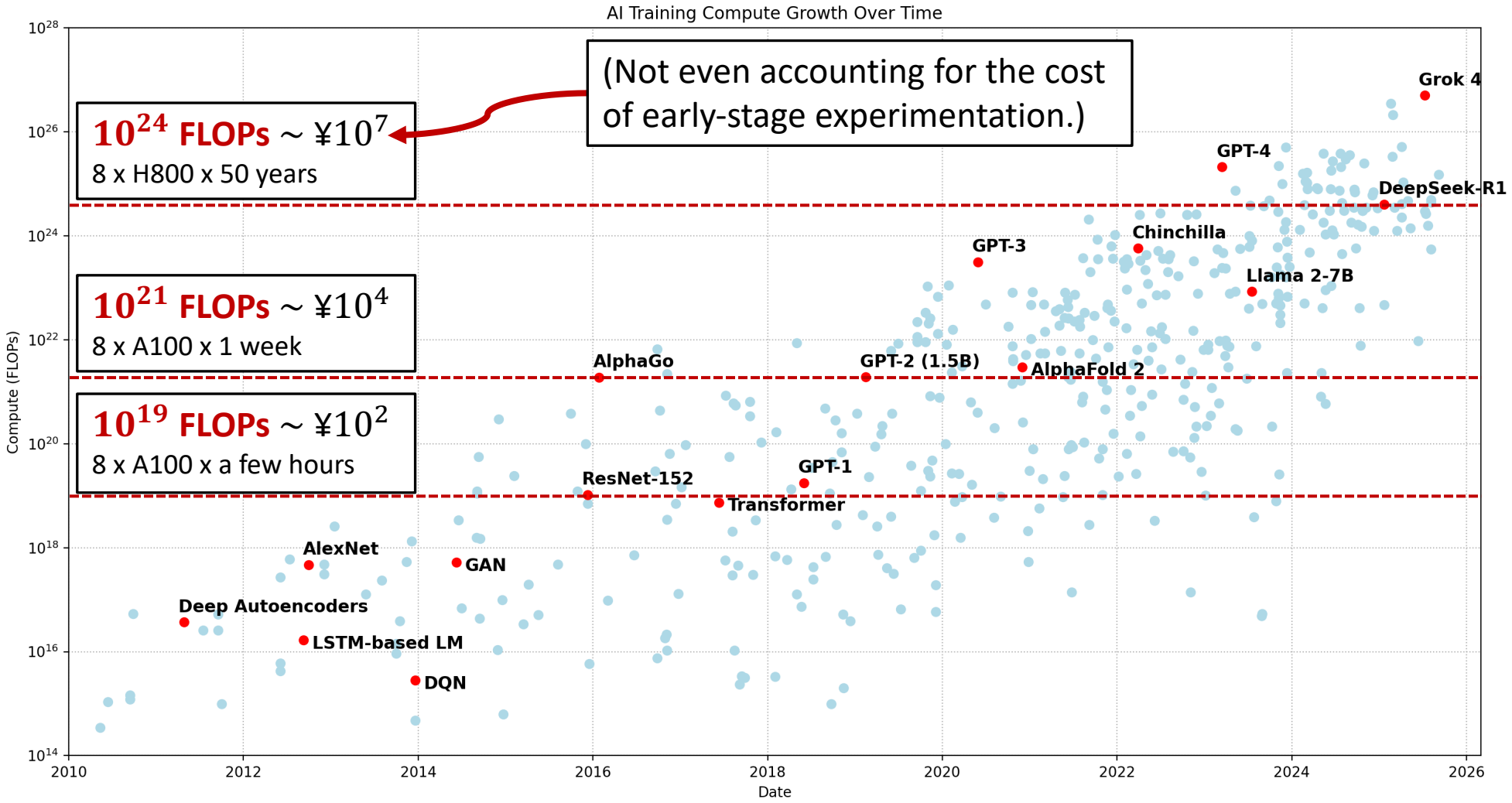
Scaling Large Language Models: The Data Problem

吕凯风 Kaifeng Lyu

清华大学 Tsinghua University

交叉信息研究院
Institute for Interdisciplinary Information Sciences

Super-smart models are super-expensive to train



Scale Matters

Small-scale Experiment:
- Efficiency is not the bottleneck
- Trial and error is cheap
- Alchemy may work just as well as science

Large-scale Experiment ≈ Rocket Launch
- You only have one shot
- Need confidence grounded in science, theory, intuition, or at least some faith

Left Figure: https://pic.rmb.bdstatic.com/bjh/241223/beautify/b224db7725802d2ddd43dcd96919f277.jpeg
Right Figure: Shenzhou 5, our first human spaceflight (https://www.12371.cn/2021/10/15/VIDE1634254560702194.shtml)

Example: Neural Scaling Laws

Neural Scaling Laws: Empirical formulas that describe how the **model performance** changes as **key factors** are scaled up and down

Key factors:

- **Model Size:** N (i.e., #parameters)
- **Dataset Size:** D (measured in tokens)
- **Total Compute:** C (measured in FLOPs)
- ……

Model Performance:

- **Pretraining loss** (test/validation)
- **Downstream loss/accuracy**
- **Accuracy after RL**
- ……

Chinchilla Scaling Law for (one-pass) LLM pretraining.

Irreducible constant

Pretraining loss $\mathcal{L}(N, T) = L_0 + A \cdot T^{-\alpha} + B \cdot N^{-\beta}$

Power law in T Power law in N

N: Model size T: Number of training steps (or tokens)

Science of Large-Scale Training?

We need to predict large-scale experiments, or make large-scale experiment predictable

Scaling Laws

Example:

- Chinchilla for model size vs. dataset size

In general:

- Visualize → Hypothesize a power law → Regression → Prediction

Our ICLR 2025 paper: A multi-power law for predicting the loss curve across LR schedules

Any Theory?

SDE Approximation: LR vs. batch size?

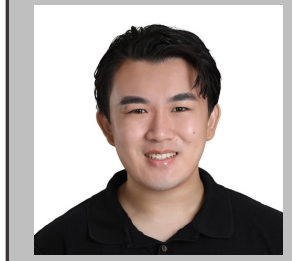
- Linear scaling rule for SGD
- Square root scaling rule for Adam (see also **our NeurIPS 2022 paper**)
- Functional scaling laws (by Lei Wu's group)

Tensor Program: Scaling up the width?

- μ-Parameterization (μP)

A Multi-Power Law for Loss Curve Prediction Across Learning Rate Schedules. Kairong Luo, Haodong Wen, Shengding Hu, Zhenbo Sun, Zhiyuan Liu, Maosong Sun, Kaifeng Lyu, Wenguang Chen. ICLR 2025.
On the SDEs and Scaling Rules for Adaptive Gradient Algorithms. Sadhika Malladi*, Kaifeng Lyu*, Abhishek Panigrahi, Sanjeev Arora. NeurIPS 2022.

Kevin Lu's Blog: The only important technology is ?



Kevin Lu

- **UC Berkeley** → **Meta AI** (2021-2022) → **Hudson River Trading** (2022-2023)
- **2024-2025: OpenAI** (led the release of **4o-mini**)
- **2025-now: Thinking Machines**

The Only Important Technology Is The Internet

How can we continue to scale large language models?

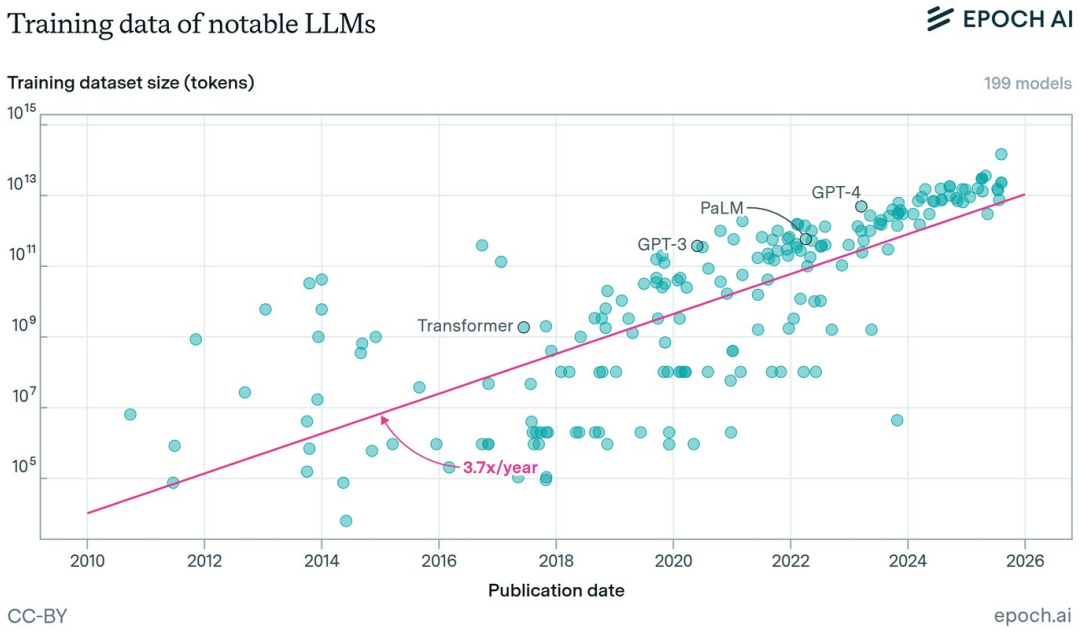
Published July 2025

Although progress in AI is often attributed to landmark papers – such as transformers, RNNs, or diffusion – this ignores the fundamental bottleneck of artificial intelligence: the data. But what does it mean to have good data?

If we *truly* want to advance AI, instead of studying deep learning optimization, we should be studying the internet. The internet is the technology that ***actually*** unlocked the scaling for our AI models.

We are running out of data for LLM pretraining

The size of datasets used to train LLMs doubles approximately every six months



- **GPT 2:** < 10 billion tokens
- **GPT 3:** 370 billion tokens
- **Llama 2:** 2 trillion tokens
- **Qwen 3:** 36 trillion tokens
- ...

Epoch.ai Report: Public data may be exhausted as early as 2028

We are running out of data for LLM pretraining

Ilya Sutskever – We're moving from the age of scaling to the age of research

"These models somehow just generalize dramatically worse than people. It's a very fundamental thing."

DWARKESH PATEL

NOV 26, 2025

Indeed, it looks like, based on various things some people say on Twitter, maybe it appears that Gemini have found a way to get more out of pre-training. At some point though, pre-training will run out of data. The data is very clearly finite. What do you do next? Either you do some kind of souped-up pre-training, a different recipe from the one you've done before, or you're doing RL, or maybe something else. But now that compute is big, compute is now very big, in some sense we are back to the age of research.

Two Options

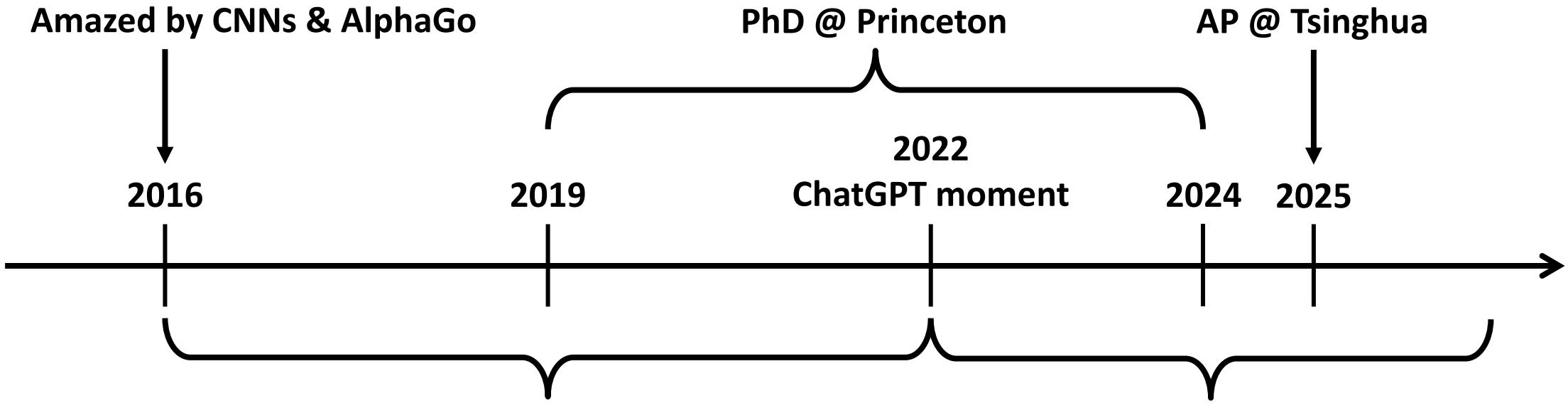
Option 1: Understand and improve generalization in the data-constrained regime
⇒ squeeze more from existing data

Option 2: Try to get more data (e.g., generating synthetic data)
⇒ enter the data-rich regime

My Research Journey

General Interest: Theoretical Foundations / Science of Deep Learning

(Understand how deep learning works and how to improve it)



Main Focus:

- Understanding the training dynamics of (overparameterized) neural networks

Main Focus:

- Science of scaling in LLM pretraining
- Principles in data-centric ML

Outline

Part 1. Scaling Laws for Multi-Epoch Training

Part 2. Towards Better Synthetic Data

Part 1: Scaling Laws for Multi-Epoch Training

One-Pass & Multi-Epoch Training at Scale

Question: How should we allocate the training budget?

Common Practice: One-pass training

- Choose model size (M); dataset size (N)

Data-constrained Regime: Multi-ecpoh training

- Choose model size (M); dataset size (N); epochs (K)

Chinchilla Scaling Law

$$L(M, N) = L_0 + A\,M^{-\alpha} + B\,N^{-\beta}$$

Pretraining loss

Power law in N

Compute-optimal training: $M \sim N^{a}$

Muennighoff et al. (2023):

$$L(M', N') = L_0 + A\,M'^{-\alpha} + B\,N'^{-\beta}$$

$$N' = (1 + R^*(1 - e^{-(K-1)/R^*})) \cdot N$$

The pre-training loss follows a power law wrt the training dataset size N

Reusing a dataset of size N for K times is equivalent to having an effective dataset size N'

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, Colin Raffel. Scaling Data-Constrained Language Models. NeurIPS 2023.

Heuristic Derivation of the Multi-Epoch Scaling Law

Effective dataset size: $N' = (1 + R^*(1 - e^{-(K-1)/R^*}) \cdot N$

Key Intuition: for every repeat of the data, the training process can extract $1 - \delta$ fraction of information contained in every data point.

→ **"Effective reuse rate":** $N'/N = 1 + R^*(1 - e^{-(K-1)/R^*})$ **— Independent of N!**

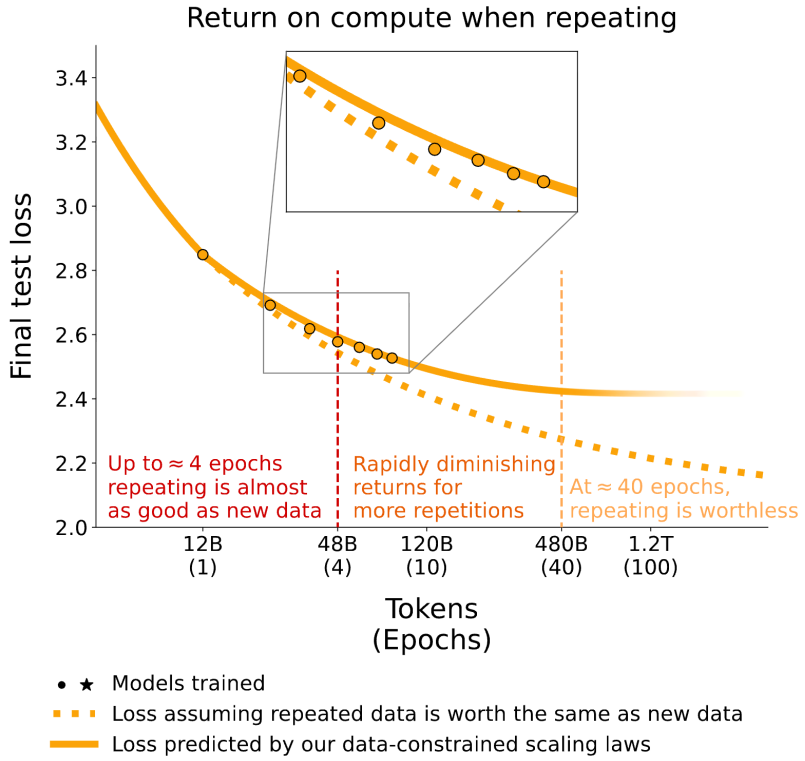
Intuitively: How many times does the dataset for one-pass training must grow to match the performance of multi-epoch training?

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, Colin Raffel. Scaling Data-Constrained Language Models. NeurIPS 2023.

Muennighoff et al. (2023)'s Derivation of the Multi-Epoch Scaling Law

Effective dataset size: $N' = (1 + R^*(1 - e^{-(K-1)/R^*}) \cdot N$

→ **"Effective reuse rate":** $N'/N = 1 + R^*(1 - e^{-(K-1)/R^*})$ **— Independent of N!**



Muennighoff et al. (2023)'s Experiments:

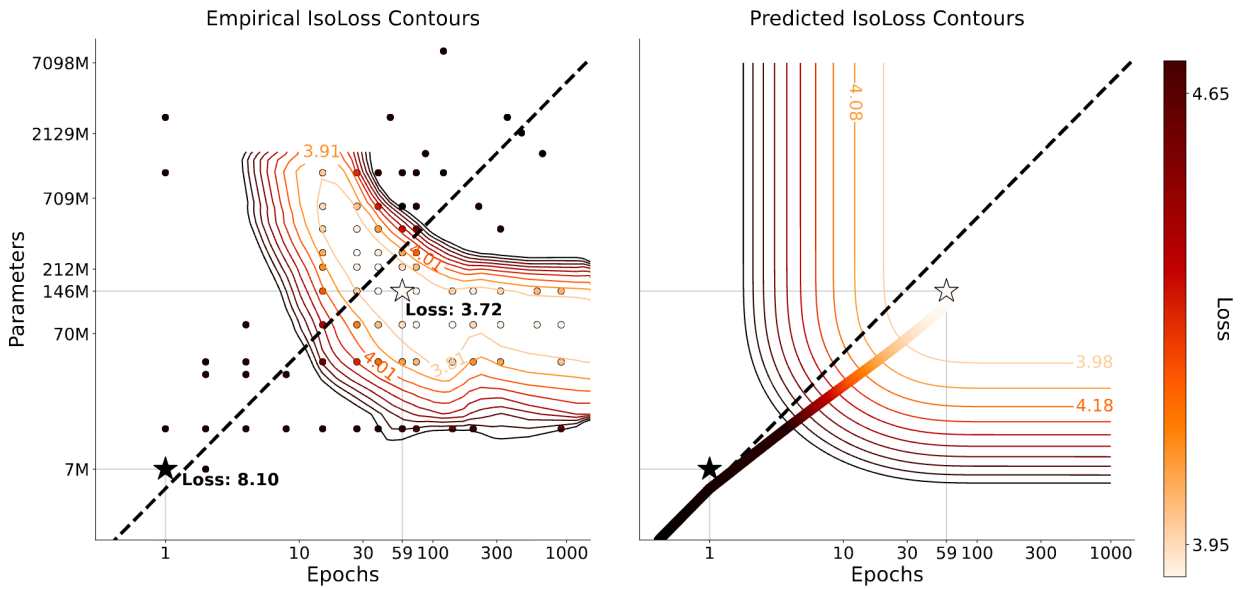
- $R^* \approx 15.39$
- $N'/N \approx K$ when $K \leq 4$
- **Major Takeaway:** Train 4 epochs!

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, Colin Raffel. Scaling Data-Constrained Language Models. NeurIPS 2023.

Muennighoff et al. (2023)'s Derivation of the Multi-Epoch Scaling Law

Effective dataset size: $N' = (1 + R^*(1 - e^{-(K-1)/R^*}) \cdot N$

→ **"Effective reuse rate":** $N'/N = 1 + R^*(1 - e^{-(K-1)/R^*})$ **— Independent of N!**



Issue I:

- The curve fit is imperfect

Issue II:

- Conceptually, is **effective reuse rate** really independent of N?

Can we **theoretically** understand this in a simple setting?

Niklas Muennighoff, Alexander M. Rush, Boaz Barak, Teven Le Scao, Aleksandra Piktus, Nouamane Tazi, Sampo Pyysalo, Thomas Wolf, Colin Raffel. Scaling Data-Constrained Language Models. NeurIPS 2023.

Larger Datasets Can Be Repeated More: A Theoretical Analysis of Multi-Epoch Scaling in Linear Regression

(ICLR 2026 Paper; NeurIPS 2025 OPT Workshop Oral)

Tingkai Yan*,Haodong Wen*, Binghui Li*, Kairong Luo
Wenguang Chen, Kaifeng Lyu

Our Work:

- N'/N is not a constant, both theoretically and empirically.
- Larger datasets can be repeated more

Insights from Linear Regression

Theoretical Setup: Linear regression

- Simple, yet serves as a useful mindset for more complex problems
- A common testbed for theoretical explanations of **scaling laws** (Bahri et al., 2021; Lin et al., 2024)

Task:

- **Linear regression:** $\mathrm{y} = \langle w^*, \mathrm{x} \rangle + \xi$
- $w^* \in \mathbb{R}^d$: ground truth weight; ξ: label noise with zero mean, $\mathbb{E}[\xi^2] = \sigma^2$
- **Strongly convex:** $\lambda_{min}(\mathbb{E}[xx^T]) \geq \text{constant} > 0$

Model:

- **Linear model:** $f(x, w) = \langle x, w \rangle$
- **MSE loss:** $\ell(w, x, y) = \frac{1}{2}(f(x, w) - y)^2$, $L(w) = \mathbb{E}_{(x,y)}[\ell(w, x, y)]$
- **Excess risk:** $R(w) = L(w) - \frac{1}{2}\sigma^2$

Theoretical Setup: Linear Regression

Training:

- K-epoch SGD training on N samples with random shuffling and constant LR η
- **Expected Excess Risk:** (expectation taken over training process)

$$\bar{R}(K,N) = \mathbb{E}[R(w)]$$

- **Optimal Expected Excess Risk:** (optimal tuning of LR)

$$R^*(K,N) = \min_\eta \bar{R}(K,N)$$

Effective Reuse Rate:

$$E(K,N) \coloneqq \frac{1}{N}\min\{N' > 0 : R^*(1,N') \leq R^*(K,N)\}$$

Intuitively: How many times does the dataset for one-pass training must grow to match the performance of multi-epoch training?

Theoretical Setup: Linear Regression

Theorem 1 (multi-epoch scaling law).

$$R^*(K,N) = \begin{cases} \frac{C_1 \log KN}{KN} \cdot \big(1 + o(1)\big) & \text{for } K = o(\log N) \\ \frac{C_2}{N} \cdot (1 + o(1)) & \text{for } K = \omega(\log N) \end{cases}$$

Theorem 2 (effective reuse rate).

$$E(K,N) = \begin{cases} K \cdot \big(1 + o(1)\big) & \text{for } K = o(\log N) \\ C \log N \cdot (1 + o(1)) & \text{for } K = \omega(\log N) \end{cases}$$

Phase transition happens at $K = \Theta(\log N)$

Two Phases:

- **Effective reuse regime:** Reusing data behaves like using fresh data
- **Limited reuse regime:** Additional epochs yield only marginal benefits

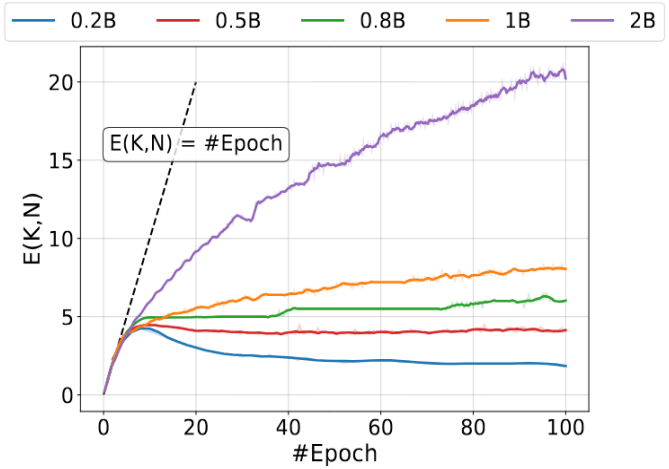
Larger datasets can be repeated more

Large Language Model Experiments

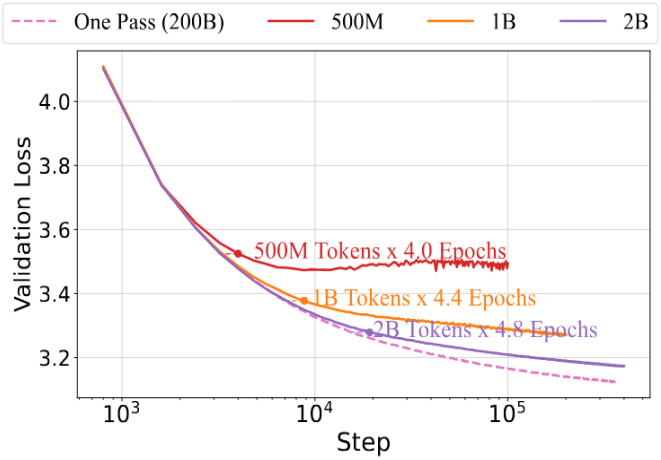
Two Phases: Effective reuse regime & Limited reuse regime

Phase Transition Point:

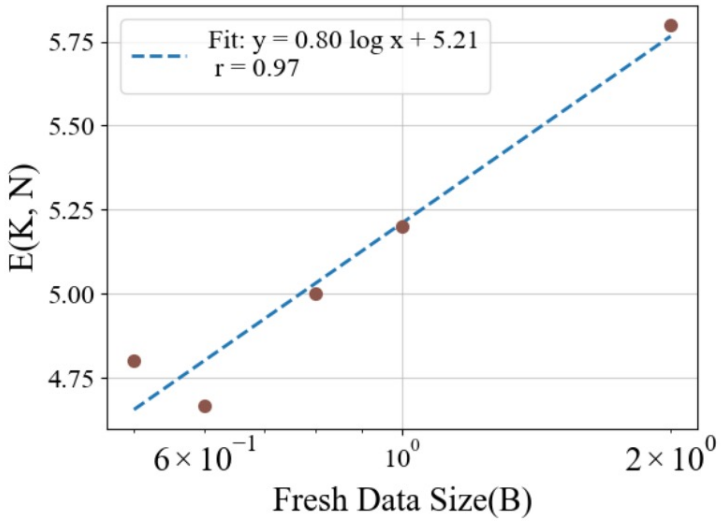
- **Our Experiments:** A log-linear law
- $K(N) = A \cdot \log N + B$



(a) The effective reuse rate $E(K, N)$ as a function of the epoch number K.



(b) Training loss as a function of training steps for different fresh data sizes.



Why can larger datasets be repeated more?

Fix the number of epochs K

Regime 1:

When the dataset size N is small → **overfitting happens** → $E(K,N) < K$

Regime 2:

When the dataset size N is large → **Random matrix product concentration** → $E(K,N) \approx K$

$$A \coloneqq \prod_{i=1}^{N}\left(I - \eta z_i z_i^T\right) \to \mathbb{E}A$$

Tool: Matrix Concentration for Products
[Huang, Niles-Weed, Tropp, Ward, 2020]

Now vary number of epochs from 1 to K , and fix N as we do in training:
Regime 2 → Regime 1: $E(K,N) \approx K \to E(K,N) = \Theta(\log N) < K$

Intuitively, why log?

① $R(w_t) =$ bias + variance

$e^{-\Theta(\eta KN)}$ $\Theta\left(\eta + \frac{1}{N}\right)$

② Very careful calculation + concentration ⟹ Optimal LR $= C \cdot \frac{\log(KN)}{KN}$

Variance dominates bias

③ As K becomes very large:

Variance = $\Theta\left(\frac{1}{N}\right)$ ⟹ Equation: $\frac{\log(N\prime)}{N\prime} \approx \frac{1}{N}$

Summary

A Theoretical Analysis of Multi-Epoch Scaling in Linear Regression

- **Muennighoff et al. (2023):** Train 4 epochs for all dataset sizes
- **Our Results:** Larger datasets can be repeated more, both empirically and theoretically

Limitation I: The effect of learning rate schedule

- The scaling laws and effective reuse rate can change with the way you do learning rate decay

Limitation II: Rewriting data instead of repeating data?

- Directly repeating the data can be ineffective. How about rewriting data?

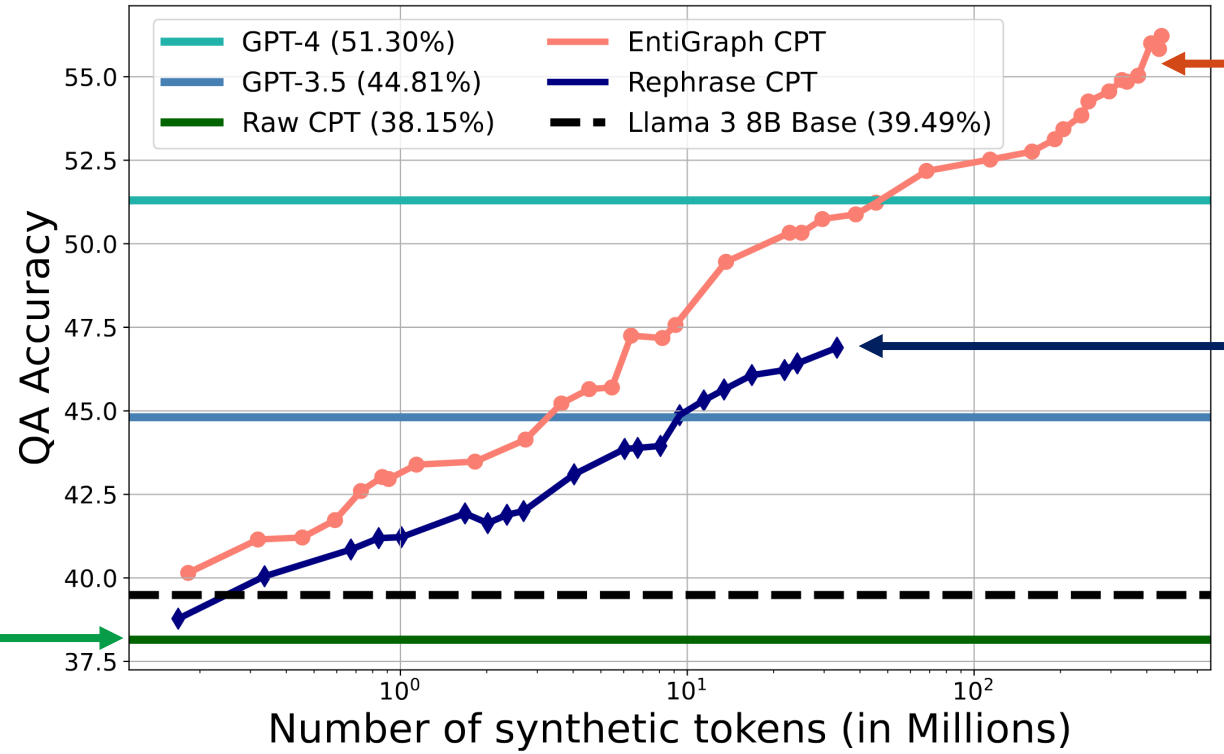
Part 2. Towards Better Synthetic Data

Part 2.1. Synthetic Data for Knowledge Injection

Ineffectiveness of multi-epoch training in knowledge injection

Knowledge Injection

- You have: a base LLM pretrained on web corpora + a small corpus for a special domain of interest
- **Goal:** Inject domain knowledge into the LLM to improve its downstream performance (e.g., QA)



Just repeat

EntiGraph: A two-phase prompting pipeline that prompts GPT-4 to identify and discuss entities and their relationships [Yang et al., 2025]

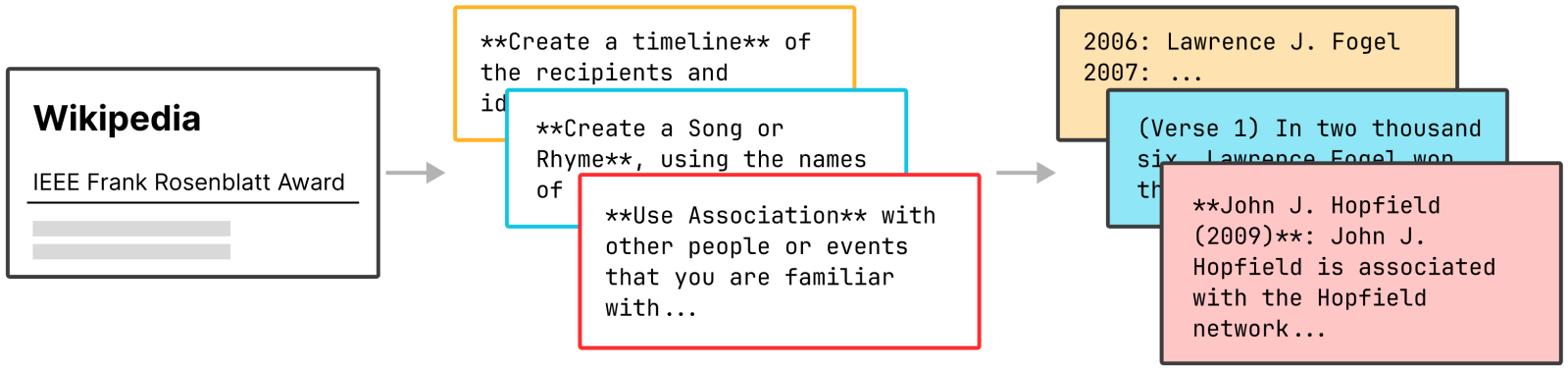
Prompt GPT-4 to rephrase data multiple times, then train on it

Another Method: Active Reading (AR)

Learning Facts at Scale with Active Reading

Jessy Lin[*1,2], **Vincent-Pierre Berges**[*1], **Xilun Chen**[1], **Wen-Tau Yih**[1], **Gargi Ghosh**[1], **Barlas Oğuz**[*1]

[1]FAIR at Meta, [2]University of California, Berkeley
[*]Equal contribution



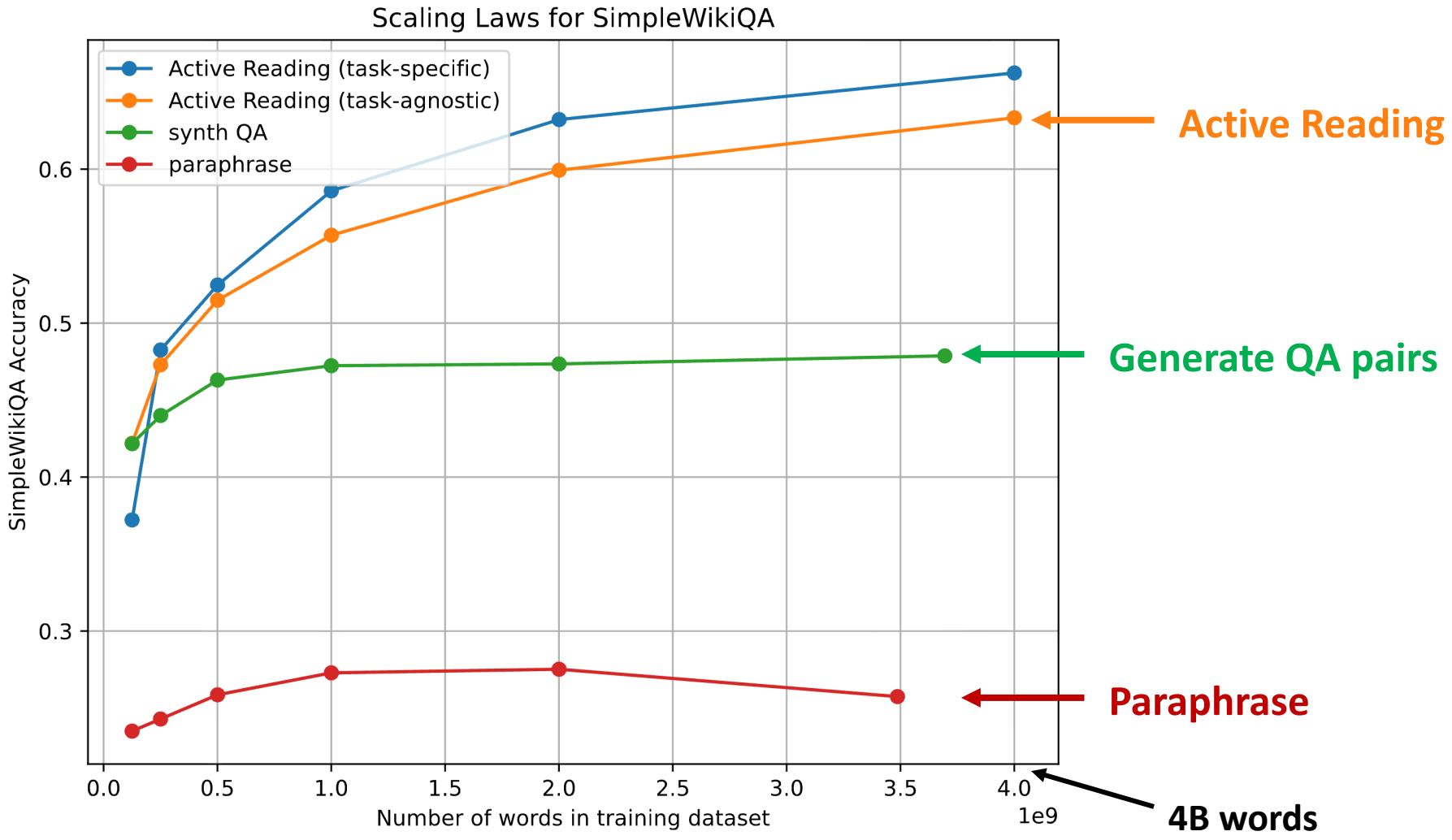
Source Document

1. Generate learning strategies to seed data generation

2. Apply strategies to generate diverse synthetic documents

Another Method: Active Reading (AR)

Injecting long-tail knowledge from Wiki



SPA: A Simple but Though-to-Beat Baseline for Knowledge Injection

Kexian Tang*, Jiani Wang*, Shaowen Wang, Kaifeng Lyu

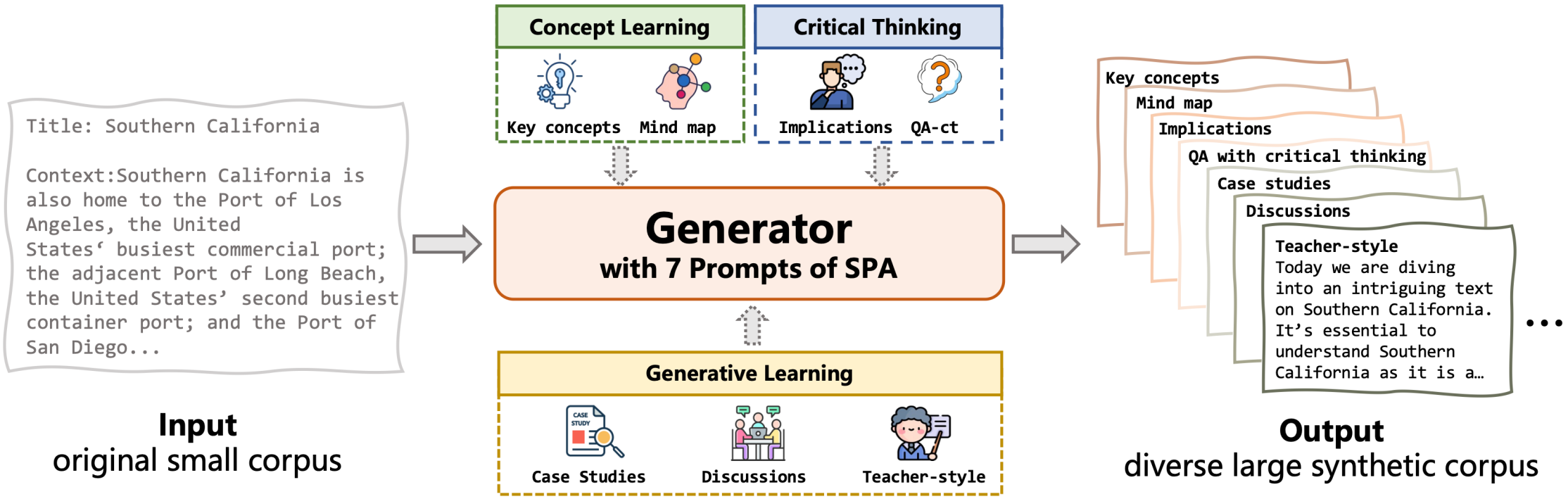
Tsinghua University

Takeaway:

- A simple baseline is able to beat many existing methods. No need for fancy pipelines.
- For future work, please compare with stronger baselines!

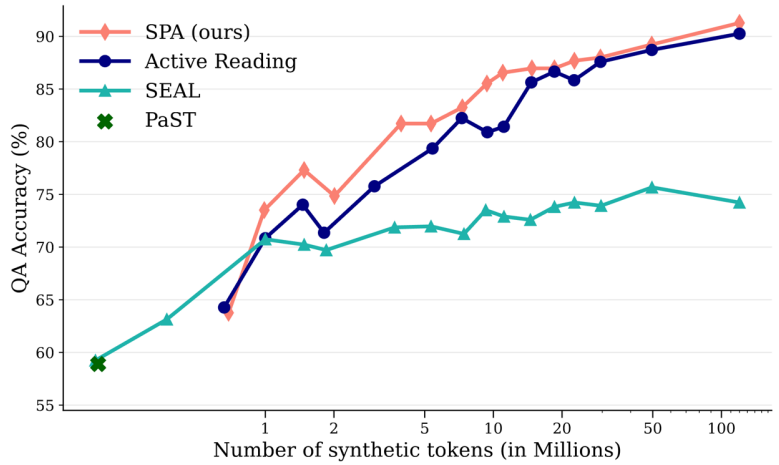
Code: https://github.com/Tangkexian/SPA

SPA: Scaling Prompt-engineered Augmentation



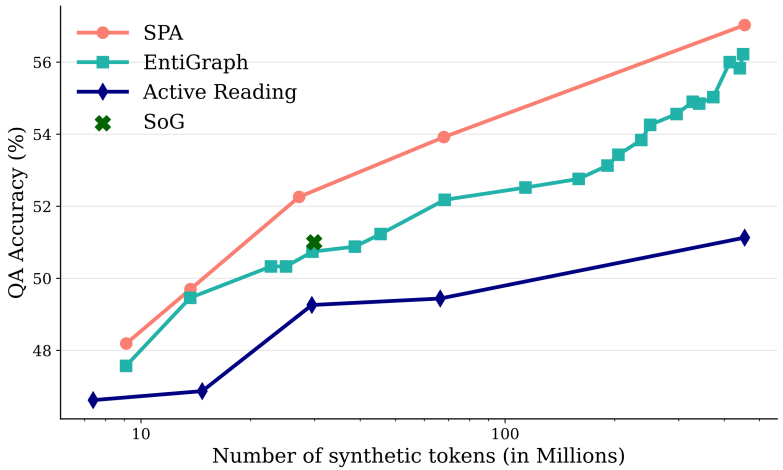
SPA beats more complex methods

Qwen2.5-7B → Qwen2.5-7B



Experiments on SQUAD

GPT-OSS-120B → Llama-3-8B



Experiments on QuALITY

MultiHop-RAG	QA Accuracy
Generator: GPT-4o-mini	
Number of Tokens: 15M	
Model: Qwen2.5-7B	
∗ Base	60.91
▷ Entigraph	85.42
▷ Active Reading	79.90
▶ SPA	**86.64**
Model: Meta-Llama-3-8B	
∗ Base	73.16
▷ Entigraph	84.31
▷ Active Reading	78.68
▶ SPA	**88.36**

Experiments on MultiHop-RAG

Failure of RL-based methods:

- Lack of diversity when scaled up

Failure of multi-stage prompting pipelines:

- Very diverse, but internal outputs (e.g., strategies in AR) are bad

Future Work: RL with entropy reg.? Using human-curated strategies as in-context examples in AR?

Part 2.2. In Principle, What Data Are Best?

Simple Answer: Know Your Downstream

D_{train}: Training distribution

Train your model f_θ on data sampled from D_{train}

Evaluate f_θ on a fixed test distribution D_{test}

Goal: Minimize the test loss

Best Way: Set $D_{\text{train}} = D_{\text{test}}$!
- If this is not feasible, make D_{train} as similar as D_{test}

In Practice: Add "high-quality" data
- High Quality: Data that are more relevant to downstream (e.g., world knowledge, math, code)
- Added by data mixing or data curriculum (see also our ICLR 2026 oral paper on data curriculum)

Kairong Luo, Zhenbo Sun, Haodong Wen, Xinyu Shi, Jiarui Cui, Chenyi Dang, Kaifeng Lyu, Wenguang Chen. How Learning Rate Decay Wastes Your Best Data in Curriculum-Based LLM Pretraining. ICLR 2026.

Is it always good to match the test distribution?

Imagine you are taking an exam next week…

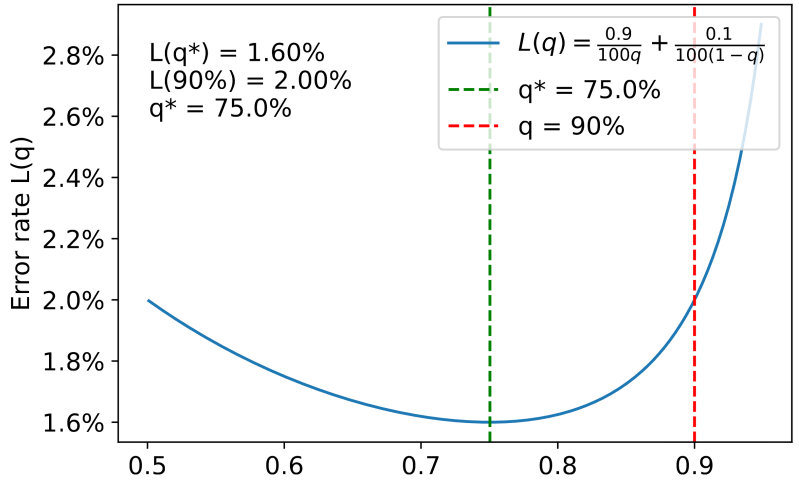
Your error rate decays as 1/n. How would you allocate your time?

Best: Introduce a "positive" distribution shift

Is it always good to match the test distribution?

Imagine you are taking an exam next week…

Your error rate decays as 1/n. How would you allocate your time?



Test distribution is rarely the best training distribution

Example: Memorizing K unique atoms

- In the test distribution, the occurrence probability of the k-th atom: $p_k \propto k^{-\alpha}$
- **Optimal training distribution** $\approx$ uniform over top $O(N)$ (N: training data size)

$$q_k^* = \max\left\{0, 1 - \beta \cdot p_k^{-1/(N-1)}\right\}$$

train = test:

$$\text{test loss} = \Theta\left(N^{-1+1/\alpha}\right)$$

train = optimal:

$$\text{test loss} = \Theta(N^{-\alpha+1})$$

A general result:

- Assume test distribution = K subpopulations weighted by $p_1, \dots, p_K$
- **Error function:**
  $e_k(n_1, \dots, n_K)$, the error on the k-th subpopulation if the numbers of samples from the K subpopulation are $n_1, \dots, n_K$.
- $\Rightarrow$ Positive distribution shift exists except for a zero-measure set of $\{e_k\}$ and $\{p_k\}$.

Where to shift?

Chinchilla Scaling Law

$$L(M, N) = L_0 + A\,M^{-\alpha} + B\,N^{-\beta}$$

Pretraining loss

Power law in N

Compute-optimal training: $M \sim N^{a}$

Many papers' view:

The power law of learning curve is induced by a power law of "skills" or "knowledge pieces" (Michaud et al., 2023; Arora & Goyal, 2023)

Conceptual Question:

If we had infinite knowledge & compute for data curation,
should we always shift the data towards a uniform distribution?

PS: if some are very hard and some are very easy, then uniform is not good.
But what if the skills are of the same difficulty?

Eric J. Michaud, Ziming Liu, Uzay Girit, Max Tegmark. The Quantization Model of Neural Scaling. NeurIPS 2023.
Sanjeev Arora, Anirudh Goyal. A Theory for Emergence of Complex Skills in Language Models. 2023.

The Power of Power Law: Asymmetry Enables Compositional Reasoning

(ICLR 2026 Latent & Implicit Thinking Workshop, Oral)

Zixuan Wang*[2], Xingyu Dang*[2], Jason D. Lee[3], Kaifeng Lyu[1]

[1]Tsinghua University, [2]Princeton University, [3]UC Berkeley

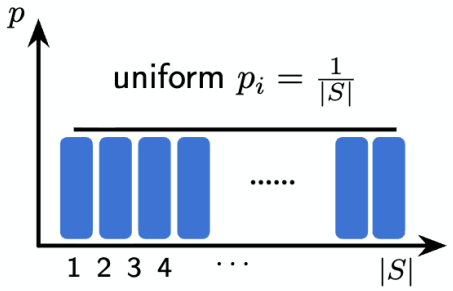
Main Result:

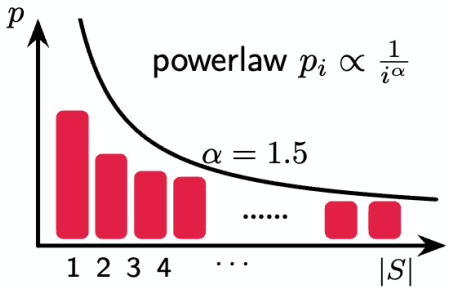
- For many implicit multi-hop tasks, power law is better
- Theoretical analysis in a minimalist task: k-multiplicative composition

Experiments: Multi-step Arithmetic & State Tracking

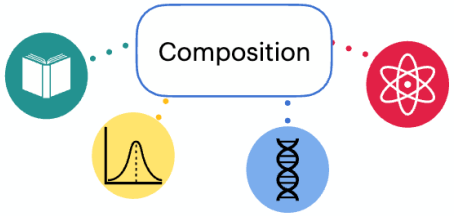
Skill distribution: $i \sim p_i$

$a \in [n]$ or $g_i \in S_n$



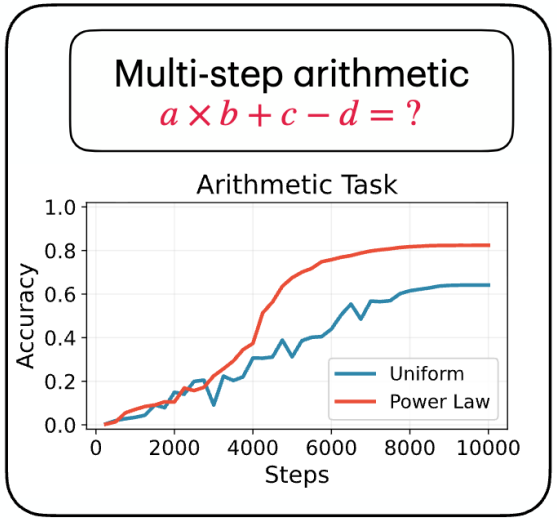


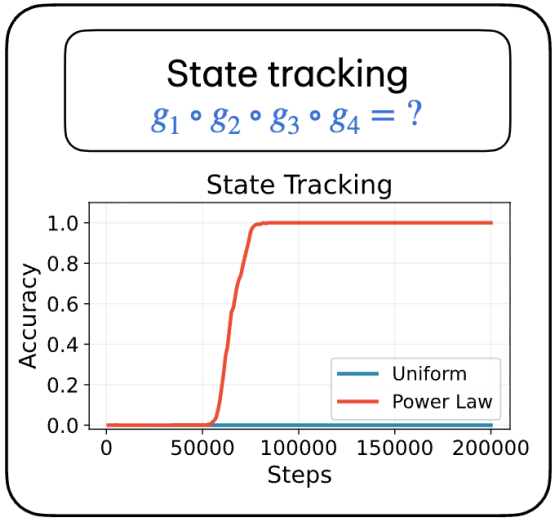
Compositional tasks



Test on uniform

- uniform 👎
- power law 👍





More Experiments

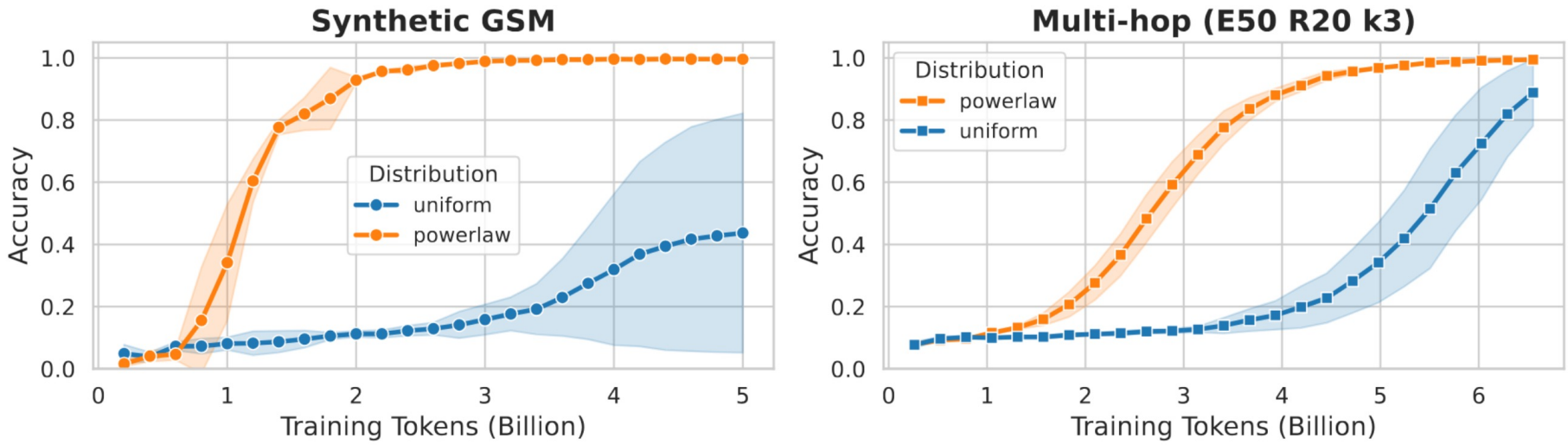


Figure 4. **Left.** Test accuracy on synthetic iGSM data. The operations are restricted within 2-8. All arithmetic calculation are done with modulo $p = 211$. **Right.** A multi-hop task with $|E| = 50$ individuals, with each person has $|R| = 20$ relations, and with hop $k = 3$.

(Empirical validation mainly on synthetic tasks; otherwise we don't know what skills are inside the multi-hop reasoning)

A Minimalist Example: k-Multiplicative Composition

Input: A sequence of skill IDs, $\boldsymbol{X} = (s_1, \dots, s_k)$

Ground Truth Skill Vector: $\boldsymbol{w}^* \in \{-1, +1\}^d$

Target: $y(\boldsymbol{X}) = \prod_{i=1}^{k} w^*_{s_i}$

Model:

- Embed each s_i to $x_i =$ (the i-th vector in the standard basis)
- Compute $f(\boldsymbol{X}; \boldsymbol{w}) = \prod_{i=1}^{k} \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle$ (can be seen as an RNN)

Connection to Function Composition:

- Each skill is a multiplication function $g_s(h) = w^*_s \cdot h$
- The task is essentially asking for $g_{s_1} \circ g_{s_2} \circ \cdots \circ g_{s_k}$

A Minimalist Example: k-Multiplicative Composition

Input: A sequence of skill IDs, $\boldsymbol{X} = (s_1, \dots, s_k)$

Ground Truth Skill Vector: $\boldsymbol{w}^* \in \{-1, +1\}^d$

Target: $y(\boldsymbol{X}) = \prod_{i=1}^{k} w^*_{s_i}$

Model:

- Embed each s_i to $x_i =$ (the i-th vector in the standard basis)
- Compute $f(\boldsymbol{X}; \boldsymbol{w}) = \prod_{i=1}^{k} \langle \boldsymbol{w}, \boldsymbol{x}_i \rangle$ (can be seen as an RNN)

Question: If you train the model on n samples from D…

- **Choice 1: Uniform** D: $s_i \sim \text{uniform}(\{1, \dots, d\})$
- **Choice 2: Power Law** $\Pr_D[s_i = j] \propto j^{-a}$

Which one will lead to a smaller test loss on the uniform distribution?

Power Law Beats Uniform

Theorem 1 (based on SQ Lower Bound):

- For some ϵ, SGD on uniform distribution needs either $\Omega\left(d^{k/2}\right)$ samples or $\Omega\left(d^{k/2}\right)$ gradient evaluations to make test loss $< \epsilon$

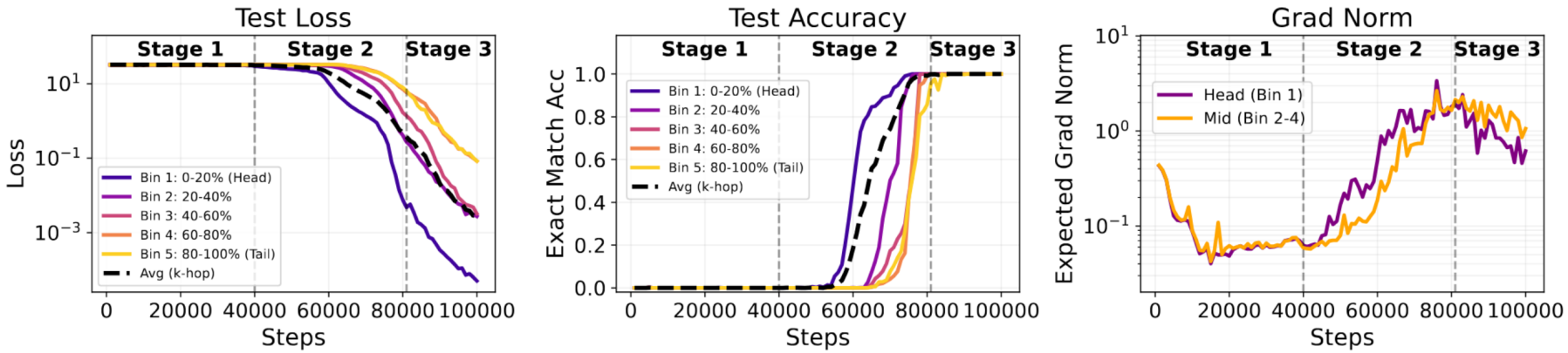
Theorem 2 (based on a careful dynamical analysis):

- For all ϵ, Online SGD on power law distribution $\Pr_D[s_i = j] \propto j^{-a}$ only needs $\tilde{O}(d^{2\alpha}/\epsilon)$ samples to make test loss $< \epsilon$

High-Level Explanation: Head skills help the tail

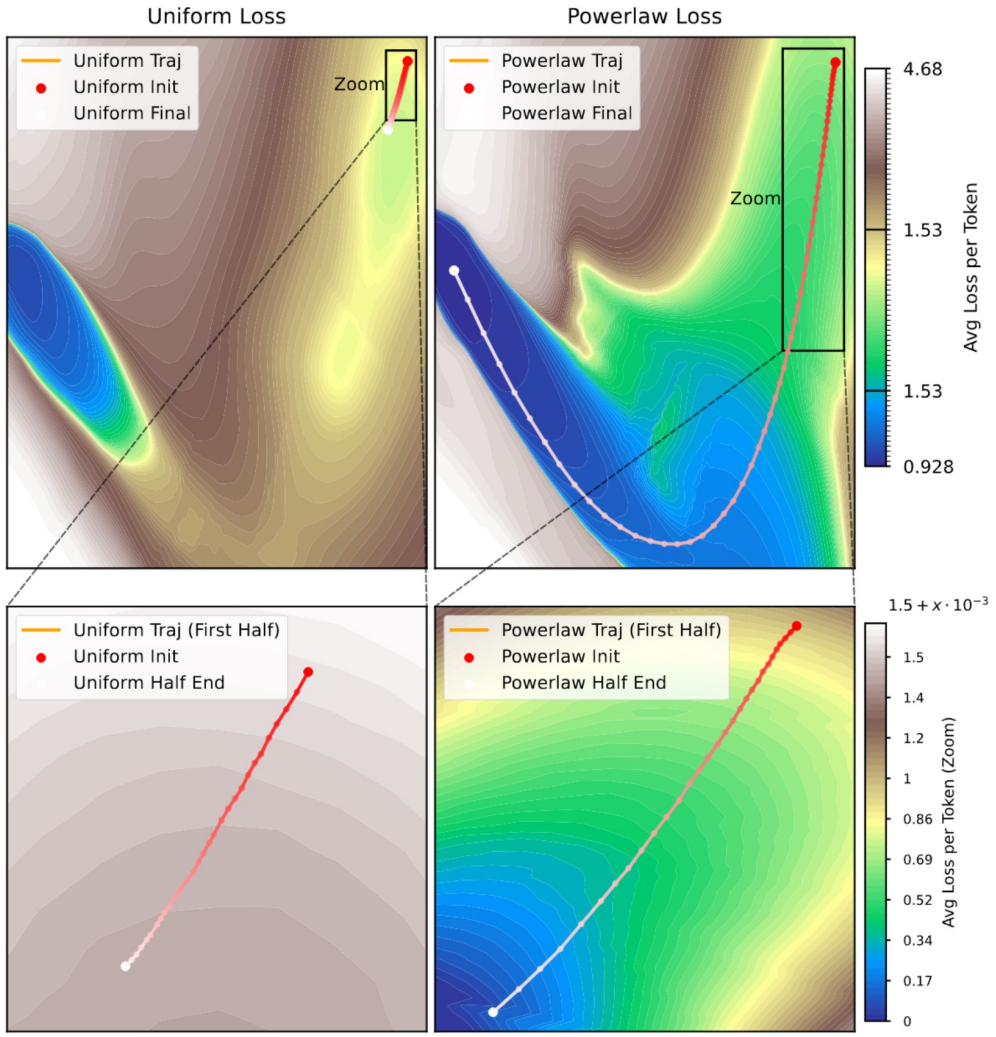
Under power law:

- A few “head skills” are sampled more frequently ⇒ Head skills are learned quickly
- “tail skills” are frequently combined with learned “head skills” ⇒ easier to learn



LLM experiments on State Tracking

Dynamical Viewpoint: Power law induces better loss Landscape



Summary: The Data Problem of Scaling Large Language Models

Scaling Laws for Multi-Epoch Training

- Larger datasets can be repeated more

SPA: A Simple Baseline for Knowledge Injection

- A mixture of fixed high-quality prompts is able to many complex methods at scale

In Principle, What Data Are Best?

- No simple answer. The test distribution may not be the best, for various reasons.
- Artificially making the data follow a power law can be good.