Training AIs for Capability and Alignment

AI 109

Previously

- Supervised Learning
- Self-Supervised Learning
  - Next-token prediction
- LLMs

Today

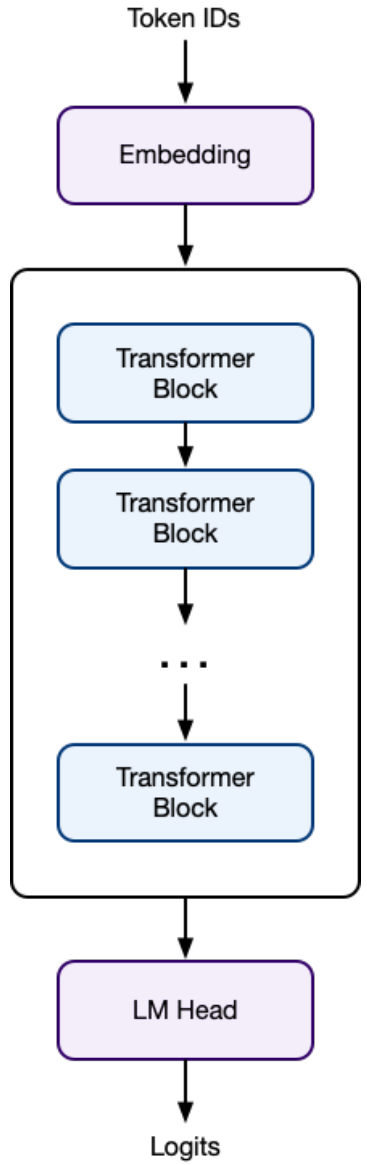
- The modern training pipeline for an LLM.
- Question: how can we prevent an AI from saying/doing horrible things?

Tay

- Introduced by Microsoft on March 23, 2016.
- Based on a similar project from China that went very well.
- Designed to post tweets and interact with other Twitter users.
- Within 24 hours, Tay was posting racist tweets.
- Tay was taken offline by March 25. Microsoft issued an apology.

Reminder: Transformers

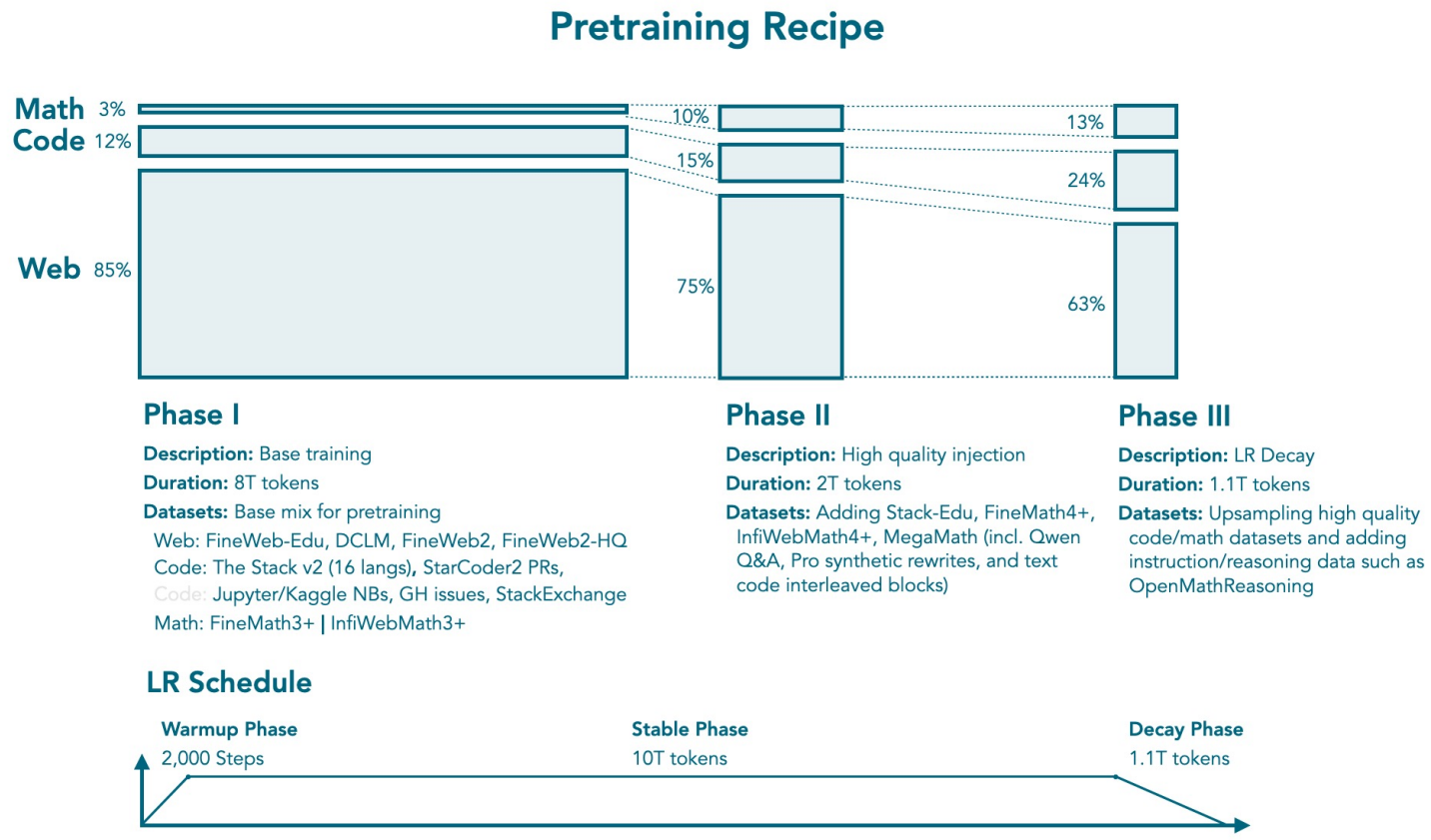
- The neural network architecture used for most AIs is the ***transformer***.
- Transformers accept tokens as inputs and produce next-token probabilities as outputs.
- Starting from a **prompt**, a process of **sampling** generates new tokens.
  - These are appended to the prompt to create the text you see as you talk to an AI.



Reminder: Pretraining

- We train models on trillions of tokens using the next-token prediction loss.
  - Modern models can be trained on 5 trillion to 50+ trillion tokens, depending on model size.
  - 1 trillion tokens of text is about 3-5 terabytes of data.
  - Example data set: fineweb-edu
- The resulting model is called a ***base*** model.
  - Because it's the starting point for other forms of more sophisticated training.

Pretraining



https://huggingface.co/blog/smollm3

Pretraining

- **Stage 1: Stable phase (0T → 8T tokens)** This foundation stage establishes strong general capabilities with our core dataset mixture:
  - Web: 85% (12% multilingual) - FineWeb-Edu, DCLM, FineWeb2 and FineWeb2-HQ
  - Code: 12% - The Stack v2 (16 programming languages), StarCoder2 pull requests, Jupyter and Kaggle notebooks, GitHub issues, and StackExchange.
  - Math: 3% - FineMath3+ and InfiWebMath3+
- **Stage 2: Stable phase (8T → 10T tokens)** We introduce higher quality math and code datasets while maintaining good web coverage:
  - Web: 75% (12% Multilingual)
  - Code: 15% - Adding Stack-Edu
  - Math: 10% - Introducing FineMath4+, InfiWebMath4+, and MegaMath (including Qwen Q&A, Pro synthetic rewrites, and text-code interleaved blocks)
- **Stage 3: Decay Phase (10T → 11.1T tokens)** The final stage further upsamples math and code data
  - Web: 63% (12% Multilingual)
  - Code: 24% - upsampling of high-quality code data
  - Math: 13% - upsampling math data and introducing instruction and reasoning datasets such as OpenMathReasoning

https://huggingface.co/blog/smollm3

Base Models Aren’t Usable

- Trained transformers don’t typically sound very good…
- They are essentially a very fancy autocomplete.
- Their style is “continuation” – they just keep going (and going, and going).
- Demo: LM Studio
- The discipline of **post-training** emerged to solve this problem.

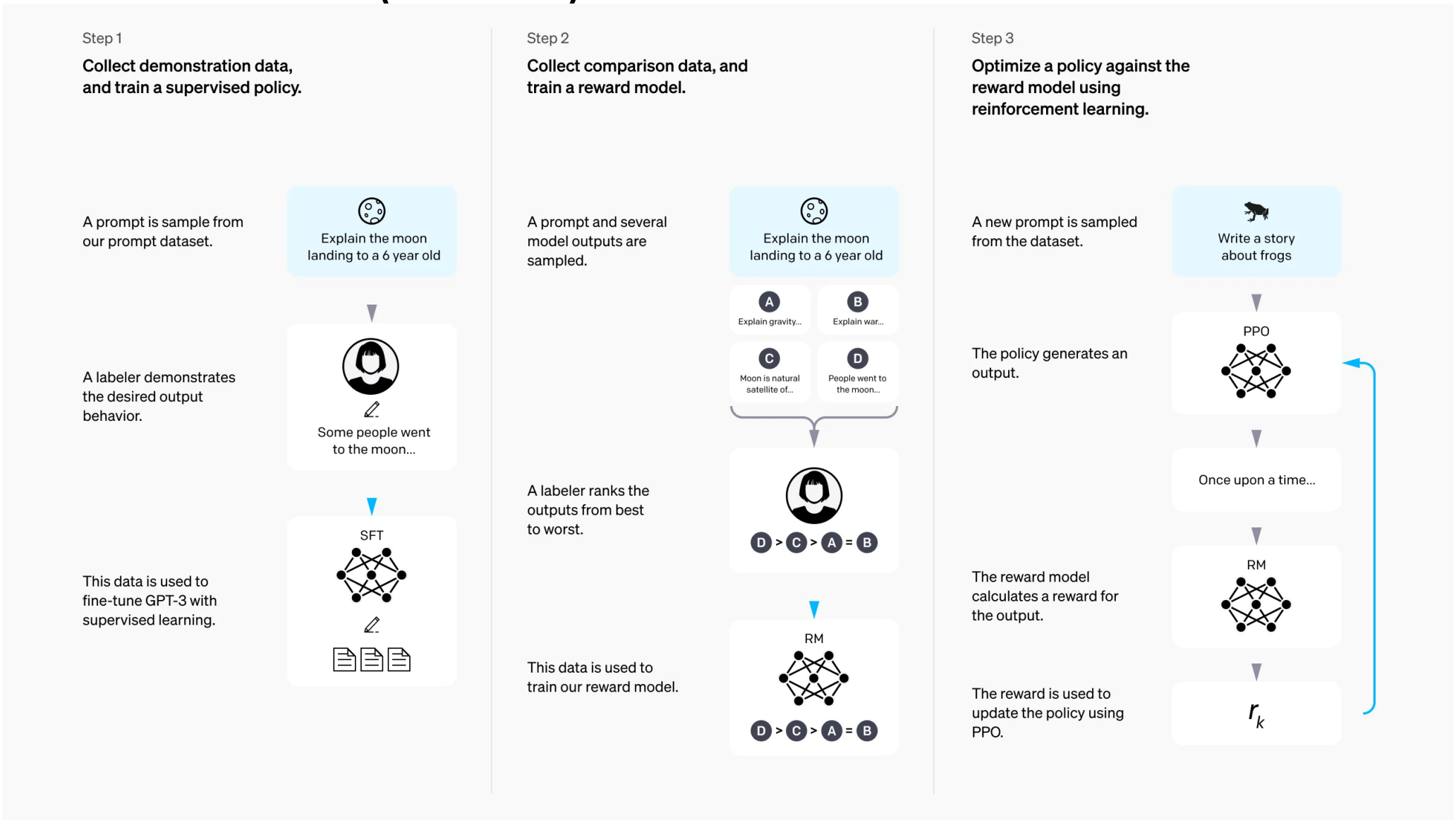
Modern AI Training

- Broken into a series of **stages**.
- Part of what determines success or failure of a model is how well these stages are implemented.
- All models start with pretraining.
- What happens next depends on the model, but there are a few patterns.
- The original recipe dates to 2022 with the **InstructGPT** model.

InstructGPT

- After pre-training, perform a series of post-training steps to improve the abilities of the model.
- 3 Stages of Post-training
  - Supervised fine-tuning
  - Reward modeling
  - **Reinforcement learning from human feedback**

InstructGPT (2022)



Step 1: Supervised Fine-tuning

- Also called **instruction fine-tuning**
- Start from the base model.
- Do supervised training using **demonstrations** of tasks.
- This is a kind of **imitation learning**.
- Examples:
  - https://huggingface.co/datasets/nvidia/Nemotron-Cascade-2-SFT-Data/viewer/chat/train?row=0

Step 2: Reward Modeling

- First, collect a data set of **comparisons** between two responses *x* and *y* for the same prompt
- Then one of two methods
  - Old: have humans rank responses.
  - New: have an AI rank responses.
- Train a neural network *r(x,y)* that gives preferred answers a higher rating.
  - This is called the **reward model.**

Step 3: RLHF

- First: sample a set of prompts from a data set.
- Second: have your model generate completions.
- Third: score each completion with your reward model
- Fourth: update the model using reinforcement learning on the scores from the reward model.

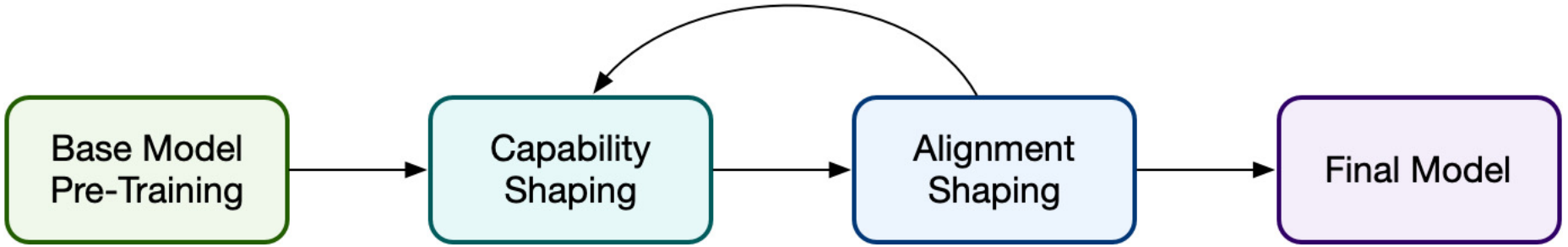
RLHF: How Much Data?

- InstructGPT (2022)
  - Instruction data
    - 10,000 examples
  - Preference data
    - 100,000 examples
  - RL
    - 100,000 prompts

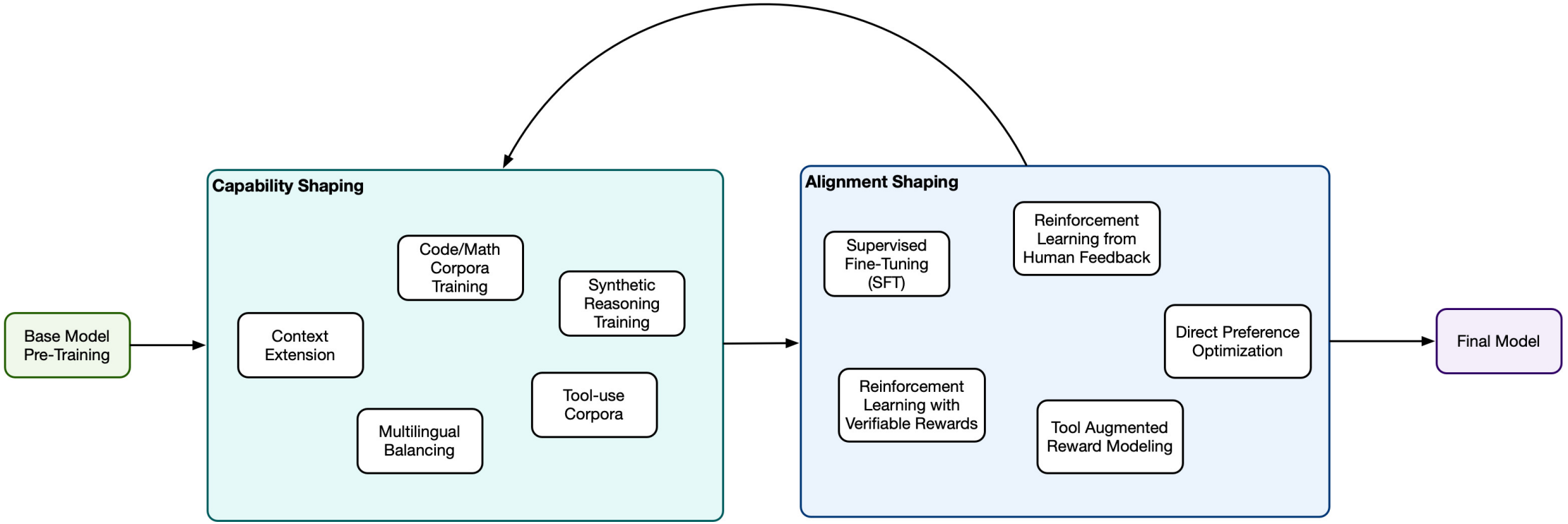
A Modern Twist RLVR

- In some areas, we don’t need human preferences.
  - Math: check the answer directly.
  - Code: run tests on the generated code.
- We can define a reward to be positive if the answer is correct and 0 otherwise.
- This is called **reinforcement learning with verifiable rewards (RLVR)**.
- Probably the main reason AI companies are focusing on code and math is because RLVR is cheaper than other forms of RL.

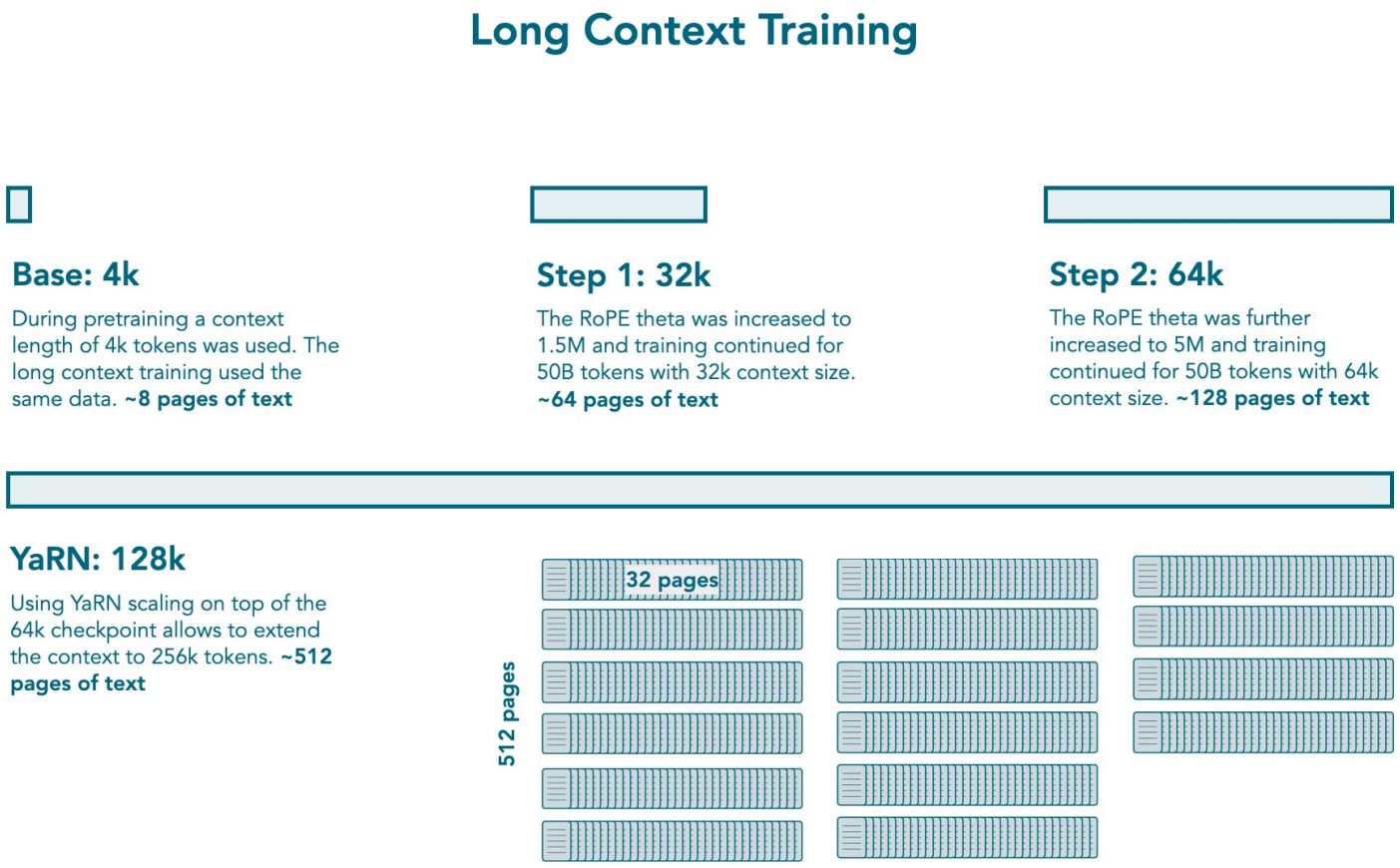
The Modern Pipeline



The Modern Pipeline

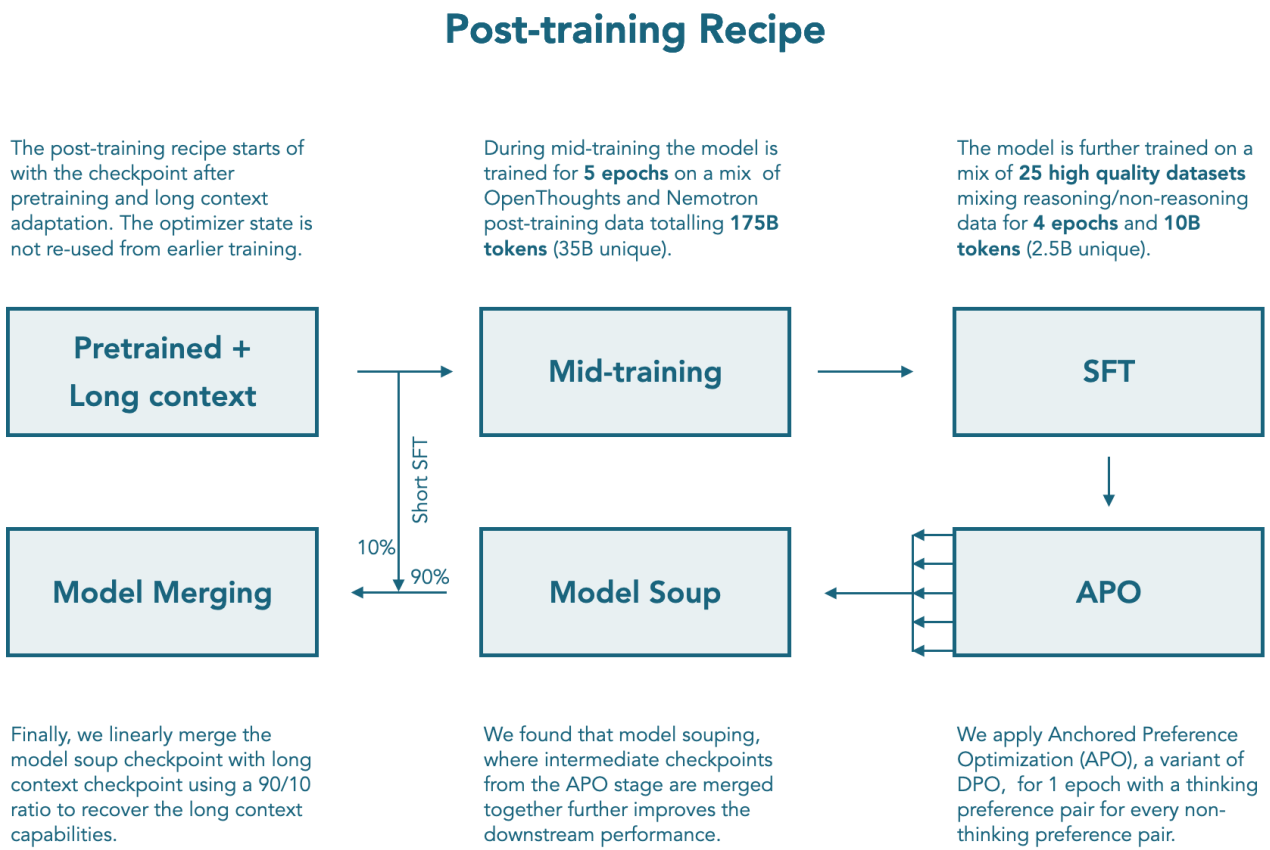


Mid-Training



https://huggingface.co/blog/smollm3

Post-Training



https://huggingface.co/blog/smollm3

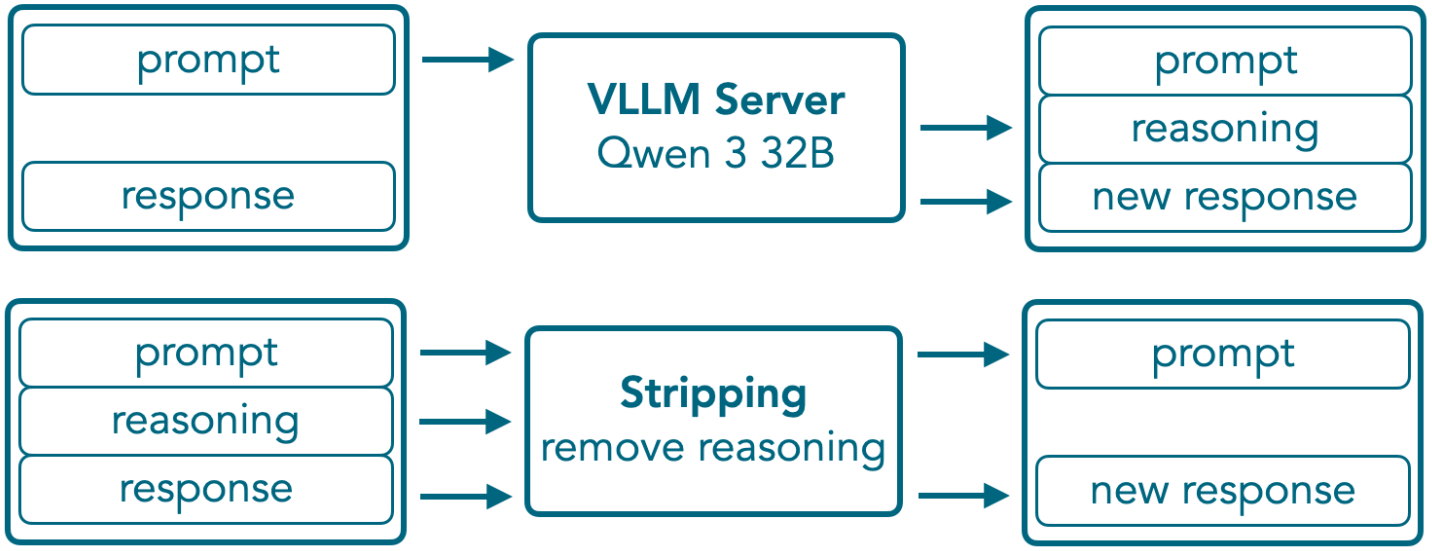
Question

- If we pretrain on trillions of tokens and post-train on millions, can post-training really do that much?
- “The **superficial alignment hypothesis** (SAH) posits that large language models learn most of their knowledge during pre-training, and that post-training merely surfaces this knowledge”
  - https://arxiv.org/abs/2602.15829
- Can we get more data?

Synthetic Data

Synthetic SFT Data

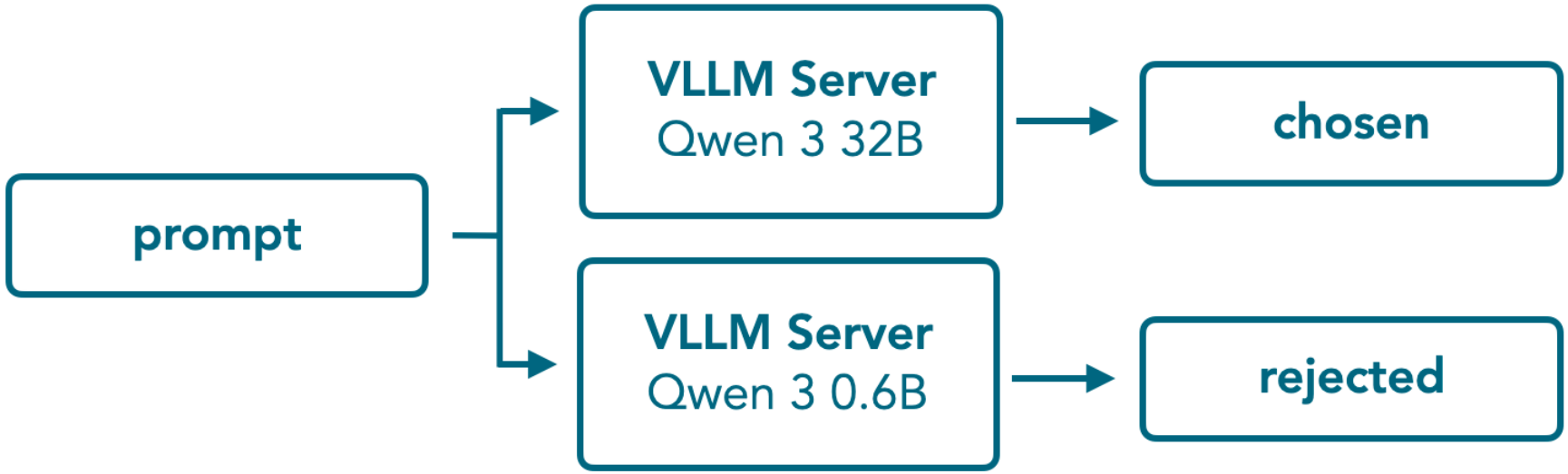
For dual reasoning mode to work reliably it was necessary to generate synthetic data for datasets which had not reasoning traces and create a reasoning stripped version of reasoning datasets:



Synthetic Data

Synthetic Preference Data

For the APO step preference data is necessary which we generated using a strong (Qwen 3 32B) and a weak model (Qwen 3 0.6B):

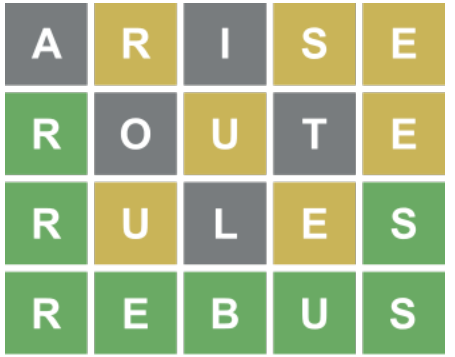


RL Environments

- We define RL problems by creating **environments**.
- These are miniature “games” the AIs play to learn a task.
- Early RL environments were based on actual games and on physics problems.
- More recently, researchers are trying to define text-based RL environments for training LLMs.

RL environments for LLMs

- Wordle
  - https://openenv-wordle.hf.space/web/
  - https://colab.research.google.com/github/huggingface/trl/blob/main/examples/notebooks/openenv_wordle_grpo.ipynb#scrollTo=euF9pZTbn9BG



Learning More

- Aligning language models to follow instructions
- Reinforcement Learning from Human Feedback
- SmolLM3: smol, multilingual, long-context reasoner