Midtraining Bridges Pretraining and Posttraining Distributions

Emmy Liu, Graham Neubig, Chenyan Xiong

mailto:emmy@cmu.edu

Language Technologies Institute

What is Midtraining?

- Midtraining is an intermediate phase of training between pretraining and posttraining, usually in the "cooldown" of pretraining, mostly using specialized data such as math, code, and instruction data
- Most frontier models now have a midtraining phase. However, why is this effective and what makes a good mix?

We conduct experiments from scratch on the role of midtraining, and find that it reduces catastrophic forgetting and improves performance on posttraining data through "bridging".

Intuition

- Pretraining→SFT causes forgetting through conflicting gradients between SFT data and pretraining data
- One mechanism that we can use to reduce this conflict is by finding a better initialization to start from a lower-conflict point
- By incorporating midtraining data, we can find a better initialization point of posttraining
- Plasticity also plays a role – better to start earlier if end posttraining is more different from pretrain data

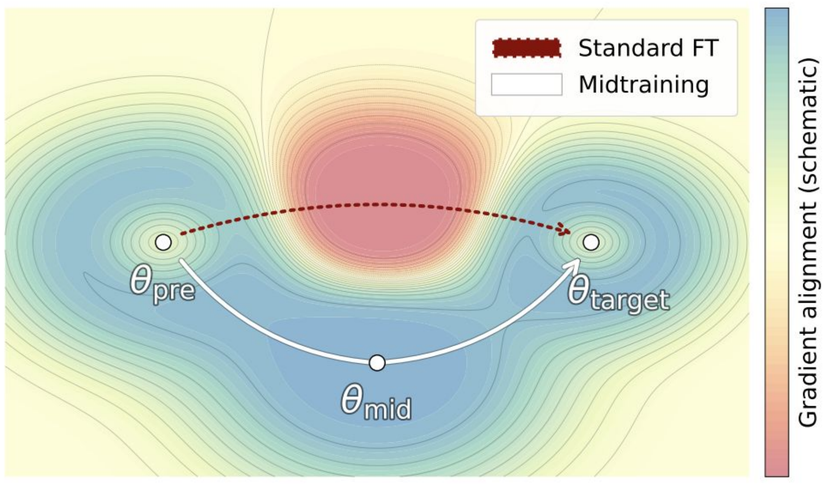


Figure 1. Schematic optimization landscape, where contours indicate gradient conflict between pretraining and posttraining objectives. Standard pretraining→SFT follows the red path, while pretraining→midtraining→SFT follows the white path. Midtraining shifts the initialization so SFT can approach the target while avoiding high-conflict regions.

Training Setup

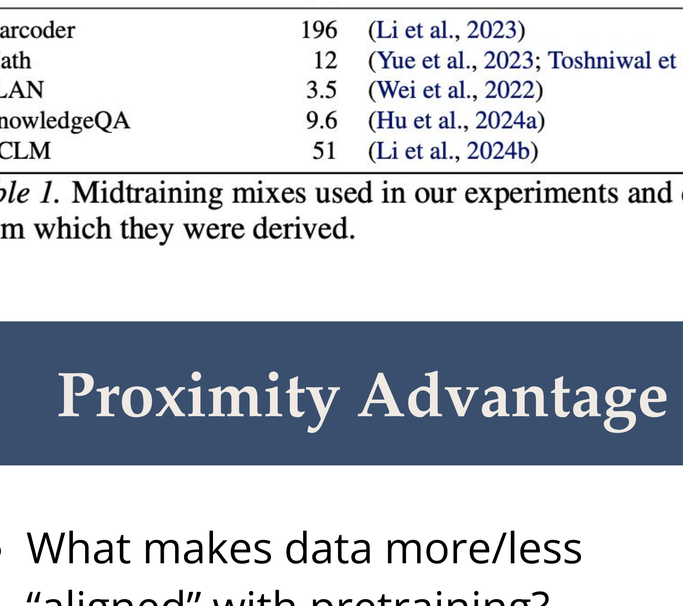
- Pretrain Pythia family models (70M-1B) from scratch on 128B tokens on C4
- Study effects of 5 common types of midtraining data, introduced at different steps in training
- Pair with domain-matched postraiin datasets for SFT (Starcoder, GSM8k, LIMA, SciQ)
- Evaluate both in-domain and C4 (forgetting) LM loss of the vanilla/midtrained model after SFT

Midtrain mix	Num. Tokens (B)	Sources
Starcoder	196	(Li et al., 2023)
Math	12	(Yue et al., 2023; Toshniwal et al., 2024)
FLAN	3.5	(Wei et al., 2022)
KnowledgeQA	9.6	(Hu et al., 2024a)
DCLM	51	(Li et al., 2024b)

Table 1. Midtraining mixes used in our experiments and dataset(s) from which they were derived.

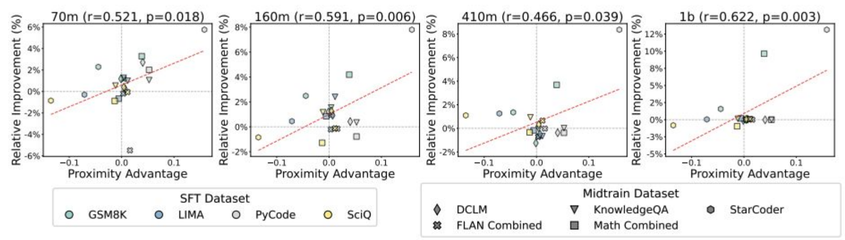
Proximity Advantage

- What makes data more/less "aligned" with pretraining?
- We stick to a simple (unigram) based definition for convenience



Takeaways

1. **Midtraining benefits are domain specific, particularly strong for math and code**
2. **Midtraining gains can be well-predicted by proximity advantage**



3. **Maintaining a mix with pretraining data >> Switching entirely to domain-specific data**
4. **Timing is important, high mixture weights of specialized data are tolerated well earlier but hurt later**

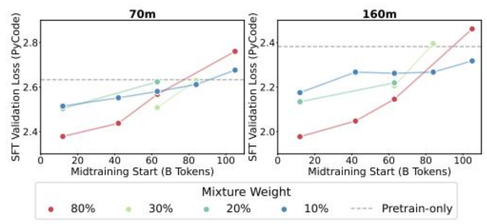


Figure 4. Effect of mixture weight and midtraining phase start on in-domain validation loss for the code mixture. A high mixture weight is beneficial when the midtraining phase begins early, but is detrimental when beginning this phase later.

Open Questions

- Are there good ways to optimize midtraining data mixtures as well as the introduction times of various types of data?
- How to leverage midtraining for larger LMs and for specialized/domain specific use cases?
- Understanding extensions to RL and multi-stage curricula

Read the paper!