DATA 8005 Advanced Natural Language Processing

The Llama 3 Herd of Models: Pre-training

Yujia Zhang Zhiheng Liu

Fall 2024

Contents

Introduction

- Data
  - 15T multilingual tokens, compared to 1.8T tokens for Llama 2.
  - Development of more careful pre-processing and curation pipelines.
- Scale
  - 405B trainable parameters on 15.6T text tokens.
- Managing complexity
  - Standard dense Transformer model architecture.
  - Simple post-training procedure.

Introduction

- Performance

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**

General Overview

- Development of our Llama 3 language models
  - Language model pre-training: training data, architecture, training fra, details.
  - Language model post-training.
- Adding multi-modal capabilities to Llama 3
  - Multi-modal encoder pre-training.
  - Vision adapter training.
  - Speech adapter training.

Pre-Training Data

- Web Data Curation
  - PII and safety filtering: personally Identifiable Information.
  - Text extraction and cleaning: code, mathematical formulas, markdown.
  - De-duplication: URL, document, line.
  - Heuristic filtering: KL divergence.
  - Model-based quality filtering
  - Code and reasoning data.
  - Multilingual data

Pre-Training Data

- Determining the Data Mix
  - Knowledge classification.
  - Scaling laws for data mix.
- Annealing Data
  - Using annealing to assess data quality.

Model Architecture

- A few small modifications compared to Llama 2:
    - Grouped query attention.
    - Using an attention mask to prevent self-attention between different documents .
    - Vocabulary with 128K tokens.
    - It increases the RoPE base frequency hyperparameter to 500,000.

Model Architecture

- Overview of the key hyperparameters of Llama 3.

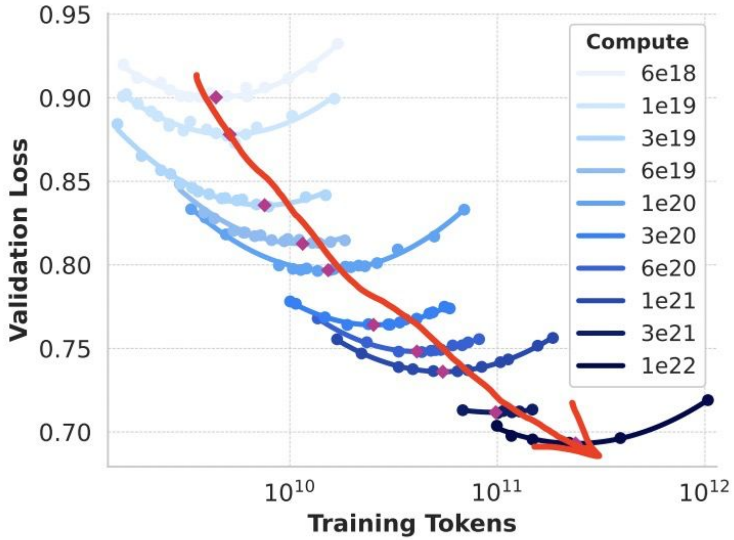
	8B	**70B**	**405B**
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Model Architecture

- Scaling Laws
  - Correlation between the compute-optimal model's negative log-likelihood on downstream tasks and the training FLOPs.
  - Correlating the negative log-likelihood on downstream tasks with task accuracy.

Model Architecture

- Scaling Laws

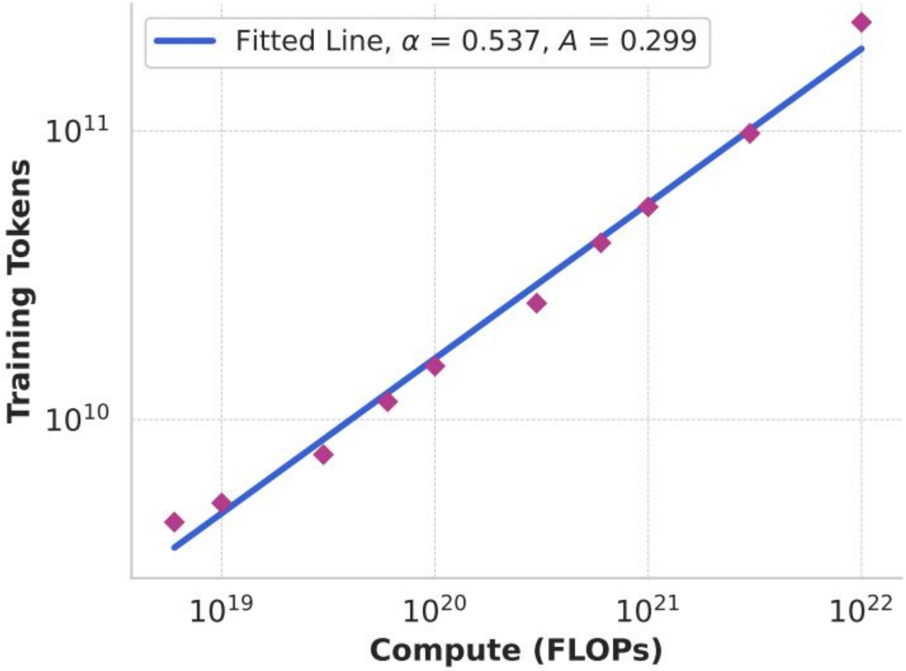


=

Scaling law IsoFLOPs curves

Model Architecture

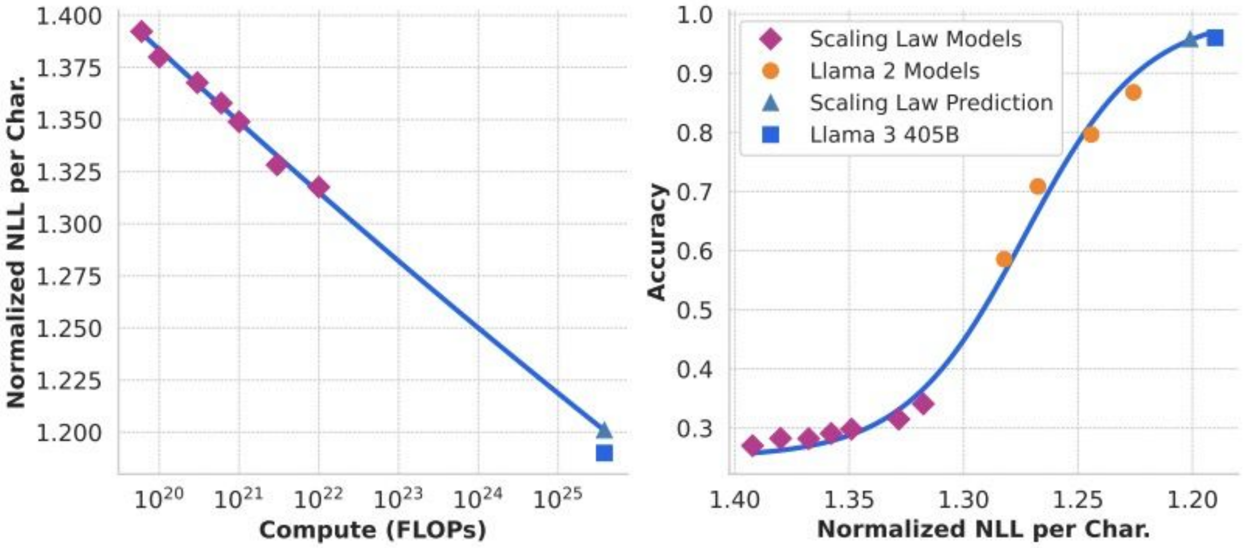
- Scaling Laws



=

Model Architecture

- Scaling Laws



=

Infrastructure, Scaling, and Efficiency

- Training Infrastructure
- Parallelism for Model Scaling
- Collective Communication
- Reliability and Operational Challenges

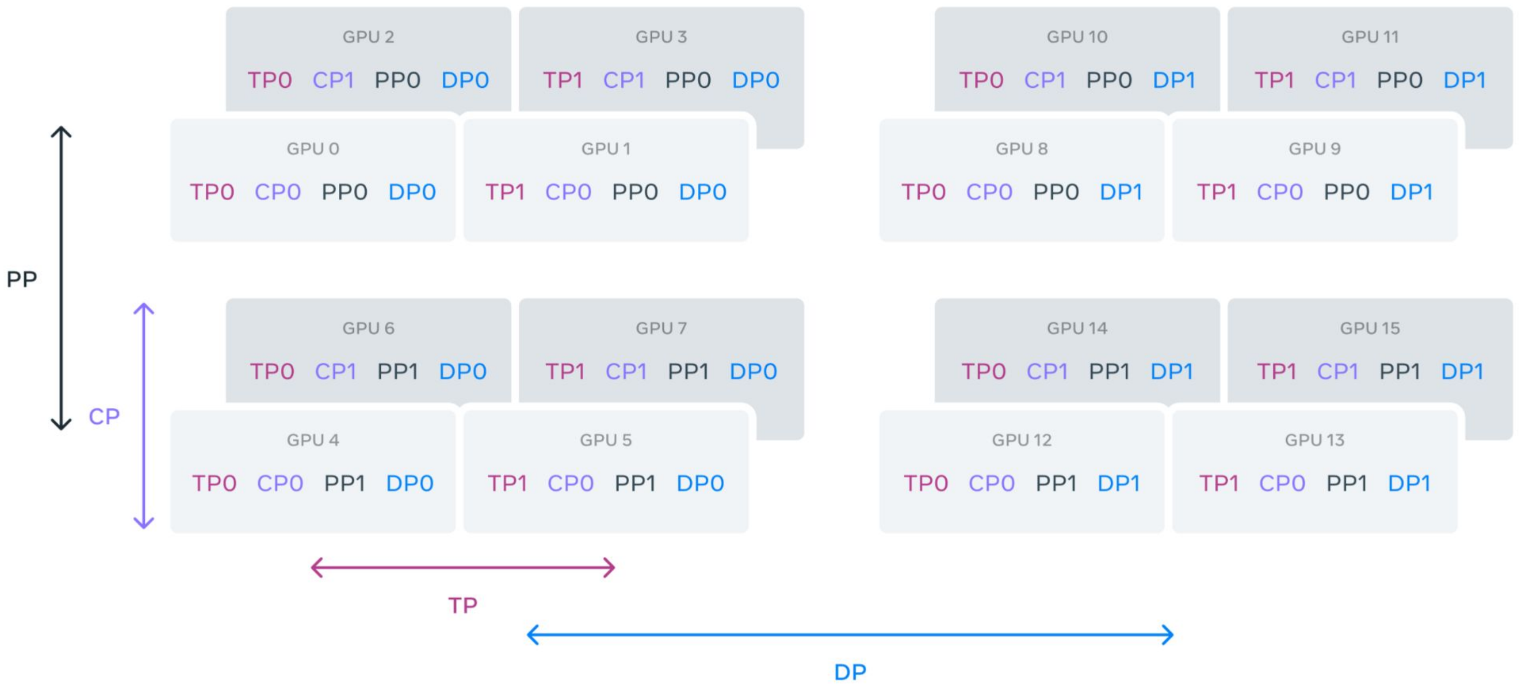
Infrastructure, Scaling, and Efficiency

Training Infrastructure

- Compute: 16K H100, scheduled using MAST
- Storage: 240PB, 7500 servers with SSDs, 2TB/s(peak 7TB/s)
- Network: 405B RoCE(Ethernet), small models(Infiniband), 400Gbps
  - Network topology: 3layers, 24K GPUs(use 16K)
  - Load balancing: 16 network flows, Enhanced-ECMP
  - Congestion control: deep-buffer switches for congestion and slow servers

Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism [TP, CP, PP, DP]



Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

= model weight tensors

- Tensor Parallelism

Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

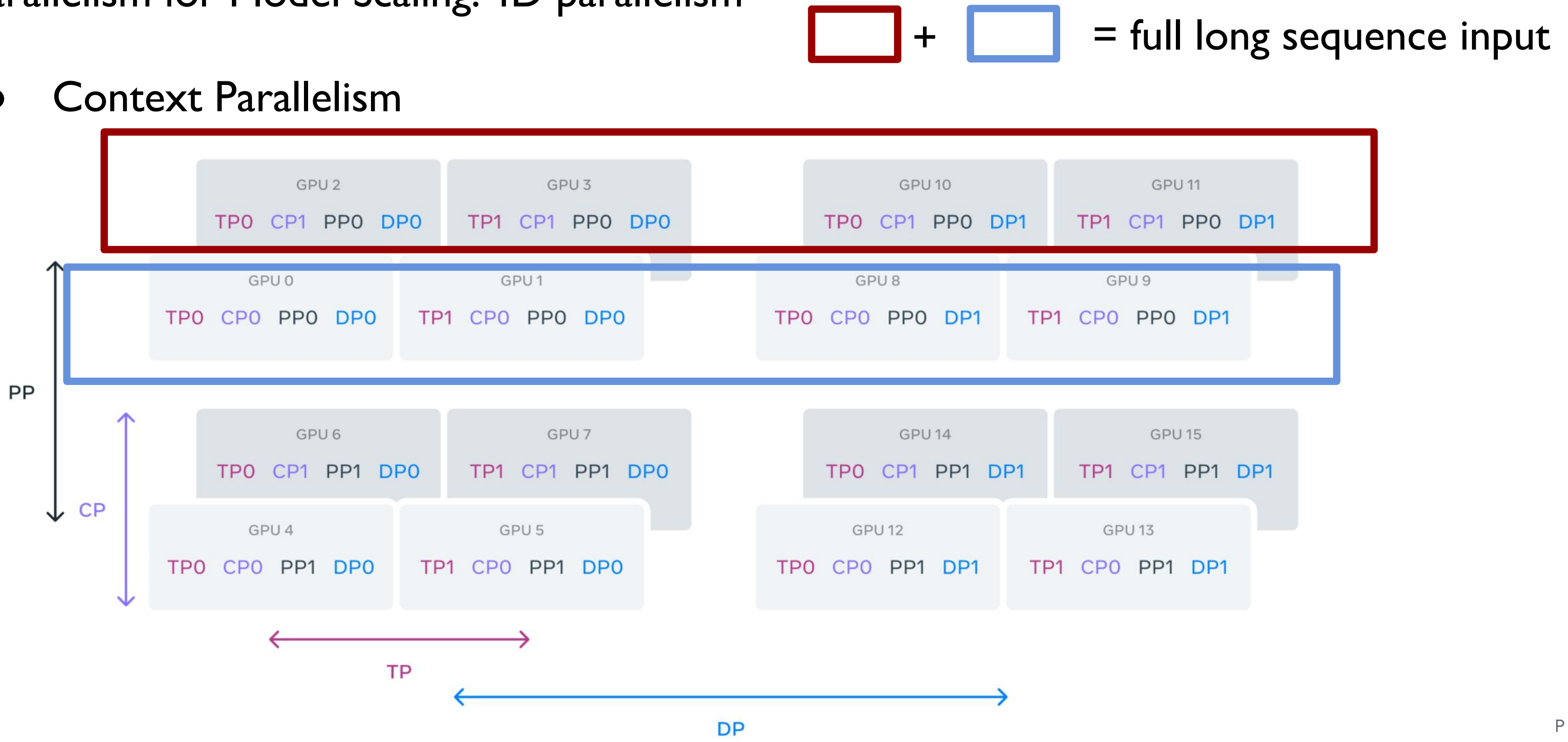
/ = a layer of weights

- Pipeline Parallelism

Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

- Context Parallelism



Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

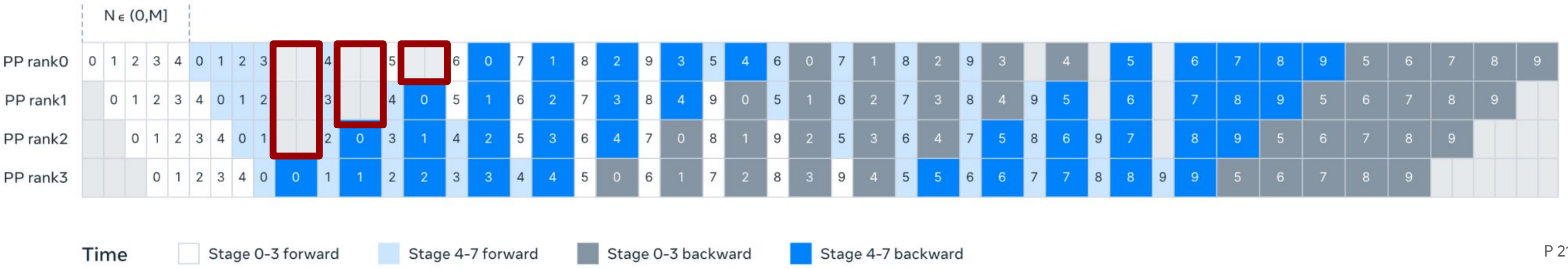
+ = data input one time

- Data Parallelism

Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

- Pipeline Parallelism: divided by layers, mini batch to micro batch
  - Batch size constraint: batch size divisible by the number of pipeline stages
  - Memory imbalance: the first stage consumes more memory for the embedding and the warm-up micro-batches
  - Computation imbalance: after the last layer, output and loss calculation



Infrastructure, Scaling, and Efficiency

Parallelism for Model Scaling: 4D parallelism

- Context Parallelism: sequence divided, all-gather
  - All-gather the key (K) and value (V) tensors, and then compute attention output for the local query (Q)
  - Support different types of attention masks (document mask)
  - Latency is small as the communicated K and V tensors are much smaller than Q tensor due to the use of GQA

Infrastructure, Scaling, and Efficiency

Collective Communication

- Nvidia's NCCL library: NCCLX

Reliability and Operational Challenges

- Higher than 90% effective training time
- 54 days, 466 job interruptions
- GPU issues 58.7%

Component	Category	Interruption Count	% of Interruptions
Faulty GPU	GPU	148	30.1%
GPU HBM3 Memory	GPU	72	17.2%
Software Bug	Dependency	54	12.9%
Network Switch/Cable	Network	35	8.4%
Host Maintenance	Unplanned Maintenance	32	7.6%
GPU SRAM Memory	GPU	19	4.5%
GPU System Processor	GPU	17	4.1%
NIC	Host	7	1.7%
NCCL Watchdog Timeouts	Unknown	7	1.7%
Silent Data Corruption	GPU	6	1.4%
GPU Thermal Interface + Sensor	GPU	6	1.4%
SSD	Host	3	0.7%
Power Supply	Host	3	0.7%
Server Chassis	Host	2	0.5%
IO Expansion Board	Host	2	0.5%
Dependency	Dependency	2	0.5%
CPU	Host	2	0.5%
System Memory	Host	2	0.5%

Training Recipe

- Initial Pre-Training
- Long Context Pre-Training
- Annealing

Training Recipe

Initial Pre-Training

- Learning rate: 8×10^{-5} , a linear warm up of 8,000 steps, and a cosine schedule decaying to 8×10^{-7} over 1,200,000 steps
- Batch size and sequence:
  - Initial batch size of 4M tokens to 8M after pre-training 252M tokens
  - Sequences of length 4,096 to 8,192 tokens after pre-training 252M tokens
  - Double the batch size again to 16M after pre-training on 2.87T tokens.
- Adjust the data mix: increase no-Eng, math and recent web data, downsample the bad

Training Recipe

Long Context Pre-Training

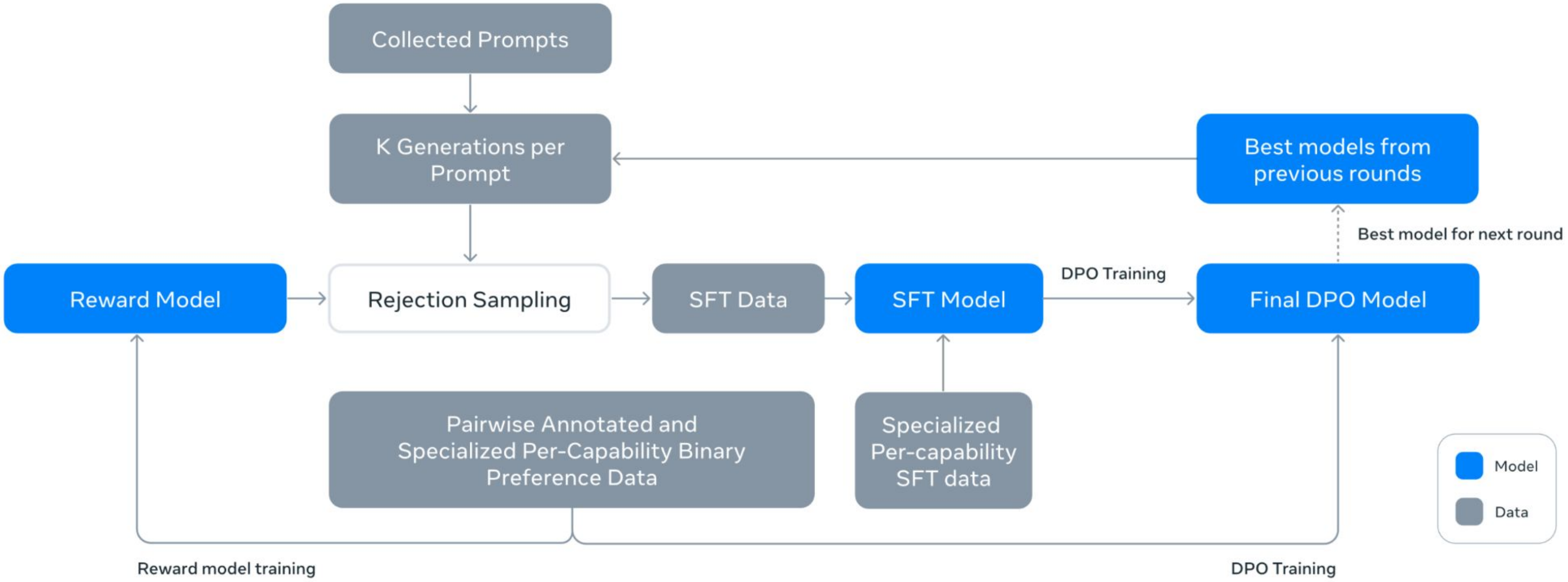
- Train on long sequences to support context windows of up to 128K tokens
  - The compute in self-attention layers grows quadratically in the sequence length.
- Assess successful adaptation:
  - Short-context evaluations
  - “needle in a haystack” tasks up to that length
- Six stages: the original 8K context window to the final 128K context window
  - 800B training tokens

Training Recipe

Annealing

- Final high-quality 40M tokens
- Learning Rate: linearly annealed the learning rate to 0
- Context length: maintaining a context length of 128K tokens
- Data mix: upsample data sources of very high quality
- Checkpoints: compute the average of model checkpoints

Post-Training



Gemma 2: Improving Open Language Models at a Practical Size

- The 27B on 13 trillion tokens, the 9B model on 8 trillion tokens, and the 2B on 2 trillion tokens.(Llama 15T tokens)
- Not trained specifically for multilingual capabilities(Llama is multilingual)
- SentencePiece tokenizer
- Train the 2B and 9B models with knowledge distillation
    - the one-hot vector seen at each token → the distribution of potential next tokens computed from a large model

$$\min_{P_S} \sum_x -P_T(x \mid x_c) \log P_S(x \mid x_c),$$

Discussion

Llama3.2 came out yesterday

- Smaller
- Multimodal
- Openly available

In the future, bigger or smaller?

Other possible abilities of LLM?