Synthetic Data and Self-Improvement

REFORM reading group 2/26

Keertana Chidambaram, Charlotte Peale

Model Collapse Demystified: The Case of Regression

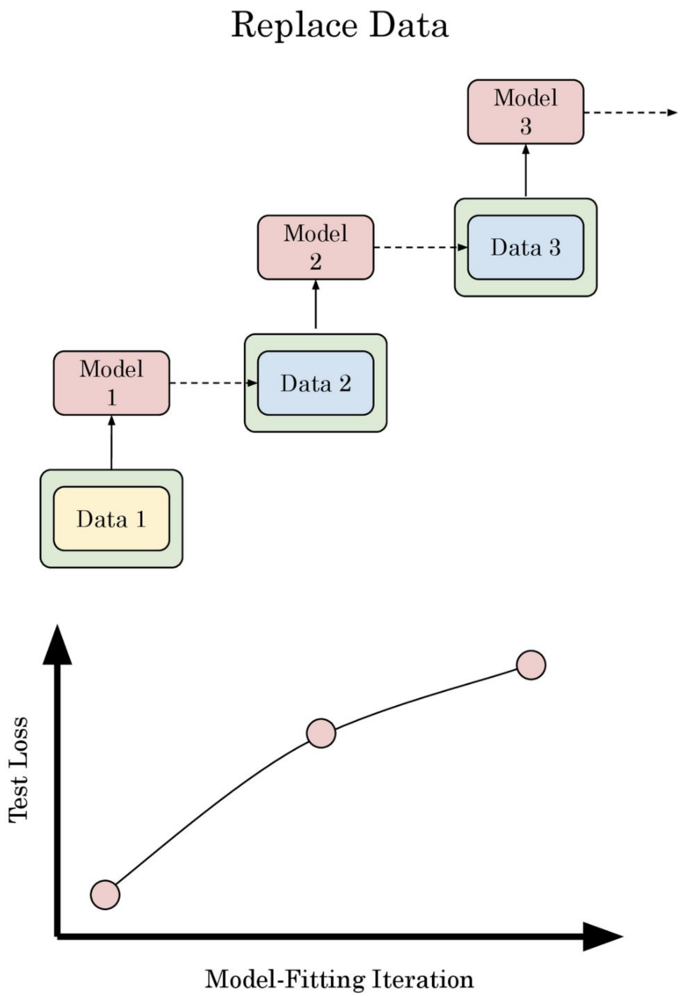
Elvis Dohmatob, Yunzhen Feng, Julia Kempe

Motivation

Internet data will be polluted with LLM generated data

Model collapse: repeated training on AI-generated data degenerates performance

This paper: theoretical explanation for this trend



Simplified Data Distribution Model

Suppose the "true" data distribution is a **linear** model

$$\left.\begin{array}{l}\textbf{(Input) } x \sim N(0, \Sigma), \\ \textbf{(Noise) } \epsilon \sim N(0, \sigma^2), \ \textbf{independent of } x \\ \textbf{(Output / Label) } y = x^\top w_0 + \epsilon.\end{array}\right\}$$

Simplified Data Distribution Model

Suppose the "true" data distribution is a **linear** model

And we want to minimize the error (excess risk)

$$E_{test}(\widehat{w}) := \mathbb{E}_{\widehat{w}}\, \mathbb{E}_{x,y}[(x^\top \widehat{w} - y)^2] - \sigma^2$$
$$= \mathbb{E}_{\widehat{w}}\, [\|\widehat{w} - w_0\|_\Sigma^2],$$

where $(x, y) \sim P_{\Sigma, w_0, \sigma^2}$ is a random clean test point.

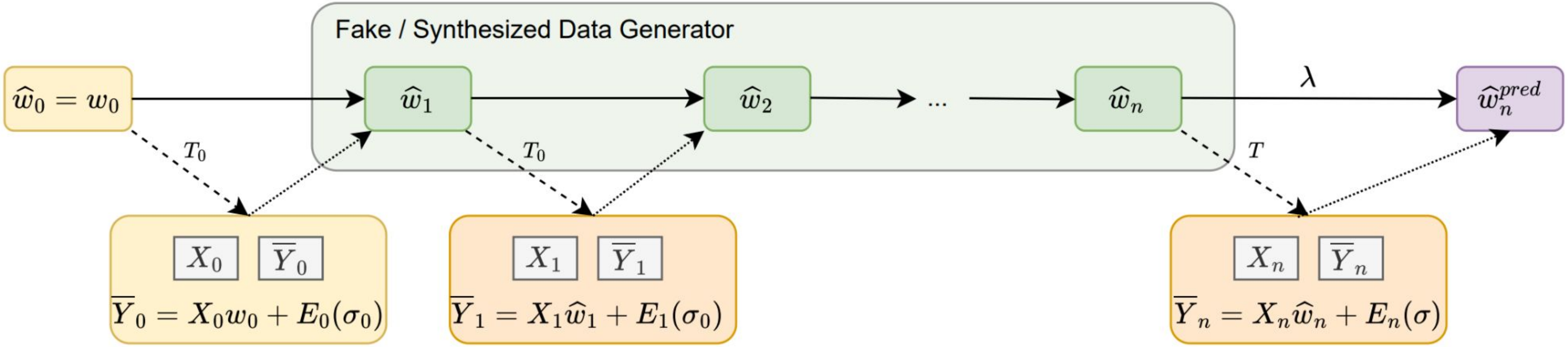
Simplified Data Distribution Model

Suppose the “true” data distribution is a **linear** model

And we want to minimize the error (excess risk)

What if we train models iteratively, with each model using the previous to generate data labels?

Data Generation Process



OLS Warmup

Setting: n models are fit in succession, $T > d + 2$ (under-parametrized regime)

OLS Warmup

Setting: n models are fit in succession, T > d + 2 (under-parametrized regime)

Algorithm: Fit OLS at each round with data labels from previous rounds!

OLS Warmup

Setting: n models are fit in succession, T > d + 2 (under-parametrized regime)

Algorithm: Fit OLS at each round with data labels from previous rounds!

Theoretical Bound:

$$E_{test}(\widehat{w}_n^{pred}) \simeq \frac{\sigma^2\phi}{1-\phi} + \frac{n\sigma_0^2\phi_0}{1-\phi_0} \qquad \textit{with}\ \phi = \frac{d}{T},\ \phi_0 = \frac{d}{T_0}$$

σ, ϕ are error noise and d/T for first round

σ0, ϕ0 are error noise and d/T for subsequent rounds

OLS Warmup

Theoretical Bound:

$$E_{test}(\widehat{w}_n^{pred}) \simeq \frac{\sigma^2 \phi}{1-\phi} + \frac{n\sigma_0^2 \phi_0}{1-\phi_0} \qquad with\ \phi = \frac{d}{T},\ \phi_0 = \frac{d}{T_0}$$

Observations:

1. Irreducible + scaling errors

OLS Warmup

Theoretical Bound:

$$E_{test}(\widehat{w}_n^{pred}) \simeq \frac{\sigma^2\phi}{1-\phi} + \frac{n\sigma_0^2\phi_0}{1-\phi_0} \qquad \textit{with } \phi = \frac{d}{T},\ \phi_0 = \frac{d}{T_0}$$

Observations:

1. Irreducible + scaling errors
2. If true data is bad then that error will carry over

OLS Warmup

Theoretical Bound:

$$E_{test}(\widehat{w}_n^{pred}) \simeq \frac{\sigma^2\phi}{1-\phi} + \frac{n\sigma_0^2\phi_0}{1-\phi_0} \qquad with\ \phi = \frac{d}{T},\ \phi_0 = \frac{d}{T_0}$$

Observations:

1. Irreducible + scaling errors
2. If true data is bad then that error will carry over
3. Another error term scales with n

Lessons so far

1. **Repeatedly training on “fake” data incurs error linearly growing with n**

OLS Warmup

Theoretical Bound:

$$E_{test}(\widehat{w}_n^{pred}) \simeq \frac{\sigma^2 \phi}{1-\phi} + \frac{n\sigma_0^2 \phi_0}{1-\phi_0} \qquad with\ \phi = \frac{d}{T},\ \phi_0 = \frac{d}{T_0}$$

Observations:

1. Irreducible + scaling errors
2. If true data is bad then that error will carry over
3. Another error term scales with n

Idea 1: what if T0 is large?

Scaling Synthetic Data Size

Say you scale the synthetic data linearly with the round, data = mT

Scaling Synthetic Data Size

Say you scale the synthetic data linearly with the round, data = mT

$$
\begin{aligned}
E_{test}(\widehat{w}_n^{pred}) &\simeq (1 + \frac{1}{2} + \frac{1}{3} + \ldots) E_{test}(\widehat{w}_0^{pred}) \\
&\simeq \log n \cdot E_{test}(\widehat{w}_0^{pred}),
\end{aligned}
$$

Scaling Synthetic Data Size

Say you scale the synthetic data linearly with the round, data = mT

$$
\begin{aligned}
E_{test}(\widehat{w}_n^{pred}) &\simeq (1 + \frac{1}{2} + \frac{1}{3} + \ldots) E_{test}(\widehat{w}_0^{pred}) \\
&\simeq \log n \cdot E_{test}(\widehat{w}_0^{pred}),
\end{aligned}
$$

Trade-off collapse with **more data** and **more compute**

The model still collapses but at a slower rate!

Lessons so far

1. Repeatedly training on “fake” data incurs error linearly growing with n

2. **Dramatically increasing generated synthetic data doesn’t fix the problem**

Motivation for Regularization

Consider the null predictor (i.e. setting all weights to 0)

$$E_{test}(w_{null}) = \|w_0\|_{\Sigma}^2$$

Motivation for Regularization

Consider the null predictor (i.e. setting all weights to 0)

$$E_{test}(w_{null}) = \|w_0\|_{\Sigma}^2$$

If we compare this with the weights learned on the nth round we get

$$\frac{E_{test}(\widehat{w}_n^{pred})}{E_{test}(w_{null})} = \frac{1}{\mathrm{SNR}} \frac{\phi}{1-\phi} + \frac{n}{\mathrm{SNR}_0} \frac{\phi_0}{1-\phi_0}$$

$$\mathrm{SNR} := \|w_0\|_{\Sigma}^2/\sigma^2 \text{ and } \mathrm{SNR}_0 := \|w_0\|_{\Sigma}^2/\sigma_0^2$$

Motivation for Regularization

Consider the null predictor (i.e. setting all weights to 0)

$$E_{test}(w_{null}) = \|w_0\|_\Sigma^2$$

If we compare this with the weights learned on the nth round we get

$$\frac{E_{test}(\widehat{w}_n^{pred})}{E_{test}(w_{null})} = \frac{1}{\mathrm{SNR}} \frac{\phi}{1-\phi} + \frac{\boxed{n}}{\mathrm{SNR}_0} \frac{\phi_0}{1-\phi_0}$$

$$\mathrm{SNR} := \|w_0\|_\Sigma^2/\sigma^2 \text{ and } \mathrm{SNR}_0 := \|w_0\|_\Sigma^2/\sigma_0^2$$

The ratio linearly scales with n!

Motivation for Regularization

Consider the null predictor (i.e. setting all weights to 0)

$$E_{test}(w_{null}) = \|w_0\|_\Sigma^2$$

If we compare this with the weights learned on the nth round we get

$$\frac{E_{test}(\widehat{w}_n^{pred})}{E_{test}(w_{null})} = \frac{1}{\text{SNR}}\frac{\phi}{1-\phi} + \frac{\boxed{n}}{\text{SNR}_0}\frac{\phi_0}{1-\phi_0}$$

$$\text{SNR} := \|w_0\|_\Sigma^2/\sigma^2 \text{ and } \text{SNR}_0 := \|w_0\|_\Sigma^2/\sigma_0^2$$

The ratio linearly scales with n!

Idea 2: Regularization

Ridge Regression

Idea: use OLS + L2 regularization (Ridge) to reduce complexity

Ridge Regression

Idea: use OLS + L2 regularization (Ridge) to reduce complexity

Case 1: T >= d + 2 (under-parametrized regime)

Error = **error (only clean data)** + **n x scaling factor**

= **bias + variance** + **n x scaling factor**

Ridge Regression

Idea: use OLS + L2 regularization (Ridge) to reduce complexity

Case 1: T >= d + 2 (under-parametrized regime)

Error = **error (only clean data)** + **n x scaling factor**

= **bias + variance** + **n x scaling factor**

Case 2: T < d + 2 (over-parametrized regime)

Error = **new bias** + **variance** + **n x another scaling factor**

Moreover **new bias** **>** **bias**

Lessons so far

1. Repeatedly training on “fake” data incurs error linearly growing with n

2. Dramatically increasing generated synthetic data doesn’t fix the problem

3. **Simple regularization also doesn’t fix the problem**

Adaptive Regularization (Simplified)

Assumption: spectral conditions on the feature covariance matrix

$$\left.\begin{aligned}&\textbf{(Capacity Condition) } \lambda_j \asymp j^{-\beta} \text{ for all } j \in [d],\\ &\textbf{(Source Condition) } \|\Sigma^{1/2-r} w_0\| = O(1),\end{aligned}\right\}$$

Capacity: how dispersed are the Xs

Source: how dispersed is w0 in relation to spectrum of feature covariance matrix

Adaptive Regularization (Simplified)

Assumption: spectral conditions on the feature covariance matrix

$$\left.\begin{aligned} &\textbf{(Capacity Condition)}\ \lambda_j \asymp j^{-\beta} \text{ for all } j \in [d], \\ &\textbf{(Source Condition)}\ \|\Sigma^{1/2-r} w_0\| = O(1), \end{aligned}\right\}$$

Algorithm: allow for adaptive (decaying with samples T) regularization rate for the L2 regularizer

$$\begin{aligned} E_{test}(\widehat{w}_n^{pred}) \asymp \max(\sigma^2, T^{1-2\underline{r}\ell-\ell/\beta}) \cdot T^{-(1-\ell/\beta)} \\ + \frac{n\sigma_0^2}{1-\phi_0} \max\left(T/T_0, \phi_0\right) \cdot T^{-(1-\ell/\beta)} \end{aligned}$$

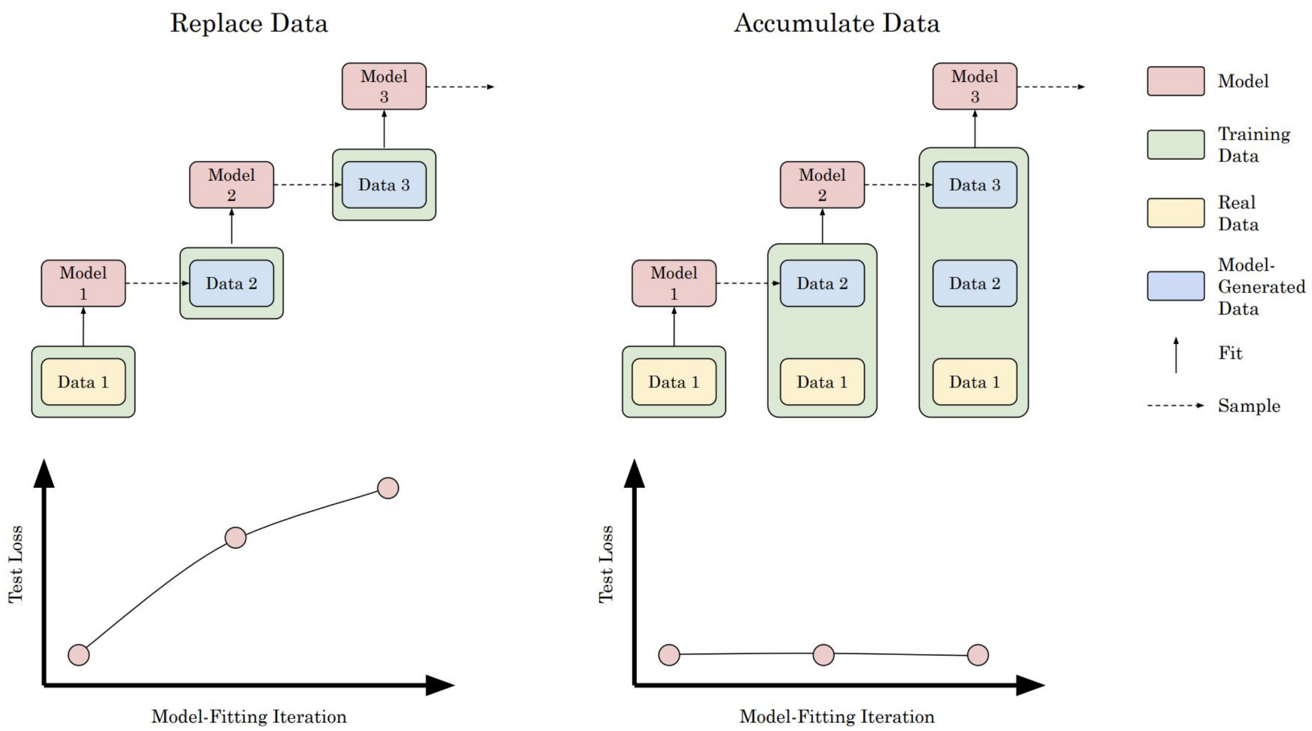
Lessons so far

1. Repeatedly training on “fake” data incurs error linearly growing with n
2. Dramatically increasing generated synthetic data doesn’t fix the problem
3. Simple regularization also doesn’t fix the problem
4. **For special cases, adaptive regularization helps alleviate model collapse**

Is Model Collapse Inevitable? Breaking the Curse of Recursion by Accumulating Real and Synthetic Data

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Henry Sleight, John Hughes, Tomasz Korbak, Rajashree Agrawal, Dhruv Pai, Andrey Gromov, Daniel A. Roberts, Diyi Yang, David L. Donoho, Sanmi Koyejo

Motivation



Replace Data
Accumulate Data
Model 1
Model 2
Model 3
Data 1
Data 2
Data 3
Test Loss
Model-Fitting Iteration
Model
Training Data
Real Data
Model-Generated Data
Fit
Sample

Data Generation Process

Same old setting model from Dohmatob et al

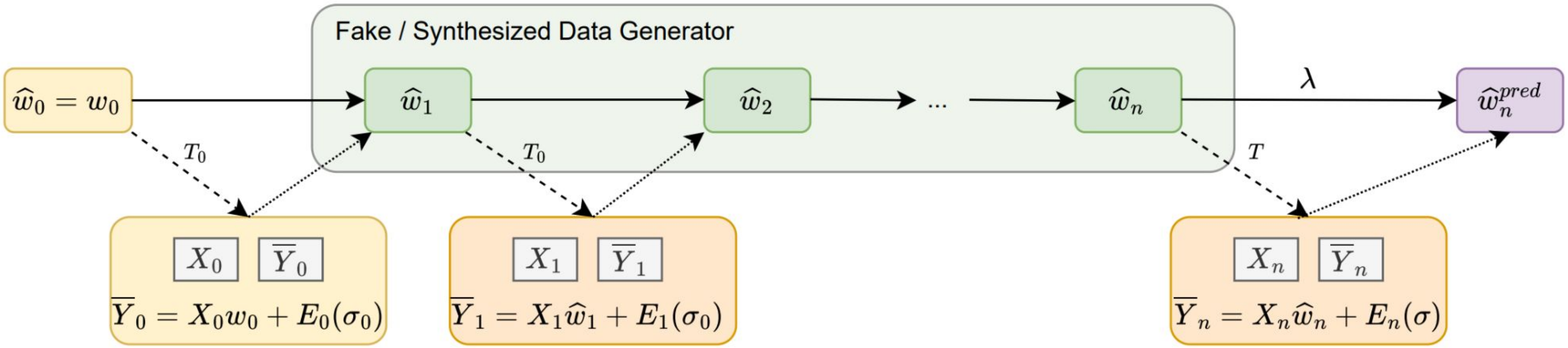
$$\left.\begin{array}{l}\textbf{(Input)}\ x \sim N(0, \Sigma),\\ \textbf{(Noise)}\ \epsilon \sim N(0, \sigma^2),\ \textbf{independent of}\ x\\ \textbf{(Output / Label)}\ y = x^\top w_0 + \epsilon.\end{array}\right\}$$

Minimize excess risk:

$$\begin{aligned} E_{test}(\widehat{w}) &:= \mathbb{E}_{\widehat{w}}\, \mathbb{E}_{x,y}[(x^\top \widehat{w} - y)^2] - \sigma^2 \\ &= \mathbb{E}_{\widehat{w}}\left[\|\widehat{w} - w_0\|_{\Sigma}^2\right], \end{aligned}$$

where $(x, y) \sim P_{\Sigma, w_0, \sigma^2}$ is a random clean test point.

Data Generation Process



… except old data is not “replaced” but new data is added & data accumulates

Theory Results

Consider the basic OLS case from before, T >= d/2 (under-parametrized) & isotropic features, the test errors without and with accumulation are:

$$E_{\text{test}}^{\text{Replace}}(\hat{w}_n) = \frac{\sigma^2 d}{T - d - 1} \times n$$

Theory Results

Consider the basic OLS case from before, T >= d/2 (under-parametrized) & isotropic features, the test errors without and with accumulation are:

$$E_{\text{test}}^{\text{Replace}}(\hat{w}_n) = \frac{\sigma^2 d}{T - d - 1} \times n$$

$$E_{test}^{Accum}(\hat{w}_n) \leq \frac{\sigma^2 d}{T - d - 1} \times \frac{\pi^2}{6}$$

Error is not longer scaling with n!

Intuition

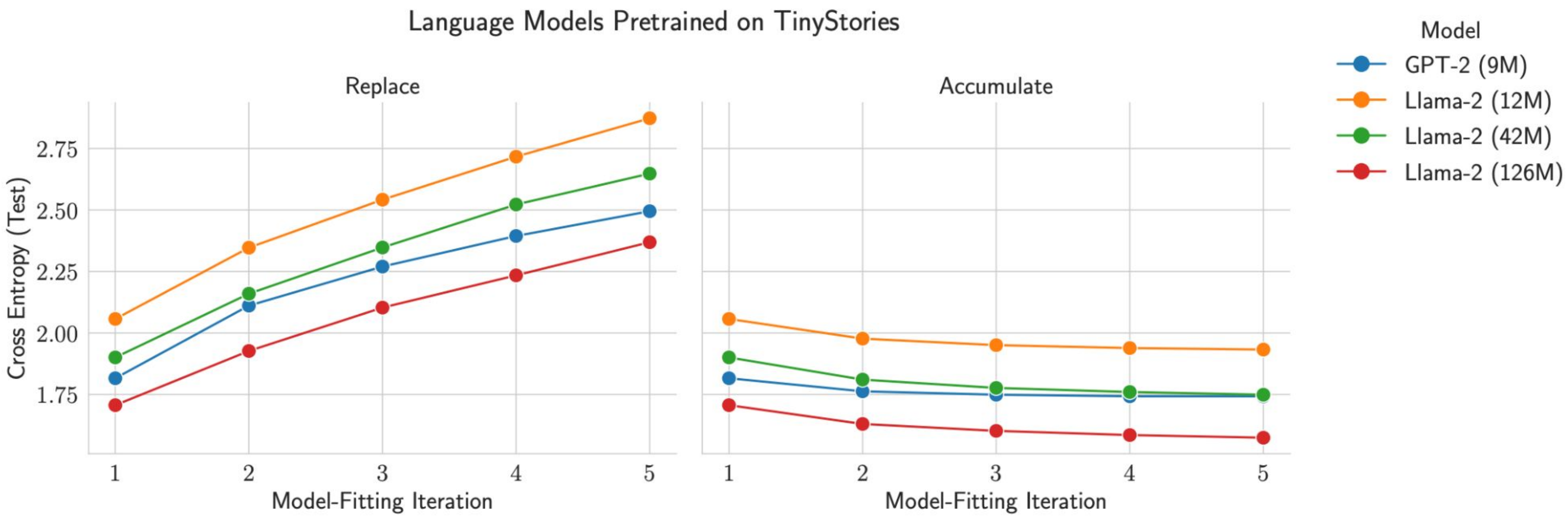
- If there is no prior data, the model is more affected by the noise from the previously generated synthetic data

- With accumulation, synthetic data is only 1/n th of the total data

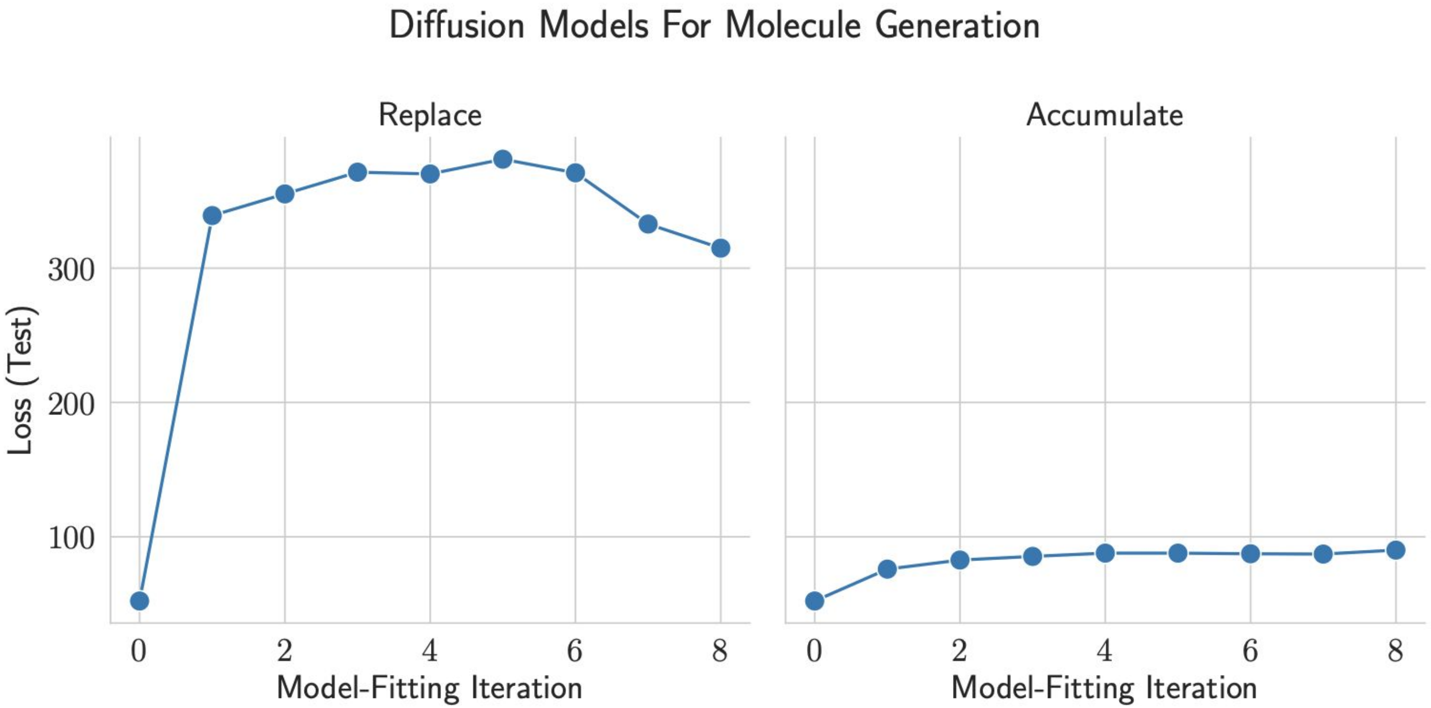
- Squared loss => effect only proportional to $(1/n)^2$

- But $(1/n)^2$ is summable!

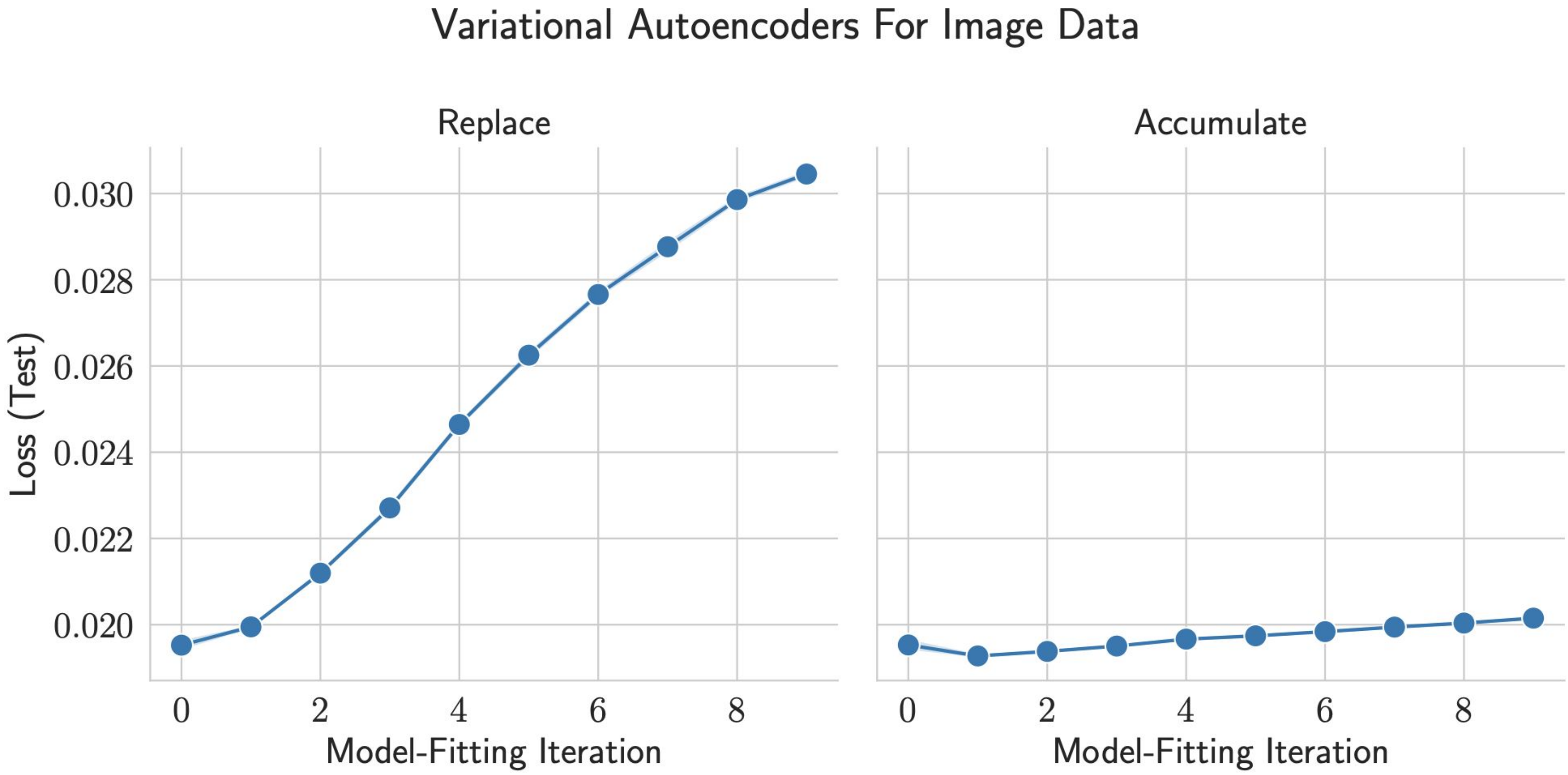
Experiments



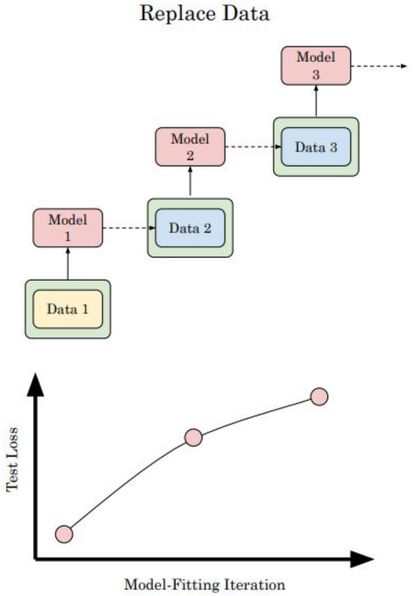
Experiments



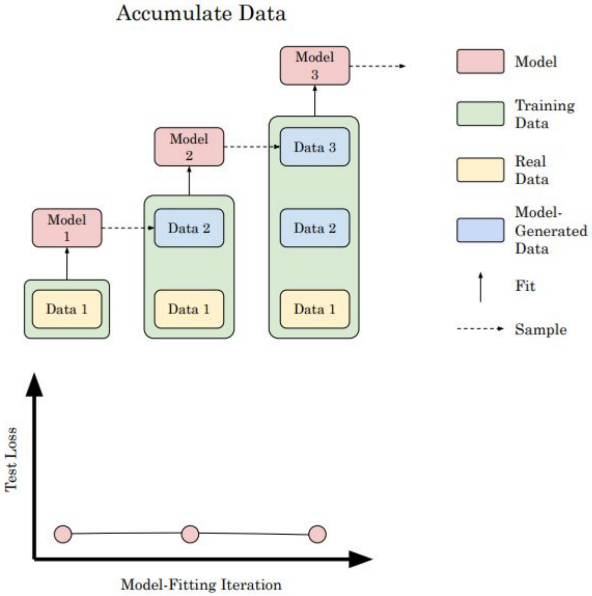
Experiments



So far, we've seen two somewhat naïve approaches to using synthetic data to train a model.

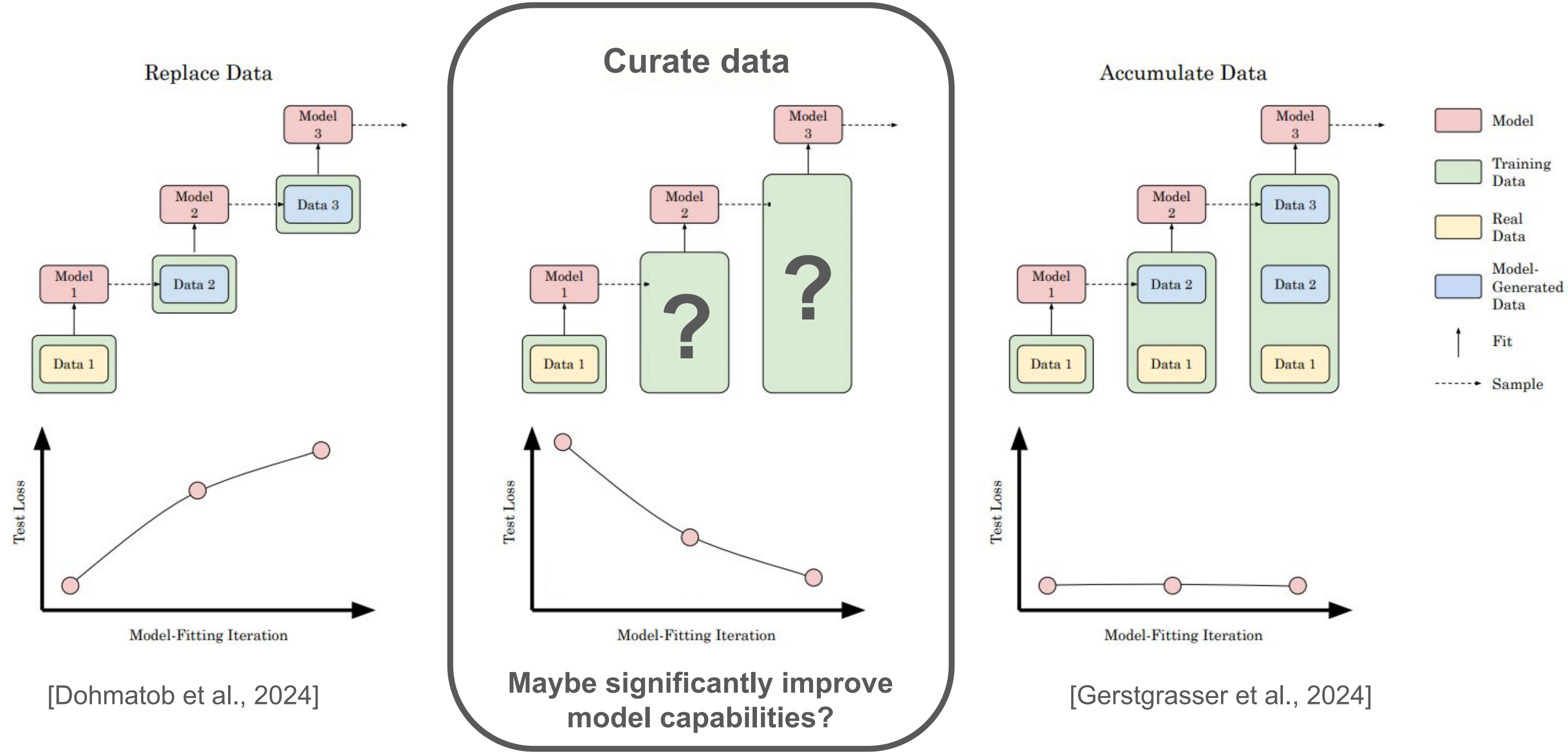


[Dohmatob et al., 2024]



[Gerstgrasser et al., 2024]

Are there alternative approaches that could allow for significant self-improvement?



[Dohmatob et al., 2024]

[Gerstgrasser et al., 2024]

Are there alternative approaches that could allow for significant self-improvement?

- A few recent works show positive results
- Today, focusing on one such recent paper:

Self-Improving Transformers Overcome Easy-to-Hard and Length Generalization Challenges

Nayoung Lee, Ziyang Cai, Avi Schwarzschild, Kangwook Lee, Dimitris Papailiopoulos

Are there alternative approaches that could allow for significant self-improvement?

- A few recent works show positive results
- Today, focusing on one such recent paper:

Self-Improving Transformers Overcome Easy-to-Hard and Length Generalization Challenges

Nayoung Lee, Ziyang Cai, Avi Schwarzschild, Kangwook Lee, Dimitris Papailiopoulos

- Key techniques:
  - Synthetic data filtering/verification
  - Carefully crafted *schedule* of synthetic data

Specific type of improvement: easy-to-hard generalization

- Math Tasks

1234 + 5313 =

↓

6547

53 * 92 =

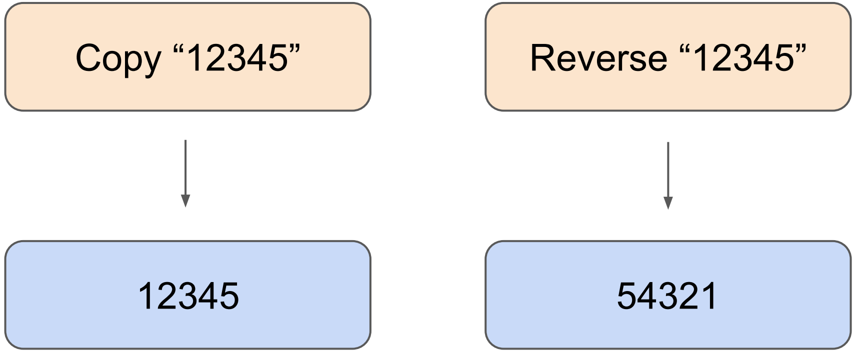
↓

4876

Generalization: Can we also do well on problems with more digits?

Specific type of improvement: easy-to-hard generalization

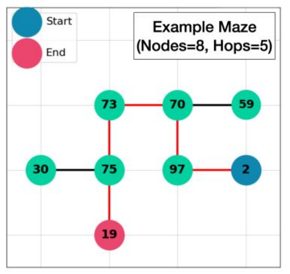
- Math Tasks
- String Tasks



Generalization: Can we also do these actions for longer strings?

Specific type of improvement: easy-to-hard generalization

- Math Tasks
- String Tasks
- Maze Solving

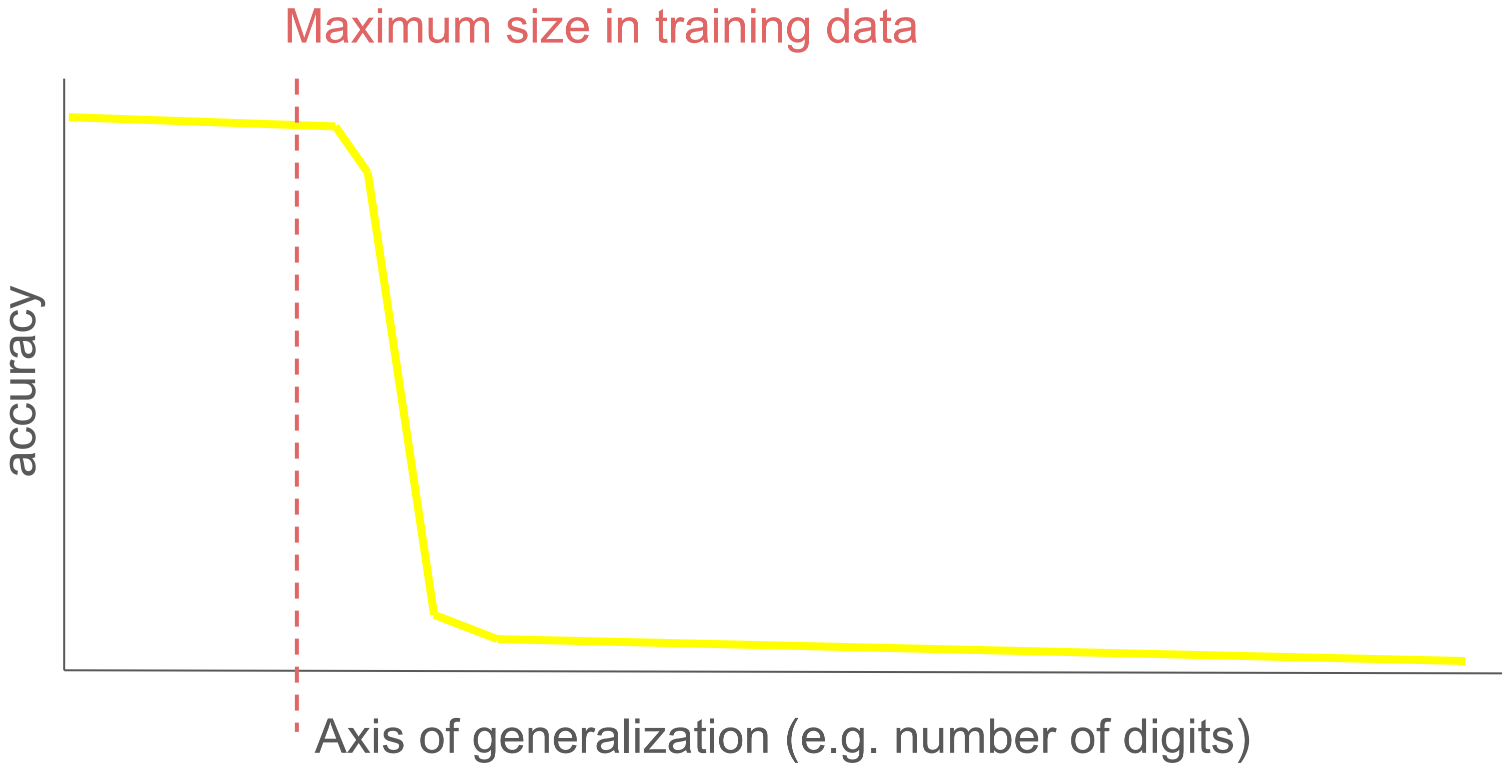


Find shortest path from node 2 to node 19.

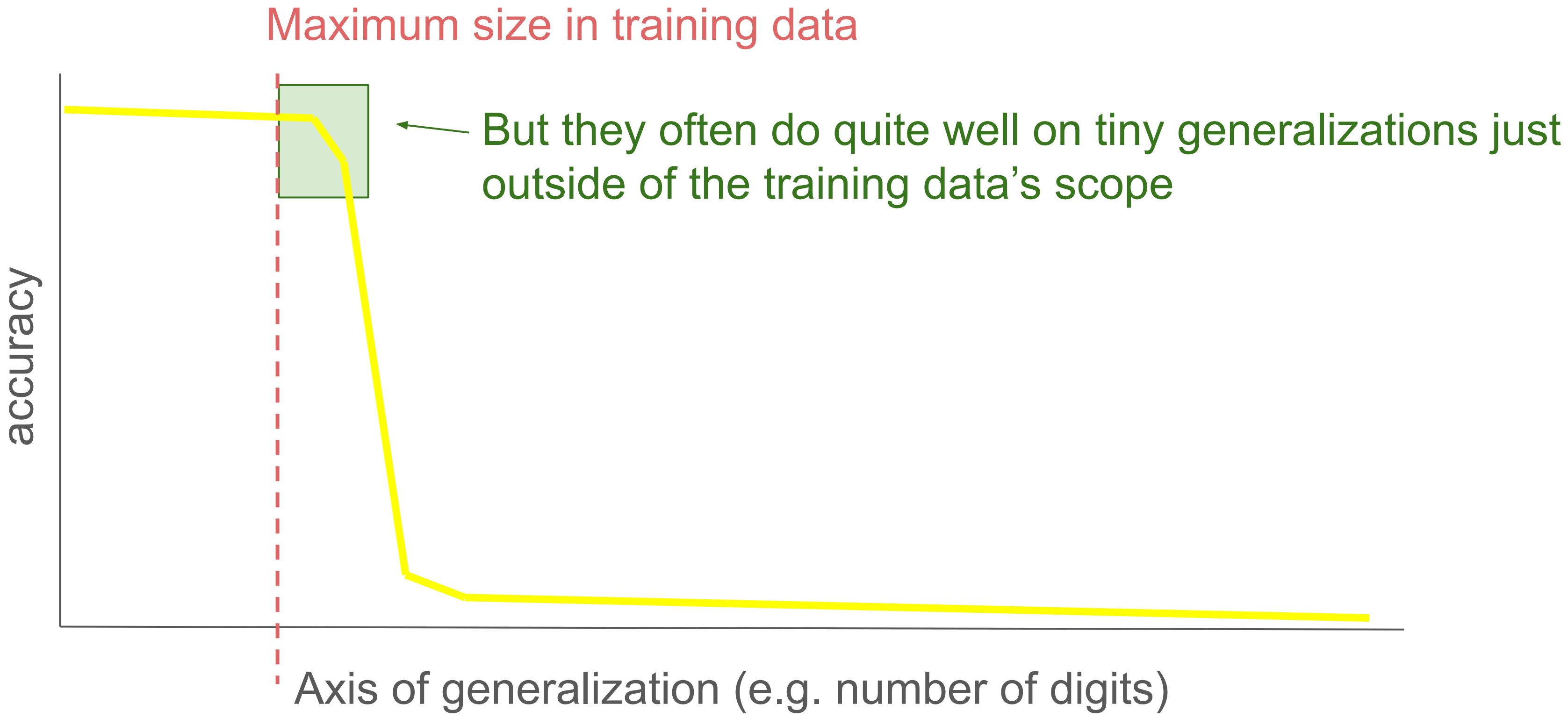
2 > 97 > 70 > 73 > 75>19

Generalization: Can we solve larger mazes?

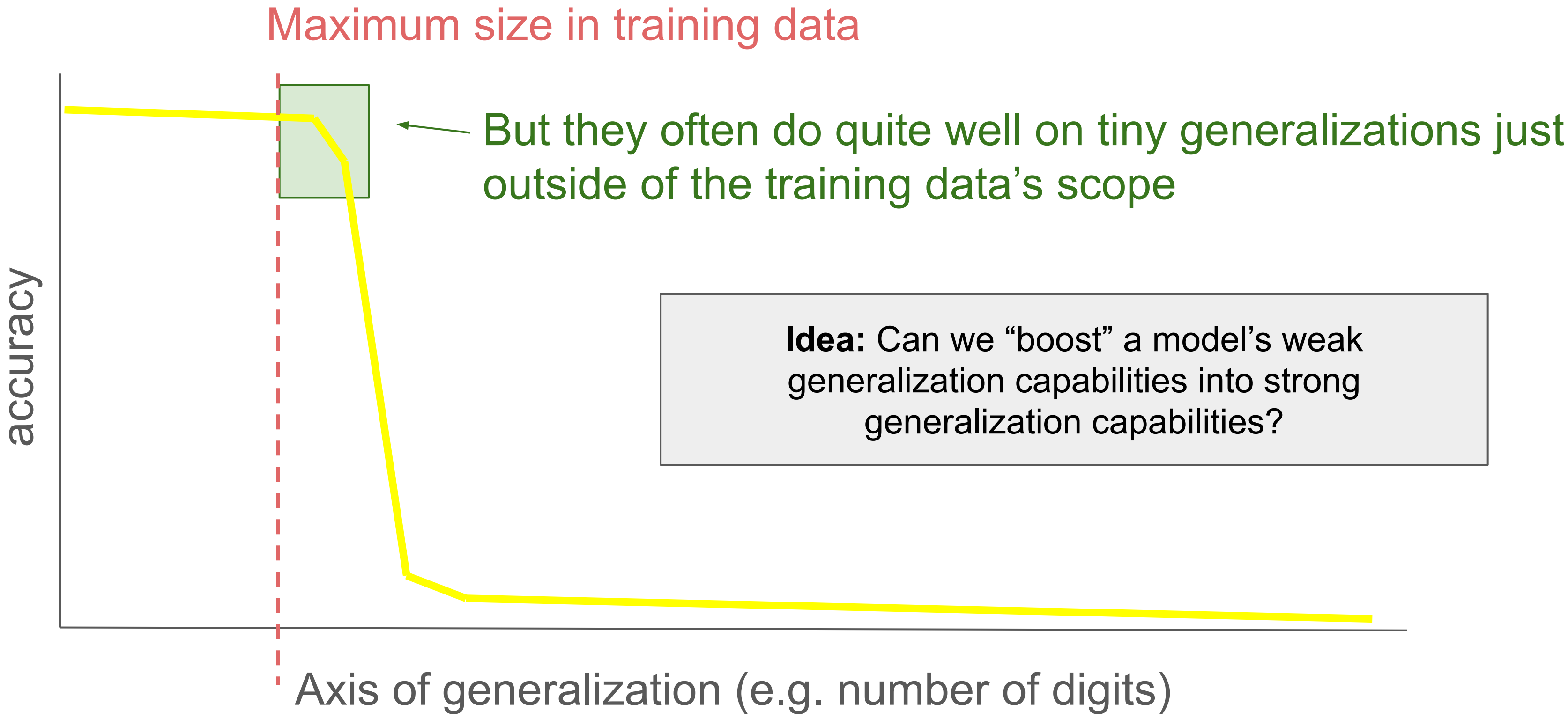
Transformers do not tend to generalize well on these tasks.



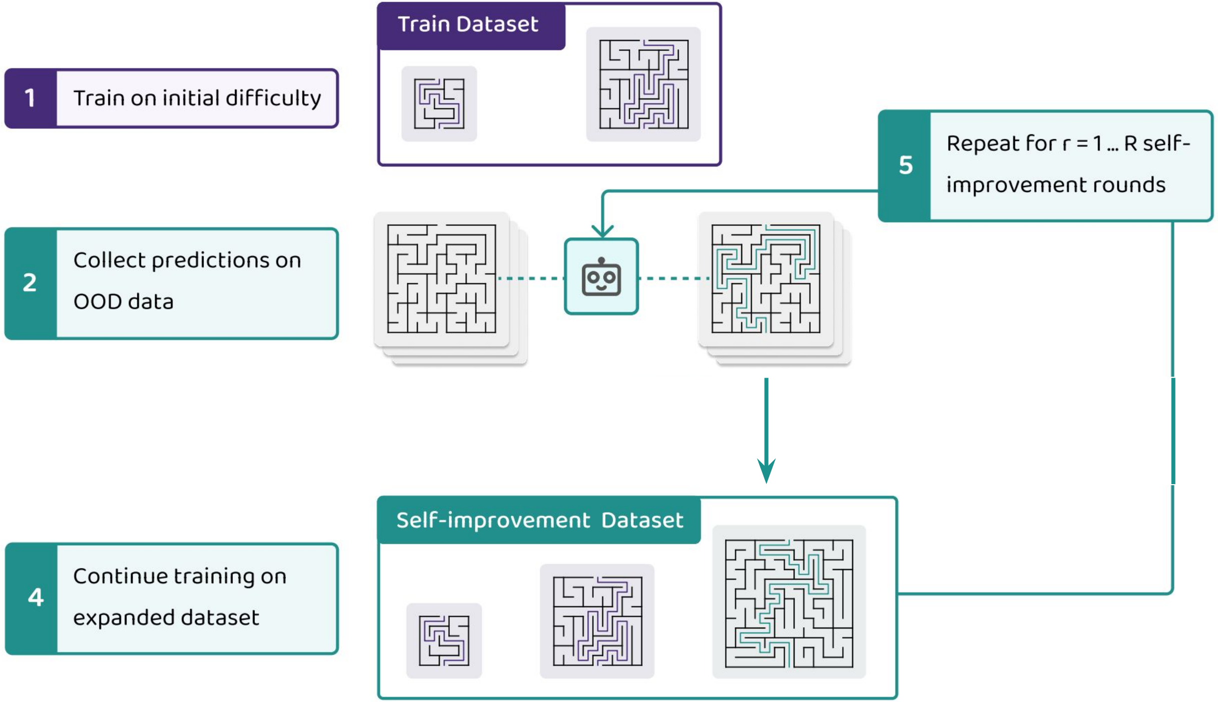
Transformers do not tend to generalize well on these tasks.



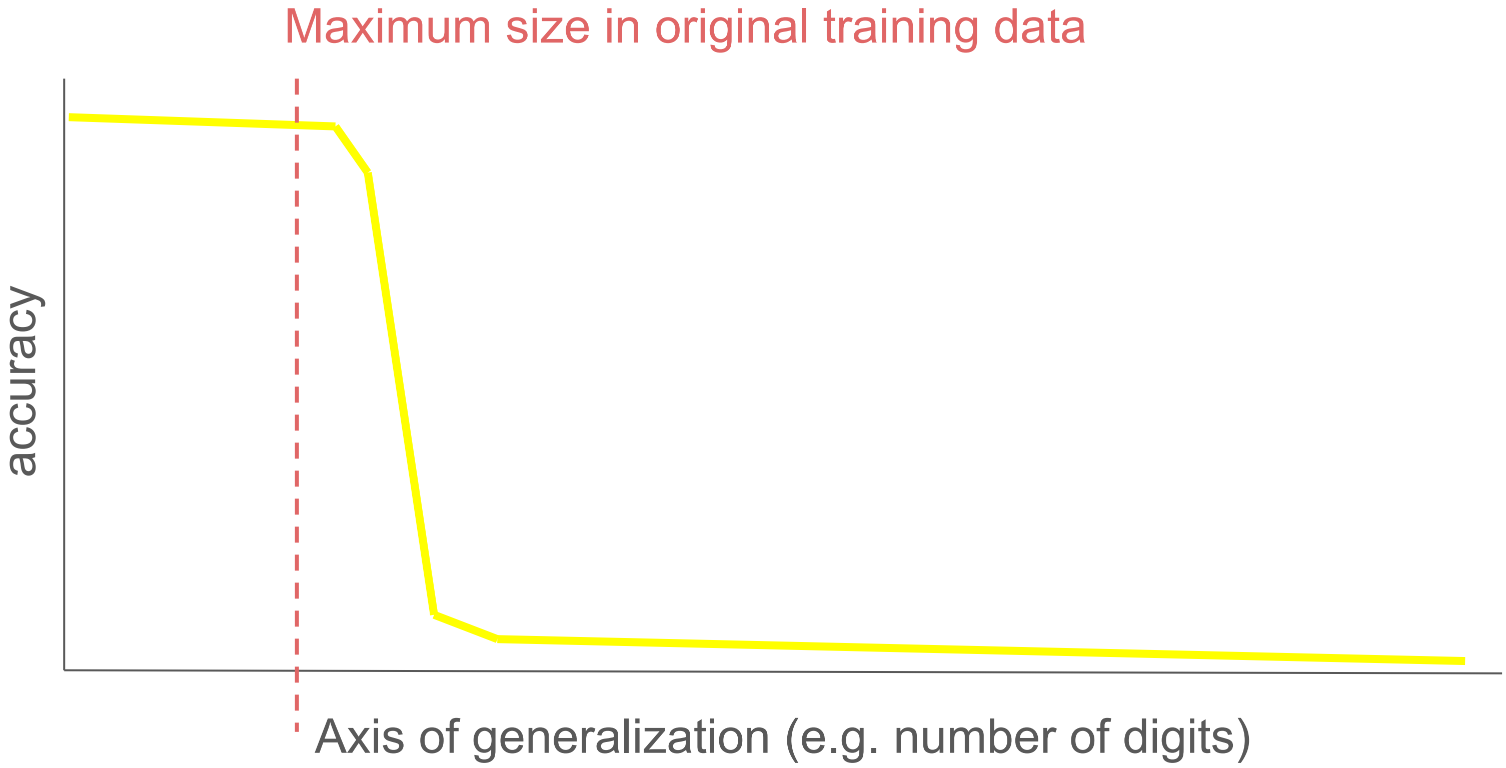
Transformers do not tend to generalize well on these tasks.



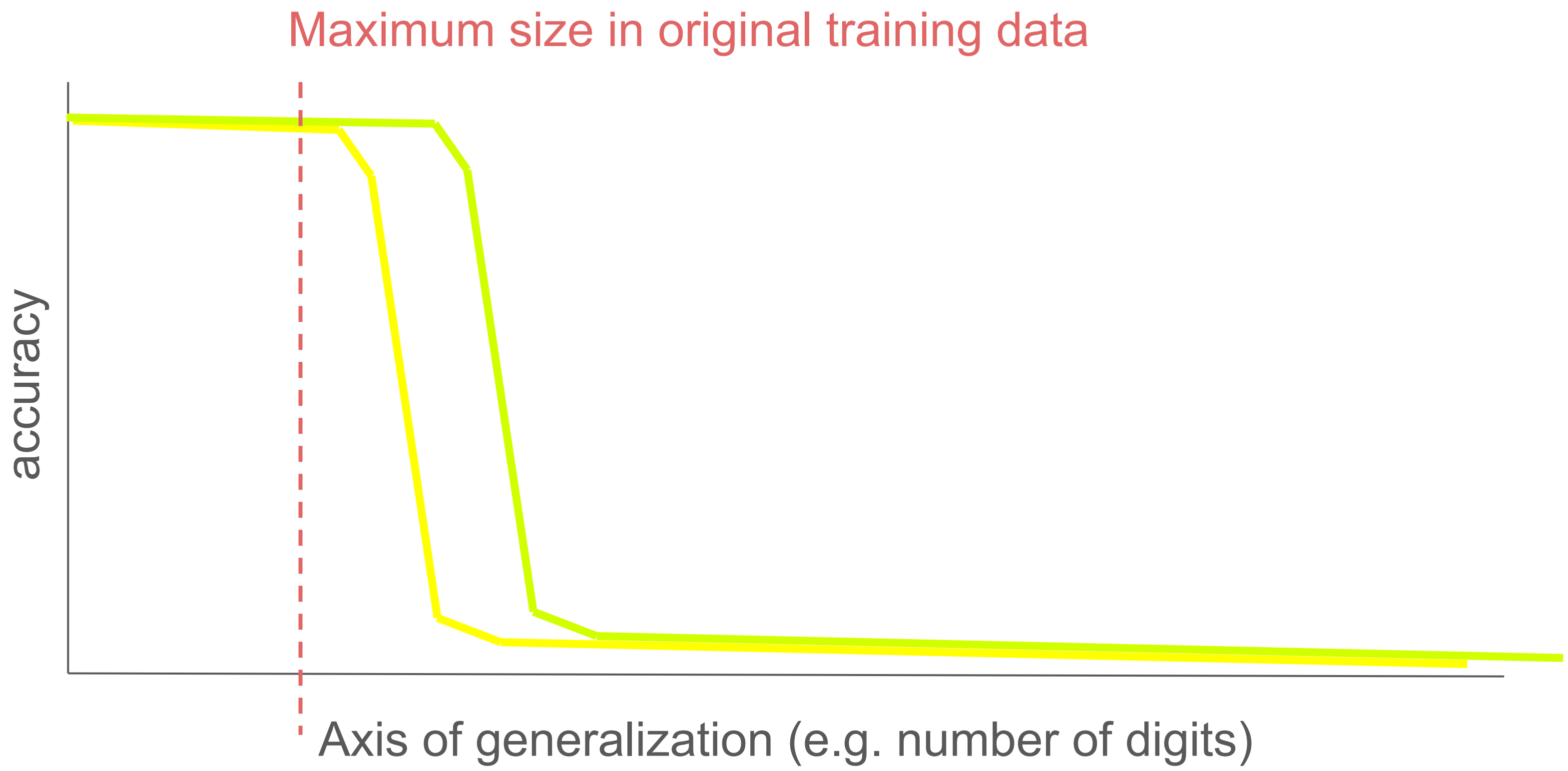
The Self-Improvement Setup



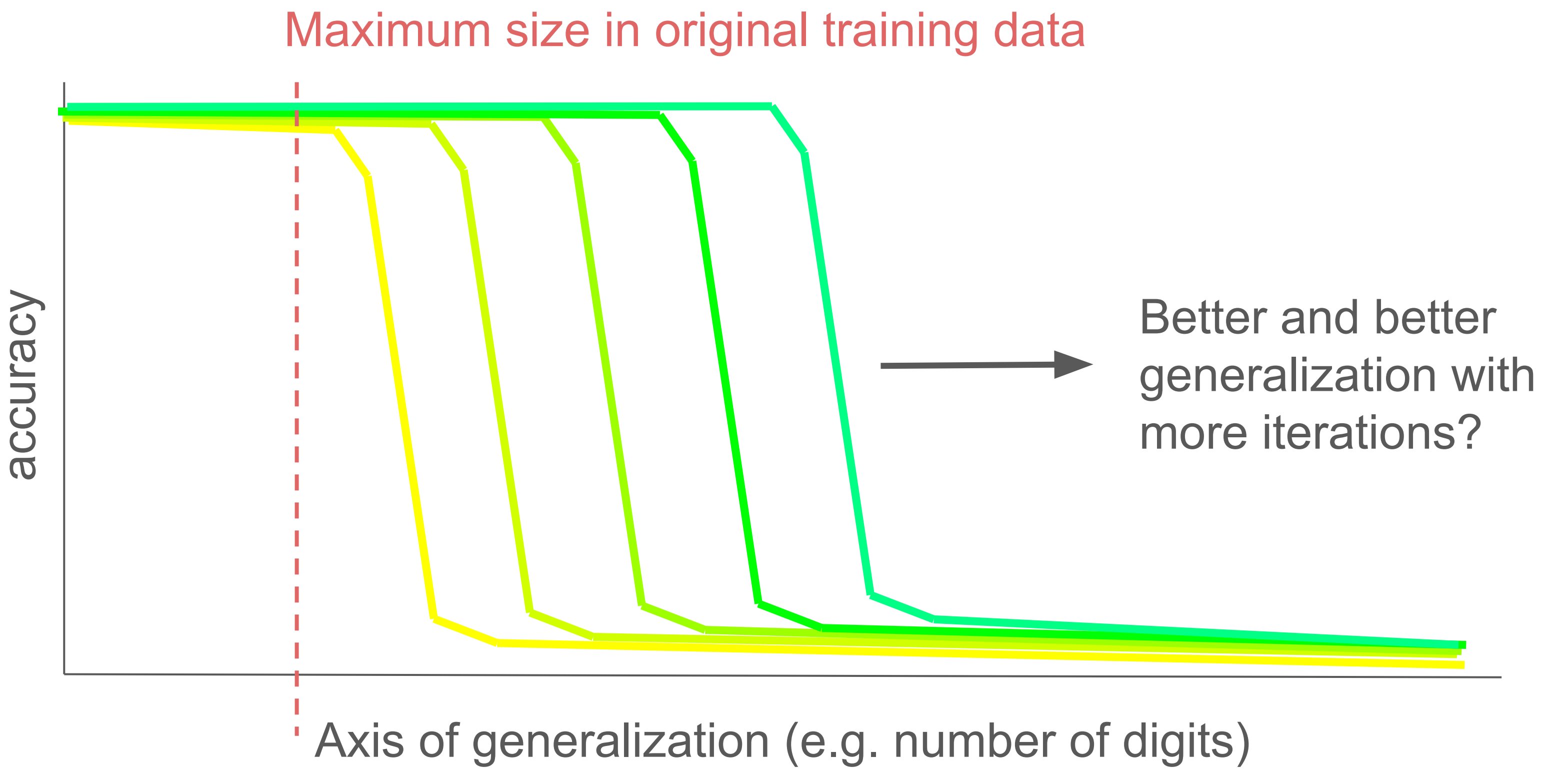
Ideal results of boosting



Ideal results of boosting



Ideal results of boosting



Actual Results on Simple Problems (very positive)

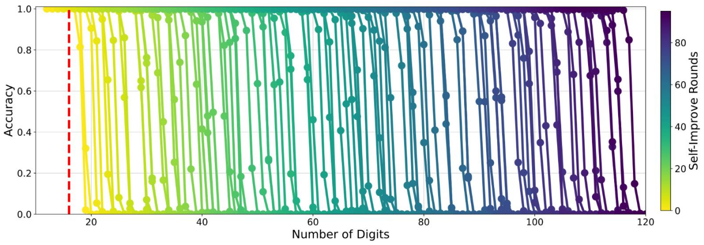


Figure 3: Results on the reverse addition task, where both operands and the output are represented in reverse order, with the least significant digit first. The self-improvement framework enables a model initially trained on 1-16 digit examples to generalize perfectly to over 100-digit addition.

Takeaway: Carefully curating the *schedule* on which synthetic data is introduced to the model can result in self-improvement gains.

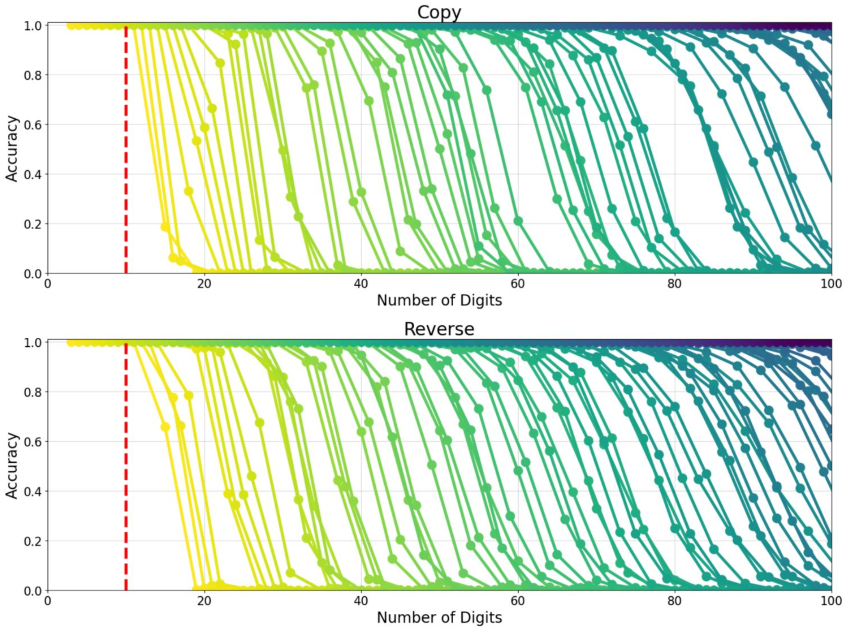
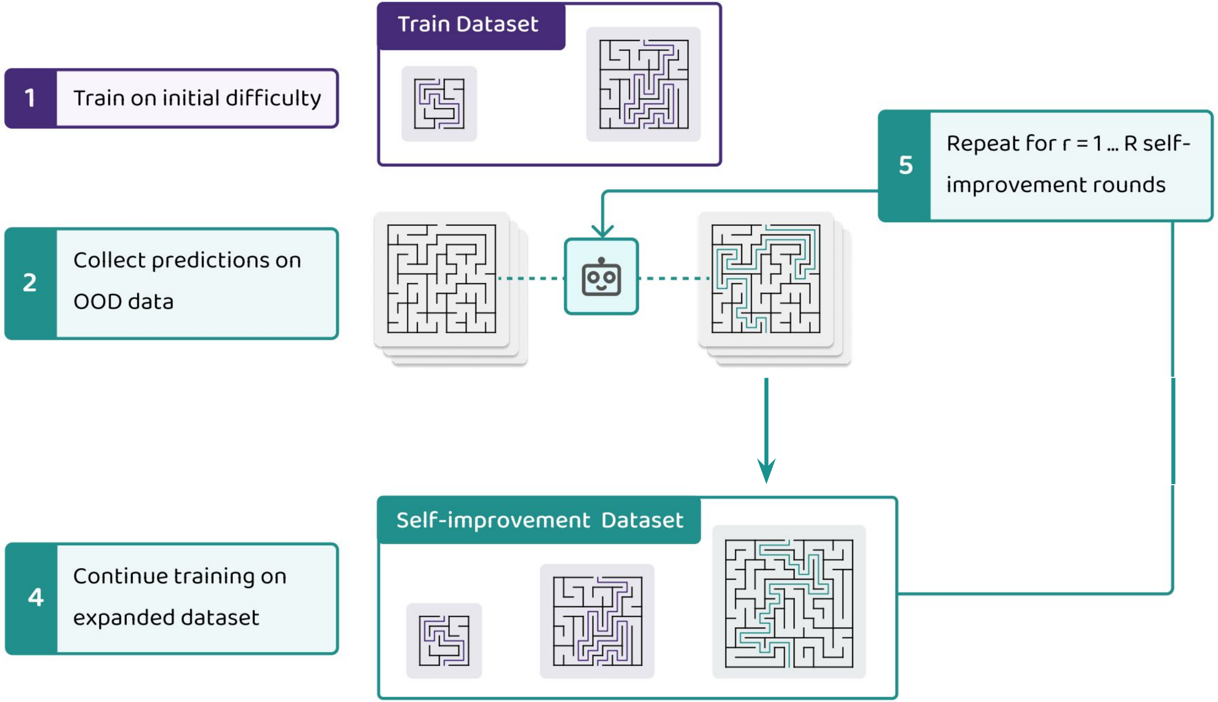
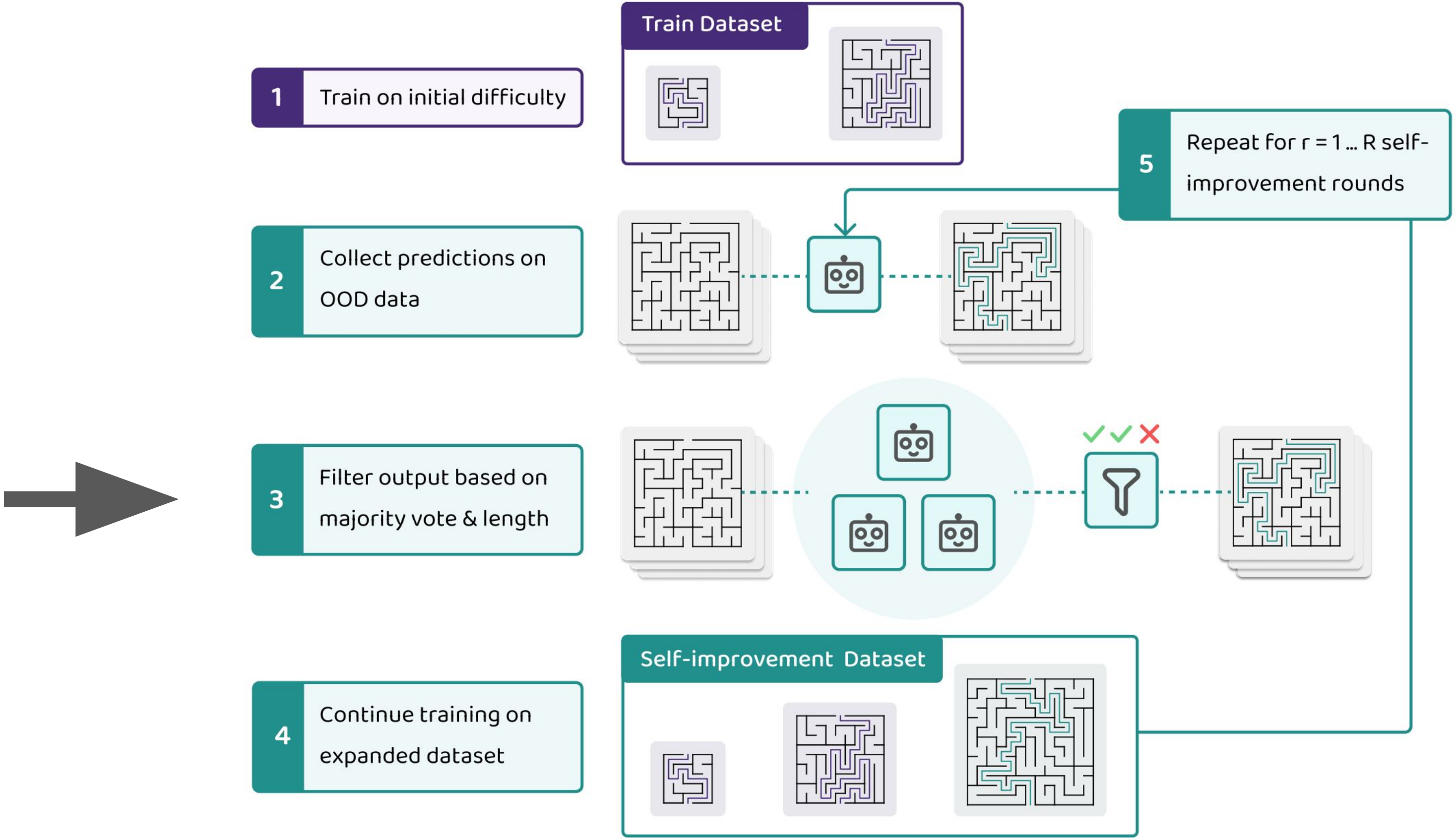


Figure 4: Results on string manipulation tasks. (Top) Copy: the model replicates the input string exactly. (Bottom) Reverse: the model outputs the input string in reverse order. The model initially trained on strings of length 1 to 10 generalizes to sequences of over 120.

An Extra Step for More Complicated Problems

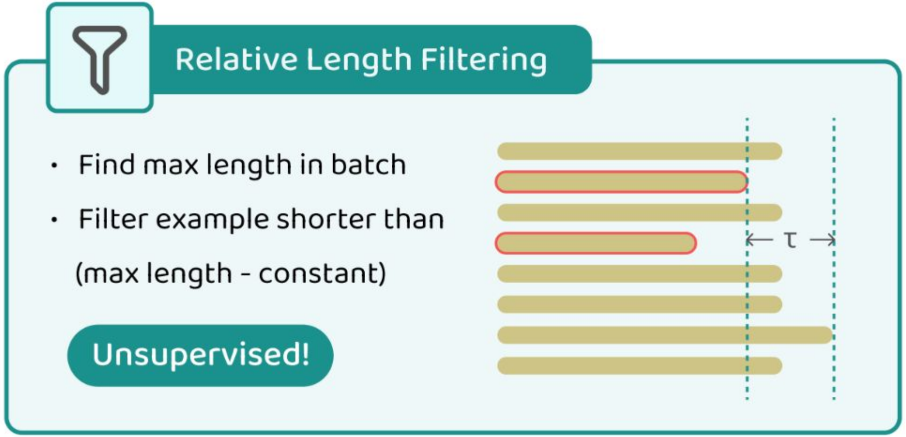


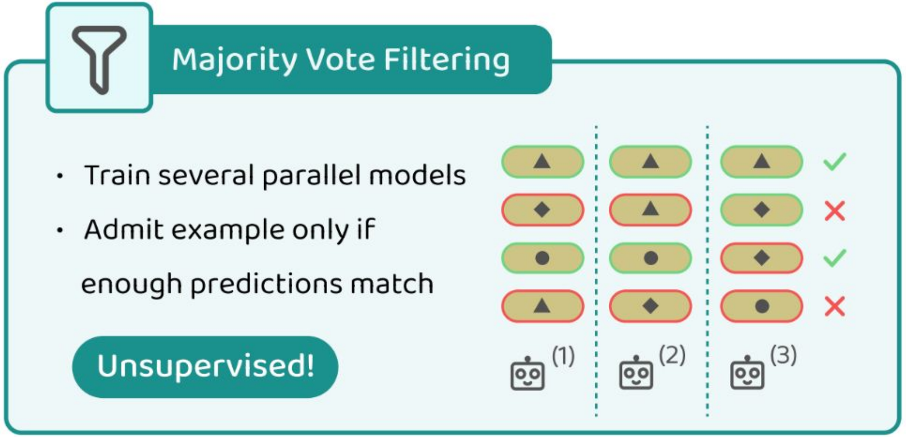
An Extra Step for More Complicated Problems



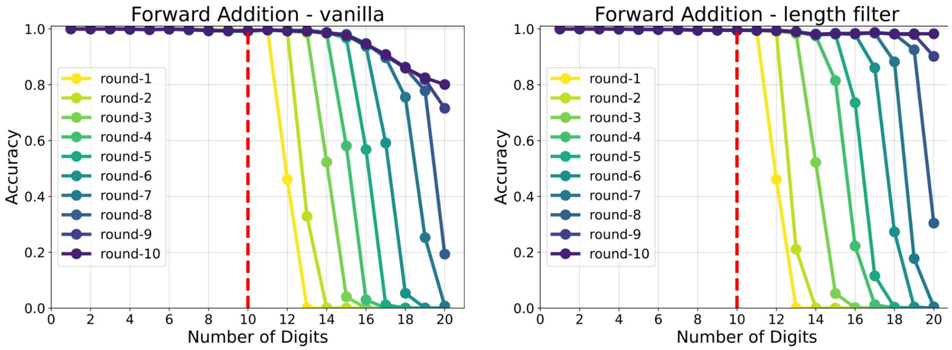
Filtering

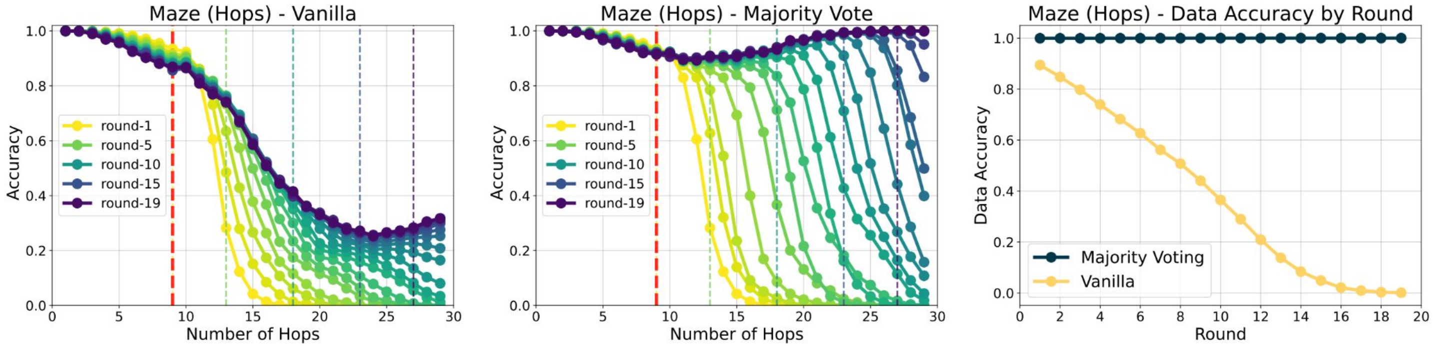
- Low-quality synthetic data leads to low-quality improvement (or degradation)
- **Idea:** Do some filtering of the synthetic data at each step to ensure better quality.
- **Important:** Want approaches to be *unsupervised*, i.e. not require an external verifier.





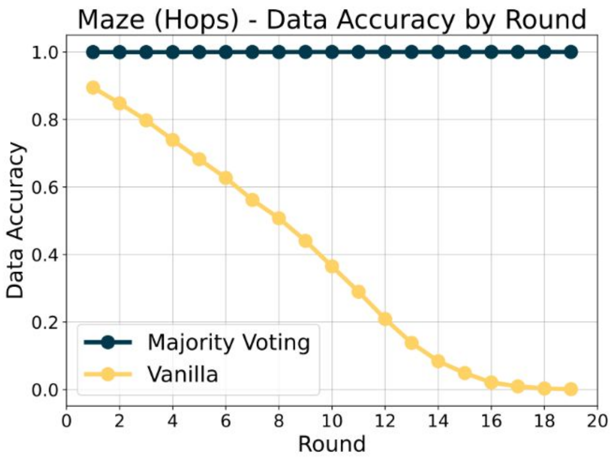
Adding filtering allows for self-improvement on more complicated tasks.

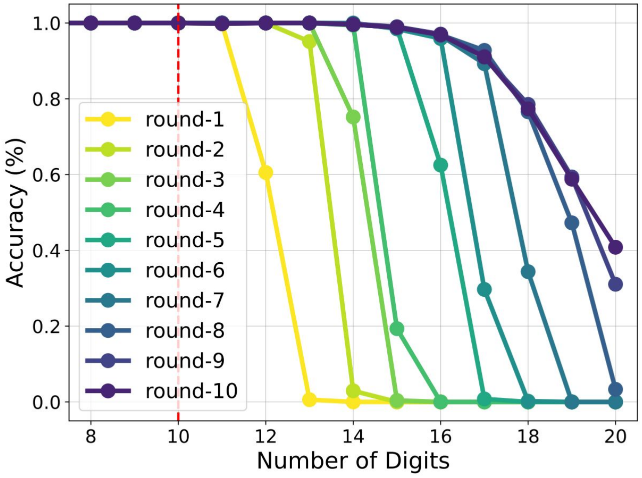




Watch out for the data avalanche

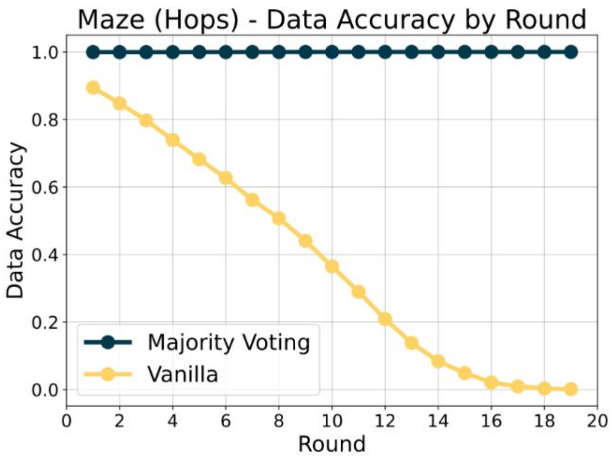
- Errors in low-quality synthetic data can accrue over rounds of self-improvement.

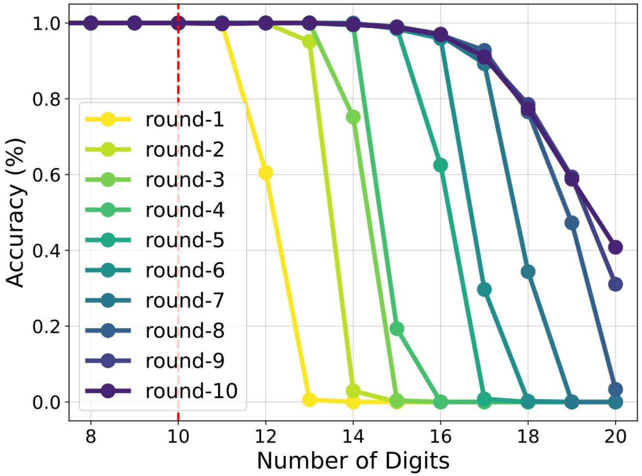




Watch out for the data avalanche

- Errors in low-quality synthetic data can accrue over rounds of self-improvement.





How much error is too much?

Watch out for the data avalanche

- Errors in low-quality synthetic data can accrue over rounds of self-improvement.
- Based on simulated mistakes, more error can be tolerated in later rounds.

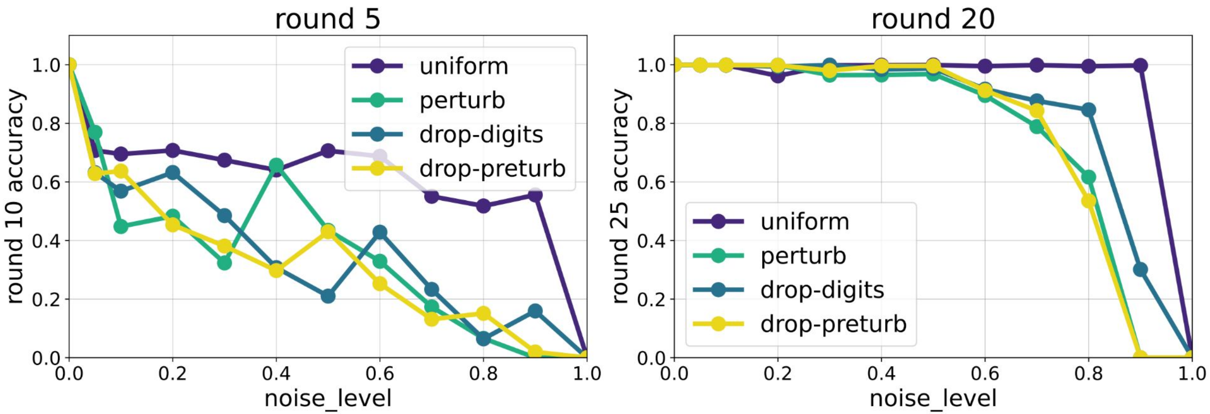


Figure 24: Simulating error avalanche. Synthetic mistakes of varying noise levels are injected at the end of rounds 5 and 20. The self-improvement process continues for 5 more rounds, and the resulting accuracy is recorded. The model tolerates errors up to a certain threshold, with greater tolerance observed in later self-improvement rounds.

Future Directions/Questions

- Identifying difficulty, "safe range" beyond toy problems
    - How to even generate example inputs?

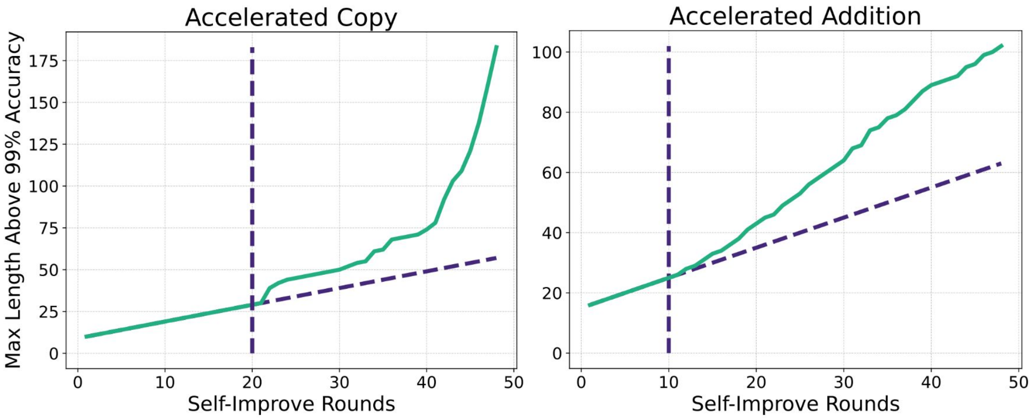


Figure 18: Maximum input length achieving over 99% accuracy at different self-improvement rounds for (Left) Copy task and (Right) Reverse addition. The dashed linear line represents the standard schedule of sampling one additional length per round. The vertical line is when we start allowing accelerated schedule. Faster self-improvement schedules allow the model to generalize to longer inputs with fewer rounds.

Future Directions/Questions

- Identifying difficulty, “safe range” beyond toy problems
- Scaling effects
    - Initial results show better self-improvement results on larger pretrained models