

Toward a Macroscopic Effective Theory of LLM Pretraining : Functional Scaling Laws

Lei Wu

School of Mathematical Sciences
Center for Machine Learning Research
Peking University

Oct. 22, 2026

Outline

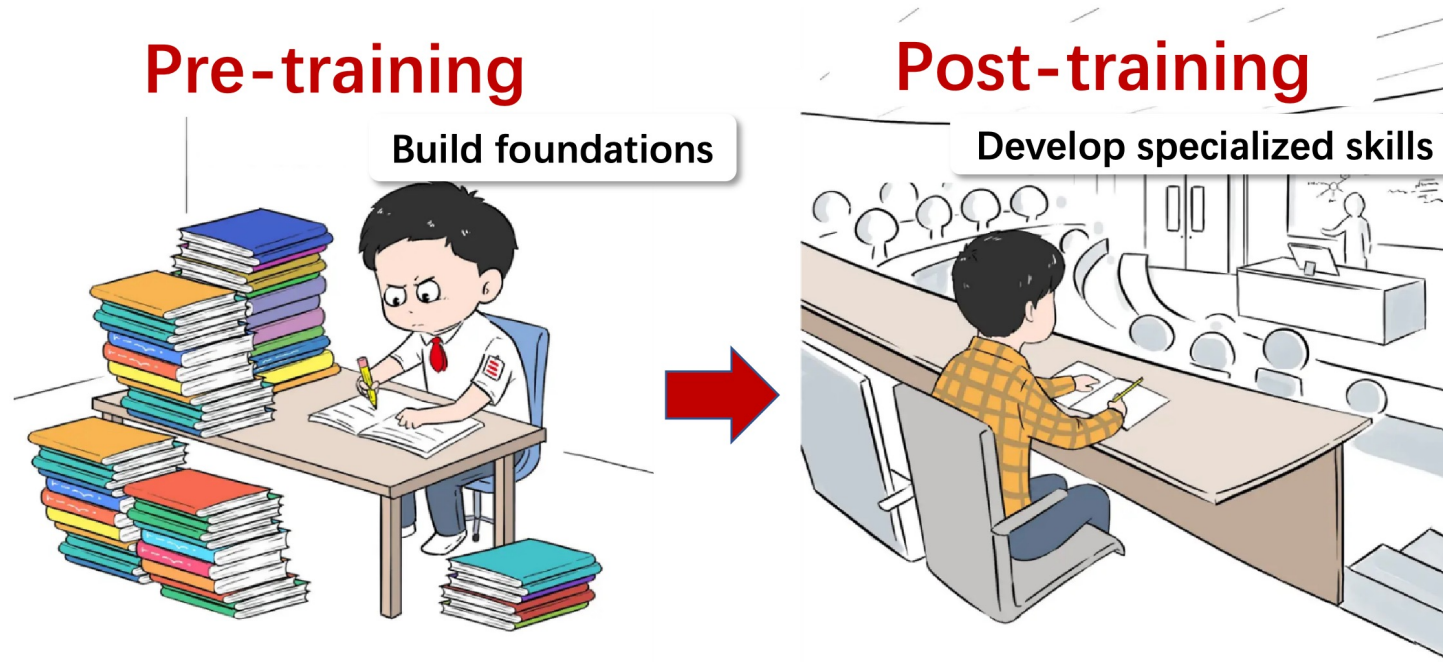
- Large language model pretraining
- Functional scaling laws (FSL)
- Prediction and design via FSL
- Beyond SGD: Universality of FSL
- Towards a macroscopic effective theory

LLM pretraining

- **The core of pretraining: scaling laws**
- **Hyperparameter**
- **Optimizer**
- **Architecture**
- **Data**

Two phases of LLM training

- ▶ **Pre-training:** Build general capability;
- ▶ **Post-training:** Strengthen reasoning & alignment.



Pre-training sets the upper bound of model capability.

Pretraining performs next-token prediction

Given the **previous tokens**, the model predicts the **next token**.

Example sentence

I love machine learning .

Model input (prefix)

Target (next token)

1	I	→	love
2	I love	→	machine
3	I love machine	→	learning
4	I love machine learning	→	.

Model's prediction for the next token

Given the prefix: "I"

love	0.70
hate	0.05
play	0.03
machine	0.01
...	...

► Training increases the probability of the **actual next token**.



Each position in a text becomes **one** supervised learning signal.

Mathematical formulation of pretraining

Goal: Learn a **general-purpose** generative model of language by solving following optimization problem:

$$\min_{\theta} -\mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \left[\sum_{l=2}^L \log P_{\theta}(x_l | x_1, x_2, \dots, x_{l-1}) \right] \quad (\text{next-token prediction}),$$

where

- ▶ **Model** P_{θ} : Large-scale Transformers (10B–1000B parameters);
- ▶ **Data** $\mathcal{D}$: Internet-scale corpora (10T–100T tokens).
- ▶ **Optimizer**: **One-pass** training, typically with **AdamW** or **Muon**.

Pretraining is extremely expensive

Pre-training is extraordinarily **compute-intensive**: it dominates the overall cost of training modern LLMs:

Training Costs	Pre-training	Post-training
in H800 GPU Hours	2664K	5K
in USD	\$5.328M	\$0.01M

Table 1: Training costs of DeepSeek-V3 (assuming H800 rental price is \$2 per GPU hour) ¹.

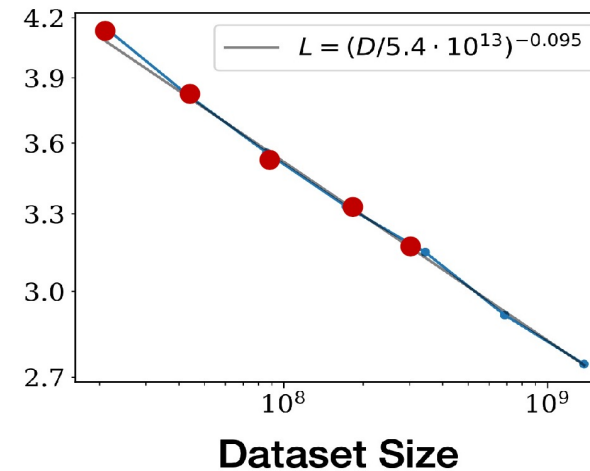
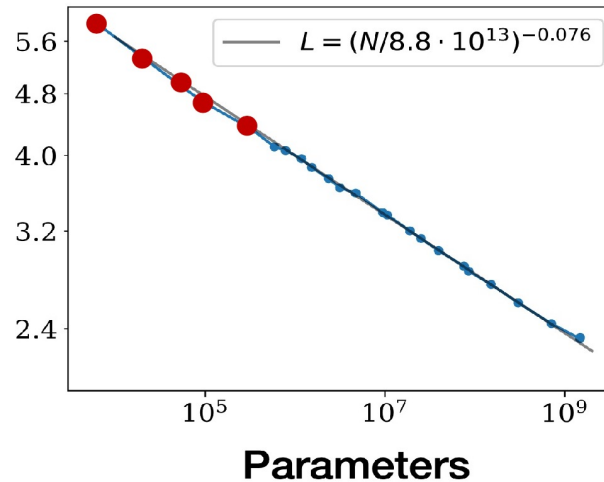
For pre-training, trial-and-error is nearly impossible.
A **principled science of pre-training** becomes essential.

Scaling laws for LLM pretraining

- ▶ The pre-training loss follows a **predictable** power-law relationship ²:

$$L(M, D) = L_0 + AM^{-a} + BD^{-b}$$

model size dataset size



- ▶ Enables **small-scale fits** to **predict large-scale performance**.

How do scaling laws emerge?

²Kaplan et al., *Scaling Laws for Neural Language Models*, OpenAI 2020.

Scaling laws guide compute-optimal training

- Given a compute budget C , how we choose model size and data size

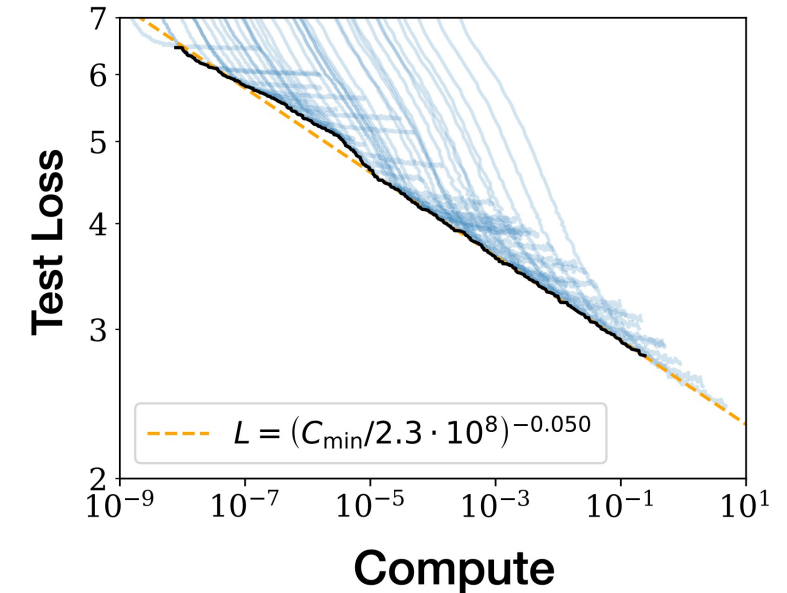
$$L(M, D) = L_0 + AM^{-a} + BD^{-b}$$

- This problem can be formulated as follows

$$\min_{M, D} L(M, D)$$

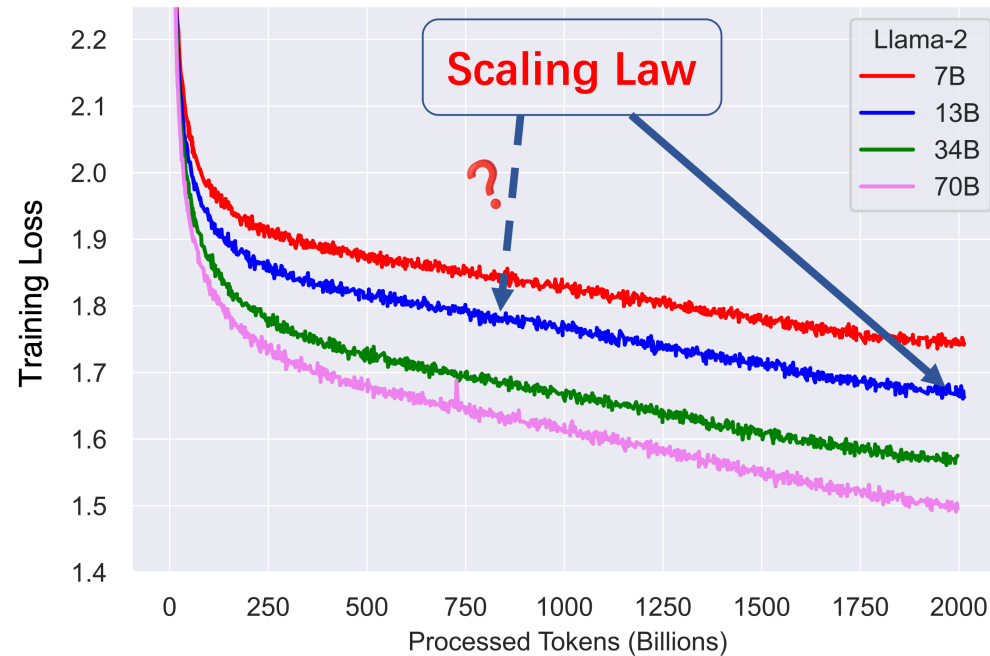
$$\text{s.t. } 6MD = C$$

Total compute



From final-loss scaling to trajectory-level laws

- Existing scaling laws only predict the final-step loss



The loss trajectories of Llama models. Here $\text{tokens} = \text{number of steps} * \text{batch size}$

Does the entire loss dynamics obey similar simple laws?

Learning rate schedule

- ▶ The **Cosine** learning rate schedule (LRS) is the default choice in many training setups.
- ▶ **Warmup–Stable–Decay (WSD)** ³ schedule has recently gained popularity (e.g., DeepSeek V3, Kimi K2).

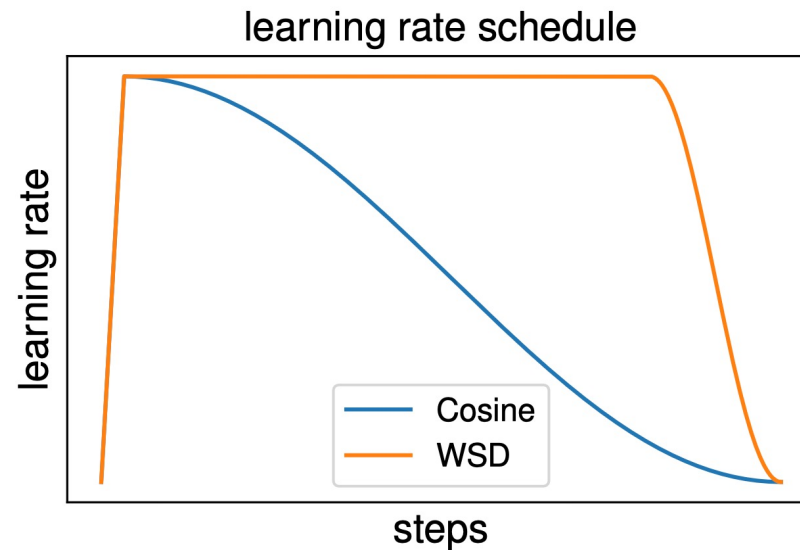


Figure 1: Comparison of Cosine and WSD learning rate schedules.

The loss dynamics under WSD

- ▶ **Stable phase:** loss decreases and then **saturates** over a long duration.
- ▶ **Decay phase:** a **fast drop** in loss occurs near the end of training.

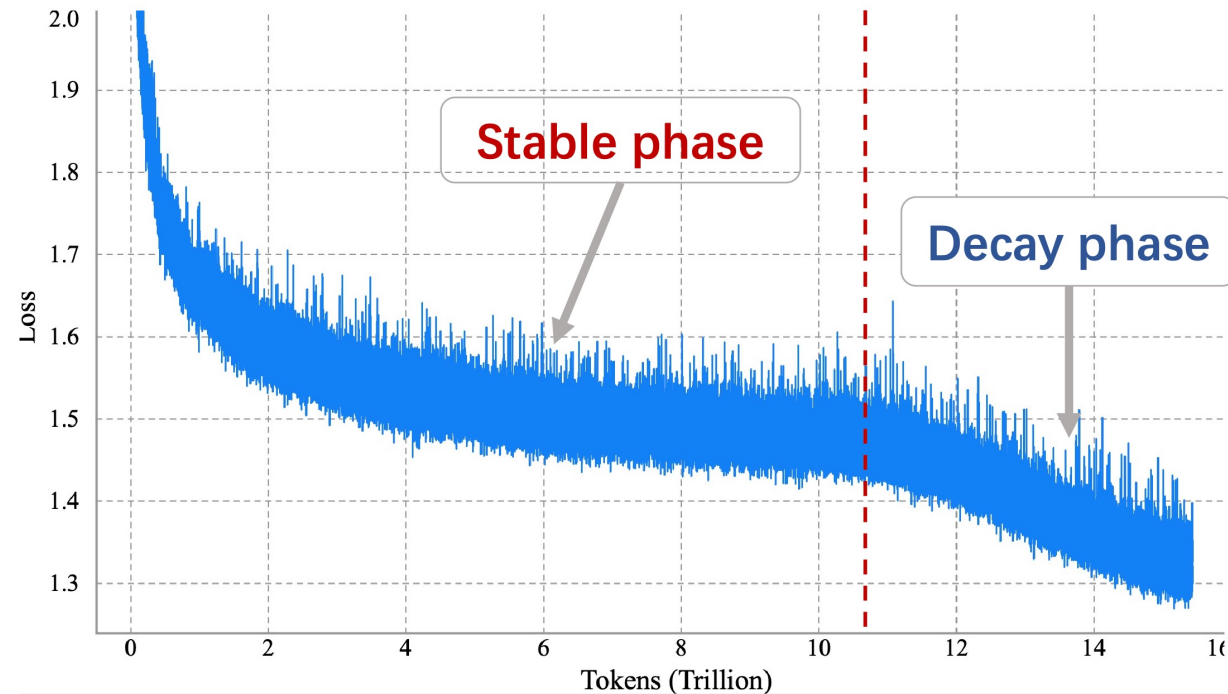
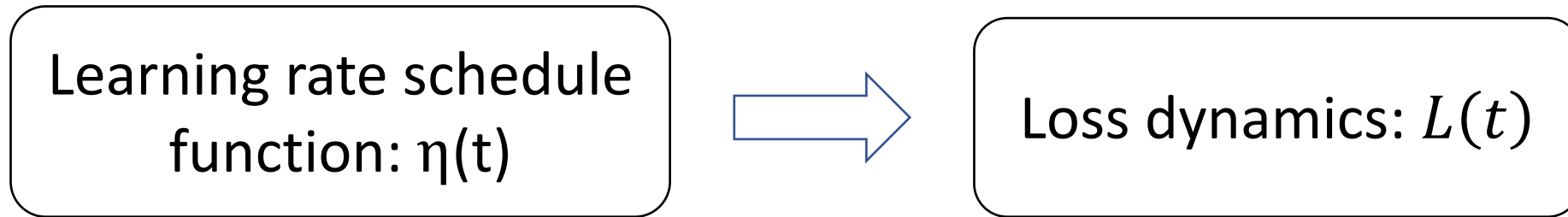


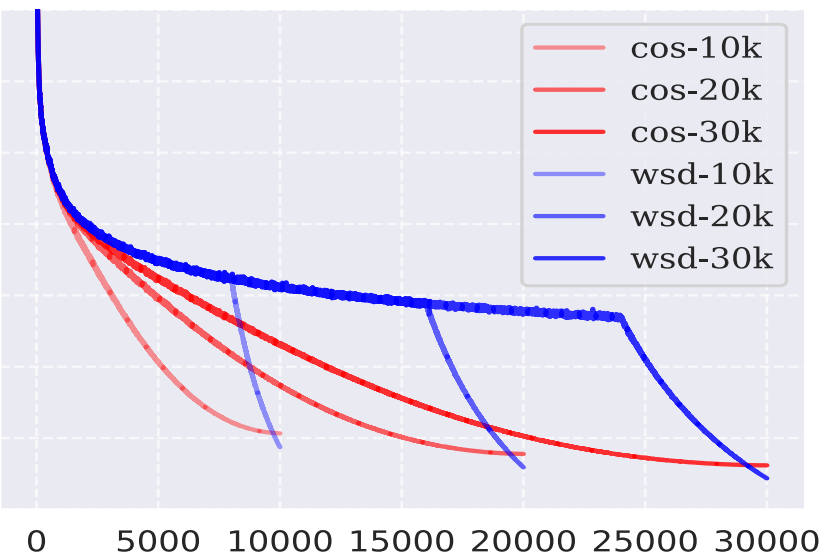
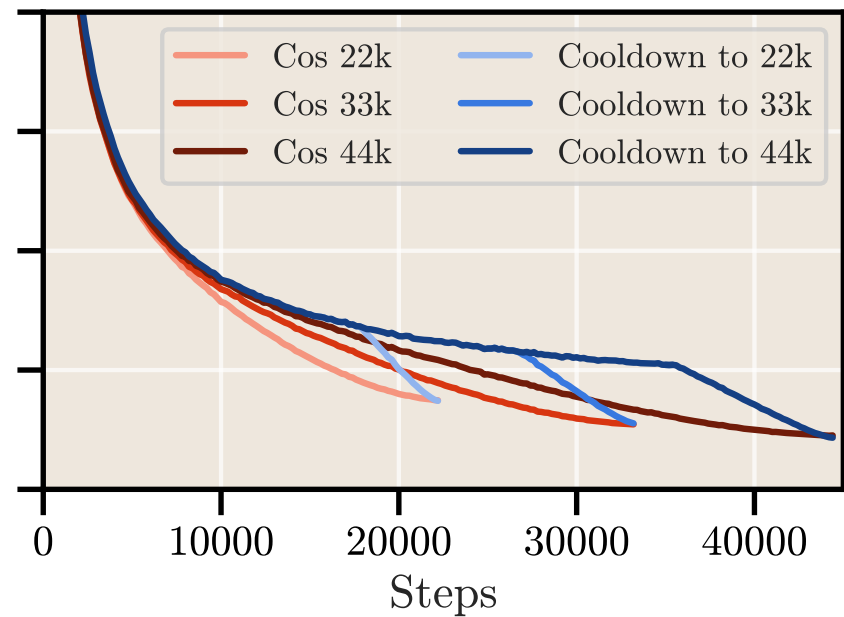
Figure 2: Loss curve of the **1T-parameter** Kimi-K2 model, where a WSD LRS is used ⁴.

Functional scaling laws



- Approach: toy models need to align with LLM pretraining
 - Scaling laws
 - LR-schedule dependent phenomena

Toy model



Toy model: Linear regression

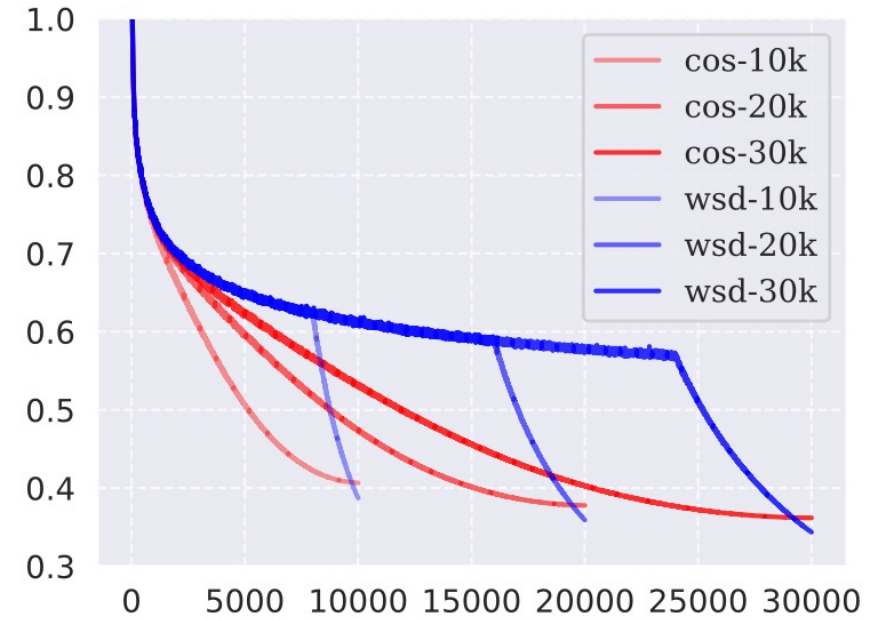
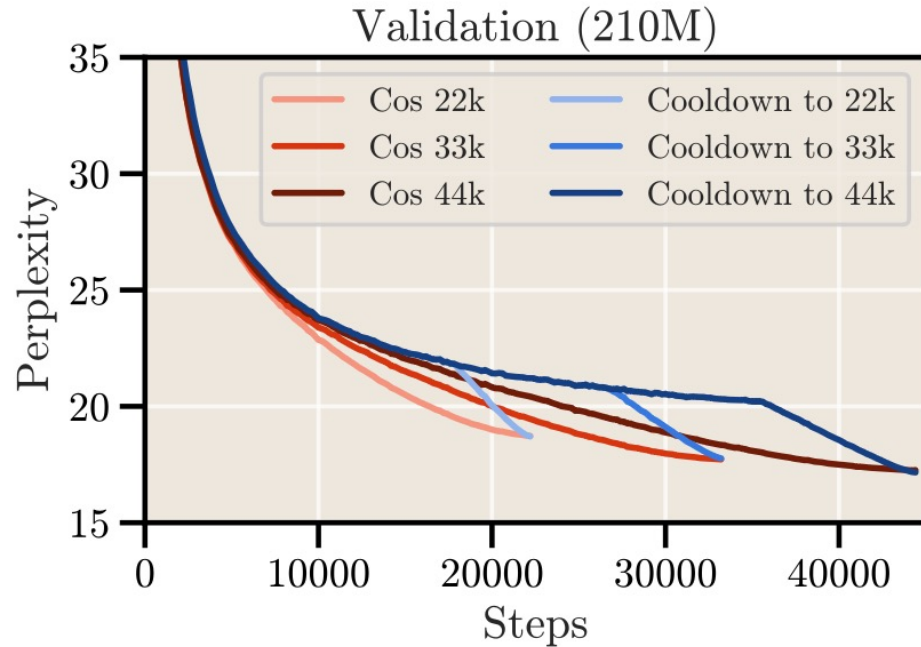


Figure 5: Left: Pre-training of a LLaMA(210M) ⁶; **Right:** The loss curve of fitting a **linear regression** using **SGD** under the same learning rate schedule.

- ▶ WSD matches or even outperform Cosine;
- ▶ Sudden loss drop in decay phase.

⁶Hägele et al., *Scaling Laws and Compute-Optimal Training Beyond Fixed Training Durations*, NeurIPS 2024.

Toy model: linear regression **with structures**

- ▶ **Data distribution:** let $\mathbf{x} \sim \mathcal{D}$ and $y = f^*(\mathbf{x}) + \epsilon$ with $\epsilon \sim \mathcal{N}(0, \sigma^2)$ and

$$f^*(\mathbf{x}) := \langle \boldsymbol{\theta}^*, \boldsymbol{\phi}(\mathbf{x}) \rangle = \sum_{j=1}^N \theta_j^* \phi_j(\mathbf{x}) \quad (N \text{ can be } \infty).$$

- ▶ **Feature distribution:** assume $\boldsymbol{\phi}(\mathbf{x}) \sim \mathcal{N}(0, \mathbf{H})$ with

$$\mathbf{H} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N), \quad \lambda_1 > \lambda_2 \geq \dots \geq \lambda_N.$$

- ▶ **Student model** (with **width M**):

$$f(\mathbf{x}; \mathbf{v}) = \sum_{j=1}^M v_j \mathbf{w}_j^\top \boldsymbol{\phi}(\mathbf{x}) = \mathbf{v}^\top \mathbf{W} \boldsymbol{\phi}(\mathbf{x}). \quad (1)$$

- ▶ **Random Features:** $\mathbf{w}_j \stackrel{iid}{\sim} \mathcal{N}(0, \mathbf{I}_N)$ for $j \in [M]$.
- ▶ **Spectral Features:** $\mathbf{w}_j = \mathbf{e}_j$ for $j \in [M]$, i.e., taking the top- M features $\{\phi_j\}_{j=1}^M$.

Model and data structures

Structural Assumptions

- ▶ **Model-capacity index** β : $\lambda_j \asymp j^{-\beta}$.
- ▶ **Task-difficulty index** s : $\theta_j^* \asymp j^{-1/2} \lambda_j^{(s-1)/2} \asymp j^{-(s\beta-\beta+1)/2}$.

Define normalized features $\hat{\phi}_j = \phi_j / \sqrt{\lambda_j}$ so that $\mathbb{E}_{\mathbf{x}}[\hat{\phi}_j(\mathbf{x})^2] = 1$.

- ▶ β quantifies the capacity of the student model:

$$f_M(\mathbf{x}; \mathbf{v}) = \sum_{j=1}^M v_j \phi_j(\mathbf{x}) \asymp \sum_{j=1}^M v_j j^{-\beta/2} \hat{\phi}_j(\mathbf{x}).$$

- ▶ s captures the relative difficulty of the task:

$$f^*(\mathbf{x}) = \sum_{j=1}^N \theta_j^* \phi_j(\mathbf{x}) \asymp \sum_{j=1}^N j^{-(s\beta+1)/2} \hat{\phi}_j(\mathbf{x}).$$

Gradient descent (GD) = multi-task learning

- ▶ Consider GD:

$$\dot{\boldsymbol{\nu}}(t) = -\nabla \mathcal{R}(\boldsymbol{\nu}(t)), \quad \boldsymbol{\nu}(0) = \mathbf{0}.$$

- ▶ Each coordinate evolves independently: For $j = 1, 2, \dots, M$:

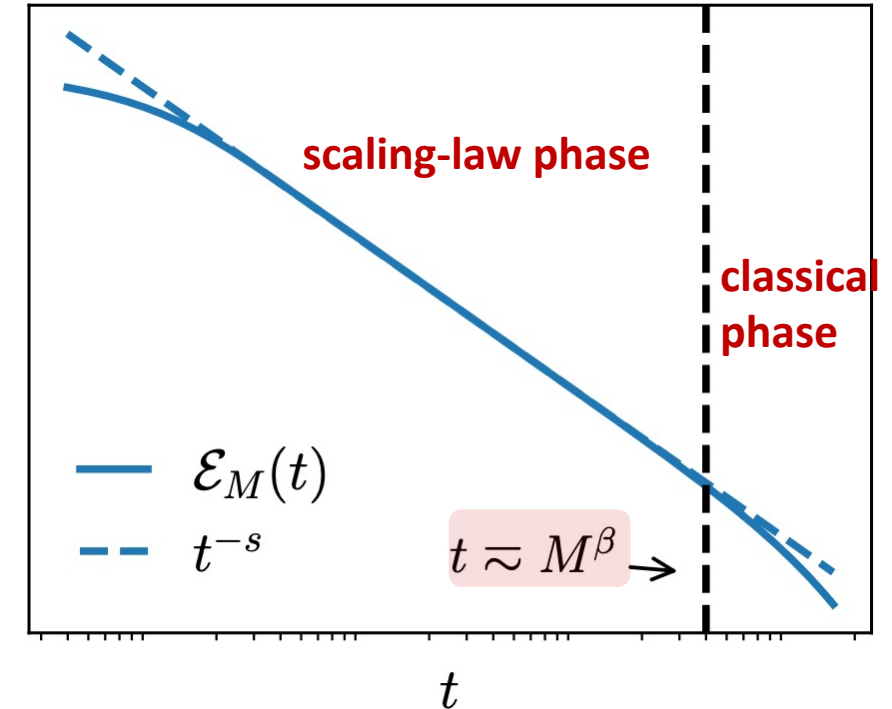
$$\dot{\nu}_j(t) = -\lambda_j(\nu_j(t) - \theta_j^*) \quad \Rightarrow \quad (\nu_j(t) - \theta_j^*)^2 = |\theta_j^*|^2 e^{-2\lambda_j t}.$$

- ▶ Population loss:

$$\begin{aligned} \mathcal{E}(t) &= \mathbb{E}_{\mathbf{x}}[|f(\mathbf{x}; \mathbf{v}_t) - f^*(\mathbf{x})|^2] = \sum_{j=1}^{\infty} \lambda_j (\theta_j(t) - \theta_j^*)^2 \\ &= \sum_{j=1}^M \lambda_j |\theta_j^*|^2 e^{-2\lambda_j t} + \sum_{j=M+1}^{\infty} \lambda_j |\theta_j^*|^2 \\ &\approx t^{-s} + M^{-s\beta}. \end{aligned}$$

Scaling law is a **collective** behavior

$$\begin{aligned} S(t) &:= \sum_{j=1}^M \lambda_j |\theta_j^*|^2 e^{-2\lambda_j t} = \sum_{j=1}^M j^{-1-s\beta} e^{-2j^{-\beta} t} \\ &\approx \int_1^M z^{-1-s\beta} e^{-2z^{-\beta} t} \\ &\asymp t^{-s} \quad \text{if} \quad t \lesssim M^\beta. \end{aligned}$$



Scaling law = superposition of **many** subtasks,
i.e., $M \gg 1$

Loss = tail mass beyond a moving cutoff

$$S(t) = \sum_{j=1}^{\infty} a_j e^{-2\lambda_j t}, \quad \lambda_j \sim j^{-\beta}, \quad a_j = j^{-(1+s\beta)}.$$

Dynamic picture:

- ▶ Each mode has a characteristic time scale $1/\lambda_j$
- ▶ Fast modes ($1/\lambda_j \ll t$): already decayed
- ▶ Slow modes ($1/\lambda_j \gg t$): essentially unchanged

Moving frontier:

$$t \sim 1/\lambda_{j_t} \Rightarrow j_t \sim t^{1/\beta}$$

Loss = tail beyond the frontier:

$$\begin{aligned} S(t) &= \sum_{j < j_t} a_j \underbrace{e^{-2\lambda_j t}}_{\text{negligible}} + \sum_{j > j_t} a_j \underbrace{e^{-2\lambda_j t}}_{\approx 1} \\ &\approx \sum_{j > j_t} a_j \approx \sum_{j > t^{1/\beta}} j^{-(1+s\beta)} \asymp t^{-s}. \end{aligned}$$

Power law = tail mass beyond a **moving cutoff**

Intrinsic-time SDE model of SGD

- ▶ SGD updates as follows

$$\mathbf{v}_{k+1} = \mathbf{v}_k - \eta_k (\nabla \mathcal{R}(\mathbf{v}_k) + \boldsymbol{\xi}_k).$$

- ▶ To isolate the effect of the learning rate schedule (LRS), we reparameterize time using the **intrinsic time**:

$$\text{training steps: } k \quad \text{vs.} \quad \text{intrinsic time: } \sum_{j=1}^k \eta_j.$$

- ▶ Then, the continuous-time limit of SGD w.r.t the intrinsic time is given by

$$d\boldsymbol{\nu}_t = -\nabla \mathcal{R}(\boldsymbol{\nu}_t) dt + \sqrt{\frac{\eta(t)}{B}} \Sigma(\boldsymbol{\nu}_t) d\mathbf{B}_t, \quad (3)$$

where $\eta(t)$ is the learning rate at the intrinsic time t , and $\mathbf{B}_t$ denotes the Brownian motion.

Functional Scaling Law

Theorem 1 (Intrinsic-time FSL)

Let $\sigma^2 = \Omega(1)$. Then, for the intrinsic-time SDE with top- M features, the following **FSL** holds for sufficiently large M and t :

$$\mathbb{E}[\mathcal{R}(\boldsymbol{\nu}_t)] - \underbrace{\frac{1}{2}\sigma^2}_{\text{irreducible}} \asymp \underbrace{\frac{1}{M^{s\beta}}}_{\text{approx. error}} + \underbrace{\frac{1}{t^s}}_{\text{signal learning}} + \underbrace{\frac{\sigma^2}{B} \int_0^t K(t-\tau) \eta(\tau) d\tau}_{\text{noise accumulation}},$$

where the forgetting kernel $K(t) = \sum_j \lambda_j^2 e^{-2\lambda_j t} \asymp (1+t)^{-(2-1/\beta)}$.

- FSL captures the **entire risk curve** and holds for **general** learning-rate schedules.

Signal learning

Theorem 1 (Intrinsic-time FSL)

Let $\sigma^2 = \Omega(1)$. Then, for the intrinsic-time SDE with spectral features and $s \in (0, 2 - 1/\beta]$, the following **FSL** holds for sufficiently large M and t :

$$\mathbb{E}[\mathcal{R}(\boldsymbol{\nu}_t)] - \underbrace{\frac{1}{2}\sigma^2}_{\text{irreducible}} \asymp \underbrace{\frac{1}{M^{s\beta}}}_{\text{approx. error}} + \underbrace{\frac{1}{t^s}}_{\text{signal learning}} + \underbrace{\frac{\sigma^2}{B} \int_0^t K(t-\tau)\eta(\tau) d\tau}_{\text{noise accumulation}},$$

where the forgetting kernel $K(t) \asymp (1+t)^{-(2-1/\beta)}$.

- ▶ The **signal learning** (learn f^* with full-batch GD) depends on the **relative difficulty** s .
- ▶ For a fixed task ($\alpha = s\beta$), **higher-capacity model** ($\beta \downarrow$) **converges faster** as $s = \alpha/\beta$ is larger.

Noise forgetting

Theorem 1 (Intrinsic-time FSL)

Let $\sigma^2 = \Omega(1)$. Then, for the intrinsic-time SDE with spectral features, the following **FSL** holds for sufficiently large M and t :

$$\mathbb{E}[\mathcal{R}(\boldsymbol{\nu}_t)] - \underbrace{\frac{1}{2}\sigma^2}_{\text{irreducible}} \asymp \underbrace{\frac{1}{M^{s\beta}}}_{\text{approx. error}} + \underbrace{\frac{1}{t^s}}_{\text{signal learning}} + \underbrace{\frac{\sigma^2}{B} \int_0^t K(t-\tau)\eta(\tau) \mathrm{d}\tau}_{\text{noise accumulation}},$$

where the forgetting kernel $K(t) := \sum_j \lambda_j^2 e^{-2\lambda_j t} \asymp (1+t)^{-(2-1/\beta)}$.

- ▶ The **noise accumulation** is characterized by a **convolution-type functional**.
- ▶ The **forgetting kernel** K quantifies **how fast the previously introduced noise is forgotten**.
- ▶ Higher-capacity ($\beta \downarrow$) models tend to forget noise slower.

FSL = dynamical **signal-noise** decomposition

$$\mathbb{E}[\mathcal{R}(\boldsymbol{\nu}_t)] - \underbrace{\frac{\sigma^2}{2}}_{\text{irreducible}} \asymp \underbrace{M^{-s\beta}}_{\text{approx. error}} + \underbrace{\left(\underbrace{t^{-s}}_{\text{signal learning}} + \frac{\sigma^2}{B} \underbrace{\int_0^t K(t-\tau) \eta(\tau) \, d\tau}_{\text{noise accumulation}} \right)}$$

where

$$K(t) \asymp (1+t)^{-(2-1/\beta)}.$$

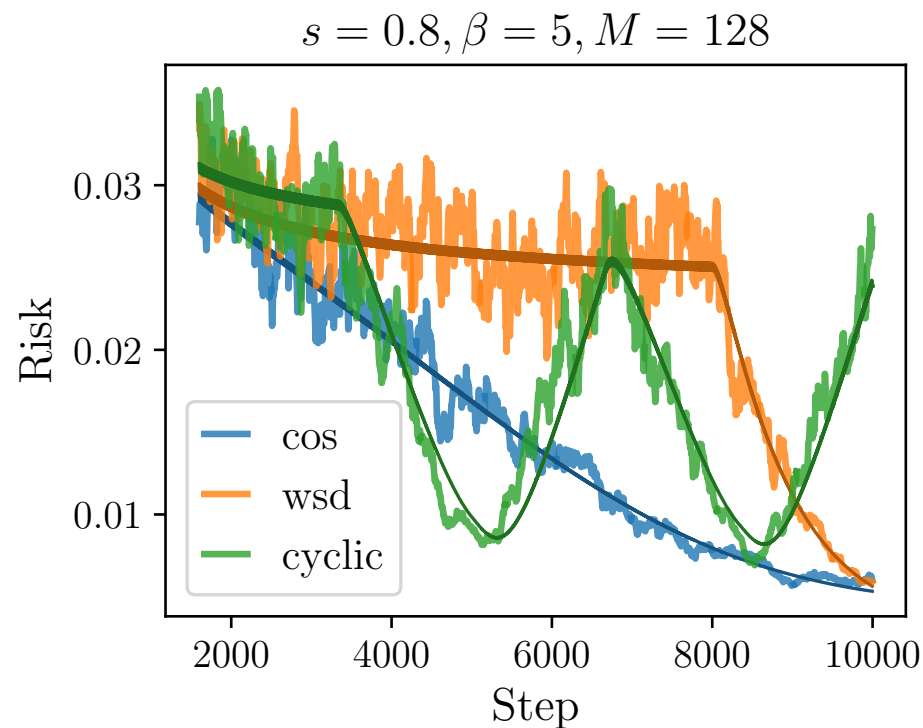
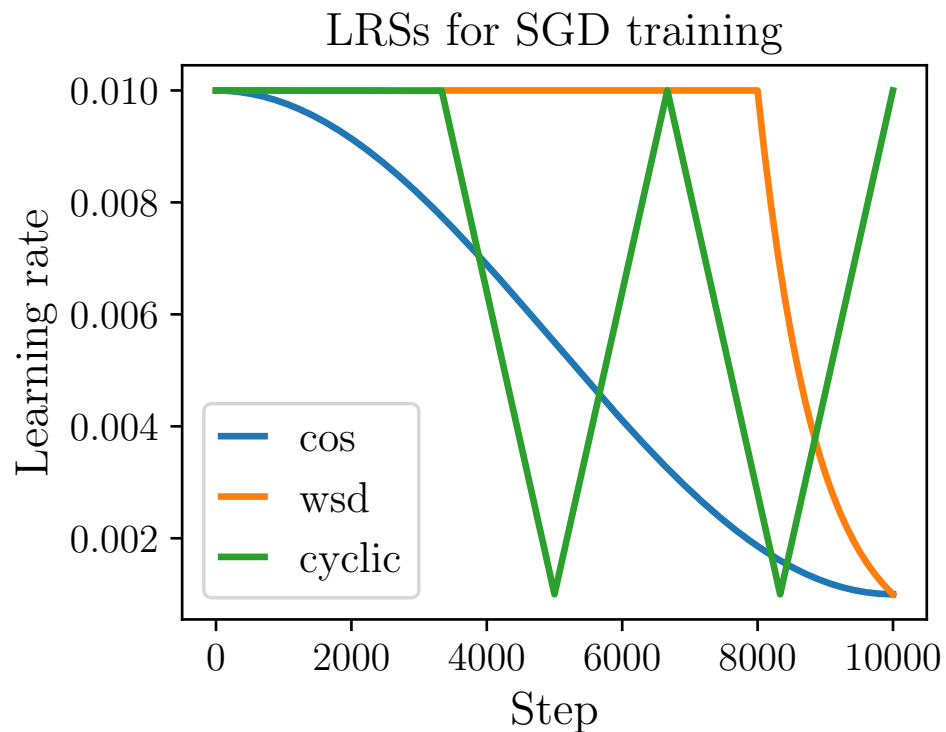
- ▶ **Decoupling.** Signal learning and noise accumulation evolve independently.
- ▶ **Power-law structure.** All components follow power laws.
- ▶ **The tradeoff between signal learning and noise forgetting** is governed by the learning rate schedule.

Prediction and design via FSL

- **Loss dynamics**
- **LR schedule**
- Batch size schedule

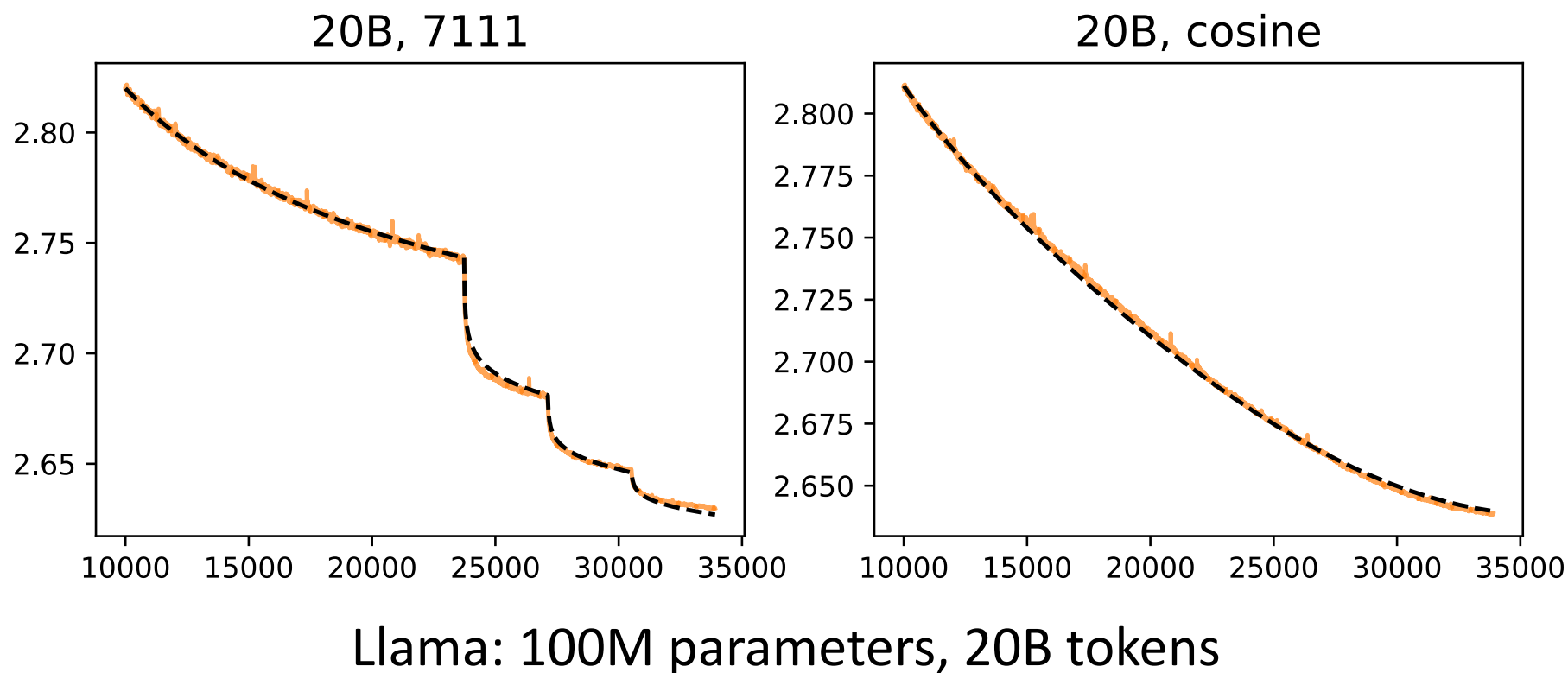
Toy model experiments

- FSL is derived for the **continuous-time SDE**, but accurately capture the dynamics of **discrete-time SGD**.

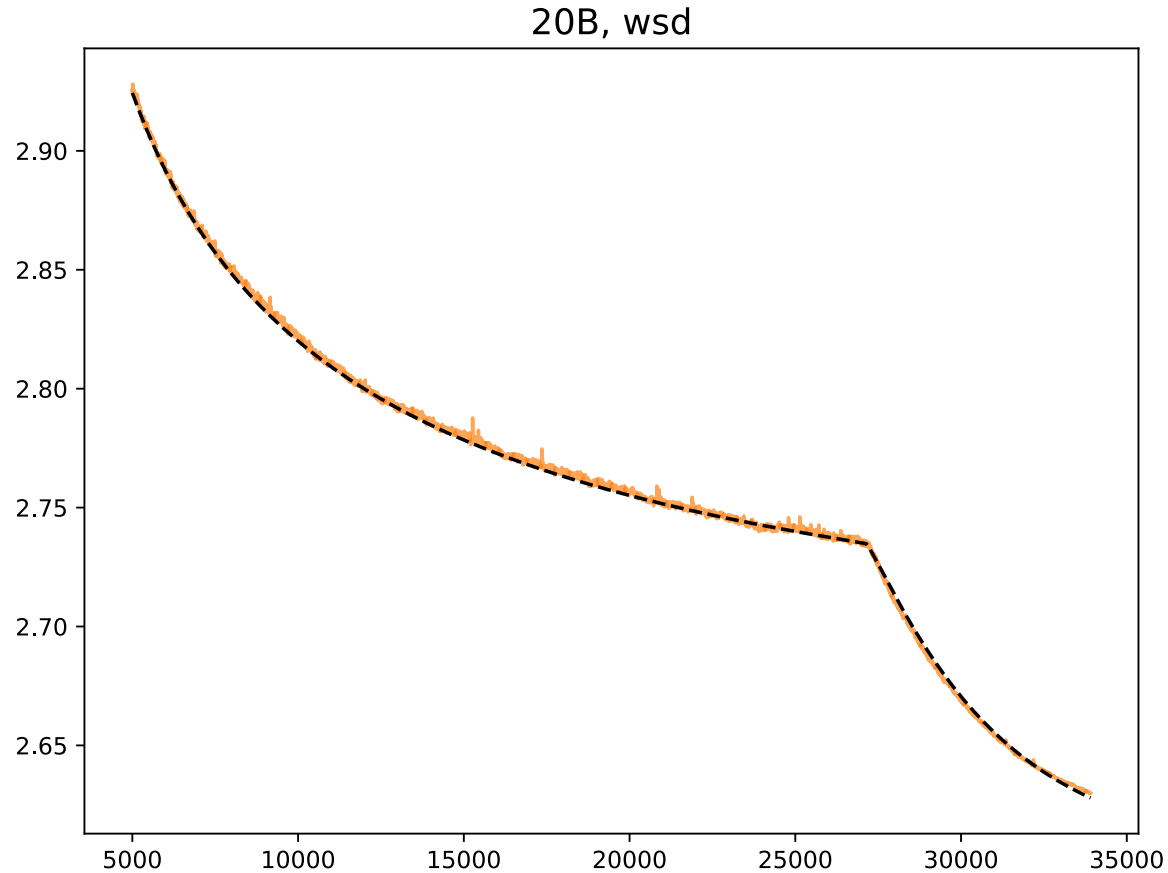


LLM pretraining experiments

- FSL also accurately capture the loss dynamics of LLM pretraining.



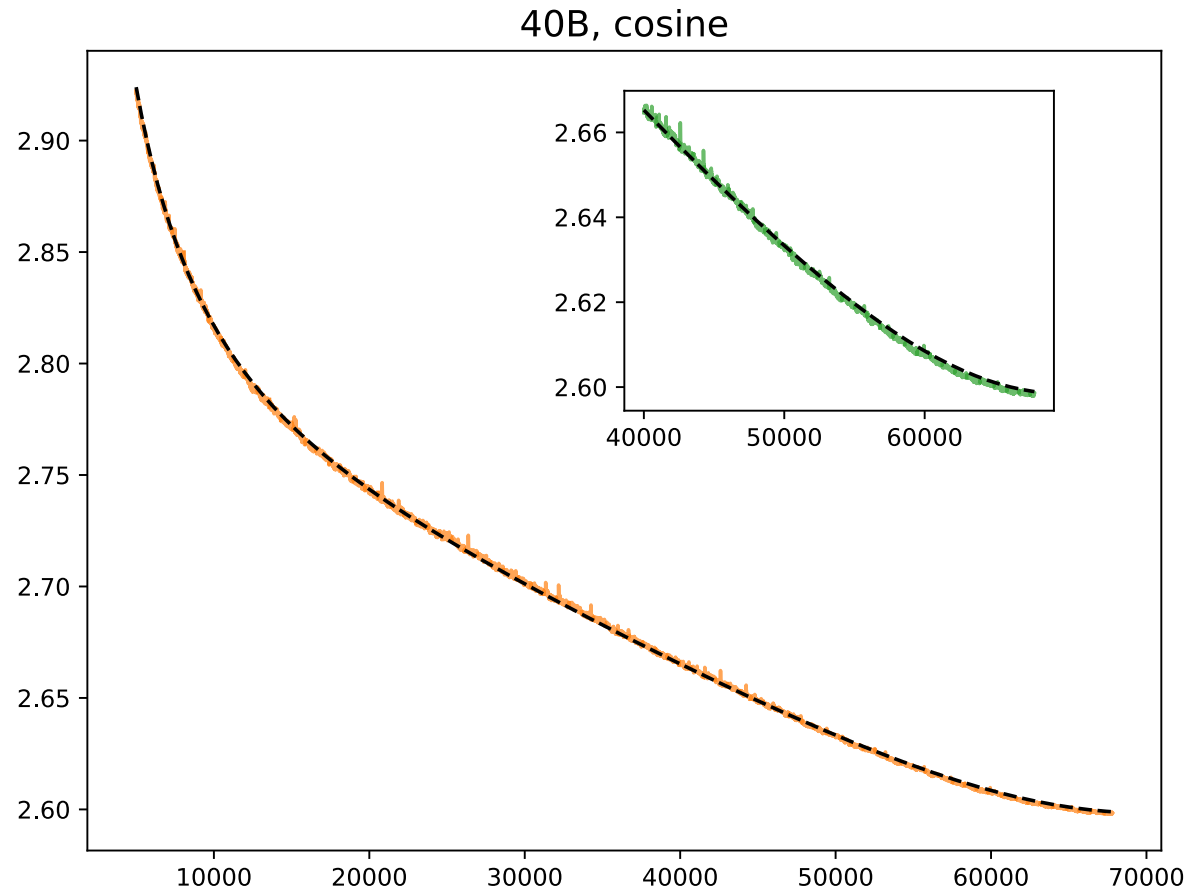
Prediction: LR shape generalization



7111+cosine -> WSD

➤ The prediction precision is on the order of **0.001**

Prediction: data extrapolation

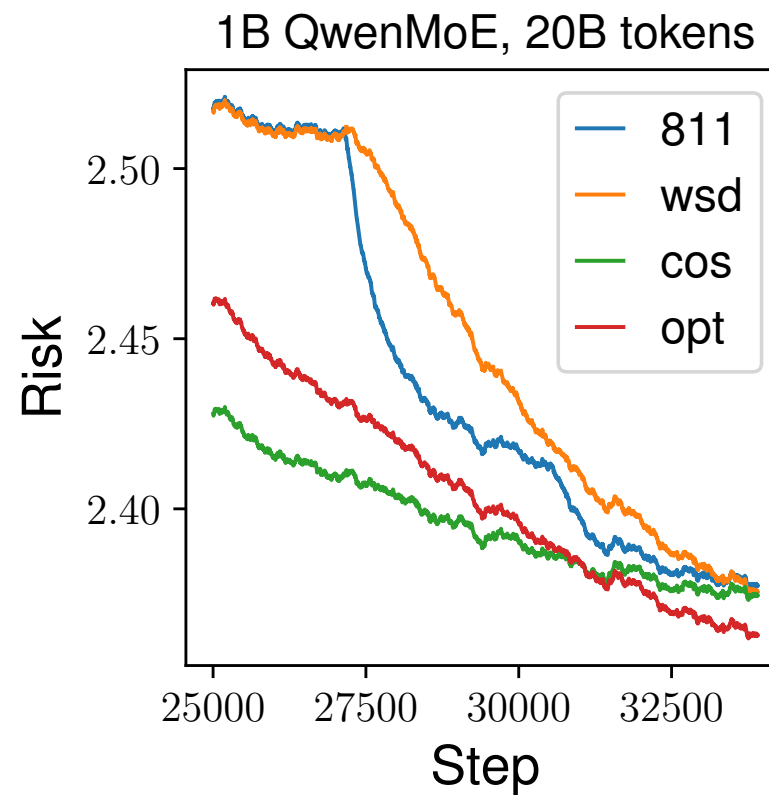
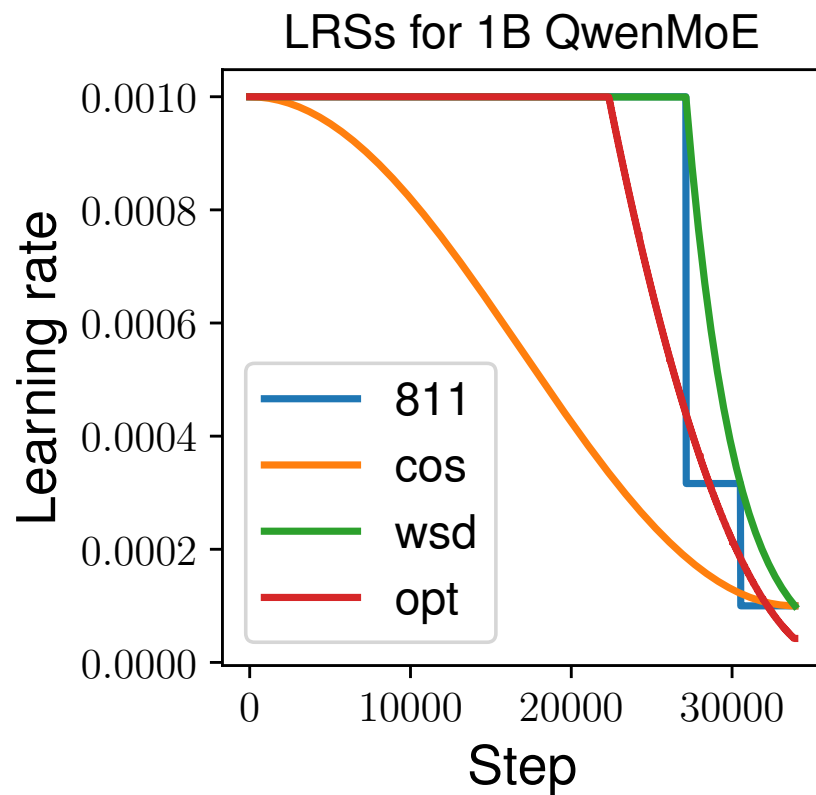


➤ The prediction precision is on the order of **0.001**

20B tokens -> 40B tokens; cosine LR schedule

LR schedule design via FSL

$$\min_{\eta \in Q} \mathcal{F}_T(\eta; \hat{\Theta})$$



Analyzing optimal LR schedule via FSL

- ▶ Let $\varphi(z)$ denotes the LRS in terms of training steps and introduce the intrinsic-time function

$$t(z) = \int_0^z \varphi(u) \, du.$$

Given a total of N steps, the optimal LRS is characterized by the following variational problem:

$$\begin{aligned} \min_{\varphi} \mathcal{F}[\varphi] &:= t(N)^{-s} + \int_0^N K (t(N) - t(z)) |t'(z)|^2 \, dz \\ \text{s.t. } t(0) &= 0, \\ 0 \leq t'(z) &\leq \eta_{\text{stability}} \quad \text{for a.e. } z \in [0, N], \end{aligned}$$

where the condition $\varphi(z) \leq \eta_{\text{stability}}$ is imposed due to stability constraint.

Easy regime: power decay from the beginning

Theorem 2 (Easy regime $\rightarrow$ Power decay)

There exists a $N_{\beta,s}$ such that when $N \geq N_{\beta,s}$, the optimal LRS φ_N^* is given by

► **Easy regime** ($s \geq 1 - 1/\beta$): We have

$$\varphi_N^*(z) = \eta_{\text{opt}}^* \left(1 - \frac{z}{N}\right)^{2\beta-1} \quad \eta_{\text{opt}}^* \asymp N^{-\frac{s-(1-\frac{1}{\beta})}{s+\frac{1}{\beta}}}$$

- The optimal schedule decays **from the beginning**.
- The optimal schedule is a **power-decay** shape.
- The **peak LR needs to scale** with the training horizon.

Hard tasks favor WSD: stay high, decay late

Theorem 2 (Hard regime $\rightarrow$ WSD)

There exists a $N_{\beta,s}$ such that when $N \geq N_{\beta,s}$, the optimal LRS φ_N^* is given by

► **Hard regime** ($s < 1 - 1/\beta$): We have

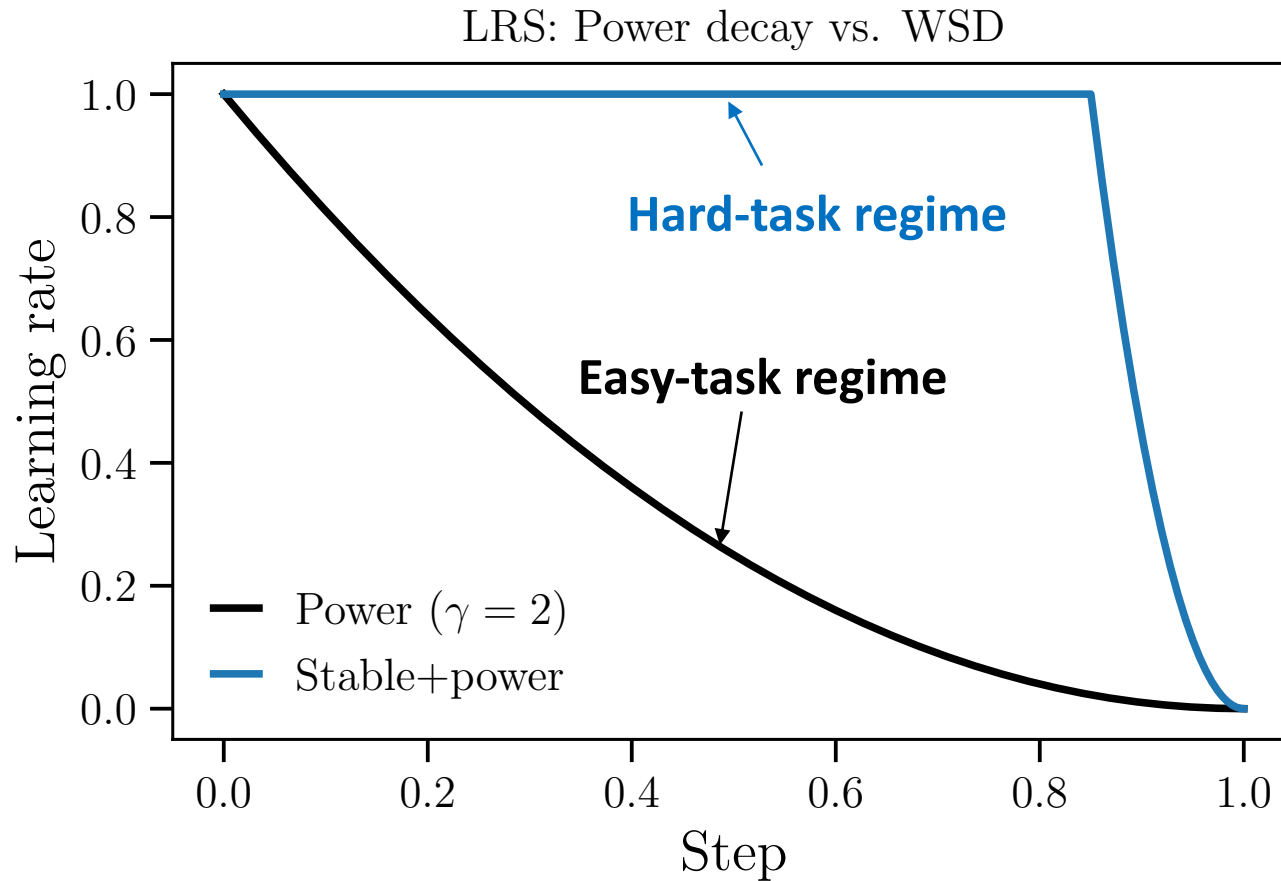
$$\varphi_N^*(z) = \begin{cases} \eta_{\text{stability}} & \text{if } 0 \leq z \leq N_1 \\ \eta_{\text{stability}} \left(1 - \frac{z - N_1}{N - N_1}\right)^{2\beta - 1} & \text{if } N_1 < z \leq N, \end{cases} \quad (6)$$

and moreover, **the duration ratio of decay phase** satisfies

$$\frac{N - N_1}{N} \asymp N^{-\alpha} \text{ with } \alpha = \frac{1 - \frac{1}{\beta} - s}{2 - \frac{1}{\beta}} > 0.$$

Optimal schedule = long stability phase + vanishing terminal power decay.

Easy vs. hard: Same decay shape, different intrinsic time



- For the same model, the same decay shape

$$(1 - u)^{2\beta - 1}$$

- For different tasks, different intrinsic times

Power decay: The forgetting kernel determines the shape

- For fixed intrinsic time, the signal-learning term is fixed

$$T^{-s} + \int_0^T K(t - \tau) \eta(\tau) d\tau$$

- We only need to minimize the noise injection:

$$\begin{array}{l} \min_{\varphi: [0, T] \rightarrow \mathbb{R}_{\geq 0}} \int_0^T \mathcal{K}(T - \tau) \varphi(\tau) d\tau \\ \text{s.t.} \quad \int_0^T \frac{1}{\varphi(\tau)} d\tau = N. \end{array} \quad \Rightarrow \quad \varphi^*(t) \propto \frac{1}{\sqrt{K(T - t)}}$$

WSD: Signal learning dominates in hard tasks

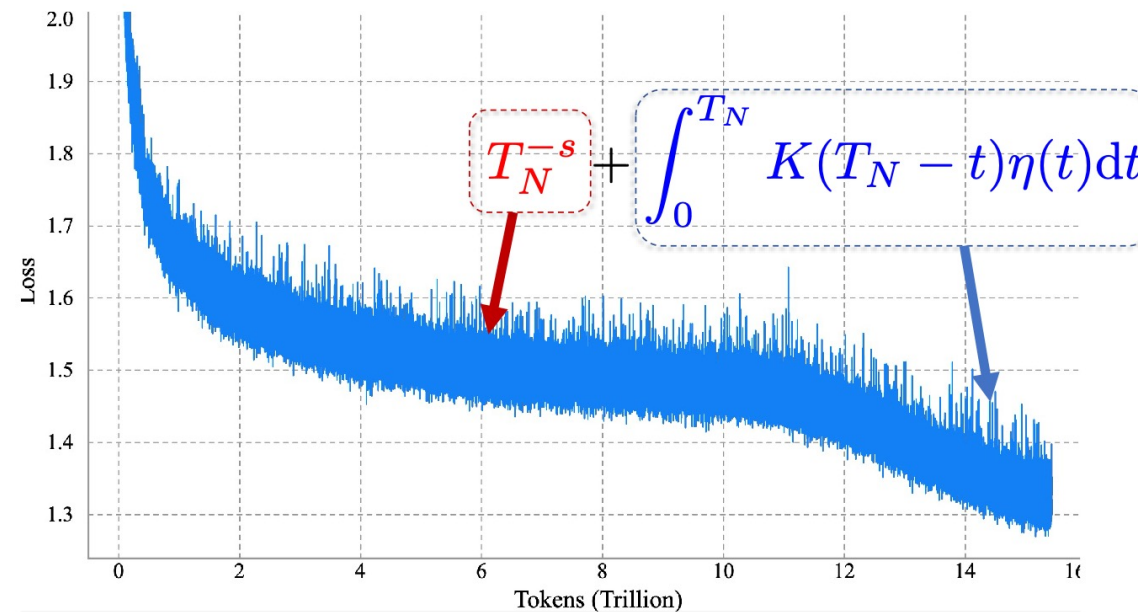
Let $T_N = \int_0^N \varphi(z) \, dz$ denote the total intrinsic time over N steps. Then,

$$\underbrace{T_N^{-s}}_{\text{Signal learning}} + \underbrace{\int_0^{T_N} (1 + T_N - t)^{-(2-1/\beta)} \eta(t) \, dt}_{\text{Noise term}}$$

- ▶ The **noise-forgetting** behavior is determined by the model capacity, *independent* of the task difficulty.
- ▶ When s is small (i.e., **hard tasks**), the signal-learning term becomes dominant, making it essential to **increase the total intrinsic time** T_N .
- ▶ Therefore, the learning rate should remain high for most of training and only decay near the end to maximize T_N .

WSD: Signal learning first, noise forgetting later

- ▶ **Stable phase:** Fit the signal aggressively; the **slow even saturate decay** arises from the gradual **accumulation of noise**.
- ▶ **Decay phase:** Forget the noise fast; the **fast drop** occurs because T_N^{-s} is already sufficiently small.



Rethink LR schedule: Polynomial decay

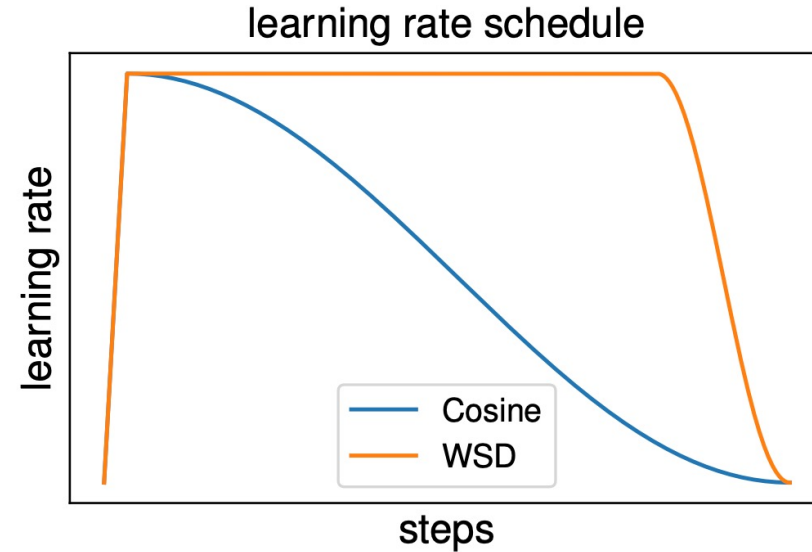
Polynomial decay performs poorly when s is small, because:

- ▶ Intrinsic time grows only **sublinearly** with N :

$$T_N = \sum_{k=1}^N k^{-\alpha} \asymp \begin{cases} N^{1-\alpha} & \text{if } \alpha \in (0, 1) \\ \log N & \text{if } \alpha = 1 \\ \text{constant} & \text{if } \alpha > 1. \end{cases}$$

Rethink LR schedule: Cosine

$$\varphi(z) = \frac{\eta_0}{2} \left(1 + \cos \left(\frac{z}{N} \pi \right) \right)$$



Cosine schedule is surprisingly effective:

- ▶ **Globally:** ensures sufficient intrinsic time: $T_N = \int_0^N \varphi(z) dz \asymp N$.
- ▶ **Locally (near final step):** induces a $(1 - z/N)^2$ -style decay, similar to optimal schedules due to

$$\varphi(z) = \frac{\eta_0 \pi^2}{4} \left(1 - \frac{z}{N} \right)^2 + o \left(\left(1 - \frac{z}{N} \right)^2 \right)$$

How Is FSL Universal?

From linear models + SGD to LLM Pretraining

- **Optimizer: SignSGD, Adam**
- Model: deep linear networks
- Regularization: Weight decay, Hyperball
- More aspects

FSL: signal learning + noise forgetting

The loss dynamics admits a noise-signal decomposition:

$$\mathcal{E}(t) \asymp \underbrace{S(t)}_{\text{signal learning}} + \underbrace{\int_0^t K(t-r) \eta(r) dr}_{\text{noise forgetting}},$$

with

Signal-learning exponent

$$S(t) \asymp t^{-\alpha},$$

Noise-forgetting exponent

$$K(t) \asymp t^{-\gamma}$$

Adam as a sign-like optimizer

➤ Adam:

$$\mathbf{m}_{k+1} = \beta_1 \mathbf{m}_k + (1 - \beta_1) \mathbf{g}_k,$$

$$\mathbf{v}_{k+1} = \beta_2 \mathbf{v}_k + (1 - \beta_2) \mathbf{g}_k^2,$$

$$\theta_{k+1} = \theta_k - \eta_k \mathbf{m}_k / \sqrt{\mathbf{v}_k},$$

➤ SignSGD is the limiting case with $\beta_1 = \beta_2 = 0$:

$$\theta_{k+1} = \theta_k - \eta_k \text{sign}(\mathbf{g}_t)$$

Why is the intrinsic time still right coordinate describe the dynamics?

$$t_k = \sum_{j=0}^k \eta_j$$

Sign-like optimizers preserve the FSL

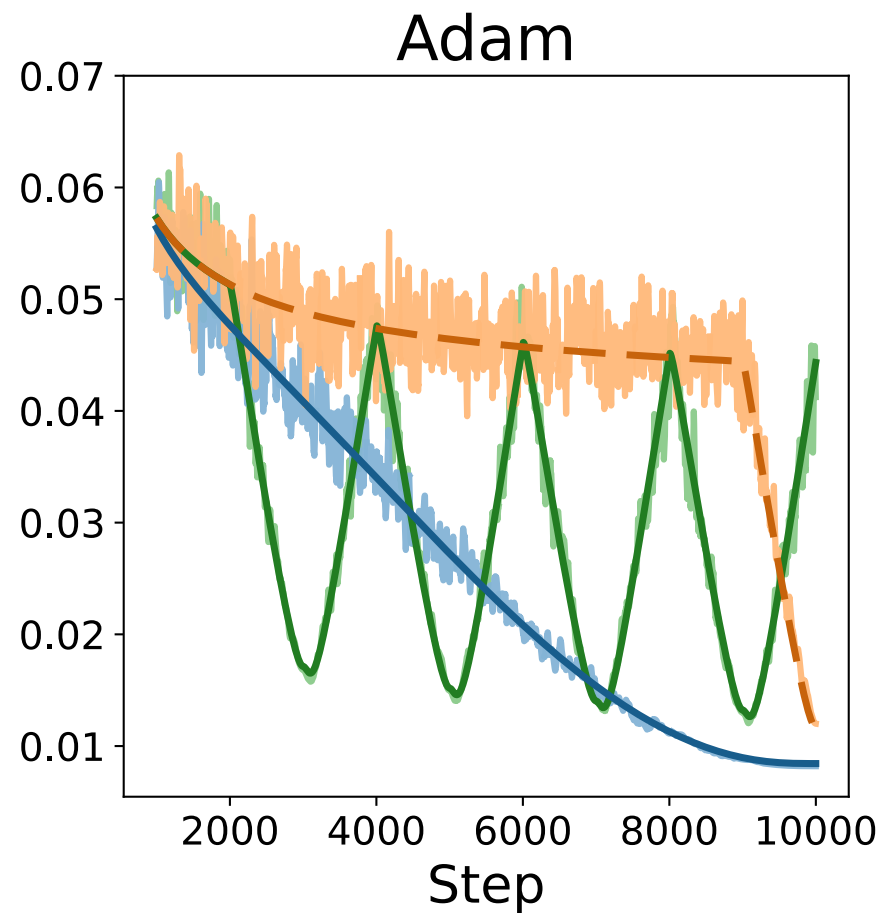
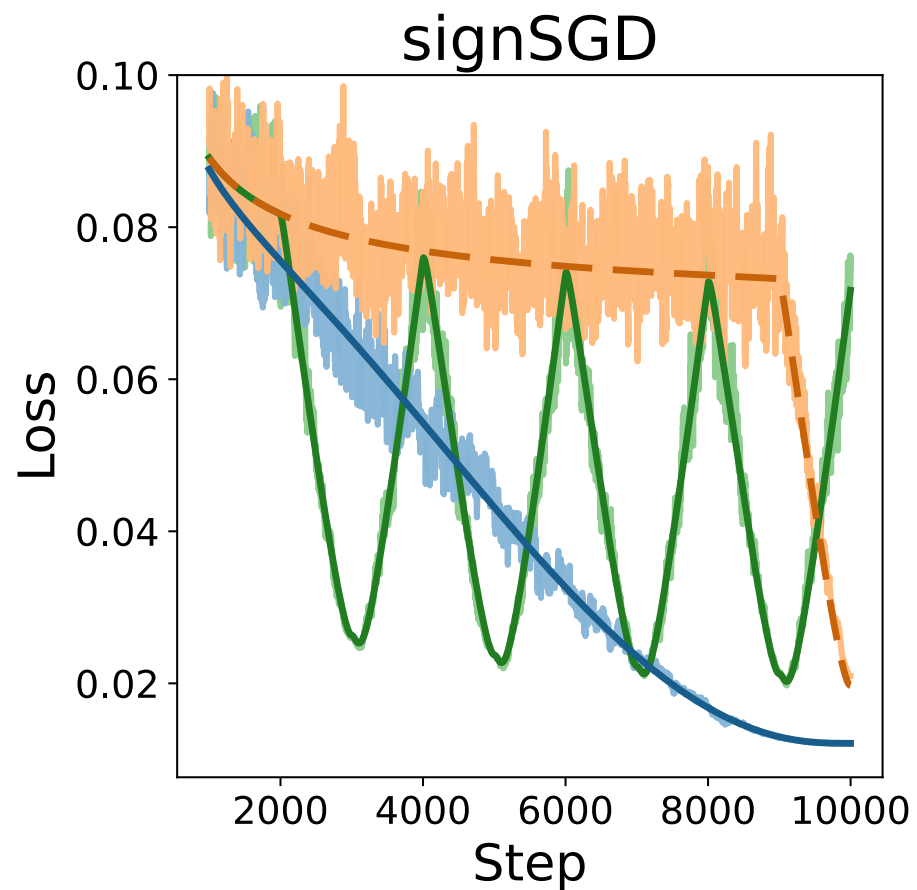
- For these sign-like optimizers, the same FSL applies but with

$$S_{\text{sgn}}(t) \asymp (\sqrt{B}t)^{-2s}, \quad K_{\text{sgn}}(t) = (\sqrt{B}t)^{-(2-2/\beta)}$$

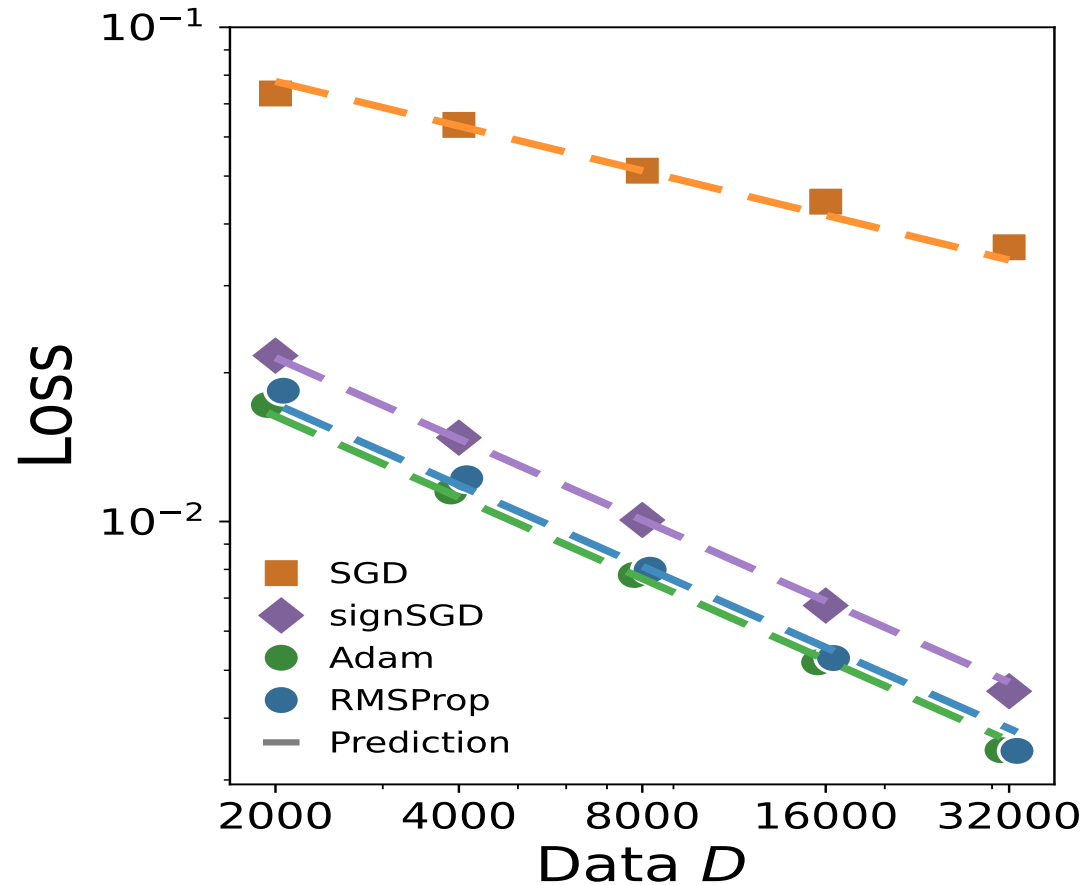
- For SGD, the same functional form but different kernels:

$$S_{\text{sgd}}(t) = t^{-s}, \quad K_{\text{sgd}}(t) = \frac{1}{B}t^{-(2-1/\beta)}$$

Toy model experiments

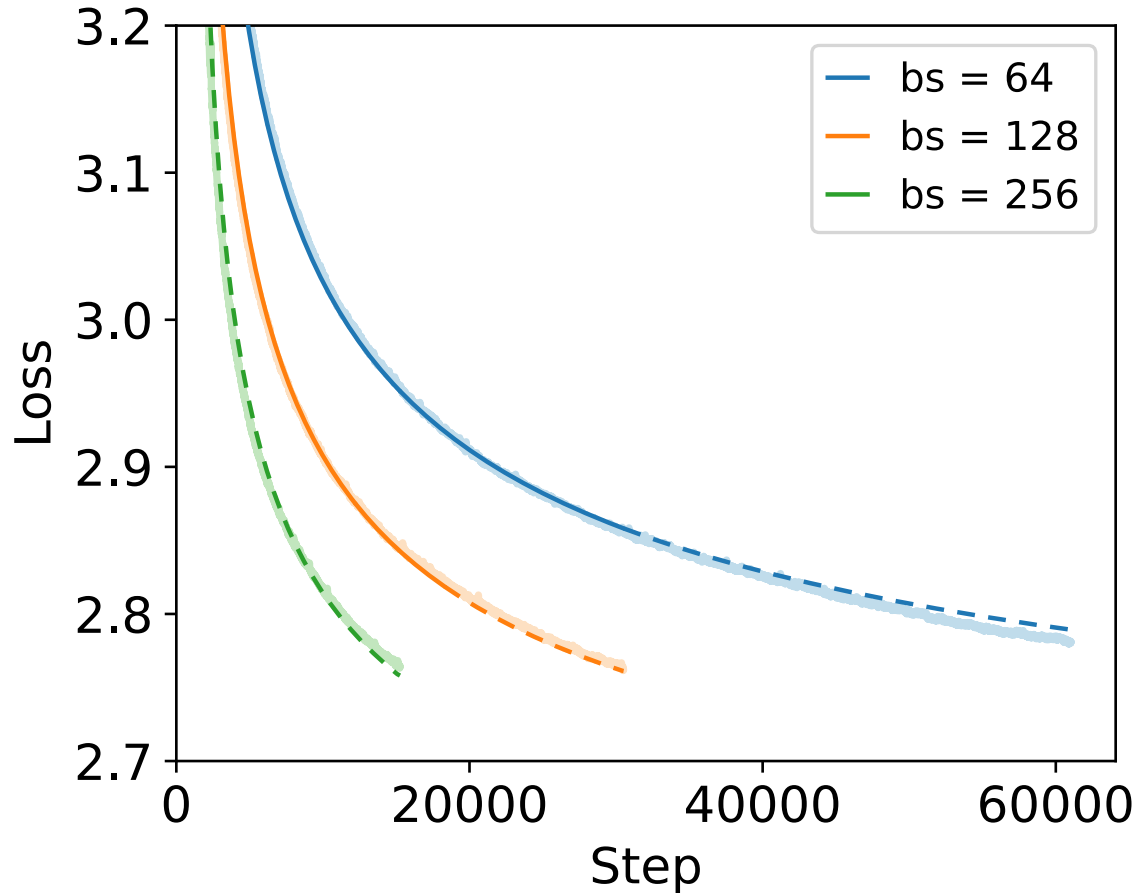


Toy model: sign-like optimizers share the same scaling



- Sign-like optimizers scale better than SGD, as predicted
- SignSGD, Adam, and RMSprop share the same scaling efficiency

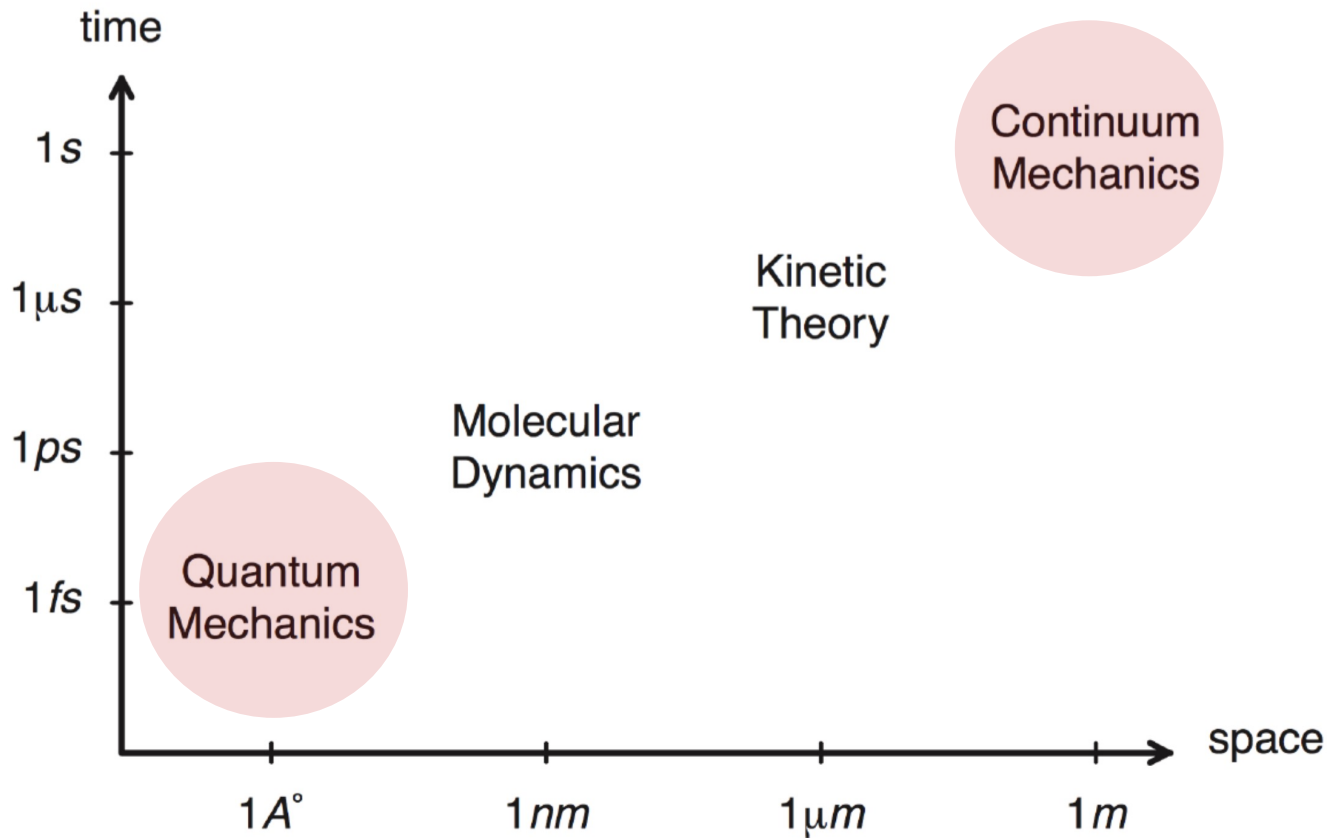
LLM: Data and batch size extrapolation



- Fit the new FSL on small-batch runs
- Extrapolate well to large-batch training

**Why do such simple laws govern
complex LLM pretraining?**

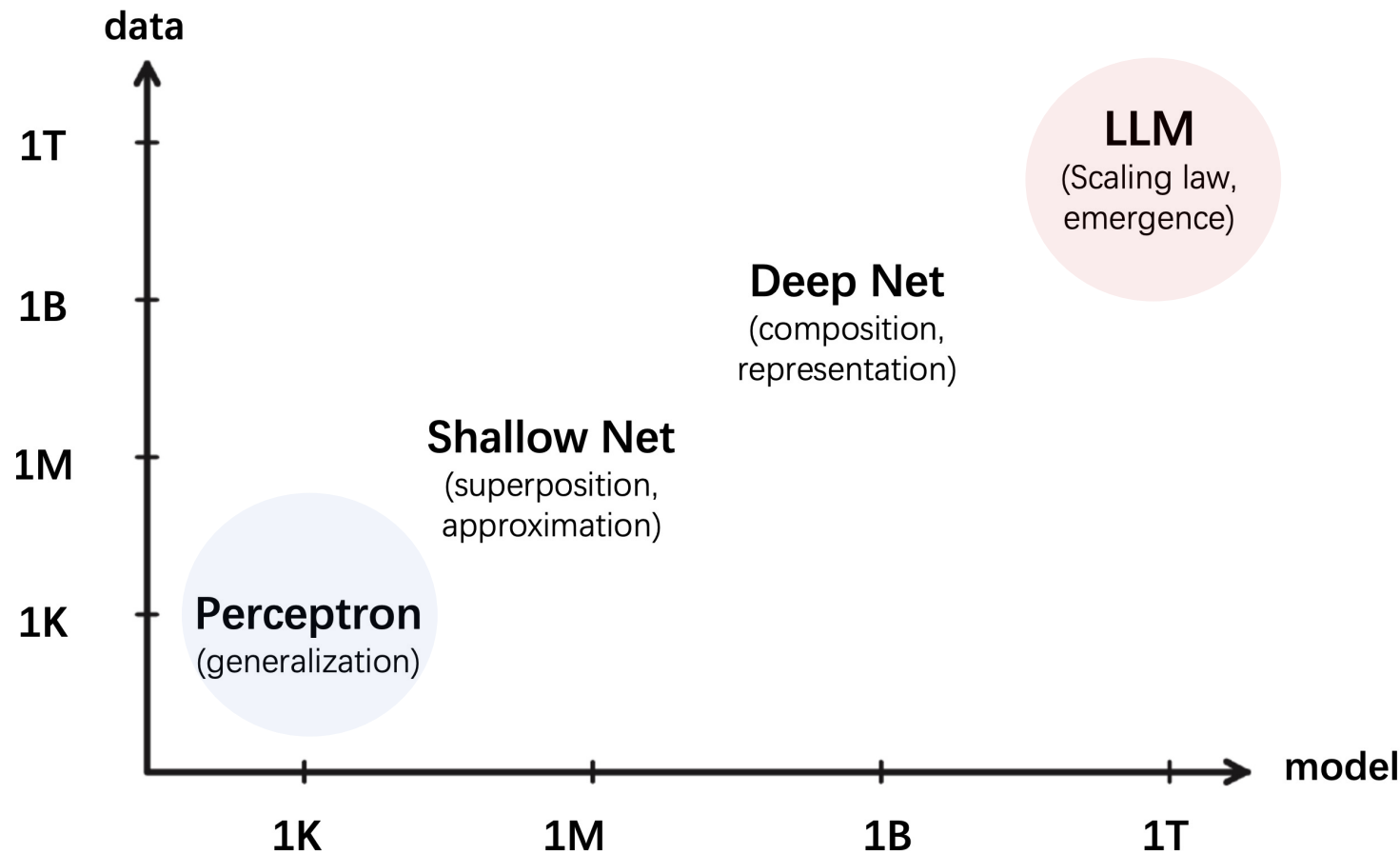
Physics is organized by scales



Taken from Weinan's slide: <https://web.math.princeton.edu/~weinan/ICIAM.pdf>

- Different scales, different **effective theories**.
- Macroscale and microscale are often simple
- The mesoscale is often where complexity lives.

Deep learning can be also organized by scales

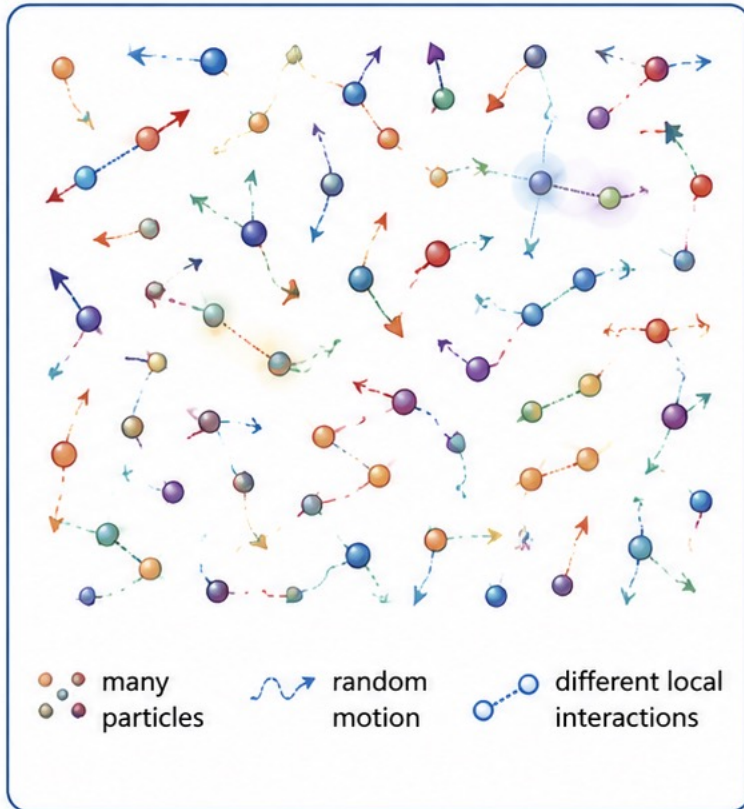


- No single-scale DL theory
- Different scales require different effective theories.
- **LLMs define a macroscale of deep learning**

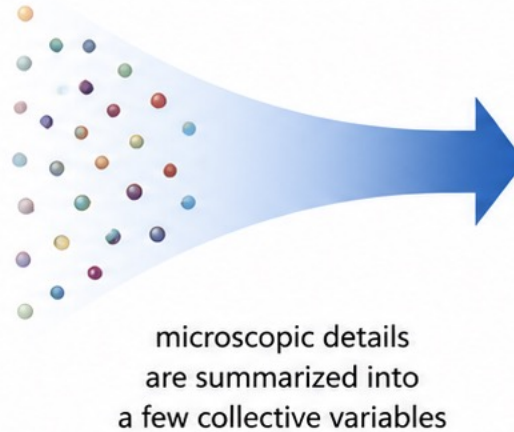
What is the effective theory of the LLM pretraining dynamics?

Microscopic diversity, macroscopic determinism

Microscopic world



averaging /
coarse-graining /
emergence

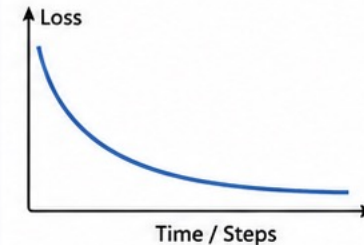


Macroscopic world

Smooth macroscopic fields



Smooth trajectories



Macroscopic observables

- Temperature $T(t)$
- Pressure $P(t)$
- Density $\rho(t)$

Simple macroscopic law (effective dynamics)

$$\frac{d\mathbf{m}}{dt} = \mathcal{F}(\mathbf{m}, t) + \xi(t)$$

few variables
predictable evolution



Complex microscopic details can give rise to **simple macroscopic laws**.

Macroscopic laws can be simple and universal

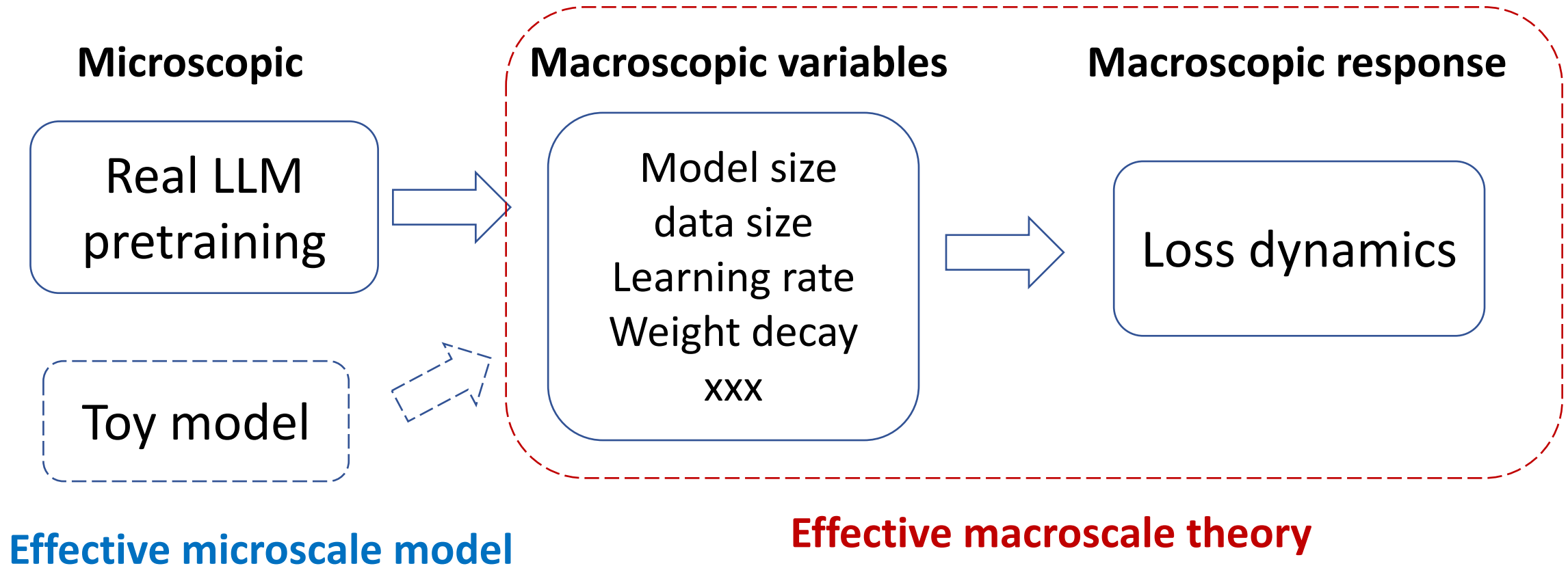
$$F_{\text{friction}} = \mu N$$

μ : friction coefficient
 N : normal force

- The same macroscopic relation holds across materials
- Microscopic details only change the friction coefficients

Toy models and LLM pretraining are **different in microscale**,
but share the **same macroscopic** FSL form.

Toward an effective theory of LLM pretraining



Summary

- LLM pretraining defines a macroscale of deep learning
- Its training dynamics admits a **macroscopic effective law**:

$$\mathcal{E}(t) \asymp \underbrace{S(t)}_{\text{signal learning}} + \underbrace{\int_0^t K(t-r) \eta(r) dr}_{\text{noise forgetting}},$$

- Different optimizers and architectures induce different signal-learning and noise-forgetting kernels.
- The kernels change, but the **functional form is universal**.

One More Thing:
The Mesoscale Is Hard

Classical DL theory lives in the mesoscale

