3. Post-training Code LMs: Supervised Fine-Tuning

Post-training: Recipe for SFT

- Instruction Data
  - Objective: Steer LLMs toward targeted behaviors (instruction following, reasoning).
  - Approach: Leverage instruction synthesis, structured response generation, and systematic quality evaluation.
- Training Curriculum
  - Data Selection: Data mixing and scaling.
  - Training Strategy: Iterative and multi-stage SFT, heavy/lightweight SFT.

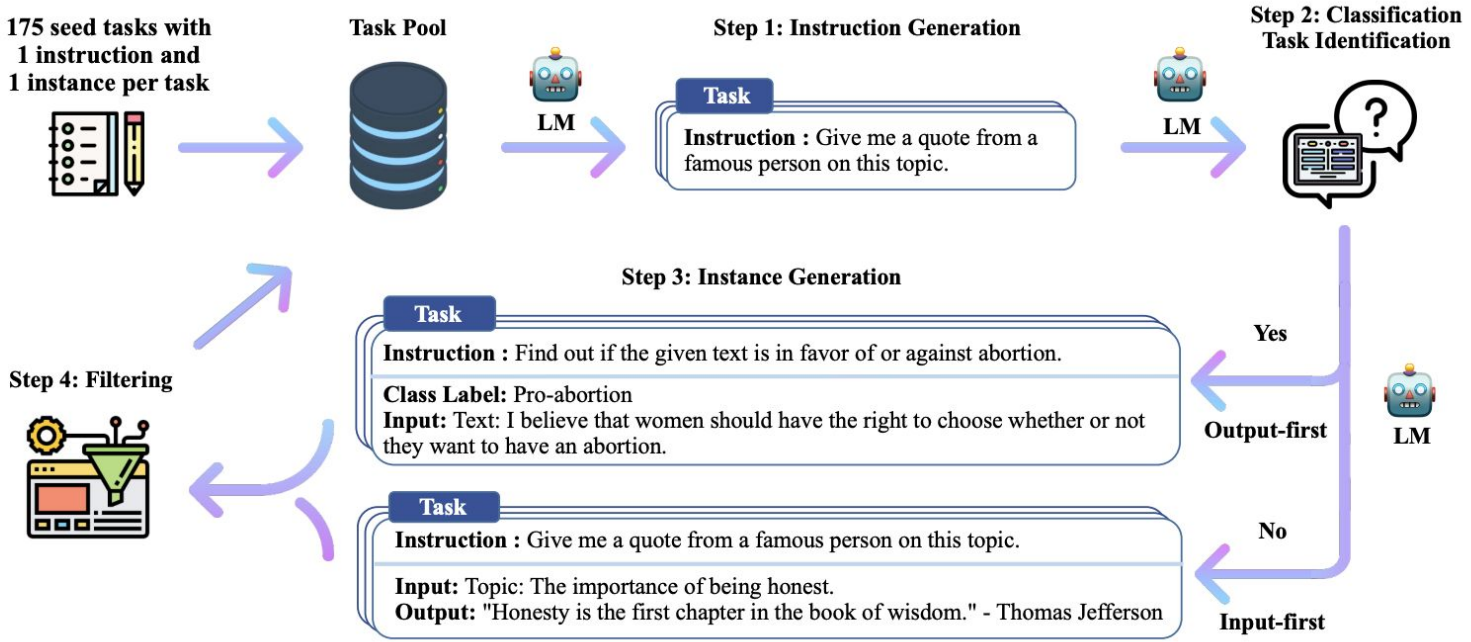
Recipe for SFT: Instruction Data

- Train code specific LLM (before)
  - Branching main pre-training run and continuing pre-training, followed by iterative SFT (Llama 3 approach)
  - Leverage to generate synthetic data, quality filtering

Recipe for SFT: Instruction Data

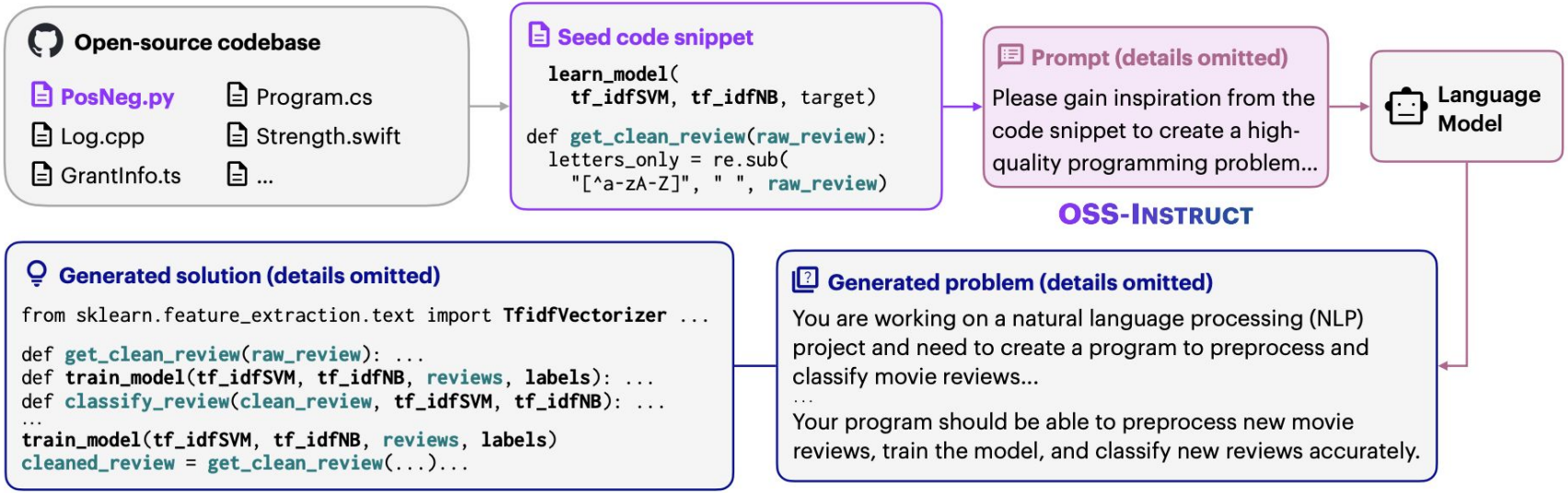
- Train code specific LLM (before)
  - Branching main pre-training run and continuing pre-training, followed by iterative SFT (Llama 3 approach)
  - Leverage to generate synthetic data, quality filtering
- Synthetic data generation
  - Techniques: Self-Instruct, Evol-Instruct, OSS-Instruct, and more.

Self-Instruct



Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

OSS-Instruct



Llama 3 adopted OSS-Instruct to generate synthetic code instruction data

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2024b. Magicoder: Empowering code generation with oss-instruct. In International Conference on Machine Learning, pages 52632–52657. PMLR.

Recipe for SFT: Instruction Data

- Train code specific LLM (before)
  - Branching main pre-training run and continuing pre-training, followed by iterative SFT (Llama 3 approach)
  - Leverage to generate synthetic data, quality filtering
- Synthetic data generation
  - Techniques: Self-Instruct, Evol-Instruct, OSS-Instruct, and more.
  - Reasoning-based data for complex tasks

Reasoning vs. Non-reasoning Mode

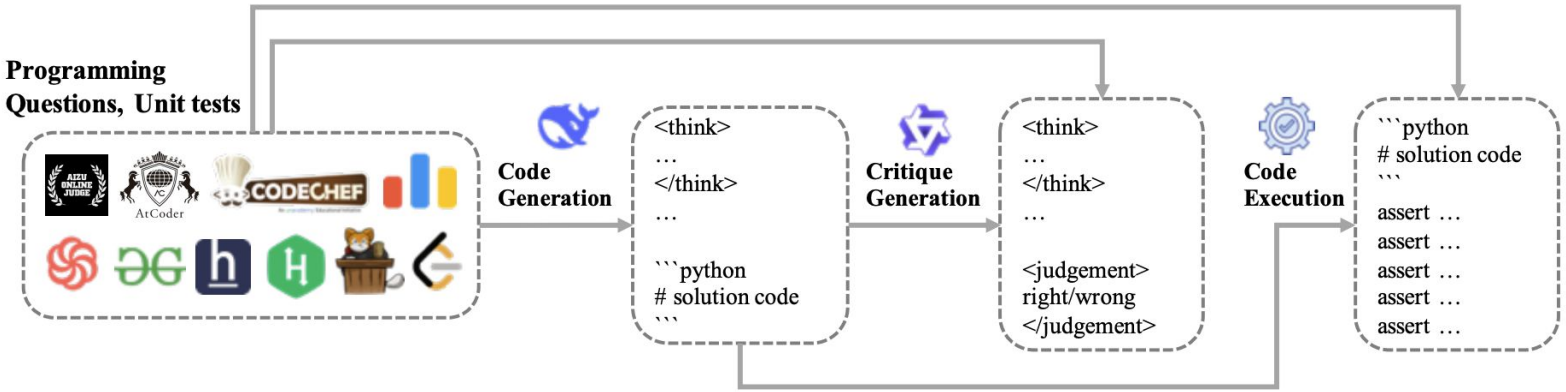
Also known as “Thinking” vs “Non-Thinking” Mode

Thinking Mode	Non-Thinking Mode
<|im_start|>user {query} /think<|im_end|> <|im_start|>assistant <think> {thinking_content} </think> {response}<|im_end|>	<|im_start|>user {query} /no_think<|im_end|> <|im_start|>assistant <think> </think> {response}<|im_end|>

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025a.

Reasoning-based Data for Competitive Coding

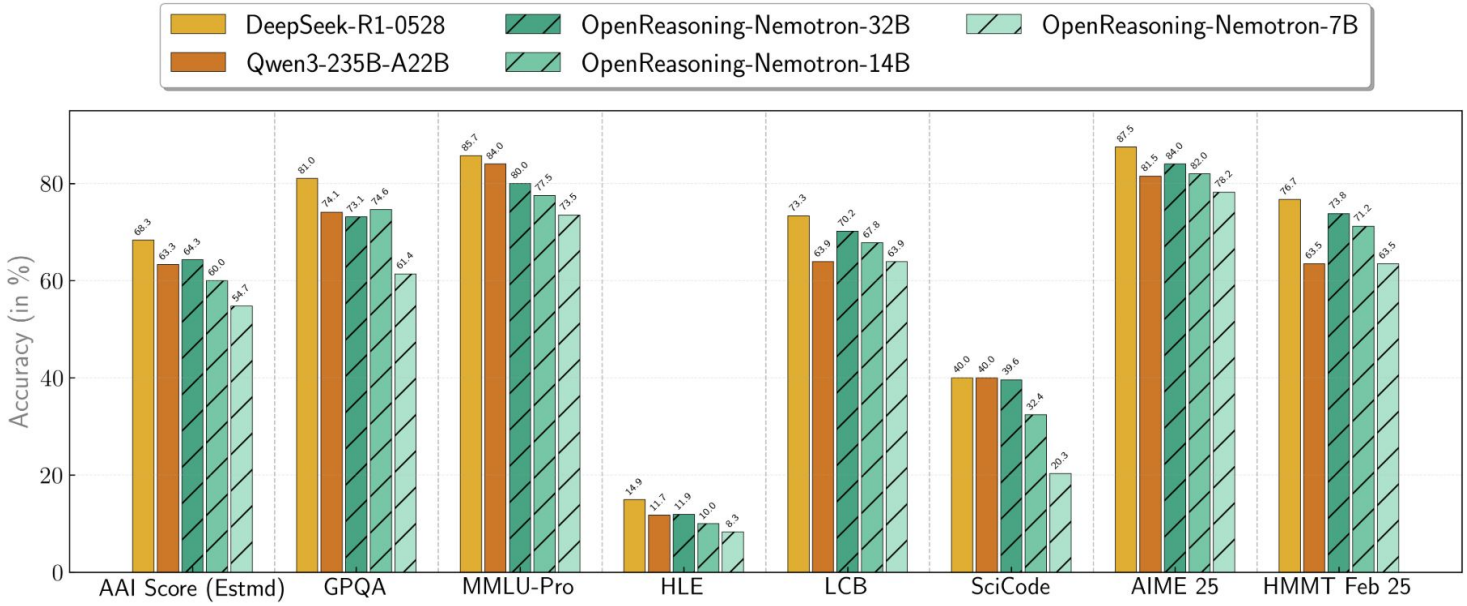
Strong-to-Weak distillation (off-policy)



Overview of the OpenCodeReasoning development pipeline.

Wasi Uddin Ahmad, Somshubra Majumdar, Aleksander Ficek, Sean Narenthiran, Mehrzad Samadi, Jocelyn Huang, Siddhartha Jain, Vahid Noroozi, and Boris Ginsburg. Opencodereasoning-ii: A simple test time scaling approach via self-critique. arXiv preprint arXiv:2507.09075, 2025.

Distillation from DeepSeek-R1-0528



Distill(Qwen2.5-*-Instruct) => OpenReasoning-Nemotron-*

https://huggingface.co/blog/nvidia/openreasoning-nemotron

Reasoning-based Data for Competitive Coding

Open Thoughts

DATA RECIPES FOR REASONING MODELS

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. arXiv preprint arXiv:2506.04178, 2025.

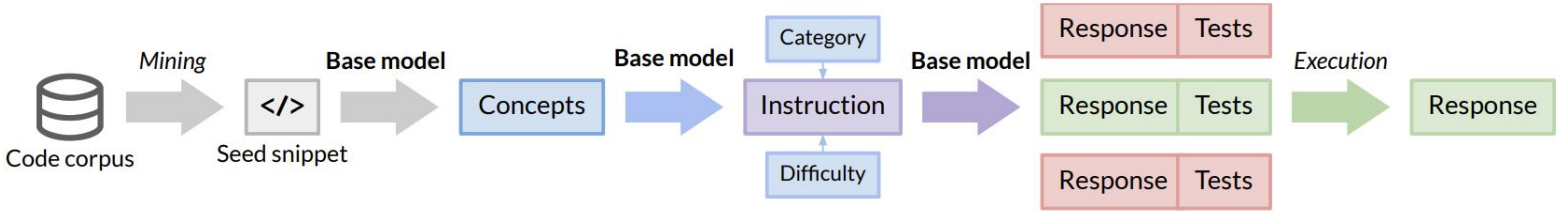
Recipe for SFT: Instruction Data

- Train code specific LLM (before)
  - Branching main pre-training run and continuing pre-training, followed by iterative SFT (Llama 3 approach)
  - Leverage to generate synthetic data, quality filtering
- Synthetic data generation
  - Techniques: Self-Instruct, Evol-Instruct, OSS-Instruct, and more.
  - Reasoning-based data for complex tasks
  - Systematic quality assessment

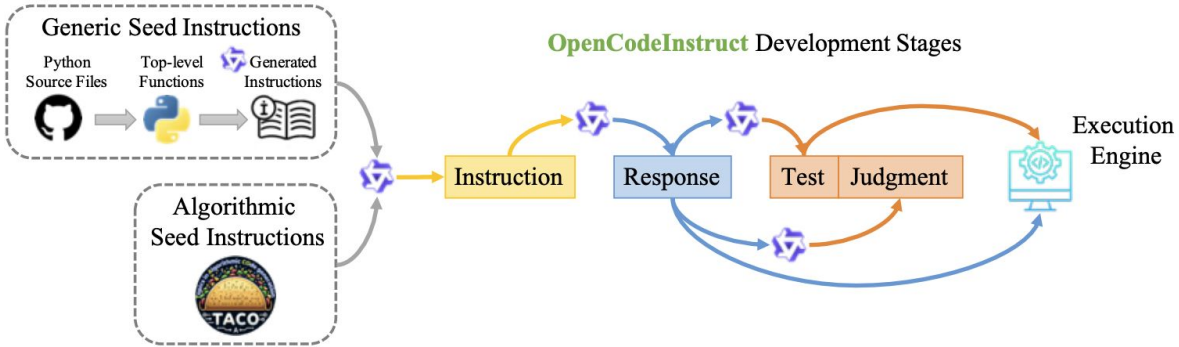
Quality Assessment

Based on execution feedback or LLM judgements

(1)



(2)

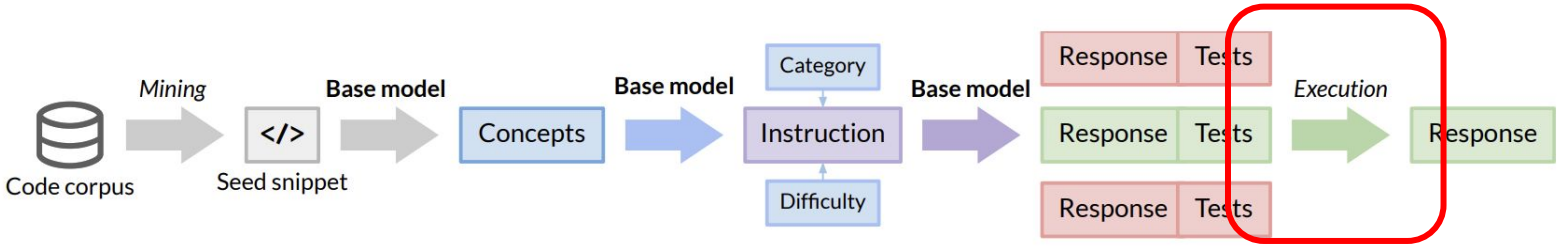


(1) Wei, Y., Cassano, F., Liu, J., Ding, Y., Jain, N., Mueller, Z., de Vries, H., Werra, L.V., Guha, A., ZHANG, L., 2024a. Selfcodealign: Self-alignment for code generation, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems. URL: https://openreview.net/ forum?id=xXRnUU7xTL.
(2 Ahmad, W.U., Ficek, A., Samadi, M., Huang, J., Noroozi, V., Majumdar, S., Ginsburg, B., 2025a. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. URL: https://arxiv.org/ abs/2504.04030, arXiv:2504.04030.

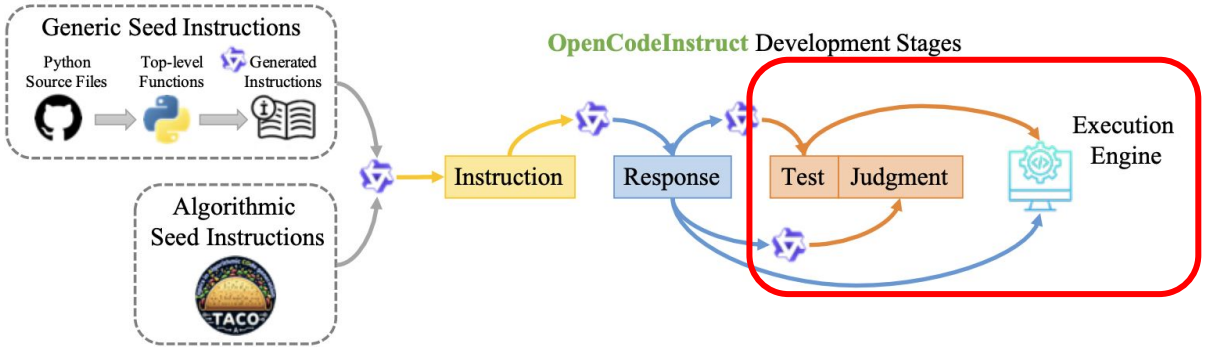
Quality Assessment

Based on execution feedback or LLM judgements

(1)



(2)



(1) Wei, Y., Cassano, F., Liu, J., Ding, Y., Jain, N., Mueller, Z., de Vries, H., Werra, L.V., Guha, A., ZHANG, L., 2024a. Selfcodealign: Self-alignment for code generation, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems. URL: https://openreview.net/ forum?id=xXRnUU7xTL.
(2 Ahmad, W.U., Ficek, A., Samadi, M., Huang, J., Noroozi, V., Majumdar, S., Ginsburg, B., 2025a. Opencodeinstruct: A large-scale instruction tuning dataset for code llms. URL: https://arxiv.org/ abs/2504.04030, arXiv:2504.04030.

Quality Assessment

Based on execution feedback or LLM judgements

Selection strategy	Data size	Execution pass rate	Pass@1
Random selection (all)	27.7k	24.1%	61.6
Random selection (subset)	15.6k	24.2%	61.6
Failures only	15.6k	0%	57.9
Passes only	15.6k	100.0%	**65.2**

Table: Pass@1 on HumanEval+ with different response selection strategies from **SelfCodeAlign**.

Wei, Y., Cassano, F., Liu, J., Ding, Y., Jain, N., Mueller, Z., de Vries, H., Werra, L.V., Guha, A., ZHANG, L., 2024a. Selfcodealign: Self-alignment for code generation, in: The Thirty-eighth Annual Conference on Neural Information Processing Systems. URL: https://openreview.net/ forum?id=xXRnUU7xTL.

Post-training: Recipe for SFT

- Instruction Data
  - Objective: Steer LLMs toward targeted behaviors (instruction following, reasoning).
  - Approach: Leverage instruction synthesis, structured response generation, and systematic quality evaluation.
- Training Curriculum
  - Data Selection: Data mixing and scaling.
  - Training Strategy: Iterative and multi-stage SFT, heavy/lightweight SFT.

Training Curriculum: Data Mixing for SFT

- Mixing code data with other sources

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: Statistics of SFT data used for **Llama 3** post-training.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Training Curriculum: Data Mixing for SFT

- Coarse-to-fine SFT
  - Coarse: Large-scale lower-quality/diverse data
  - Fine: Small-scale high-quality data

Stage	Data Source	# Examples
Stage1	RealUser-Instruct	0.7 M
	Large-scale Instruct	2.3 M
	Infinity-Instruct	1.0 M
Stage2	McEval-Instruct	36 K
	Evol-Instruct	111 K
	Verified-Instruct	110 K
	Package-Instruct	110 K

Table: Detailed SFT data statistics for **OpenCoder** post-training.

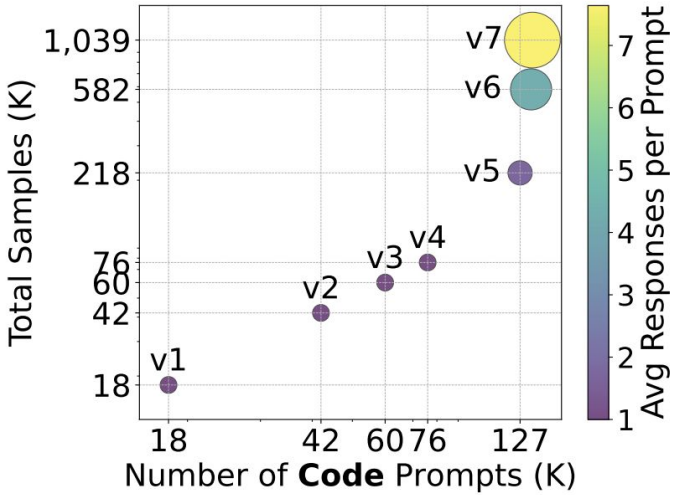
Siming Huang, Tianhao Cheng, Jason Klein Liu, Jiaran Hao, Liuyihan Song, Yang Xu, J Yang, JH Liu, Chenchen Zhang, Linzheng Chai, et al. 2024. Opencoder: The open cookbook for top-tier code large language models. arXiv preprint arXiv:2411.04905.

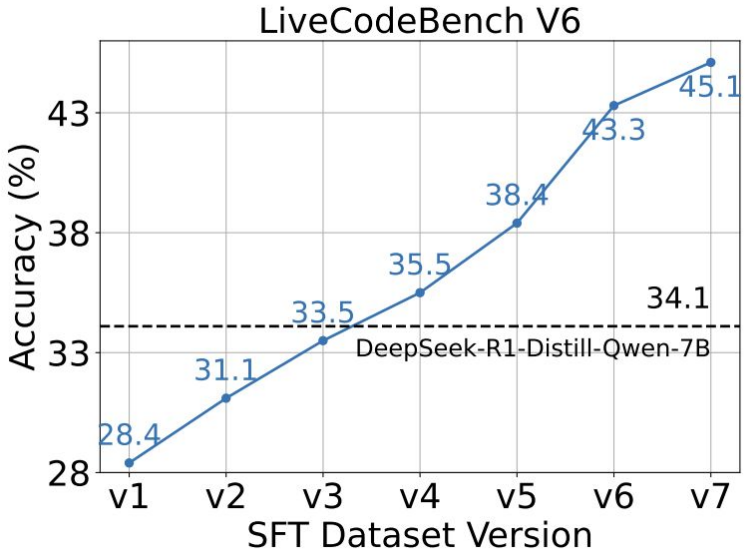
Training Curriculum: Data Mixing for SFT

- Reasoning vs. Non-reasoning data
  - In general, **reasoning-based samples > non-reasoning samples**
  - DeepSeek-R1 => 600k reasoning, 200k non-reasoning SFT samples
  - Llama-Nemotron => 900k reasoning, 9M non-reasoning SFT code samples
  - Nemotron-H => 5:1 and 1: 1 ratio of reasoning to non-reasoning samples for stage 1 and 2

Training Curriculum: Data Scaling

Supervised Fine-tuning of Qwen2.5-7B



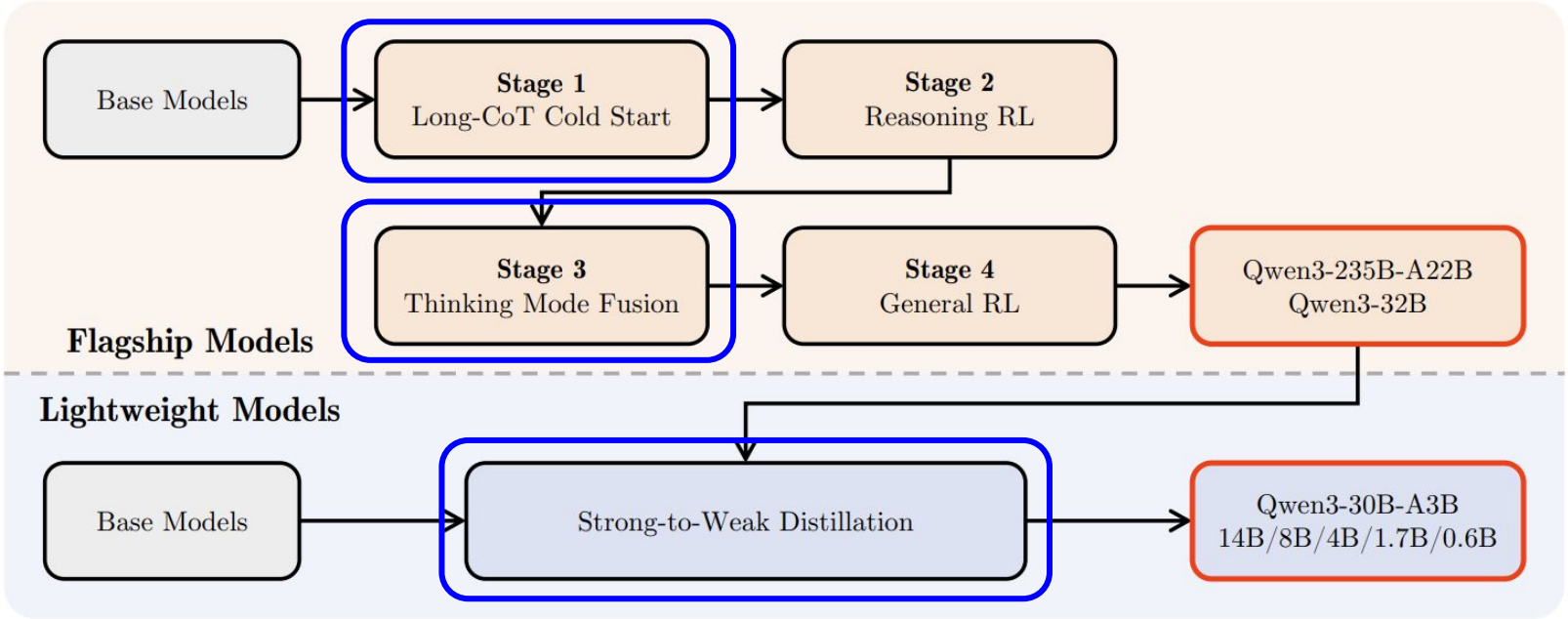


Zihan Liu, Zhuolin Yang, Yang Chen, Chankyu Lee, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. Acereason-nemotron 1.1: Advancing math and code reasoning through sft and rl synergy. arXiv preprint arXiv:2506.13284, 2025i.

Training Curriculum: SFT Strategy

- Iterative SFT (before)
  - Model improves and generates better synthetic data for subsequent iterations
- Multi-stage SFT (now)
  - Coarse-to-fine SFT: initial stages leverage large-scale, lower-quality data, while later stages refine the model using smaller but higher-quality datasets
- Lightweight vs. heavy SFT
  - SFT can over-constrain the model, restricting exploration during the online RL stage (lightweight SFT is suggested in Llama 4 post-training recipe)

Training Curriculum: SFT Strategy



An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025a.