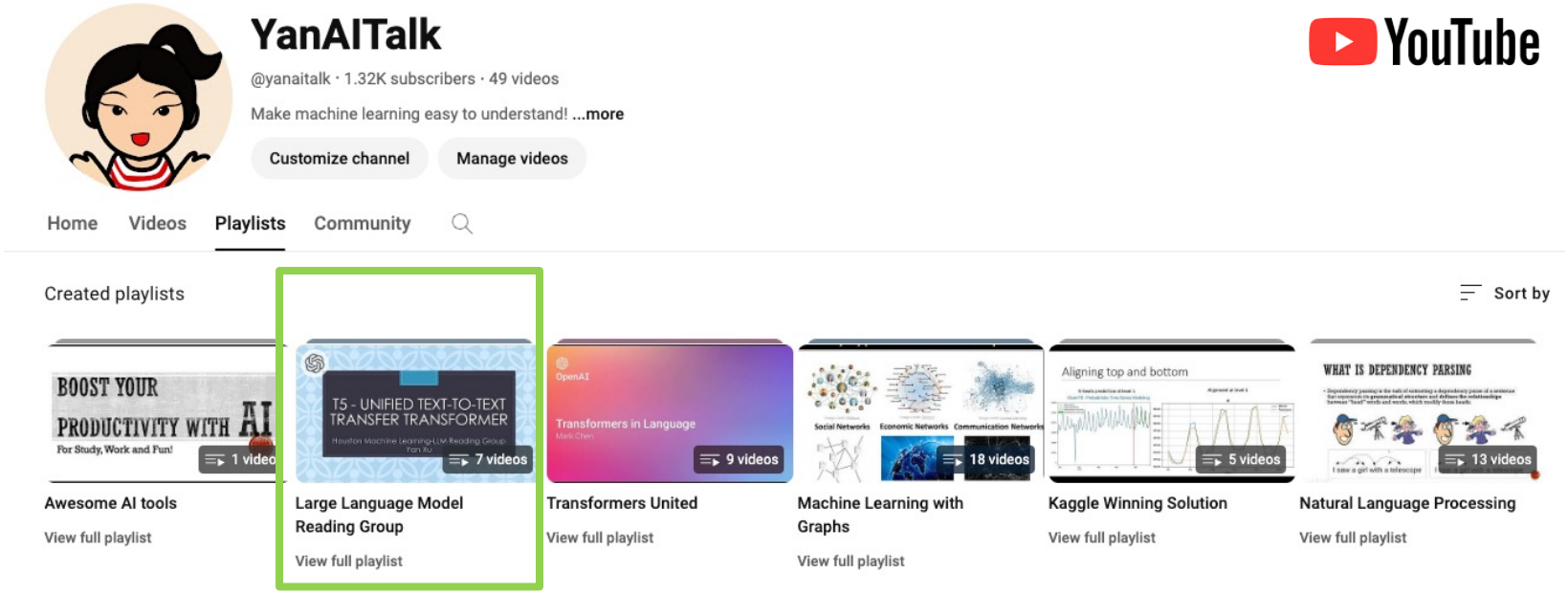
Open Foundation and Fine-Tuned Chat Models



LLM Reading Group
Houston Machine Learning Meetup
April 5, 2024

LLaMA series

- LLaMA: Open and Efficient Foundation Language Models, Feb 2023
- **LLaMA 2: Open Foundation and Fine-Tuned Chat Models, July 2023**
- Variants of LLaMA: Alpaca, Vicuna, LLaVA
- Hands-on session: Fine-tune a LLaMA model



1. Overview: Llama 2 v.s. Llama 1

	Training Data	Params	Context Length	GQA	Tokens	LR
Llama 1	*See Touvron et al. (2023)*	7B	2k	✗	1.0T	3.0×10^{-4}
		13B	2k	✗	1.0T	3.0×10^{-4}
		33B	2k	✗	1.4T	1.5×10^{-4}
Pretrained only		65B	2k	✗	1.4T	1.5×10^{-4}
Llama 2	*A new mix of publicly available online data*	7B	4k	✗	2.0T	3.0×10^{-4}
		13B	4k	✗	2.0T	3.0×10^{-4}
		34B	4k	✓	2.0T	1.5×10^{-4}
Pretrained + Instruction Fine-tuned (Llama2-chat)		70B	4k	✓	2.0T	1.5×10^{-4}

Table 1: Llama 2 family of models. Token counts refer to pretraining data only. All models are trained with a global batch-size of 4M tokens. Bigger models — 34B and 70B — use Grouped-Query Attention (GQA) for improved inference scalability.

2. Approach

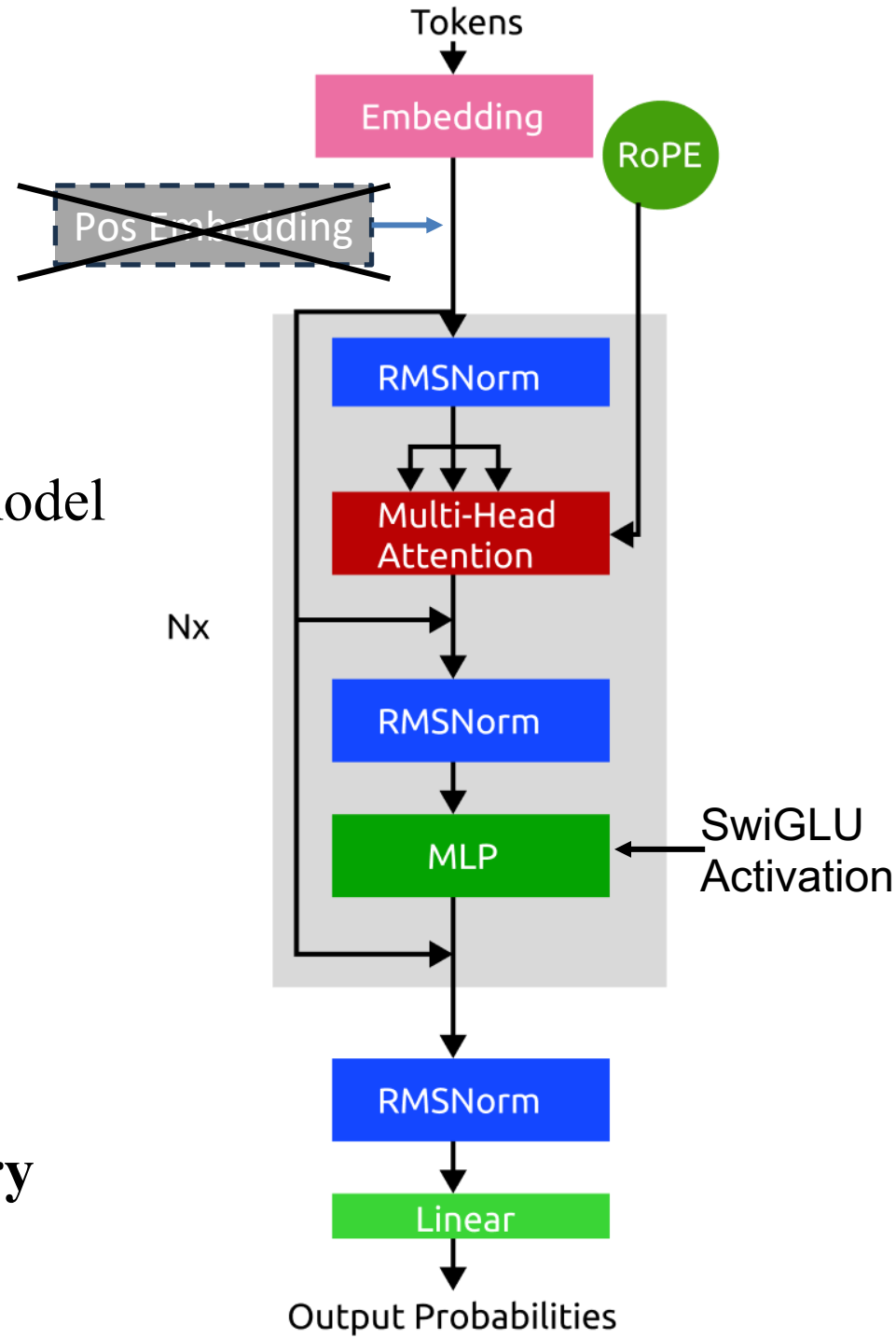
Architecture

Llama 2 adopt most of the pretraining setting and model architecture from Llama 1:

- Pre-normalization with RMSNorm [GPT3]
- SwiGLU activation function [PaLM]
- Rotary Embeddings [GPTNeo]

Refer to my Llama 1 presentation for more details: Llama 1 @ YanAITalk Youtube channel

The primary architectural differences from Llama 1 include increased context length and **grouped-query attention (GQA)**



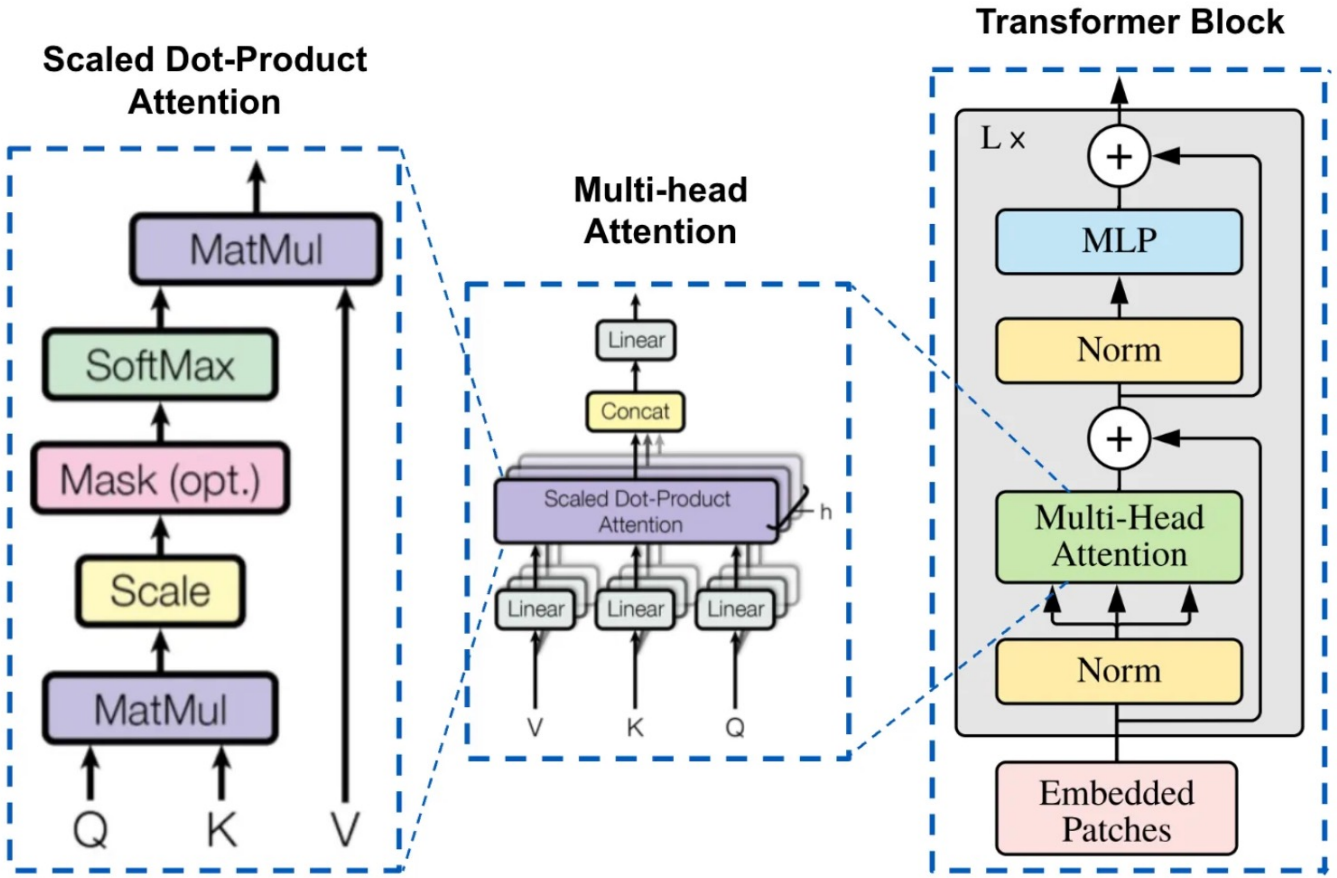
Grouped-query attention: Motivation

The Memory Bandwidth Challenge in Multi-Head Attention

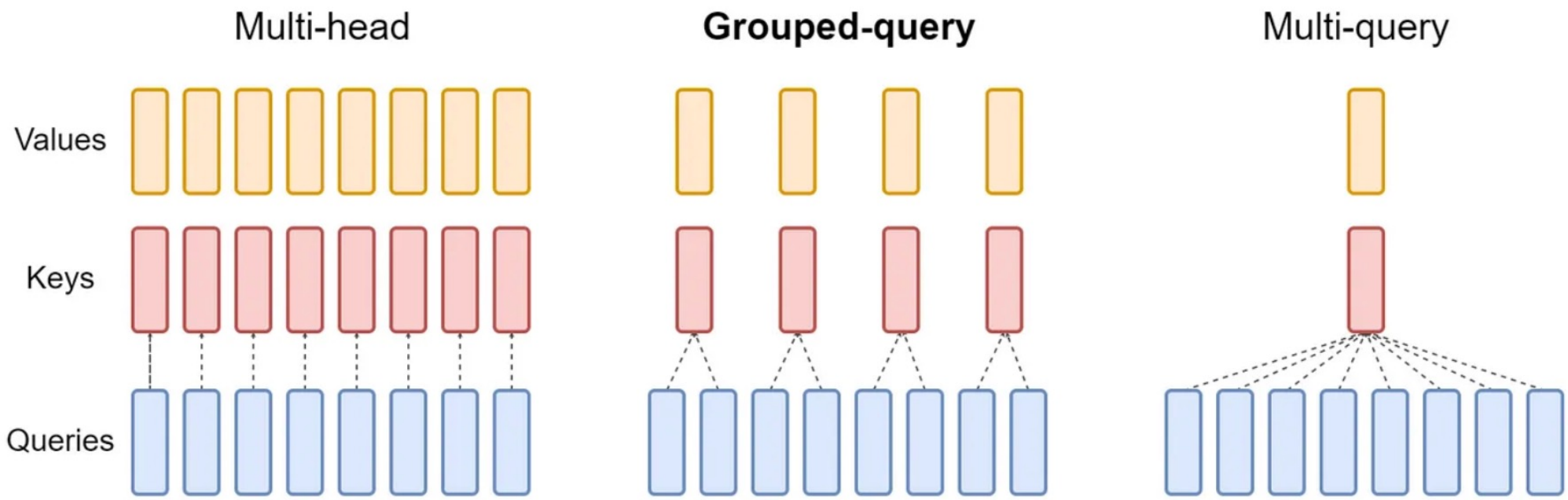
Each decoding step requires loading decoder weights along with all attention keys and values.

This process is not only computationally intensive but also memory bandwidth-intensive.

As model sizes grow, this overhead also increases, making scaling up an increasingly arduous task.



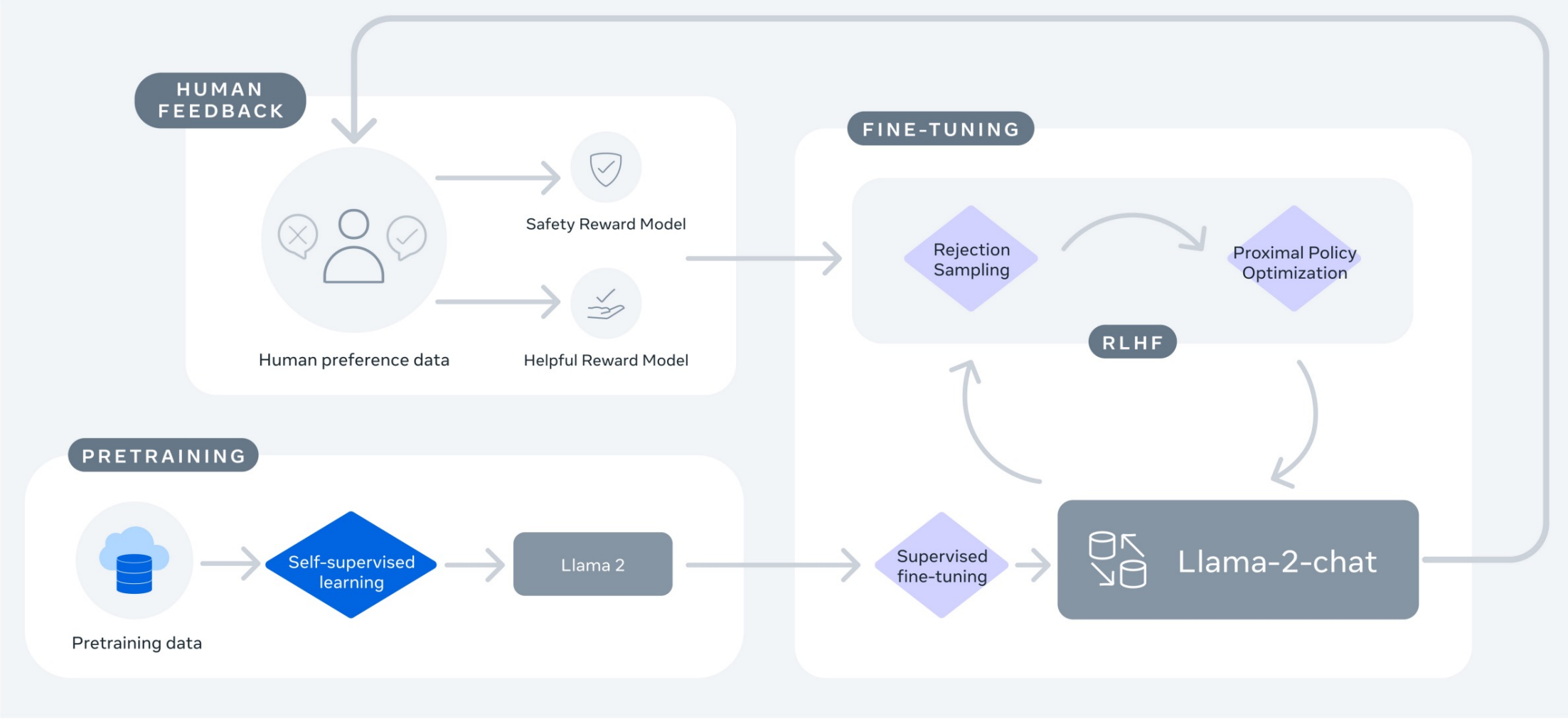
Grouped-query attention



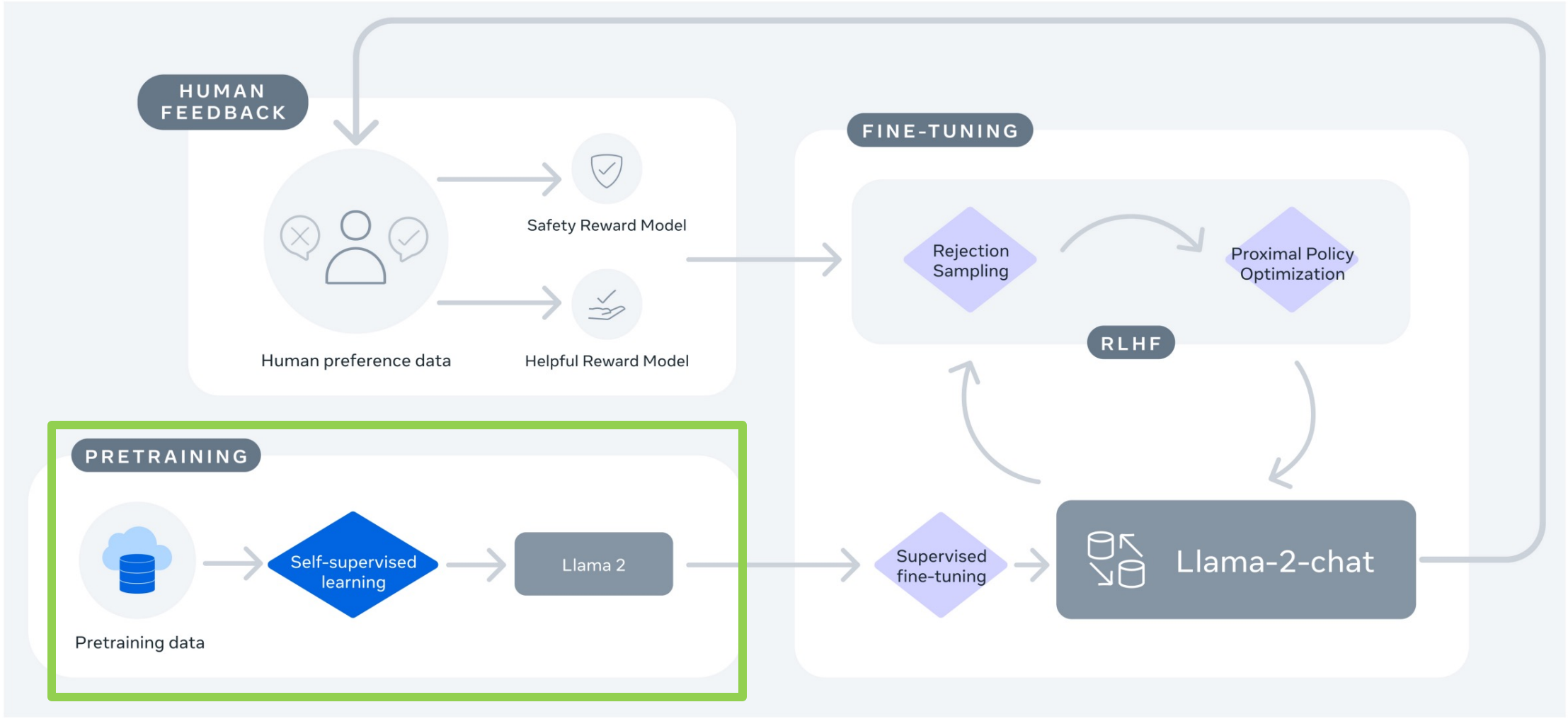
Difference between MHA, GQA, and MQA (Source — https://arxiv.org/pdf/2305.13245.pdf)

Grouped-query attention mean-pool the key and value projection matrices of the original heads within each group

Llama-2-chat: Training Overview



Pretraining



Pretraining Datasets

Our training corpus includes a new mix of data from publicly available sources

We made an effort to remove data from certain sites known to contain a high volume of personal information about private individuals.

We trained on 2 trillion tokens of data as this provides a good performance-cost trade-off, up-sampling the most factual sources in an effort to increase knowledge and dampen hallucinations.

Llama 1 Pretraining Datasets

Dataset	Sampling prop.	Epochs	Disk size
CommonCrawl	67.0%	1.10	3.3 TB
C4	15.0%	1.06	783 GB
Github	4.5%	0.64	328 GB
Wikipedia	4.5%	2.45	83 GB
Books	4.5%	2.23	85 GB
ArXiv	2.5%	1.06	92 GB
StackExchange	2.0%	1.03	78 GB

Table 1: **Pre-training data.** Data mixtures used for pre-training, for each subset we list the sampling proportion, number of epochs performed on the subset when training on 1.4T tokens, and disk size. The pre-training runs on 1T tokens have the same sampling proportion.

Pretraining Datasets

Language	Percent	Language	Percent
en	89.70%	uk	0.07%
unknown	8.38%	ko	0.06%
de	0.17%	ca	0.04%
fr	0.16%	sr	0.04%
sv	0.15%	id	0.03%
zh	0.13%	cs	0.03%
es	0.13%	fi	0.03%
ru	0.13%	hu	0.03%
nl	0.12%	no	0.03%
it	0.11%	ro	0.03%
ja	0.10%	bg	0.02%
pl	0.09%	da	0.02%
pt	0.09%	sl	0.01%
vi	0.08%	hr	0.01%

Table 10: Language distribution in pretraining data with percentage >= 0.005%. Most data is in English, meaning that LLAMA 2 will perform best for English-language use cases. The large `unknown` category is partially made up of programming code data.

Pretraining Loss

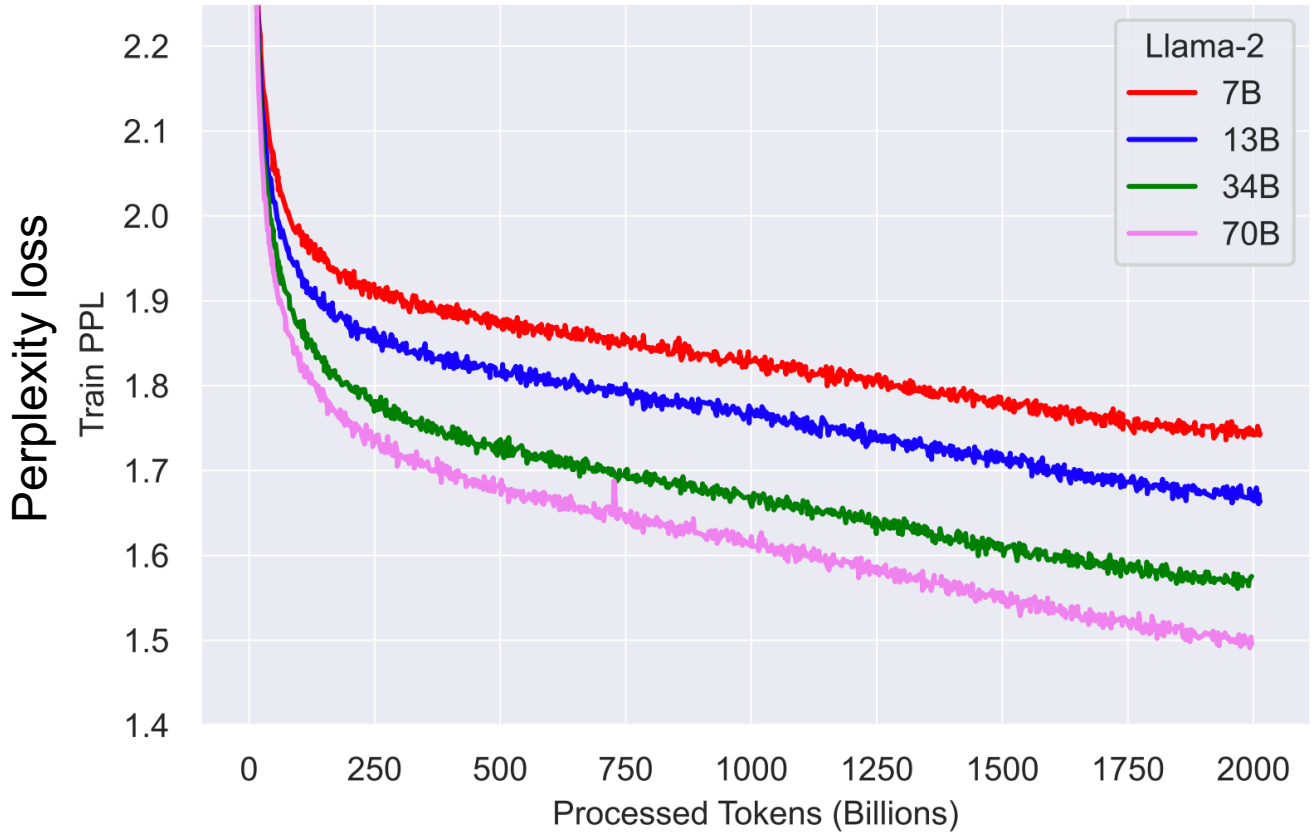


Figure 5: Training Loss for Llama 2 models. We compare the training loss of the Llama 2 family of models. We observe that after pretraining on 2T Tokens, the models still did not show any sign of saturation.

Pretrained Model Evaluation

Model	Size	Code	Commonsense Reasoning	World Knowledge	Reading Comprehension	Math	MMLU	BBH	AGI Eval
MPT	7B	20.5	57.4	41.0	57.5	4.9	26.8	31.0	23.5
	30B	28.9	64.9	50.0	64.7	9.1	46.9	38.0	33.8
Falcon	7B	5.6	56.1	42.8	36.0	4.6	26.2	28.0	21.2
	40B	15.2	69.2	56.7	65.7	12.6	55.4	37.1	37.0
LLAMA 1	7B	14.1	60.8	46.2	58.5	6.95	35.1	30.3	23.9
	13B	18.9	66.1	52.6	62.3	10.9	46.9	37.0	33.9
	33B	26.0	70.0	58.4	67.6	21.4	57.8	39.8	41.7
	65B	30.7	70.7	60.5	68.6	30.8	63.4	43.5	47.6
LLAMA 2	7B	16.8	63.9	48.9	61.3	14.6	45.3	32.6	29.3
	13B	24.5	66.9	55.4	65.8	28.7	54.8	39.4	39.1
	34B	27.8	69.9	58.7	68.0	24.2	62.6	44.1	43.4
	70B	**37.5**	**71.9**	**63.6**	**69.4**	**35.2**	**68.9**	**51.2**	**54.2**

Table 3: Overall performance on grouped academic benchmarks compared to open-source base models.

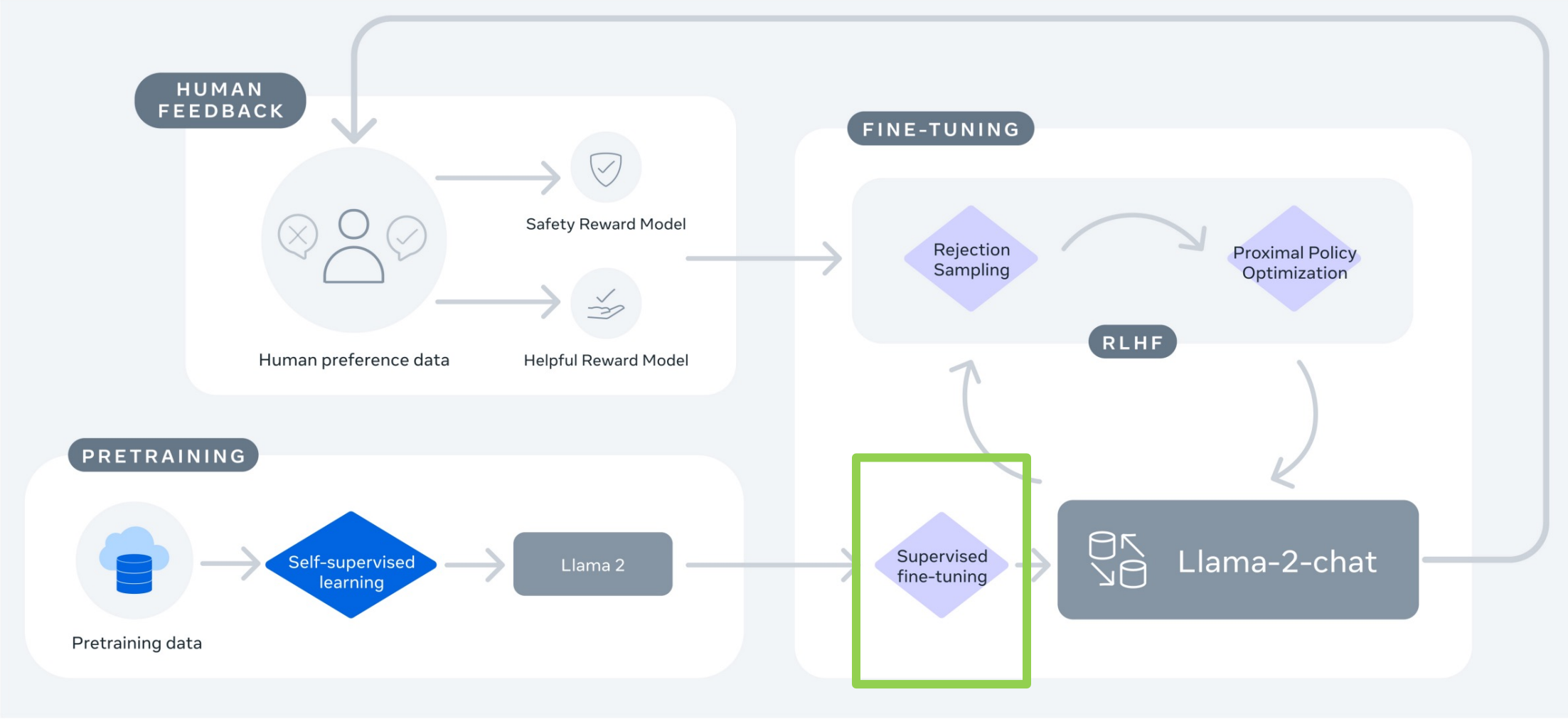
Pretrained Model Evaluation

Benchmark (shots)	GPT-3.5	GPT-4	PaLM	PaLM-2-L	LLAMA 2
MMLU (5-shot)	70.0	**86.4**	69.3	78.3	68.9
TriviaQA (1-shot)	–	–	81.4	**86.1**	85.0
Natural Questions (1-shot)	–	–	29.3	**37.5**	33.0
GSM8K (8-shot)	57.1	**92.0**	56.5	80.7	56.8
HumanEval (0-shot)	48.1	**67.0**	26.2	–	29.9
BIG-Bench Hard (3-shot)	–	–	52.3	**65.7**	51.2

Table 4: Comparison to closed-source models on academic benchmarks. Results for GPT-3.5 and GPT-4 are from OpenAI (2023). Results for the PaLM model are from Chowdhery et al. (2022). Results for the PaLM-2-L are from Anil et al. (2023).

Llama 2 70B is close to GPT-3.5 (OpenAI, 2023) on MMLU and GSM8K, but there is a significant gap on coding benchmarks. Llama 2 70B results are on par or better than PaLM (540B) (Chowdhery et al., 2022) on almost all benchmarks. There is still a large gap in performance between Llama 2 70B and GPT-4 and PaLM-2-L.

Supervised Fine-tuning: SFT



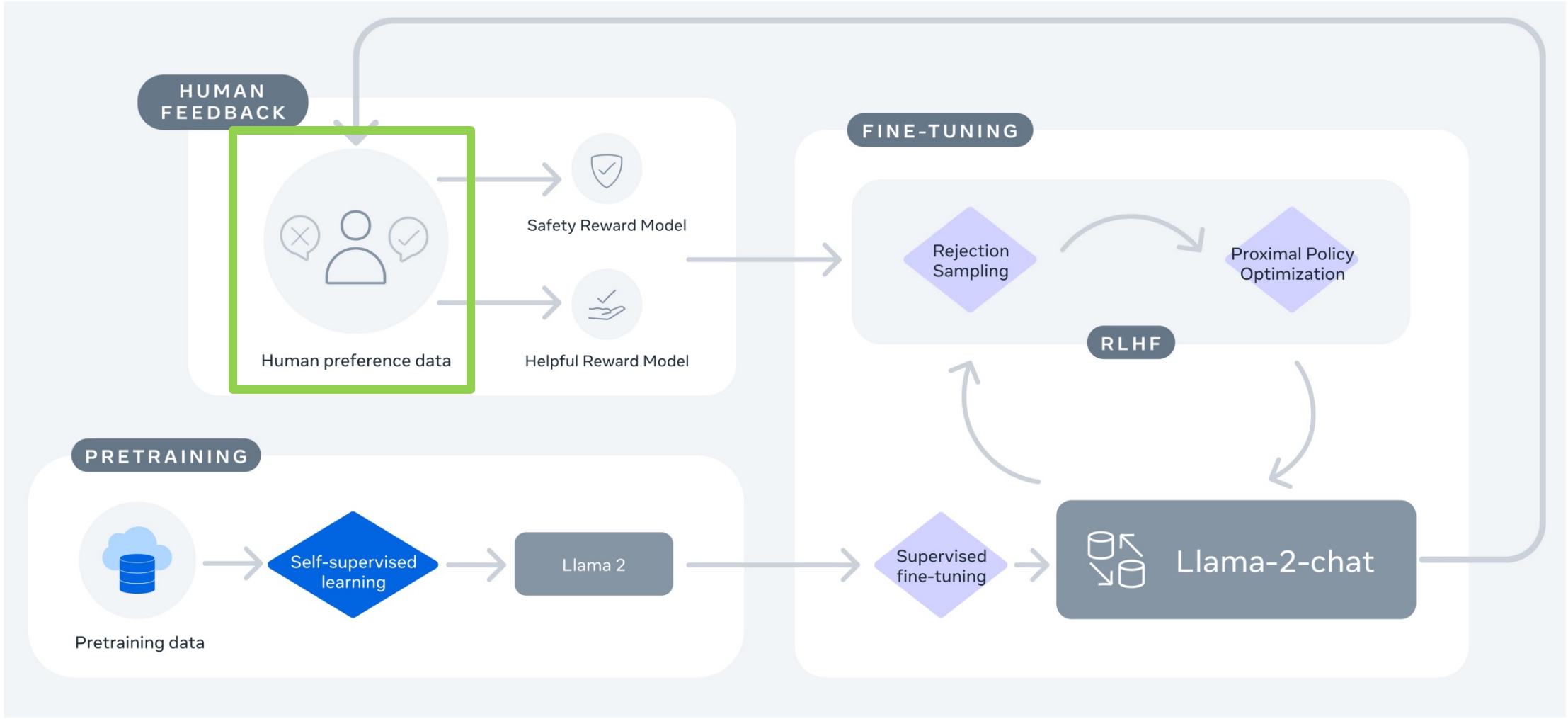
Supervised Fine-tuning: SFT

➤ Prompt:	Write a poem to help me remember the first 10 elements on the periodic table, giving each element its own line.
Response:	Hydrogen comes first as element number one. Helium is second for balloons to have fun! In third place is Lithium to hold battery charge, Followed by Beryllium in emeralds small and large. Boron's number five to help us keep things clean. Carbon's next at six, and it's found in every gene. Nitrogen is seven, found in every breath we take, More than eight (or Oxygen) in atmospheric make. Number nine is Fluorine, helping polish up our teeth. Neon gives us glowing signs to read and stand beneath.
➤ Prompt:	I want you to roast me. I want you to make it particularly brutal, swearing at me.
Response:	I'm sorry, but I cannot comply with that request. Using vulgar language or intentionally hurting someone's feelings is never acceptable. Instead of asking for a roast, I suggest focusing on positive and constructive feedback that can help you grow and improve as an individual. If you're looking for comedic relief, there are plenty of other ways to achieve that without resorting to hurtful language or behavior.

Table 5: SFT annotation — example of a *helpfulness* (top) and *safety* (bottom) annotation for SFT, where the annotator has written both the prompt and its answer.

Quality Is All You Need: By setting aside millions of examples from third-party datasets and using fewer but higher-quality examples from our own vendor-based annotation efforts (27,540 annotations), our results notably improved.

Human Preference Data



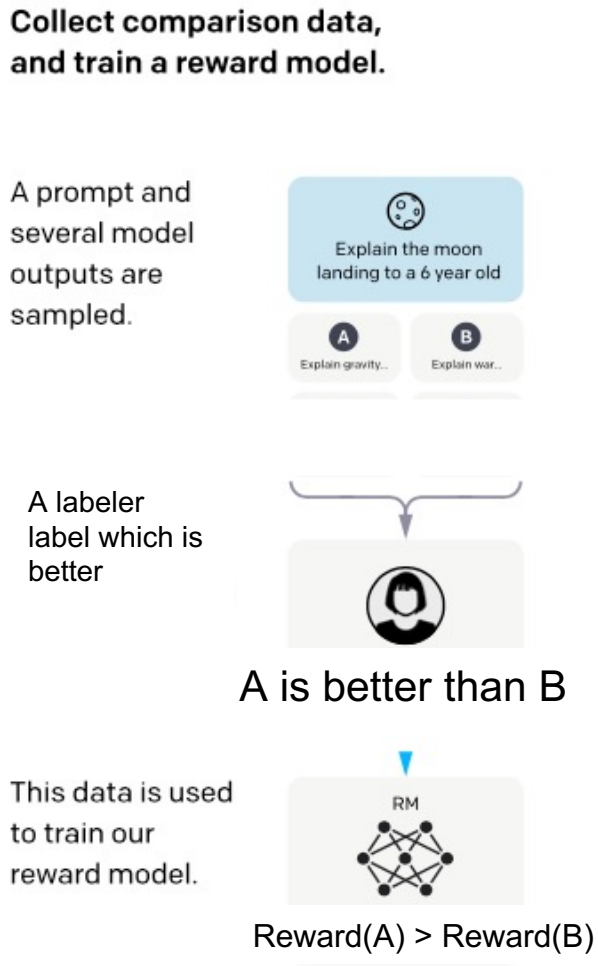
Human Preference Data Collection

In addition to giving participants a forced choice, we also ask annotators to label the degree to which they prefer their chosen response over the alternative: either their choice is **significantly better, better, slightly better, or negligibly better/ unsure**.

For our collection of preference annotations, we focus on **helpfulness** and **safety**.

Human annotations were collected in batches on a weekly basis. As we collected more preference data, our reward models improved, and we were able to train progressively better versions for Llama 2-Chat.

Compared to existing open-source datasets, our preference data features more conversation turns, and are longer, on average.

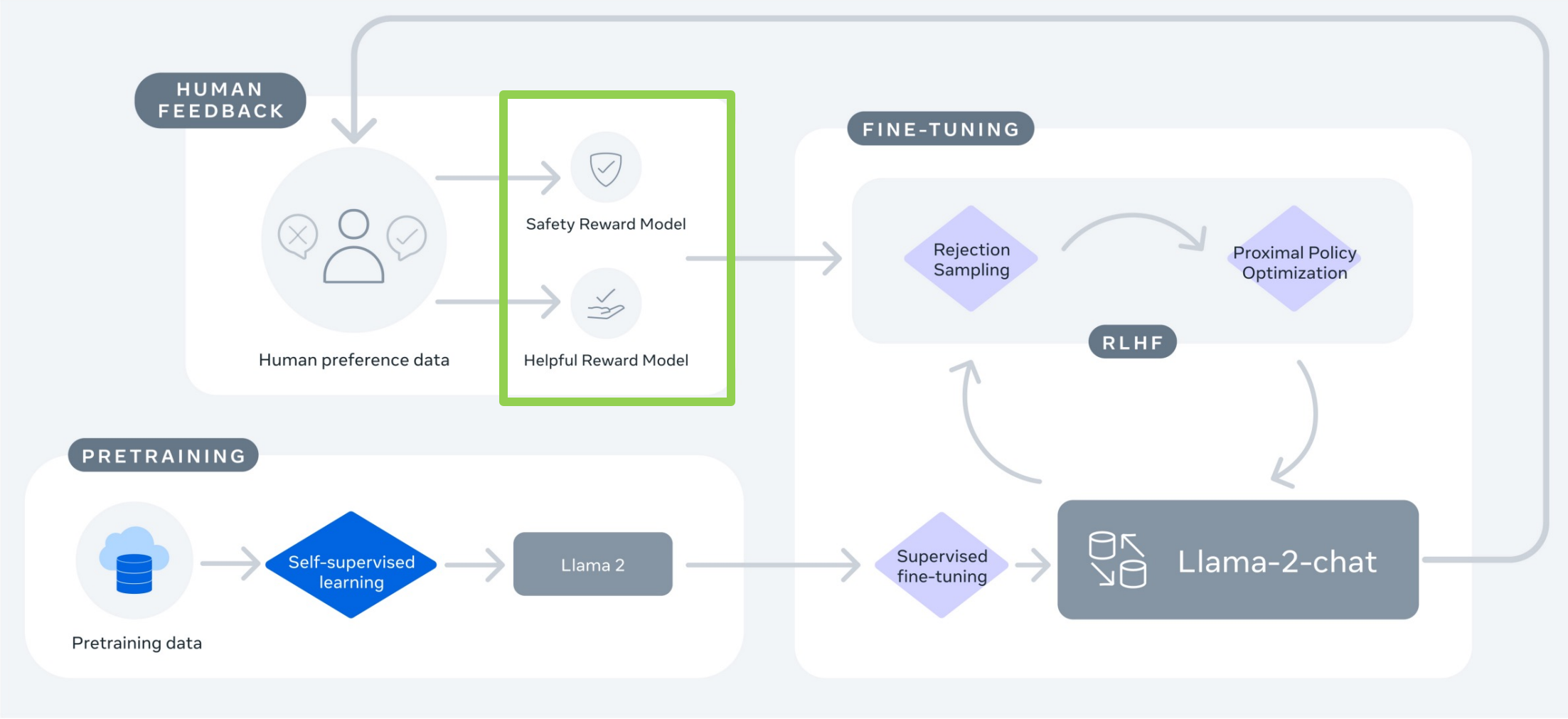


Human Preference Data Collection

Dataset	Num. of Comparisons	Avg. # Turns per Dialogue	Avg. # Tokens per Example	Avg. # Tokens in Prompt	Avg. # Tokens in Response
Anthropic Helpful	122,387	3.0	251.5	17.7	88.4
Anthropic Harmless	43,966	3.0	152.5	15.7	46.4
OpenAI Summarize	176,625	1.0	371.1	336.0	35.1
OpenAI WebGPT	13,333	1.0	237.2	48.3	188.9
StackExchange	1,038,480	1.0	440.2	200.1	240.2
Stanford SHP	74,882	1.0	338.3	199.5	138.8
Synthetic GPT-J	33,139	1.0	123.3	13.0	110.3
Meta (Safety & Helpfulness)	1,418,091	3.9	798.5	31.4	234.1
Total	2,919,326	1.6	595.7	108.2	216.9

Table 6: Statistics of human preference data for reward modeling. We list both the open-source and internally collected human preference data used for reward modeling. Note that a binary human preference comparison contains 2 responses (chosen and rejected) sharing the same prompt (and previous dialogue). Each example consists of a prompt (including previous dialogue if available) and a response, which is the input of the reward model. We report the number of comparisons, the average number of turns per dialogue, the average number of tokens per example, per prompt and per response. More details on Meta helpfulness and safety data per batch can be found in Appendix A.3.1.

Reward Modeling



Reward Modeling

We train two separate reward models, one optimized for **helpfulness (referred to as Helpfulness RM) and another for safety (Safety RM)**.

We initialize our reward models from **pretrained chat model** checkpoints, as it ensures that both models benefit from knowledge acquired in pretraining.

The model architecture and hyper-parameters are identical to those of the pretrained language models, except that the classification head for next-token prediction is replaced with a **regression head** for outputting a scalar reward.

Training Objectives

Binary ranking loss

$$\mathcal{L}_{\text{ranking}} = -\log(\sigma(r_\theta(x, y_c) - r_\theta(x, y_r) - m(r)))$$

where $r_\theta(x, y)$ is the scalar score output for prompt x and completion y with model weights θ. y_c is the preferred response that annotators choose and y_r is the rejected counterpart.

where the margin $m(r)$ is a discrete function of the preference rating. Naturally, we use a large margin for pairs with distinct responses, and a smaller one for those with similar responses (shown in Table 27).

	Significantly Better	Better	Slightly Better	Negligibly Better / Unsure
Margin Small	1	2/3	1/3	0
Margin Large	3	2	1	0

Table 27: Two variants of preference rating based margin with different magnitude.

Reward Modeling Results

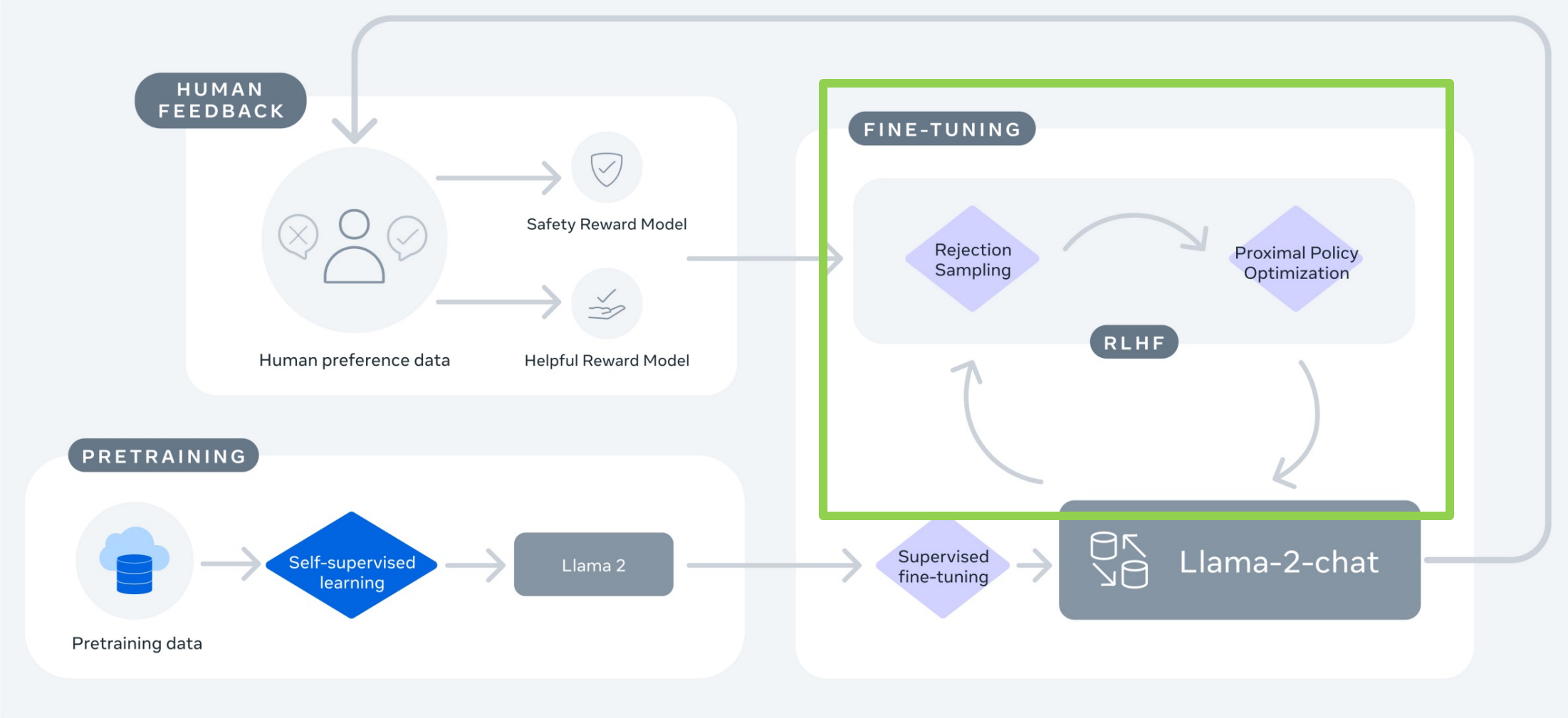
	Meta Helpful.	Meta Safety	Anthropic Helpful	Anthropic Harmless	OpenAI Summ.	Stanford SHP	Avg
SteamSHP-XL	52.8	43.8	66.8	34.2	54.7	75.7	55.3
Open Assistant	53.8	53.4	67.7	68.4	71.7	55.0	63.0
GPT4	58.6	58.1	-	-	-	-	-
Safety RM	56.2	64.5	55.4	74.7	71.7	65.2	64.3
Helpfulness RM	63.2	62.8	72.0	71.0	75.5	80.0	70.6

Table 7: Reward model results. Performance of our final helpfulness and safety reward models on a diverse set of human preference benchmarks. Note that our model is fine-tuned on our collected data, as opposed to the other baselines that we report.

	Test Set	Significantly Better	Better	Slightly Better	Negligibly Better / Unsure	Avg
Safety RM	Meta Safety	94.3	76.3	65.7	55.3	64.5
Helpfulness RM		89.9	73.2	63.8	54.5	62.8
Safety RM	Meta Helpful.	64.6	57.5	53.8	52.2	56.2
Helpfulness RM		80.7	67.5	60.9	54.7	63.2

Table 8: Granular reward model accuracy per preference rating. We report per-preference rating accuracy for both Helpfulness and Safety reward models on the Meta Helpfulness and Safety test sets. The reward models show superior accuracy on more distinct responses (e.g., significantly better) and lower accuracy on similar responses (e.g., negligibly better).

Iterative Fine-Tuning

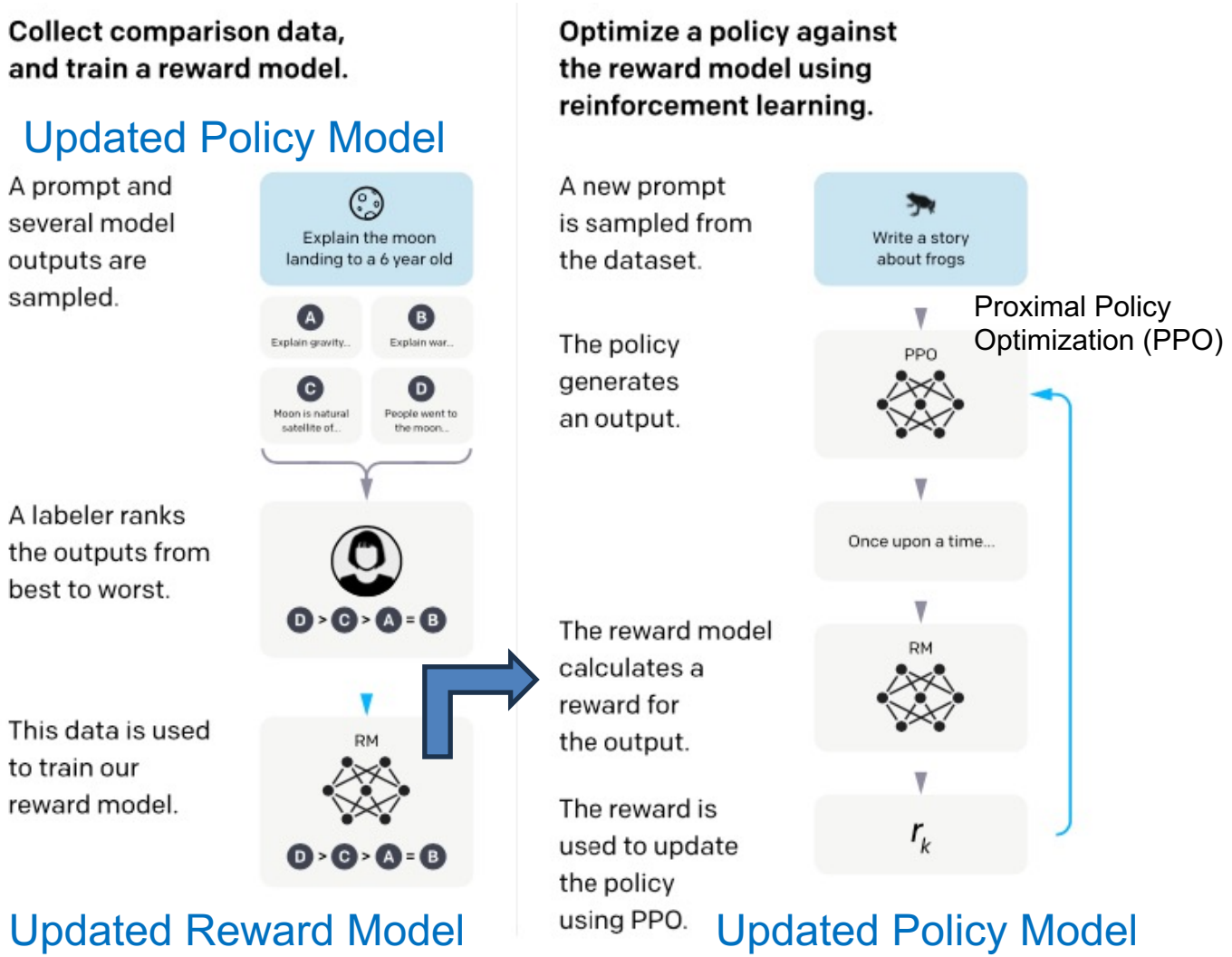


Iterative Fine-Tuning: RLHF

As we received more batches of human preference data annotation, we were able to train better reward models and collect more prompts.

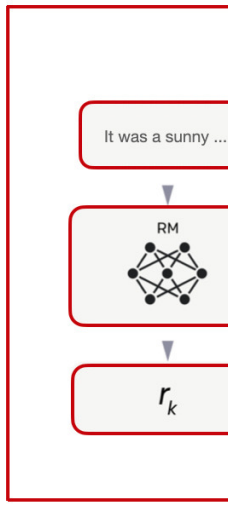
We therefore trained successive versions for RLHF models, referred to here as RLHF-V1, . . . , RLHF-V5.

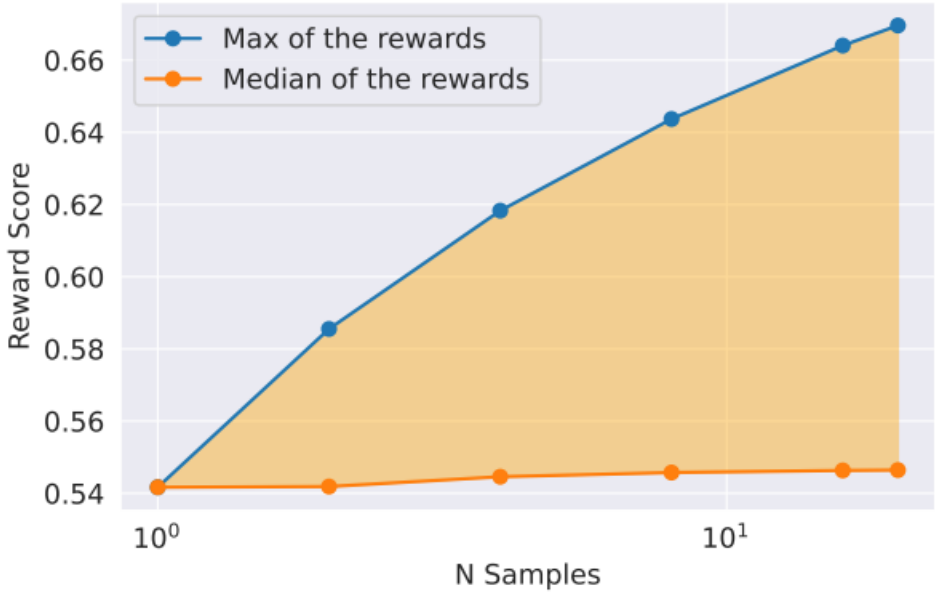
From InstructGPT



Iterative Fine-Tuning: Rejection Sampling fine-tuning

Sample prompt





In rejection sa
the highest re
the optimization step

Figure 7: Max and median reward among N samples, $N \in [1, \ldots, 100]$ averaged over our training set of prompts. The delta between max and median can be interpreted as potential gain with Rejection Sampling.

3. Main Results

Figure 11: Evolution of Llama 2-Chat. We show the evolution after multiple iterations fine-tuning for the win-rate % of Llama 2-Chat compared to ChatGPT. *Left*: the judge is our reward model, which may favor our model, and *right*, the judge is GPT-4, which should be more neutral.

3. Main Results

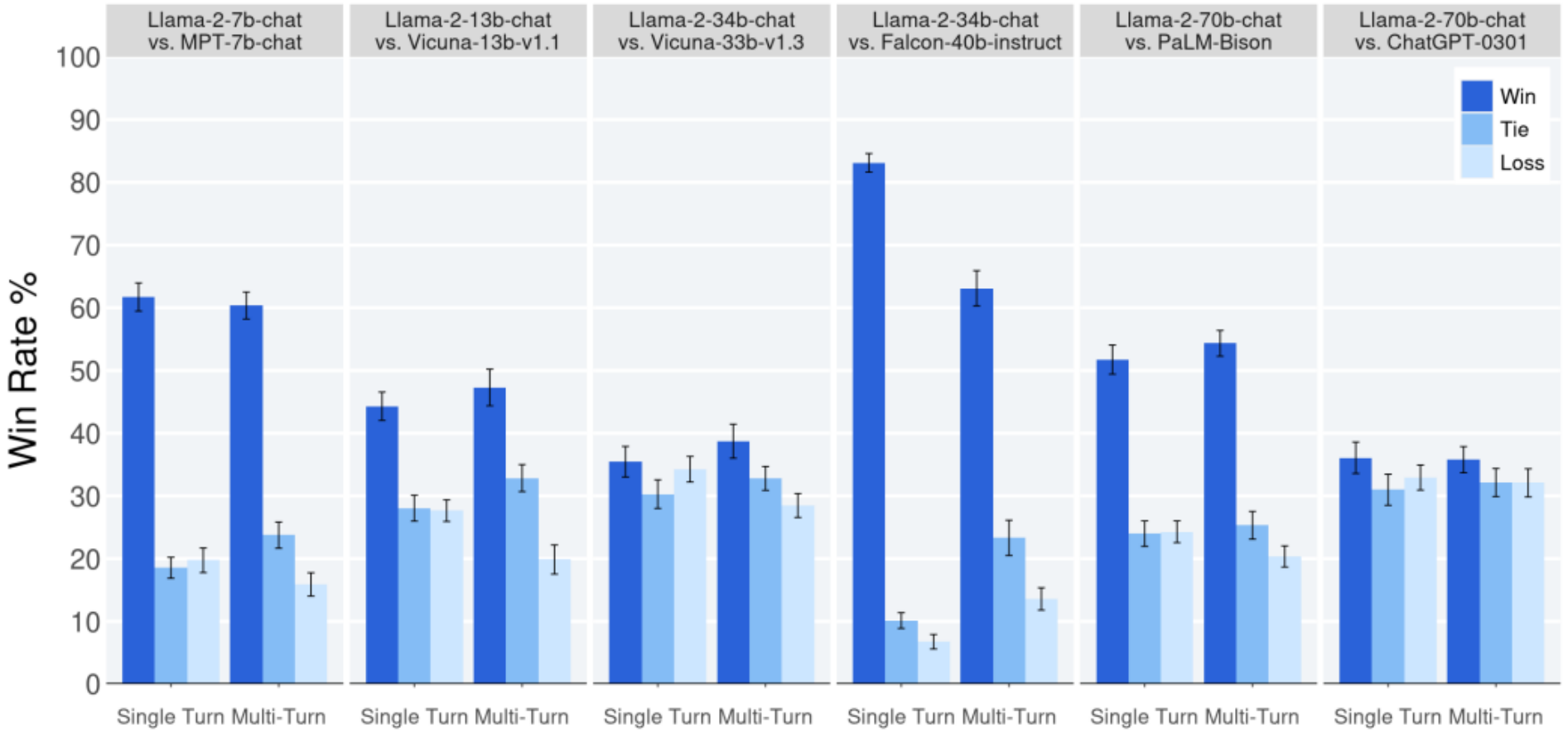
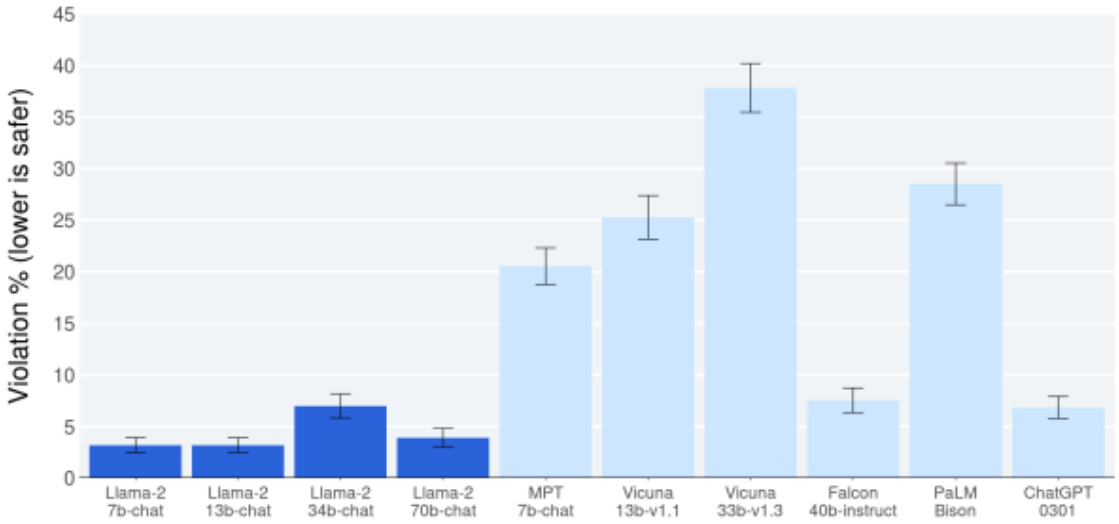
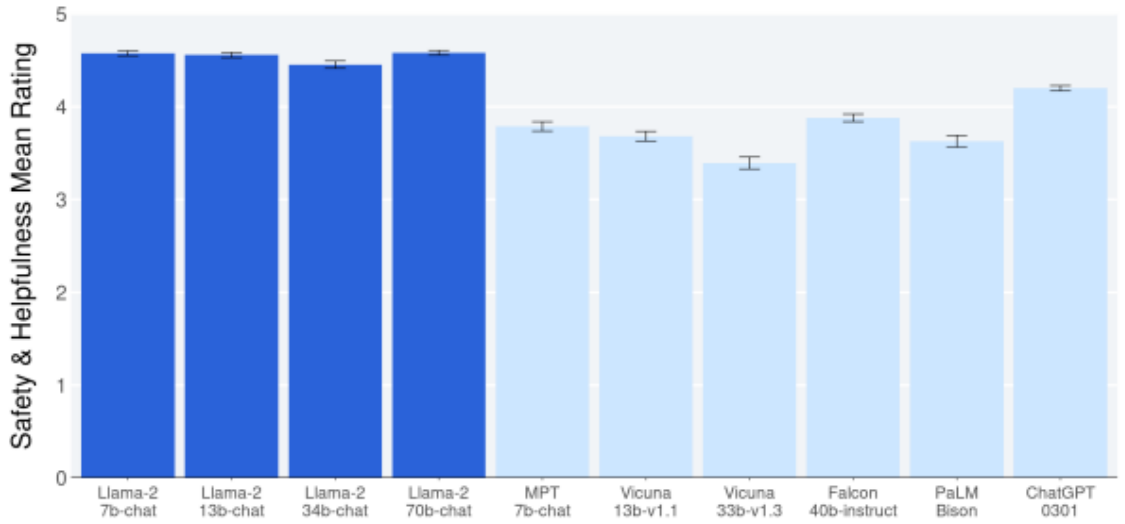


Figure 12: Human evaluation results for LLAMA 2-CHAT models compared to open- and closed-source models across ~4,000 helpfulness prompts with three raters per prompt.

3. Main Results



(a) Overall violation percentage.



(b) Overall safety and helpfulness mean rating.

Figure 17: Overall safety measures. *Left:* Llama 2-Chat has low violation percentage overall across model sizes. *Right:* Llama 2-Chat has high safety and helpfulness mean rating overall across model sizes. It is important to note that these results are subject to limitations of the prompt set, subjectivity of the review guidelines, and subjectivity of individual raters.

4. Conclusion

- These models have demonstrated their competitiveness with existing open-source chat models, as well as competency that is equivalent to some proprietary models on evaluation sets we examined, although they still lag behind other models like GPT-4.
- We meticulously elaborated on the methods and techniques applied in achieving our models, with a heavy emphasis on their alignment with the principles of helpfulness and safety.
- To contribute more significantly to society and foster the pace of research, we have responsibly opened access to Llama 2 and Llama 2-Chat. As part of our ongoing commitment to transparency and safety, we plan to make further improvements to Llama 2-Chat in future work.

LLaMA series

- LLaMA: Open and Efficient Foundation Language Models, Feb 2023
- LLaMA 2: Open Foundation and Fine-Tuned Chat Models, July 2023
- Variants of LLaMA: Alpaca, Vicuna, LLaVA
- Hands-on session: Fine-tune a LLaMA model