The Llama3 Herd of Models

Llama Team, AI @ Meta

Student: Jun-Chen Hung
Advisor: Wen-Chih Peng

2024/09/30

Slide here! 

Outline

- Introduction
- Related Work
- Problem
- Solution
- Experiment
- Conclusion
- References
- Q & A

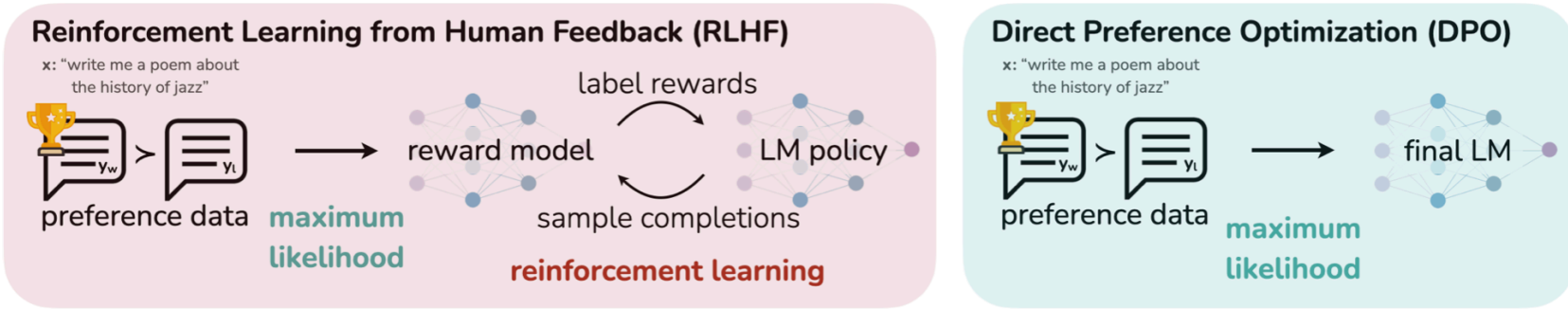
Introduction

- The Llama 3 herd of models natively supports multilinguality, coding, reasoning, and tool usage.
- Processing information in a context window of up to 128K tokens.
- Seek to optimize for 3 levers in our development process:
  - Data: 15T multilingual tokens (llama2 with 1.8T)
  - Scale:
    - pre-trained using 3.8×10^{25} FLOPs (50x more than llama2)
    - Train smaller models for much longer than is compute-optimal.
  - Managing complexity
    - Opt for a standard dense Transformer model architecture, rather than MoE.
    - Simple post-training procedure based on SFT, RS, and DPO.

Related Work

- **Scale**: Scaling Laws.
- **Small models**: Compute-optimal v.s. Distill larger models.
- **Architectures**: Dense Transformer v.s. MoE.
- **Open source**: Mistral, Falcon, Gemma, Grok, OpenELM, Qwen etc.
- **Post-training**: Reject Sampling, Supervised Finetuning, Direct Preference Optimization.

DPO - Direct Preference Optimization

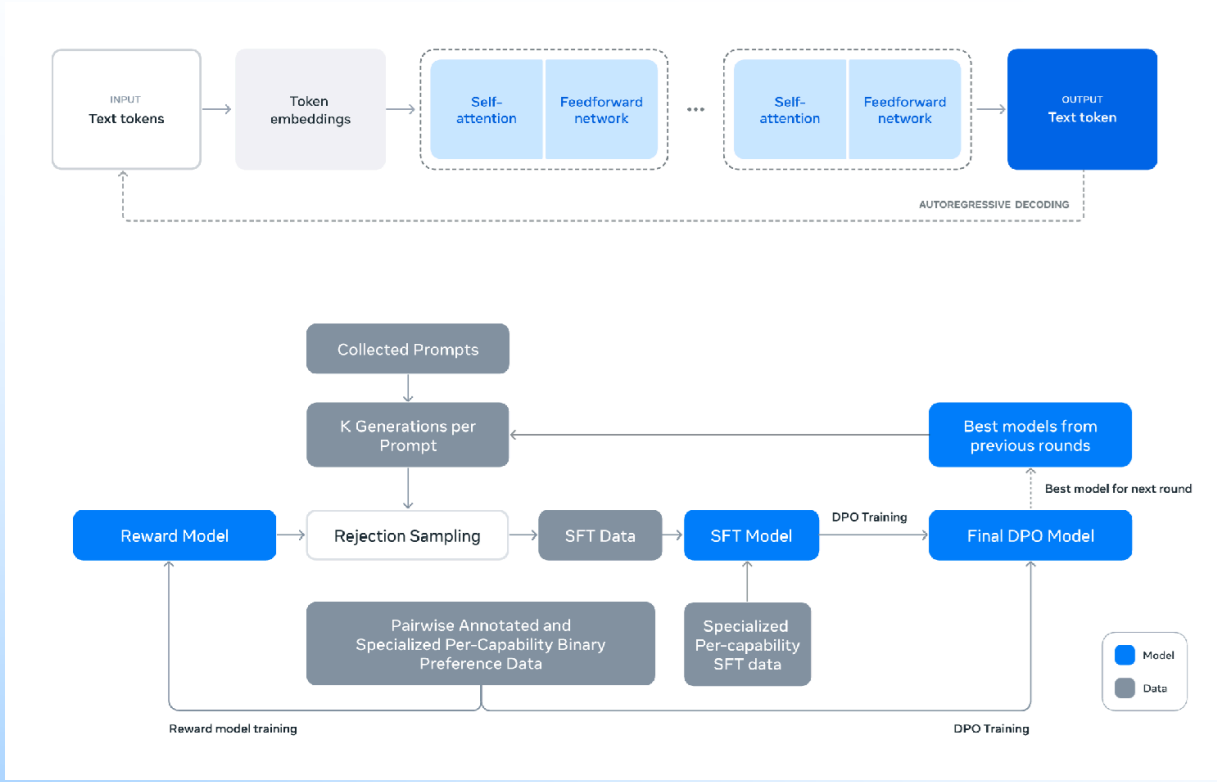


- Rather than using reward modeling and reinforcement learning, DPO leverages a unique parameterization reward functions to extract the optimal policy directly.

Problem

- Embracing the open source
- Improve overall performance across core LLM capabilities such as reasoning and coding
- Multilingual (llama3.1)
- Longer Context (llama3.1)
- Multimodal (llama3.2 11b & 90b)

Solution


INPUT
Text tokens
Token embeddings
Self-attention
Feedforward network
Self-attention
Feedforward network
OUTPUT
Text token
AUTOREGRESSIVE DECODING
Collected Prompts
K Generations per Prompt
Best models from previous rounds
Best model for next round
Reward Model
Rejection Sampling
SFT Data
SFT Model
DPO Training
Final DPO Model
Pairwise Annotated and Specialized Per-Capability Binary Preference Data
Specialized Per-capability SFT data
Model
Data
Reward model training
DPO Training

Pre-training - Data

- Knowledge-cutoff: the end of 2023
- Web Data Curation:
  - PII & safety filtering, Text extraction & cleaning, De-duplication, Heuristic filtering
  - Model-based quality filtering
    - Use `fasttext` to recognize if a given text would be referenced by Wikipedia.
    - Roberta-based classifiers trained on Llama 2 predictions.
  - Code and reasoning data
    - build domain-specific pipelines that extract code and math-relevant web pages.
    - DistilRoberta models trained on web data annotated by Llama 2.
  - Multilingual data: `fasttext`-based model to categorize documents into 176 languages.

Pre-training - Data

- Annealing Data
  - Annealing on small amounts of high-quality code and mathematical data can boost the performance of pre-trained models on key benchmarks
  - Annealing the learning rate of a 50% trained Llama 3 8B model linearly to 0 on 40B tokens, assign 30% weights to the new dataset and remaining 70% weight to the default data mix.
  - The improvements on the 405B model are negligible => has strong in-context learning and reasoning capabilities

Pre-training - Model

- Mostly follow llama2
- All models use GQA now (34B and 70B only in llama2)
- Use attention mask between different document in the same sequence.
- 128k volcabulary size = 100k from `tiktoken` + 28k for non-English languages
- RoPE frequency increased to 500k => support longer context

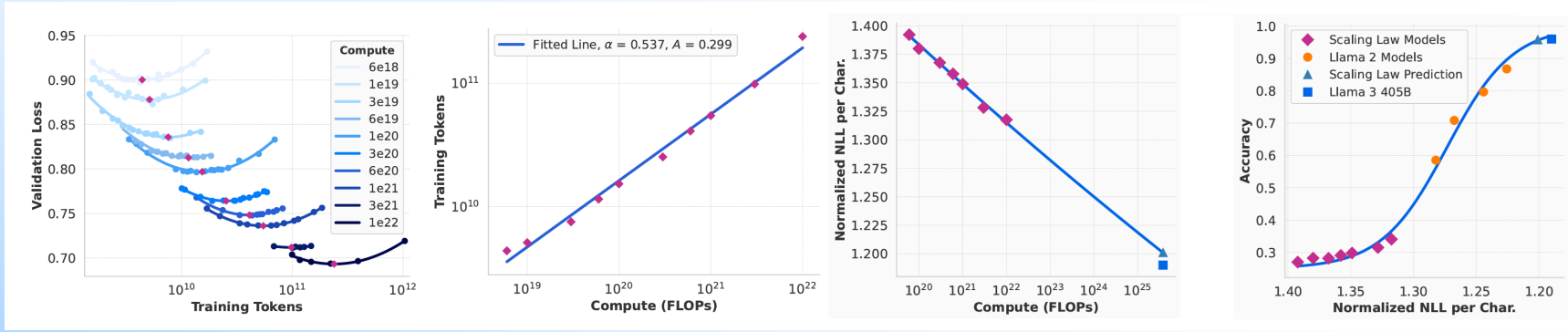
	8B	70B	405B
Layers	32	80	126
Model Dimension	4,096	8192	16,384
FFN Dimension	14,336	28,672	53,248
Attention Heads	32	64	128
Key/Value Heads	8	8	8
Peak Learning Rate	3×10^{-4}	1.5×10^{-4}	8×10^{-5}
Activation Function	SwiGLU		
Vocabulary Size	128,000		
Positional Embeddings	RoPE ($\theta = 500,000$)		

Pre-training - Scaling Rule

- the model size and the number of training tokens should be scaled equally.
- Target:
  - Determine optimal model size.
  - Forcast flagship's model performance on downstream tasks
- Current issue:
  - Existing one use training loss rather than enchmark performance.
  - May be noisy and unreliable because of small compute budgets.

Pre-training - Scaling Rule

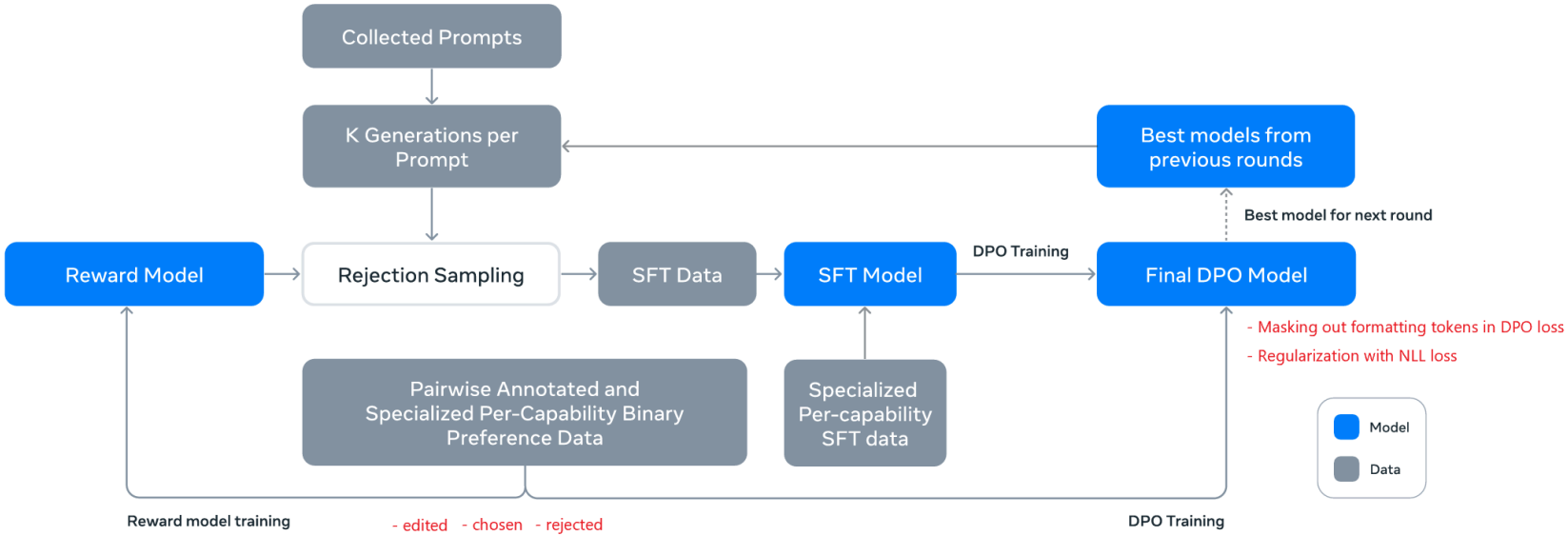
- Accurately predict downstream benchmark performance:
  - Establish a correlation between negative log-likelihood on down stream tasks and the training FLOPs
  - Correlate the negative log-likelihood on downstream tasks with task accuracy with scaling law models and llama2 models.



Pre-training - Training Recipe

- Initinal pre-training
- long-context pre-training 800B tokens
  - Scaling up context window from 8k to 128k in 6 stagees
- Annealing 40M tokens
  - Linearly annealed the learning rate to 0
  - Compute the average of model checkpoints as the final pre-trained model.

Post-training



Doing 6 rounds. In each cycle, collect new preference annotations and SFT data, sampling synthetic data from the latest models.

Post-training - Data

Prefrence Data

- 4 levels: significantly better, better, slightly better, or marginally better.
- New `edit` field: further improve the preferred response.
- In each round:
  - use all the preference data that is available for reward modeling.
  - using the latest batches from various capabilities for DPO training
  - Only use significantly better, better pairs.

Dataset	% of comparisons	Avg. # turns per dialog	Avg. # tokens per example	Avg. # tokens in prompt	Avg. # tokens in response
General English	81.99%	4.1	1,000.4	36.4	271.2
Coding	6.93%	3.2	1,621.0	113.8	462.9
Multilingual	5.19%	1.8	1,299.4	77.1	420.9
Reasoning and tools	5.89%	1.6	707.7	46.6	129.9
Total	100%	3.8	1,041.6	44.5	284.0

Post-training - Data

SFT Data

- Prompts from our human annotation collection with rejection-sampled responses
- Synthetic data targeting specific capabilities
- Small amounts of human-curated data

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

Post-training - Data

Quality Control

- **Data cleaning**: a series of rule-based data removal and modification strategies to filter or clean problematic data. (e.g. emoji, `I'm sorry`, `I apologize`)
- **Topic classification**: Finetune Llama 3 8B into a topic classifier, classify all data into coarsely-grained and fine-grained buckets.
- **Quality scoring**: Obtain scores from reward model and Llama-based signals, select examples that are marked as high quality
- **Difficulty scoring**: Instag (with lama3 70B) and Llama-based scoring
- **Semantic deduplication**: using RoBERTa to cluster dialogs and keep the ones that have maximum cosine similarity less than a threshold

Post-training - Capabilities

- Code
- Multilinguality
- Math and Reasoning
- Long Context
- Tool Use
- Factuality
- Steerability

Post-training - Code Capability

- Code Export: Continuing pre-training on a 1T token mix of mostly (>85%) code data.
  - Its SFT and DPO data mixes primarily targeting code.
  - Also use for reject sampling with coding prompts.
- Synthetic data generation
  - Execution feedback
  - Programming language translation
  - Backtranslation
- System prompt steering during rejection sampling.
- Filtering training data with execution and model-as-judge signals
  - code correctness and code style

```
public static int ClimbStairs(int n)
{
    if (n == 1)
    {
        return 1;
    }

    if (n == 2)
    {
        return 2;
    }

    int[] dp = new int[n + 1];
    dp[1] = 1;
    dp[2] = 2;

    for (int i = 3; i <= n; i++)
    {
        dp[i] = dp[i - 1] + dp[i - 2];
    }

    return dp[n];
}
```

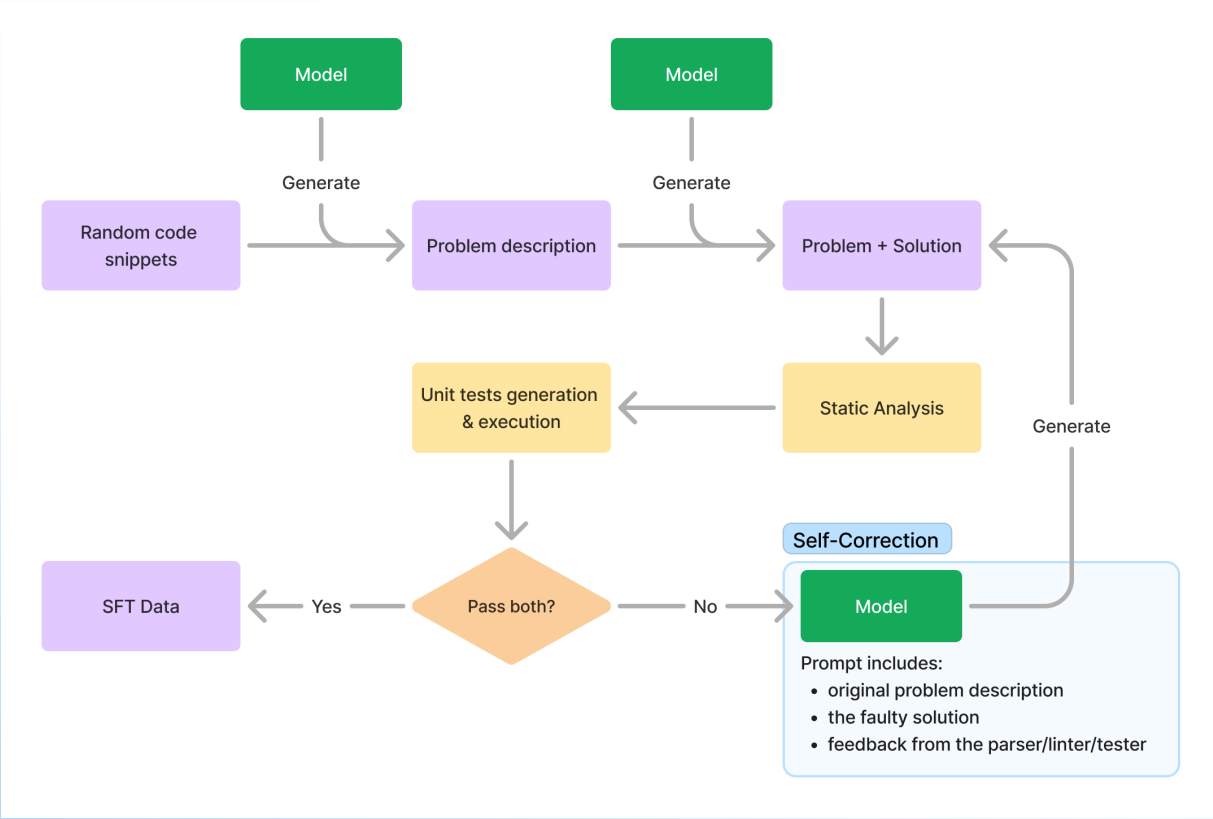
```
public static int ClimbStairs(int n)
{
    // Base cases
    if (n == 1) return 1;
    if (n == 2) return 2;

    // Initialize variables to store the number
of ways to climb
    int prev = 1;
    int curr = 2;

    // Calculate the number of ways to climb for
n > 2
    for (int i = 3; i <= n; i++)
    {
        int temp = curr;
        curr = prev + curr;
        prev = temp;
    }

    return curr;
}
```

Synthetic data generation: Execution feedback



Post-training - Multilingual Capability

- Multilingual Expert: continuing to pre-train on a data mix that consists of 90% multilingual tokens.
- Multilingual data collection:
  - Human annotations (2.4%)
  - Data from other NLP tasks (44.2%)
  - Rejection sampled data (18.8%)
  - Translated reasoning data (34.6%)

Post-training - Math and Reasoning

- Challanges
  - Lack of prompts
    - Convert mathematical context to QA format for SFT
  - Lack of ground truth chain of thought
    - Generate step-by-step solutions for a set of prompts and self-verification
  - Incorrect intermediate steps
    - Use step-wise reward models to filter
  - Teaching models to use external tools
    - Prompt llama3 to solve problems using textual reasoning and associated Python code
  - Discrepancy between training and inference