Investigating How Small Transformers Learn Arithmetic: Self-Improvement and Beyond

Zifei Bai*, Fang Yu*, Zhiwei Xu, Yixin Wang

Abstract

Large Language Models (LLMs) often struggle with length generalization, and numerous self-improvement frameworks have been proposed to address this issue. However, training on wrong answers accumulated at each round will lead to model collapse in the end. Yet, what if the model could self-correct those errors? In our research, we reproduced a self-improvement framework and explore how transformers can achieve self-correction.

Model Architecture

The architecture follows modern LLM design principles with pre-normalization, multi-head causal attention, and SwiGLU activation similar to LLaMA. Flash Attention optimization is used when available for improved computational efficiency.

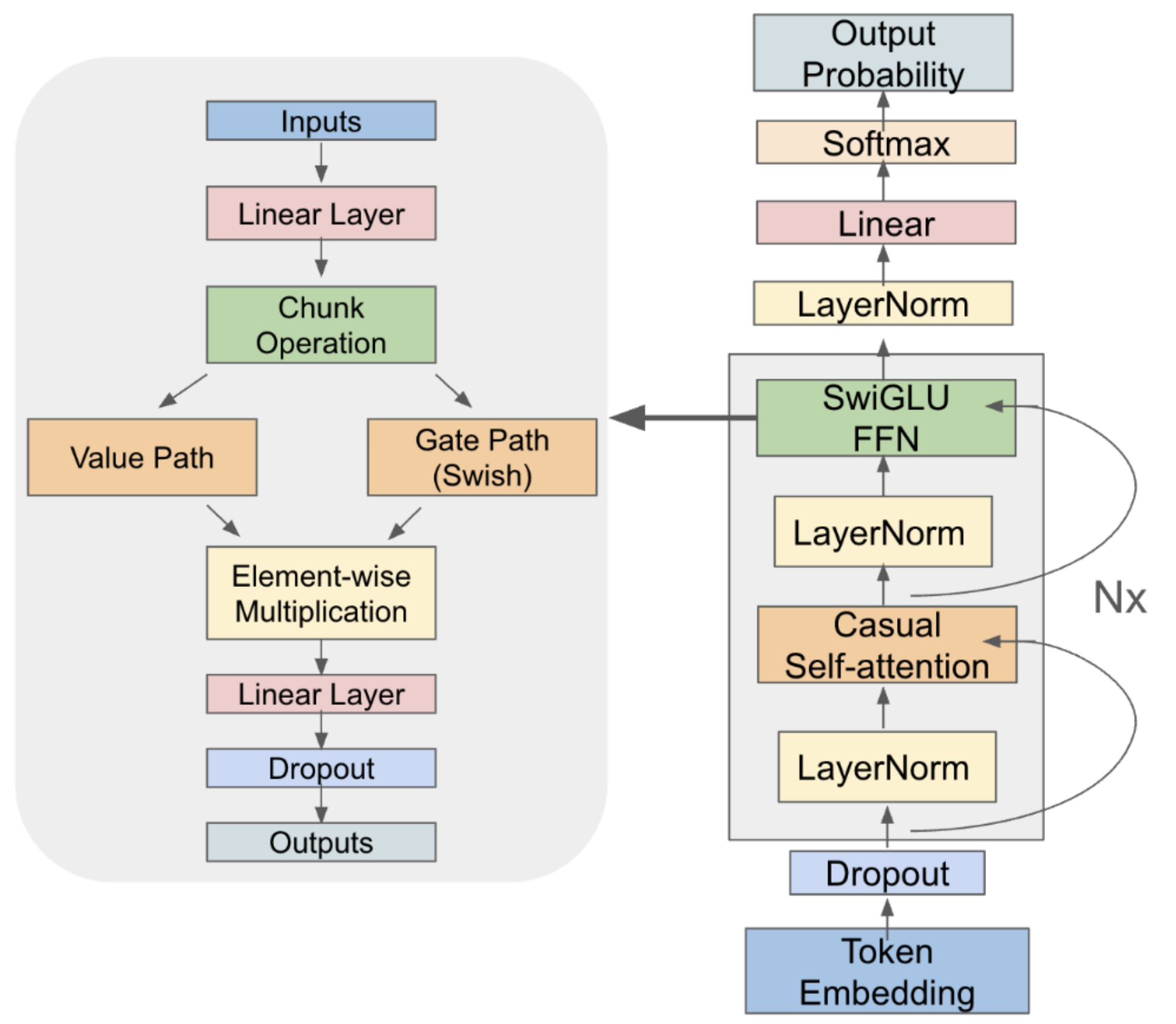


Figure 1. Model Architecture

Self-Improvement Framework

For each self-improvement round r:

⇒ Use M_{r-1} to generate OOD samples, with length $10+r$

⇒ Go through filters until reach 50,000 synthetic data:

⇒ Majority Voting

⇒ Length Filter

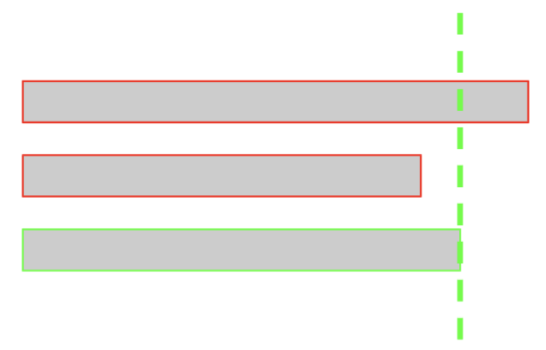
⇒ Train on a combined dataset to get M_r:

$$\underbrace{20\mathrm{M} + (r-1)\times 50000}_{50\%} + \underbrace{50000}_{50\%}$$

⇒ Evaluate M_r on 11-21 digits

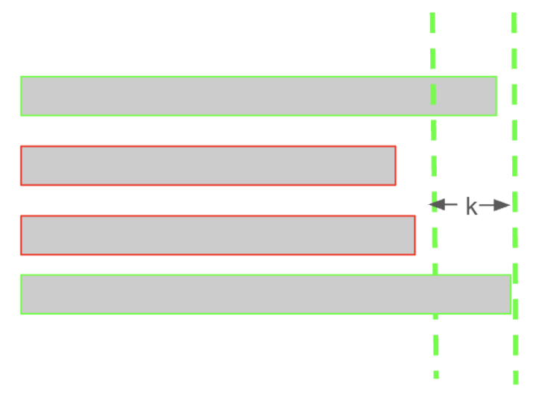
Length Filter

Supervised Length Filtering:



- Filter out the outputs that does not match exactly as the prompt length.
- Strong supervised length filtering for novelty.

Unsupervised Length Filtering:



- Find max length $\mathcal{L}$ within a batch
- Filter examples shorter than ($\mathcal{L}$ - k)

Majority Voting

- Initialize 5 models with different random seeds and train them in parallel.
- For each prompt: if at least 3 out of 5 models generate the same answer, it is saved as a synthetic sample.
- **Note:** If 3 out of 5 models agree on an incorrect answer, the sample is still accepted.

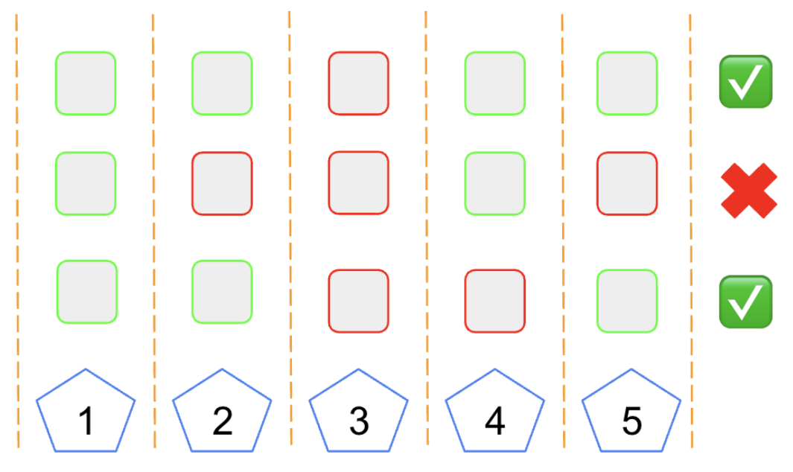


Figure 2. Majority Voting

Results

We performed ten rounds of self-improvement and saved the model checkpoint after each round. For each model M_r, we evaluated its accuracy on inputs of length $10+r$, where $r \in \{1,\ldots,10\}$.

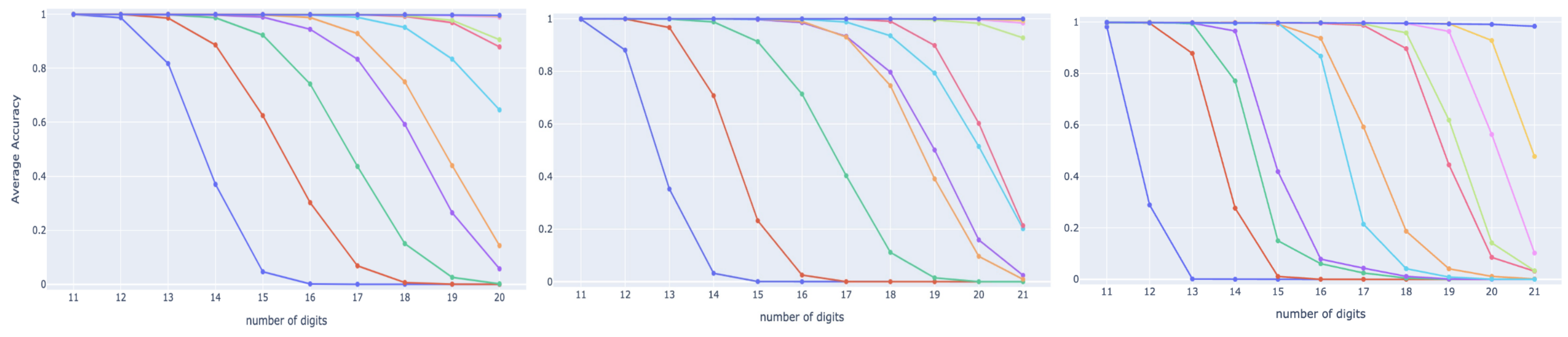


Figure 3. String Copy task: left is advanced no-filter; middle is length filter; right is majority voting

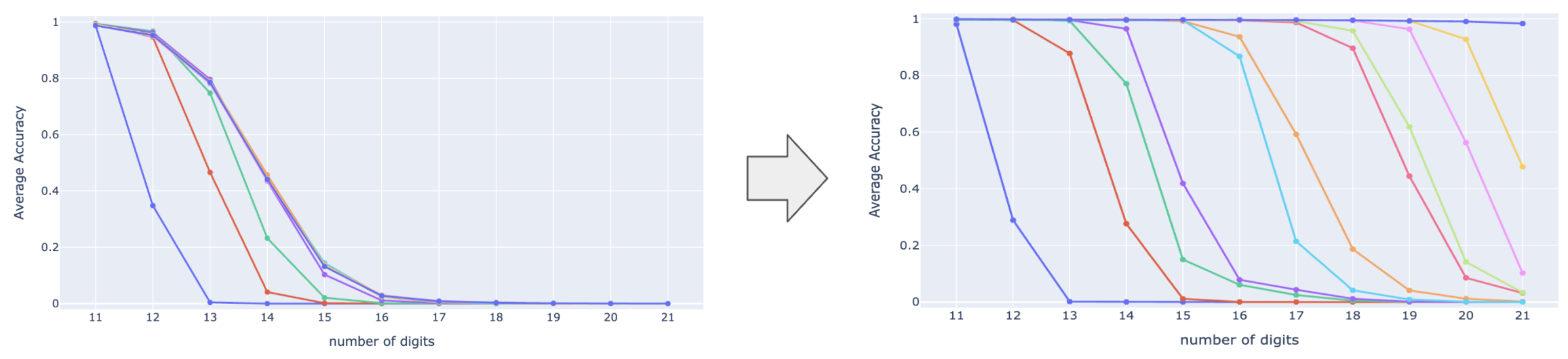


Figure 4. Reverse Addition task: left is no-filter; right is majority voting

Self-Correction Behavior

- As self-improvement rounds progress, error accumulation can also increase, leading to the model being trained on an increasing number of incorrect samples — which may ultimately result in model collapse.
- To analyze this risk, we extract and store incorrect outputs from 50,000 self-improvement samples at each round r.
- Evaluate whether the models trained from round $r+1$ to $r+10$ are able to correct those wrong answers in earlier rounds.
- For example, the blue line in Figure 4 right shows that the model trained at round 3 can correct 90% of the errors made at round 2.
- The positive upward trend across curves in Figure 4 suggests that LLMs exhibit self-correction behavior, potentially offsetting the risks of error accumulation.

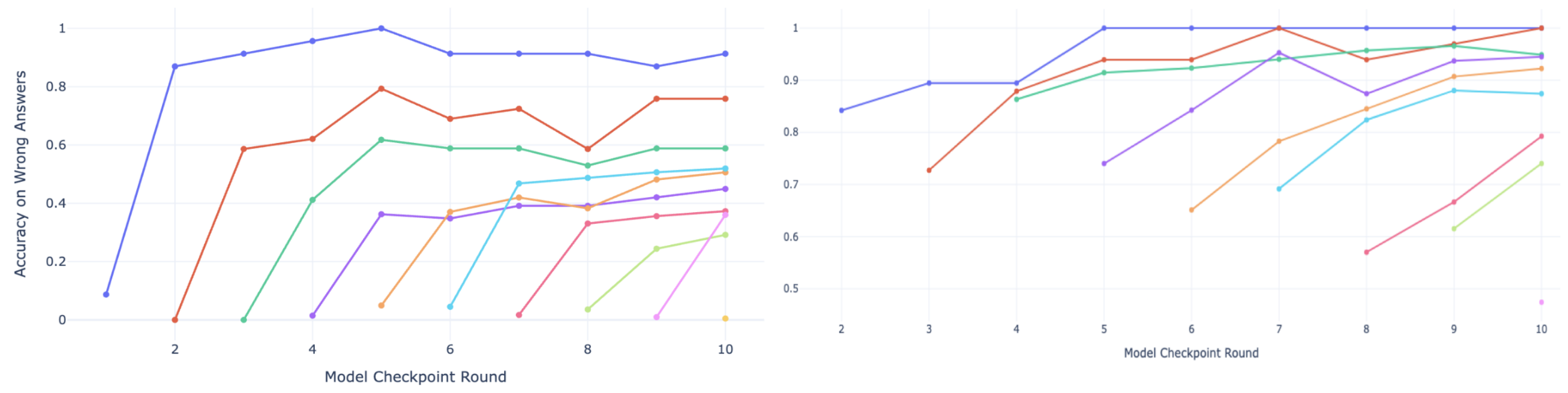


Figure 5. Self-Correction behavior, left: String Copy; right: Reverse Addition

Discussion

- Explore why the Transformer model without positional encoding naturally has a certain length generalization ability, while the model with RoPE as positional encoding does not have such ability.
- Explore the potential phenomenon: where does the ability to self-correct come from — whether the model can self-correct on easier tasks by training on harder tasks, or whether the model can acquire this ability by continuing to train on tasks of equal difficulty.
- Conduct more experiments on complex tasks such as multi-operand addition and multiplication.

References

[1] Lee, Nayoung, et al. *Teaching arithmetic to small transformers.* arXiv preprint arXiv:2307.03381 (2023).

[2] Lee, Nayoung, et al. *Self-Improving Transformers Overcome Easy-to-Hard and Length Generalization Challenges.* arXiv preprint arXiv:2502.01612, 2025.

zifeibai@umich.edu