Q&A 3: Post-Training Recipe Evolution and Advice

RLHFBOOK.COM/COURSE

Nathan Lambert

Corrections and questions from the GitHub, Discord, and YouTube comments through Lecture 11.

Q1: “Should capabilities be trained separately, then merged?”

A viewer asks:

> *Hi Nathan, thanks for another great video on post-training! At @24:21, this got me thinking (and I might be misunderstanding): if each capability is sort of its own “mode,” would it make sense to post-train stage by stage — coding → math → agentic tool use → etc. — learning one mode at a time? My guess is this intuition is wrong somewhere, since it seems like people don’t really do this and still put a lot of care into the data mixture. Would love to understand what I’m missing here.*

Q1: “Should capabilities be trained separately, then merged?”

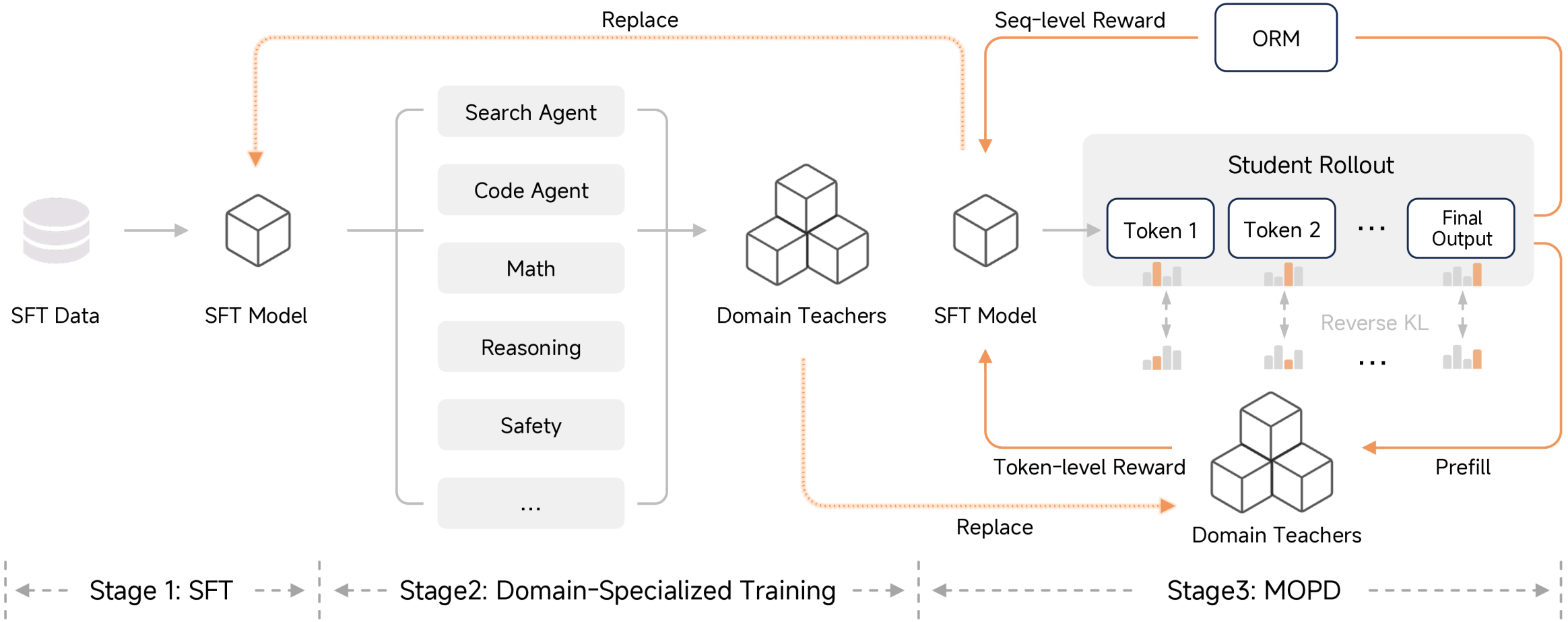
A viewer asks – should post-training be sequential rather than a strong mix.

In industry post-training has a shocking number of stages and sequencing. It is usually not just per-capability in a line, rather with forking & merging, with some sequence too.

The emerging version trains **specialist teachers in parallel**, then consolidates them into one student with multi-teacher on-policy distillation (MOPD). See Conversation 1 (w/ Finbarr Timbers).

Source: @mcmc6052 on YouTube (Lecture 10)

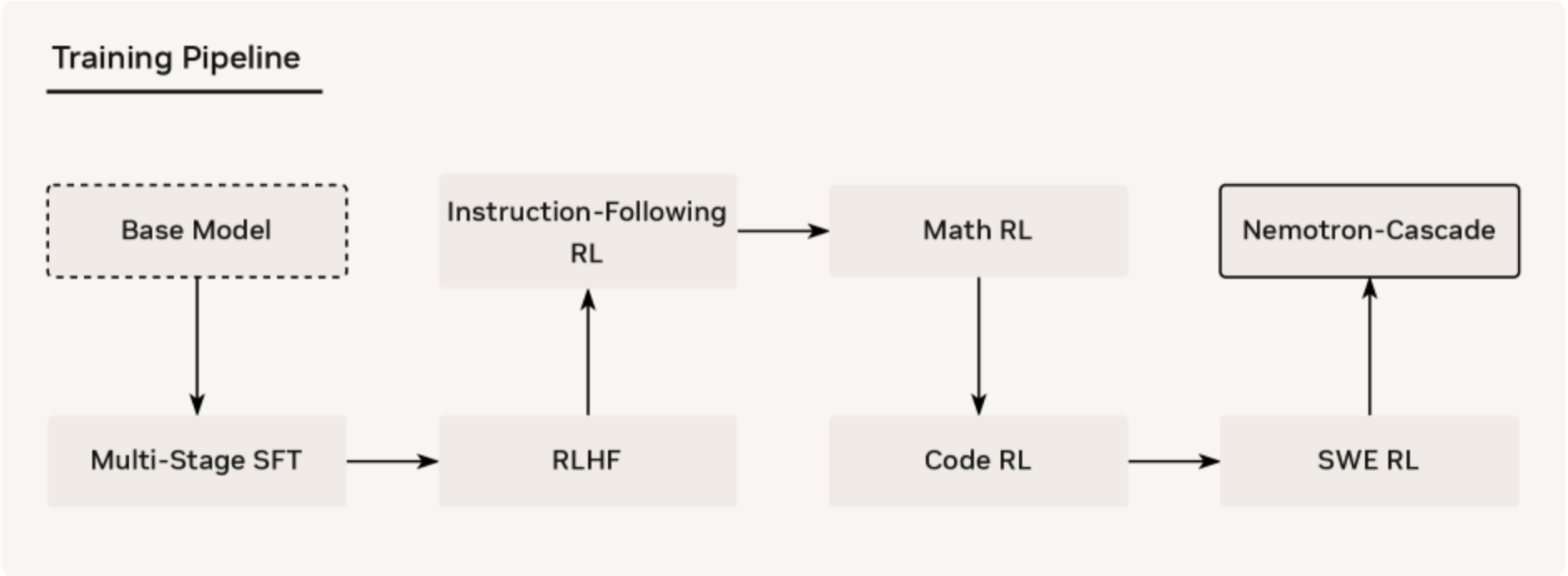
MOPD example (covered more in Conversation 1 / Lecture 7)



MiMo-V2-Flash trains domain specialists before consolidating them into one student with MOPD.

Sources: MiMo-V2-Flash, MOPD

Cascade RL: A sequential recipe

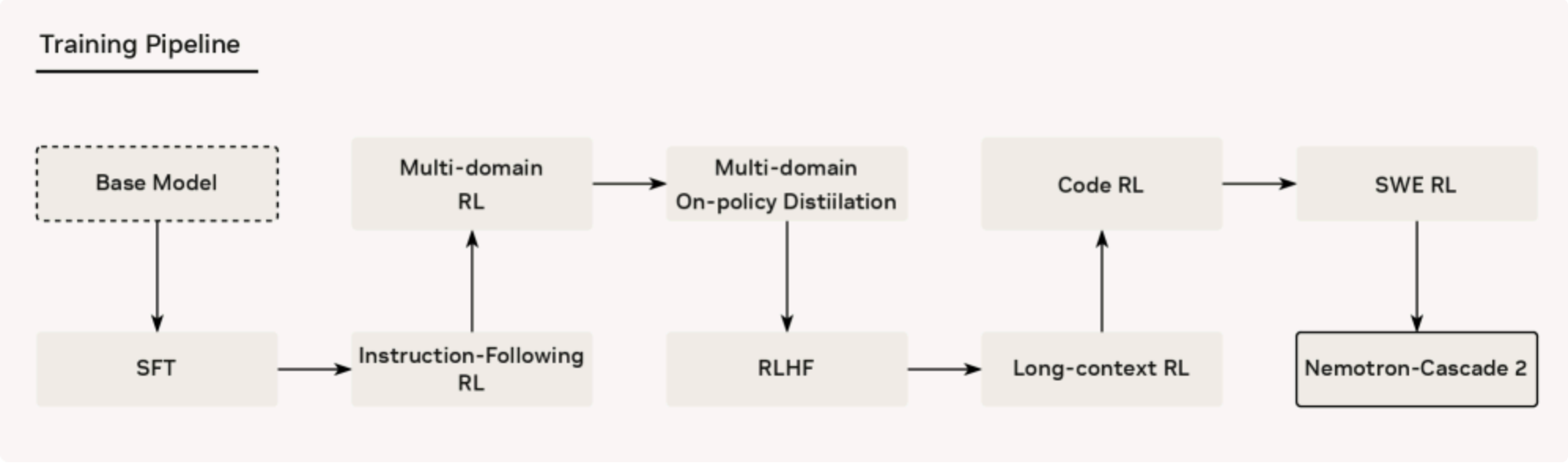


Nemotron-Cascade: sequential, domain-wise RL stages after SFT.

A fun paper! Integrated a bit into Nemotron, but not accepted as state-of-the-art.

Source: (NVIDIA, 2025)

Cascade RL 2: A sequential recipe + on-pol distillation



Nemotron-Cascade 2 folds multi-domain on-policy distillation into the cascade.

Read: Nemotron-Cascade 2 on arXiv.

Source: (NVIDIA, 2026)

Q2: “Why does RLVR often omit the KL penalty?”

A follow-up from the Lecture 10 comments:

> *And on a related note, it seems like for RLVR the default is to drop the KL term entirely. My understanding is that this is because in RLVR the reward is verifiable ground truth, so we don’t run into the over-optimization problem you get in RLHF, where a learned reward model is only a proxy (and arguably a proxy of a proxy, since the preference data itself approximates what we actually care about). Is that the right way to think about it?*

Overall yes. We even found during Tülu 3 that in RLVR the models are very good at selectively updating behaviors – e.g. learning math without OOD degradation.

Source: @mcmc6052 follow-up on YouTube (Lecture 10)

Some more β choices

Dropped it

- Open-Reasoner-Zero (Hu et al., 2025), Magistral (Mistral AI, 2025), and Skywork OR-1 (He et al., 2025) ran RLVR at $\beta = 0$ (Lecture 5's list).
- DAPO removed it explicitly so long reasoning chains can explore (Yu & others, 2025).
- GLM-5 (2026): "we remove the KL regularization term to accelerate RL improvement" (report, §3.2).

Kept (a version of) it

- DeepSeek-R1's GRPO objective ships with a $\beta\, D_{\mathrm{KL}}$ term to the reference model (Guo et al., 2025).
- Kimi K3's report never mentions KL in its RL objective — the frontier default is quiet omission.

Q3: “How does the ~1M SFT prompt budget scale with model size?”

From the Lecture 2 comments:

> *You mention some pocket rules for prompt budgets, e.g. ~1M prompts for SFT. In your experience, how does this number scale with the size of the model being post-trained?*

Mostly, bigger models tend to need less data (e.g. Tülu 3 405B). They’re more sample efficient at RL (in terms of steps, not total GPU hours) and need less SFT data (or much of SFT data doesn’t help them). I’d read this as bigger models need fewer tokens/capability, which fits scaling laws.

This has confounders today, as midtraining and SFT are almost indistinguishable in some ways. They both seed the reasoning behaviors in the models.

Source: @secmt on YouTube (Lecture 2); budgets from (Lambert et al., 2024)

Q4: “RL is basically just SFT with a regularization term?”

A great question from the Lecture 3 comments:

> *Hi Nathan. i have a question about some recent papers about RL vs. SFT for LLMs. So the conventional belief is that RL is good for hardcore reasoning tasks whereas SFT is good for getting the structure and formatting, which is basically why we had 2-3 phases in post-training for so long so we believe they do different things. But there are always new papers popping up saying RL is basically just SFT with some regularized term, and if you do their regularized version of SFT, you can get the same result and then they give you a bunch of plots and graphs in their paper demonstrating that to you. How believable/generalizable is this? I can’t imagine it generalizes to big models; otherwise, DeepSeek and GLM would already be doing it…*

On the premise: I’d say it’s less “SFT for formatting, RL for reasoning” and more that **SFT is the foundation** (and very compute efficient these days), then **RL is how the model learns a lot more from there** — on harder problems, and adapting to the harnesses it’ll be used in at deployment.

Both are impossible to get around right now. Still, **great models can be made with mostly SFT** as the field moves so fast. Any good data work will get you far.

Source: @entropic4439 on YouTube (Lecture 3)

The similarities in all the loss functions are very real

Yes — the two updates share their core term, and on curated targets SFT is the "all-positive" special case (same can be shown for DPO). That shared algebra is what drives iw-SFT (Qin & Springenberg, 2025), DFT (Wu et al., 2025), and the unified-view papers (Lv et al., 2025).

A huge difference: **negative gradients**. RL pushes below-baseline samples *down* — it learns from its own mistakes, which SFT structurally cannot do. + the power of on-policy learning to regularize and nudge a model.

For the math see Lecture 10.

Sources: (Qin & Springenberg, 2025), (Wu et al., 2025), (Lv et al., 2025)

Q5: “When should in-house traces be trained into the weights vs. handled with ICL?”

From the Lecture 7 comments:

> *When should in-house traces be learned into the weights rather than handled with in-context learning?*

Mostly it comes down to cost, how much inference you’re doing, and what the improvements are worth monetarily. Build evaluations to see where you’re at, but assume any meaningful training effort is at least millions or tens of millions.

Source: @Decocoa on YouTube (Lecture 7)

Q6: “Can continued pretraining on a distillation-only dataset make a model worse?”

From the Lecture 7 comments.

Distillation data as a category doesn’t make it good or bad. It depends on the teacher model, the student model, the filtering, the generation procedure etc. I’m sure people have made both failed models and SOTA models with continued pretraining on synthetic data.

Source: @sankalpsingh7150 on YouTube (Lecture 7)

Thanks for watching

Questions, comments, and future Q&A prompts:

- Course: rlhfbook.com/course
- Discord: discord.gg/yz5AwK4gBR
- GitHub: github.com/natolambert/rlhf-book
- Newsletter: interconnects.ai
- Contact: nathan@natolambert.com

References (1/2)

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., et al.. "*Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*." *arXiv preprint arXiv:2501.12948*, 2025.

He, J., Liu, J., Liu, C., and others. "*Skywork Open Reasoner 1 Technical Report*." *arXiv preprint arXiv:2505.22312*, 2025.

Hu, J., Zhang, Y., Han, Q., Jiang, D., Zhang, X., et al.. "*Open-Reasoner-Zero: An Open Source Approach to Scaling Up Reinforcement Learning on the Base Model*." *arXiv preprint arXiv:2503.24290*, 2025.

Lambert, N., Morrison, J., Pyatkin, V., Huang, S., Ivison, H., et al.. "*Tülu 3: Pushing Frontiers in Open Language Model Post-Training*." *arXiv preprint arXiv:2411.15124*, 2024.

Lv, X., Zuo, Y., Sun, Y., Liu, H., Wei, Y., et al.. "*Towards a Unified View of Large Language Model Post-Training*." 2025. [link]

Mistral AI. "*Magistral: Scaling Reinforcement Learning for Reasoning in Large Language Models*." 2025. [link]

NVIDIA. "*Nemotron-Cascade: Scaling Cascaded Reinforcement Learning for General-Purpose Reasoning Models*." *arXiv preprint arXiv:2512.13607*, 2025. [link]

NVIDIA. "*Nemotron-Cascade 2: Post-Training LLMs with Cascade RL and Multi-Domain On-Policy Distillation*." *arXiv preprint arXiv:2603.19220*, 2026. [link]

References (2/2)

Qin, C., and Springenberg, J.. "*Supervised Fine Tuning on Curated Data is Reinforcement Learning (and can be improved)*." 2025. [link]

Wu, Y., Zhou, Y., Zhou, Z., Peng, Y., Ye, X., et al.. "*On the Generalization of SFT: A Reinforcement Learning Perspective with Reward Rectification*." 2025. [link]

Yu, Q., and others. "*DAPO: An Open-Source LLM Reinforcement Learning System*." *arXiv preprint arXiv:2503.14476*, 2025.