Overtrained Language Models Are Harder to Fine-Tune

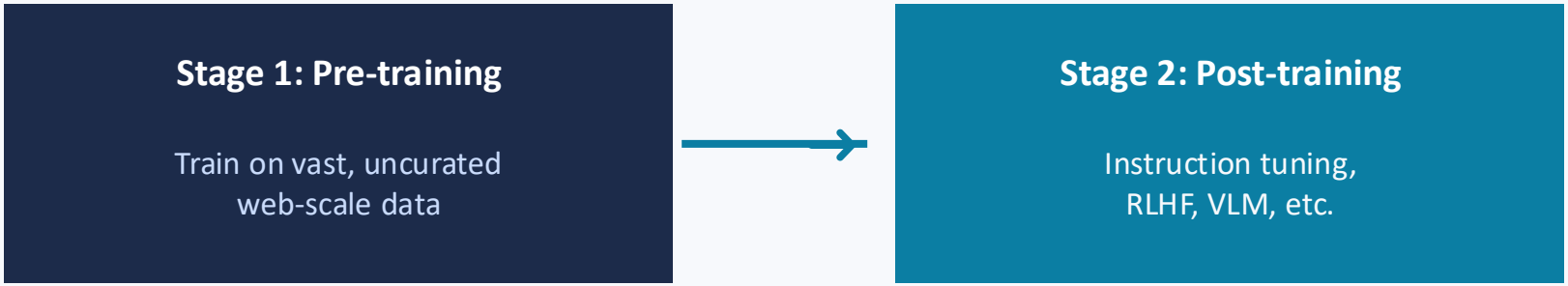
Springer · Goyal · Wen · Kumar · Yue · Malladi · Neubig · Raghunathan

Presenter : Lin Che

Mentor: Andreas Plesner

Springer, J. M., Goyal, S., Wen, K., Kumar, T., Yue, X., Malladi, S., ... & Raghunathan, A. (2025). Overtrained language models are harder to fine-tune. ICML 2025

The Standard LLM Training Pipeline



🧠 Prevailing assumption: "More pre-training tokens → Better downstream model"

Stage 1: LLM Pre-training

C4 web data: a massive, cleaned, and open-source text dataset developed by Google (introduced with the T5 model). It is created by scraping and heavily filtering the public web via Common Crawl.

Stage 1: Pre-traning - Benchmark

ARC: Science QA and reasoning.

HellaSwag: Commonsense text completion.

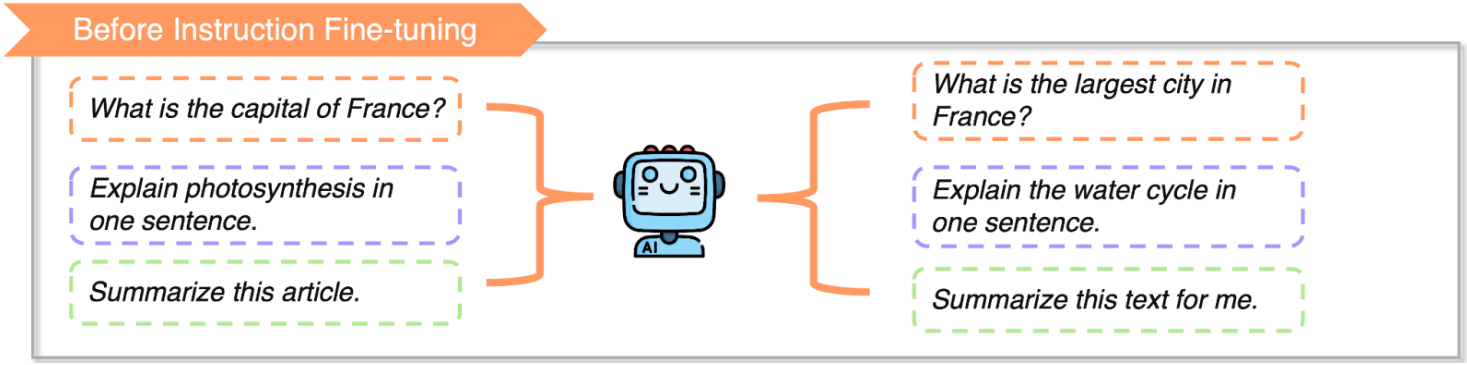
PIQA: Physical commonsense reasoning.

Winogrande: Commonsense pronoun resolution.

SIQA: Social commonsense reasoning.

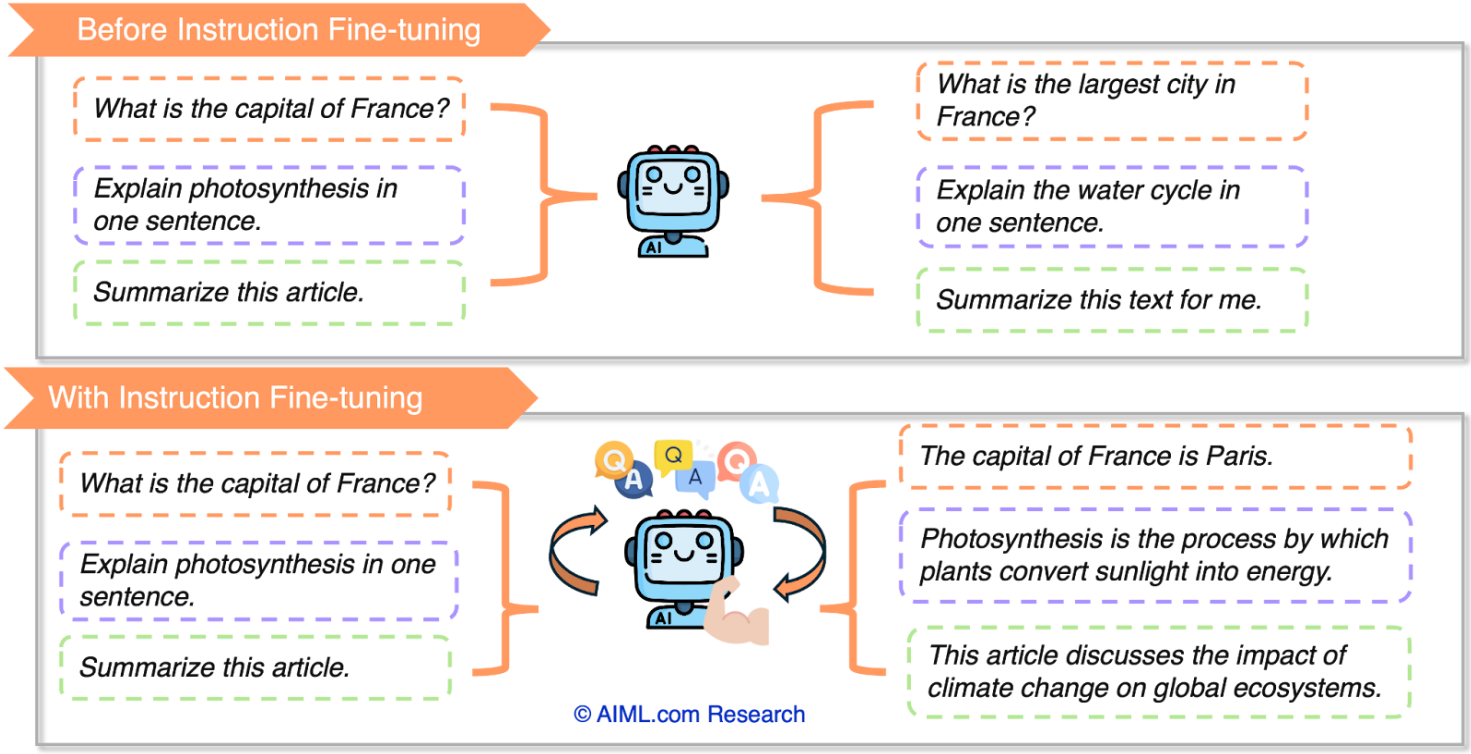
Stage 2: Post-training / Fine-tuning – 1. Instruction tuning

- Anthropic-Helpful-Harmless dataset
- TULU dataset



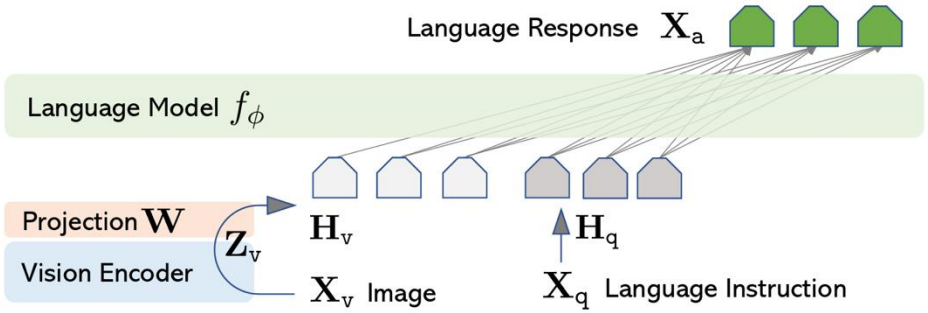
Stage 2: Post-training / Fine-tuning – 1. Instruction tuning

- Anthropic-Helpful-Harmless dataset
- TULU dataset



Stage 2: Post-training / Fine-tuning – 2. Multimodal fine-tuning

LLaVA/Visual Instruction tuning



Liu, H., Li, C., Wu, Q., & Lee, Y. J. (2023). Visual instruction tuning. *Advances in neural information processing systems*, *36*, 34892-34916.

Stage 2: Post-training / Fine-tuning - Evaluation

In Distribution Evaluation

1. Instruction Fine-tuning

- AlpacaEval

2. VLM fine-tuning

- VLM Score from five vision-language benchmark (Average)
  (MME / GQA / AI2D / POPE / TextVQA)

Out of Distribution Evaluation

- generalist benchmarks (ARC, PIQA, HellaSwag, Winogrande, BoolQ, SCIQ)

ID: tuning learning rate for the best performance model

OOD: Evaluate if fine-tuning degrade the general ability

Stage 2: Post-training / Fine-tuning - Evaluation

In Distribution Evaluation – Instruction fine-tuning

- AlpacaEval

Out of Distribution Evaluation

- generalist benchmarks

(ARC, PIQA, HellaSwag, Winogrande, BoolQ, SCIQ)

In Distribution Evaluation – VLM fine-tuning

VLM Score from five vision-language benchmark (Average)

- MME / GQA / AI2D / POPE / TextVQA

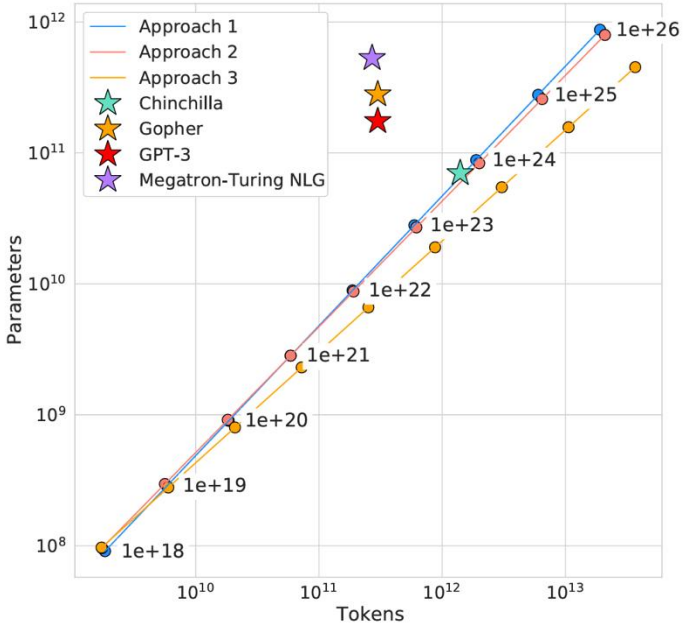
Motivation - Scaling Law

For compute-optimal training, model size and training tokens should scale approximately equally.

$$N_{\text{opt}} \propto C^{0.5}, \quad D_{\text{opt}} \propto C^{0.5}$$

C: the training compute budget, i.e., the number of training FLOPs.
N_{opt}: the compute-optimal number of model parameters.
D_{opt}: the compute-optimal number of training tokens.
$\propto$: means "is proportional to," not "is equal to."

Chinchilla: approximately 20 training tokens per parameter. $D \approx 20N$

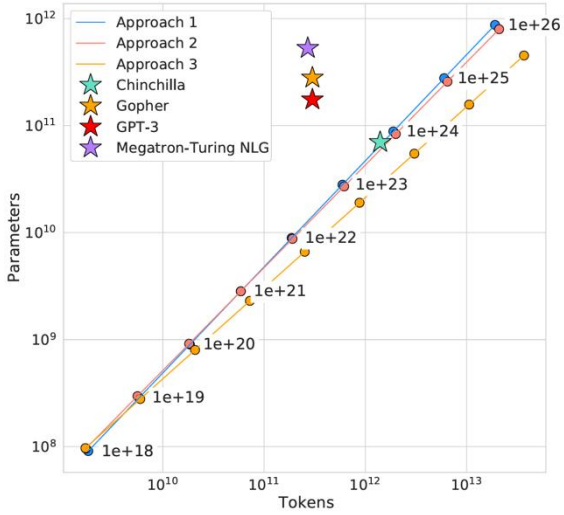


Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. Neurips 2022

Motivation - Scaling Law

$$N_{\text{opt}} \propto C^{0.5}, \quad D_{\text{opt}} \propto C^{0.5}$$

C: the training compute budget, i.e., the number of training FLOPs.
N_{opt}: the compute-optimal number of model parameters.
D_{opt}: the compute-optimal number of training tokens.
$\propto$: means "is proportional to," not "is equal to."



Shows that many large language models at that time are over-sized and under-trained.

LLM: More pre-training tokens → Better base model → Better post-trained model

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., ... & Sifre, L. (2022). Training compute-optimal large language models. Neurips 2022

The Surprising Answer

Does more pre-training always help after fine-tuning?

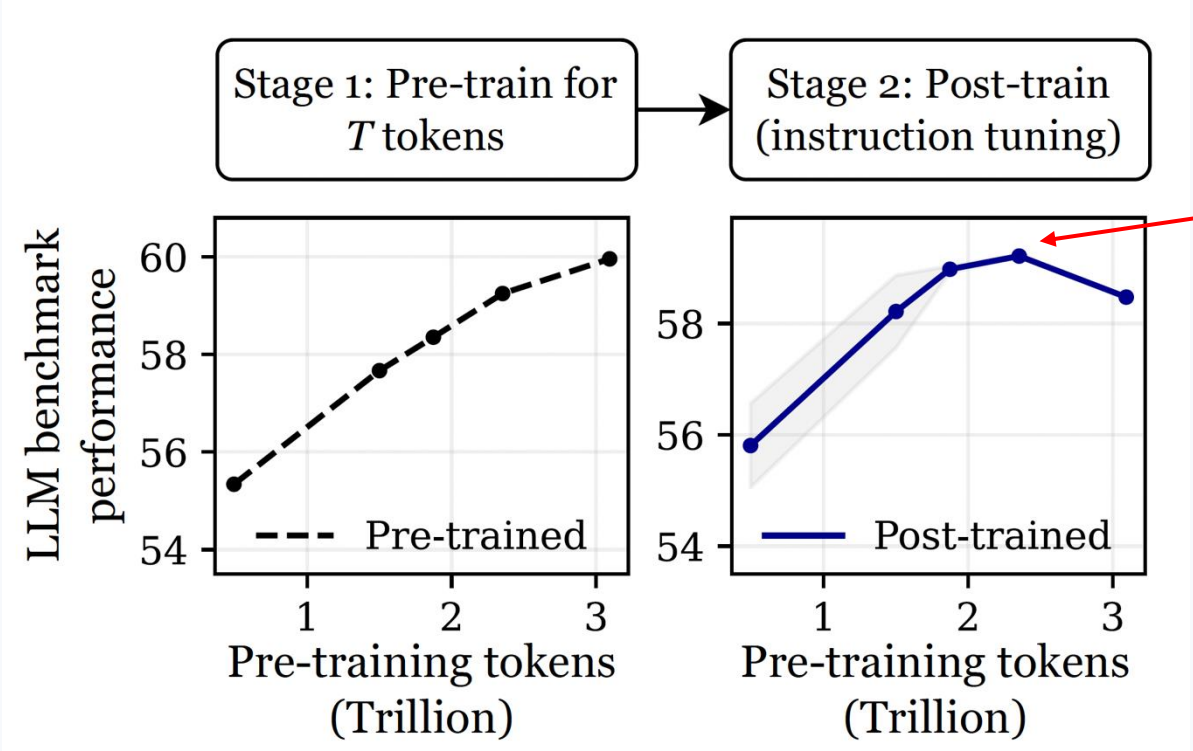
⚠ **No — Extended pre-training can DEGRADE post-training performance**

Evidence: OLMo-1B pre-trained on 3T tokens →

- 3% lower AlpacaEval score vs. 2.3T checkpoint
- 2% lower ARC reasoning benchmark
- Equivalent to training on only 1.5T tokens (50% less data)

→ They call this: Catastrophic Overtraining

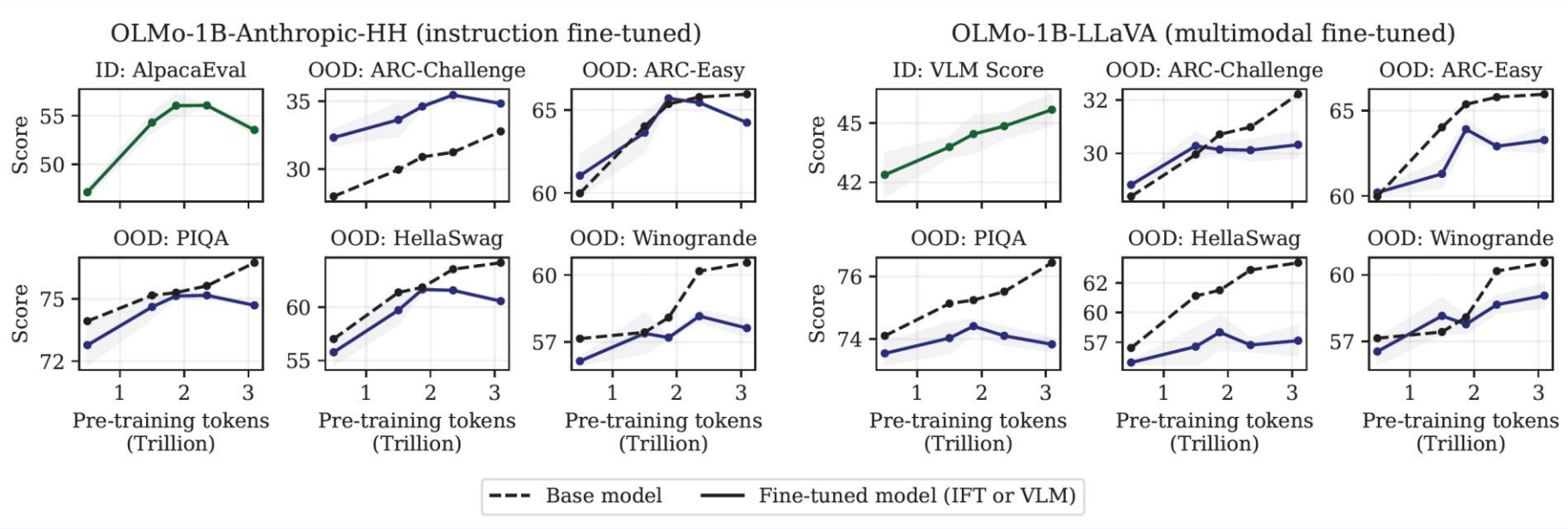
Catastrophic Overtraining — The Core Observation



Inflection Point

Dashed = Base model (always improves) **Solid = Post-trained (degrades after ~2.3T tokens)**

Extended Pre-training Hurts Post-training Performance



In-distribution (ID) Out-of-distribution (OOD) --- Base model

- Extended pre-training always improves base models.
- After instruction tuning, 3T token OLMo-1B worse than 2.3T (ID), Inflection point
- Catastrophic overtraining not necessarily for both ID and OOD at the same time, ID can also drop

Defining Catastrophic Overtraining

📌 **Definition — Catastrophic Overtraining**

Extending pre-training beyond a certain token budget results in a decrease in the model's performance after subsequent modifications.

🔍 **Inflection Point**

Token budget where model’s performance first begins to degrade

📉 **Progressive Sensitivity**

Longer-trained models are increasingly fragile to any parameter modification

🔄 **Scope**

Fine-tuning · RL · Quantization · Pruning · Noise perturbations

Catastrophic Overtraining - Progressive Sensitivity

Progressive Sensitivity:

as pre-training is extended, the model becomes increasingly sensitive to parameter changes.

Catastrophic Overtraining - Progressive Sensitivity

Progressive Sensitivity:

as pre-training is extended, the model becomes increasingly sensitive to parameter changes.

Intuitively:

• early checkpoints are still relatively “rough” and their parameter space may be more flexible;

• later checkpoints have acquired more fine-grained capabilities, but these capabilities may rely on more delicate parameter configurations;

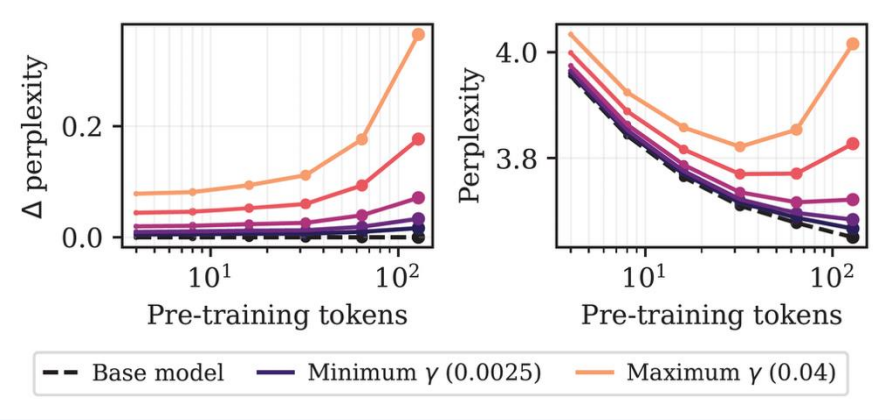
• Therefore, a parameter perturbation of the same magnitude can cause more damage to a later-stage model than to an earlier-stage model.

Progressive Sensitivity

Adding Gaussian noise to base models pre-trained to various token budgets

$\tilde{\theta} = \theta + \epsilon \text{ where } \epsilon \sim \mathcal{N}(\mathbf{0}, \gamma^2 \boldsymbol{\Sigma})$ OLMo-30M Model

For the same perturbation magnitude, longer-trained models degrade MORE



Gaussian noise perturbation

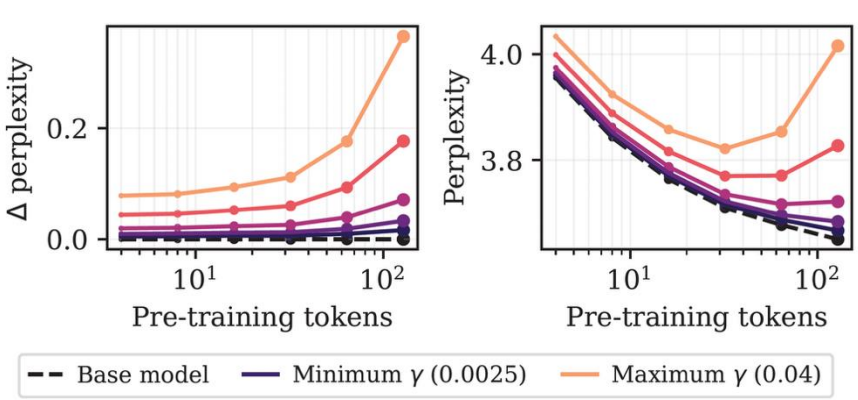
- *No fine-tune, add Gaussian noise on pretrained model*
- *Δ Perplexity monotonically increasing*
- Catastrophic overtraining : U shape

Progressive Sensitivity

Adding Gaussian noise to base models pre-trained to various token budgets

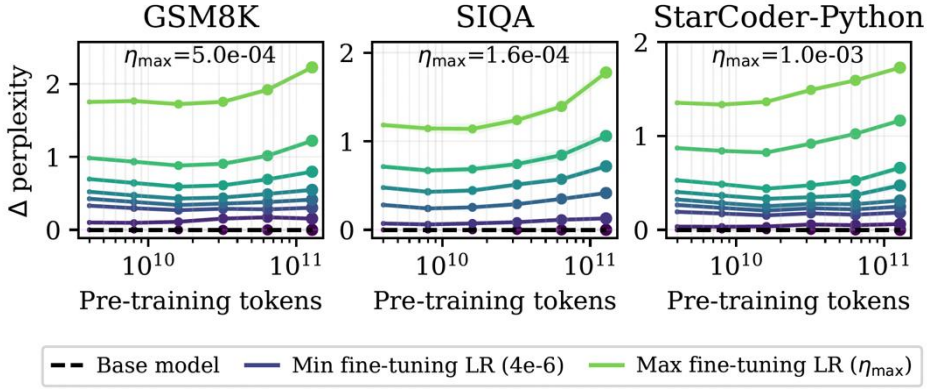
$$\tilde{\theta} = \theta + \epsilon \text{ where } \epsilon \sim \mathcal{N}(\mathbf{0}, \gamma^2 \boldsymbol{\Sigma})$$ OLMo-30M Model

For the same perturbation magnitude, longer-trained models degrade MORE



Gaussian noise perturbation

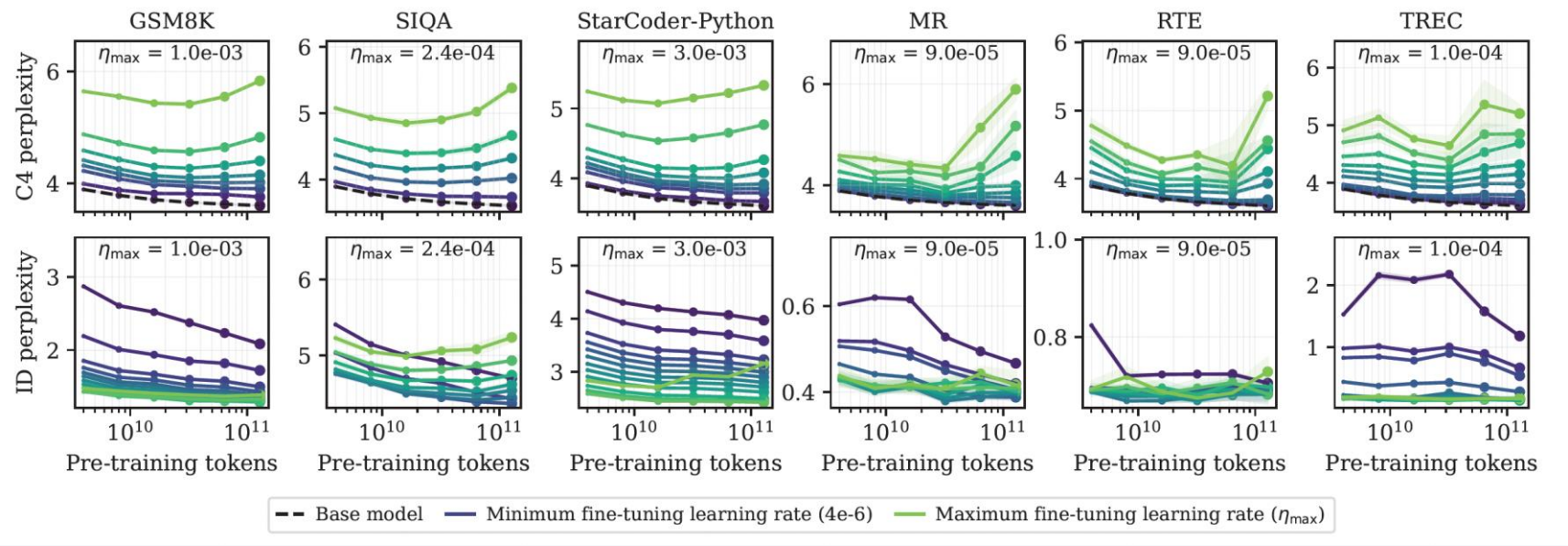
- *No fine-tune, add Gaussian noise on pretrained model*
- *Δ Perplexity monotonically increasing*
- Catastrophic overtraining : U shape



Fine-tuning (fixed learning rate)

- Math (GSM8K), Code (Starcoder-Python), QA(SIQA)
- LR range from 4e-6 to dataset-specific Maximum
- Learning rate ↑, degradation ↑ , inflection point earlier

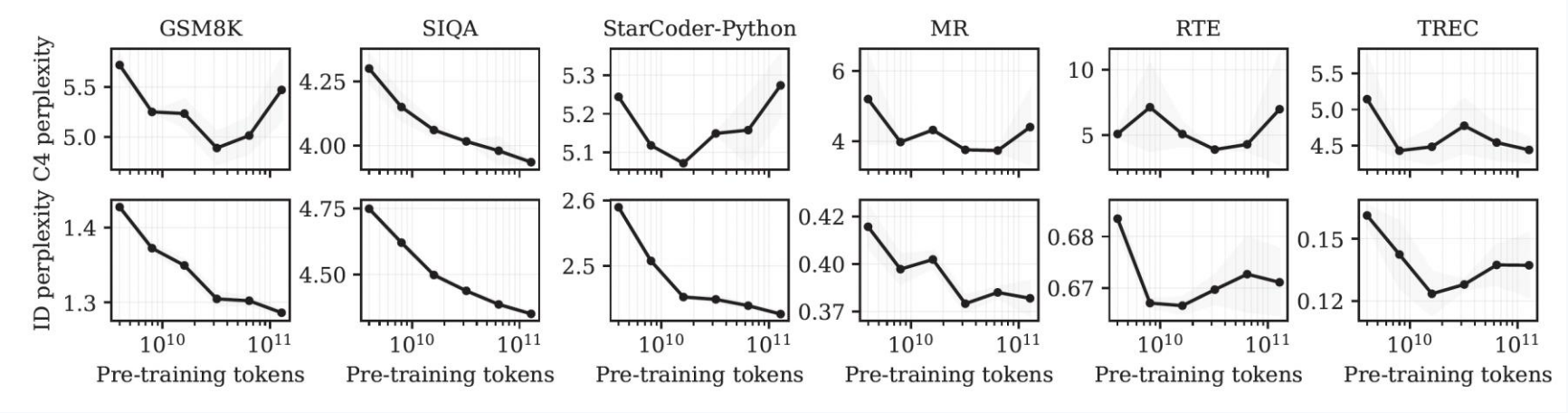
The U-Shaped Curve of Catastrophic Overtraining



Top: C4 perplexity (OOD) **Bottom:** Task perplexity (ID) **U-shape = Catastrophic Overtraining**

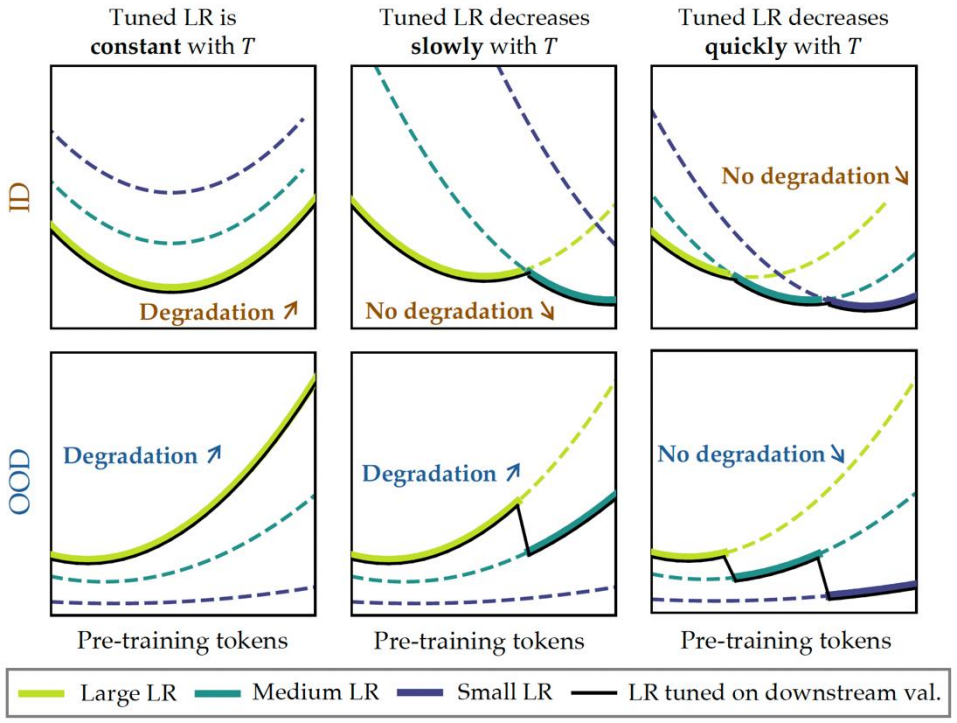
- Fixed learning rate
- C4 Perplexity (OOD) always in U shape -> general ability
- Sometimes ID also shows U shape -> fine-tuning task

Can Tuning the Learning Rate Save Us? – Schematic Illustration



Even though we pick the best LR of the checkpoint, it still sometimes happens even for some ID tasks.

Can Tuning the Learning Rate Save Us? – Schematic Illustration



① Constant optimal LR

ID + OOD both degrade ✗

② Slowly decreasing LR

OOD degrades, ID may improve ⚠

③ Quickly decreasing LR

Both improve ✓ (but limits adaptation)

Key insight: Tuning can delay but not always prevent catastrophic overtraining — and smaller LR **limits adaptation capacity**

Theoretical Analysis: Linear Transfer Learning

Setup: Two-layer linear network $\theta = W_1 W_2$ (Simplified)

Pre-training: $\mathbf{y} = \mathbf{A}^{pre}\mathbf{x}$ (features learned incrementally)
Fine-tuning: $\mathbf{y} = \mathbf{A}^{ft}\mathbf{x}$ (shared singular vectors, different spectrum)

Theorem

There exists a sequence of timesteps t1 < . . . < ti < . . . td such that at timestep ti

$\mathbf{A}^{pre}$ -> SVD

$$\theta(t_i) \approx \boldsymbol{U}\boldsymbol{\Sigma}^{\mathrm{pre}}_{:i}\boldsymbol{V}^T. \qquad \Sigma^{pre}_{:i} = \mathrm{diag}(\sigma^{pre}_1, \ldots, \sigma^{pre}_i, 0, \ldots, 0)$$

Model learning features gradually at the order of strength

Pretraining time ↑ -> features ↑ -> sensitive/fragile features ↑

Saxe, A. M., McClelland, J. L., and Ganguli, S. A mathematical theory of semantic development in deep neural networks. CoRR, abs/1810.10531, 2018. URL http://arxiv.org/abs/1810.10531

Theoretical Analysis: Inflection Point Definition

Setup: Two-layer linear network $\theta = W_1 W_2$ (Simplified)

Pre-training: $\mathbf{y} = \mathbf{A}^{pre}\mathbf{x}$ (features learned incrementally)
Fine-tuning: $\mathbf{y} = \mathbf{A}^{ft}\mathbf{x}$ (shared singular vectors, different spectrum)

Inflection Point – Catastrophic Overtraining

The inflection point with respect to the pre-training loss is defined as the smallest r such that

$$\mathcal{L}_{\text{pre}}(\mathcal{A}(\boldsymbol{\theta}(t_r))) < \mathcal{L}_{\text{pre}}(\mathcal{A}(\boldsymbol{\theta}(t_{r+1}))).$$

Study two different post-training Modification :

- Gaussian parameter perturbations
- Fine-tuning on a canonical family of tasks.

1. Gaussian parameter perturbations

$$Z_1, Z_2 \sim \mathcal{N}(0, \gamma^2 \mathrm{I}_{d^2 \times d^2})$$

$$\theta(t_n) = W_1(t_n) W_2(t_n) \quad \Longrightarrow \quad \tilde{\theta}(t_n) = (W_1(t_n) + Z_1)(W_2(t_n) + Z_2)$$

1. Gaussian parameter perturbations

$$Z_1, Z_2 \sim \mathcal{N}(0, \gamma^2 \mathrm{I}_{d^2 \times d^2})$$

$$\theta(t_n) = W_1(t_n)W_2(t_n) \Longrightarrow \tilde{\theta}(t_n) = (W_1(t_n) + Z_1)(W_2(t_n) + Z_2)$$

Proposition – Inflection Point

$$\widetilde{\mathcal{L}}_{\mathrm{pre}}(t_n) - \widetilde{\mathcal{L}}_{\mathrm{pre}}(t_{n-1}) \geq (2d\gamma^2 - \sigma_n^{\mathrm{pre}})\sigma_n^{\mathrm{pre}}.$$

$$2d\gamma^2\sigma_n^{pre} - (\sigma_n^{pre})^2$$

gaussian noise damage on feature n

benefit of learning feature n

Theorem - Inflection Point

For some γ > 0, there exists an inflection point r ∈ [1, d) such that e Lpre(n) increases monotonically for n ≥ r.

2. Fine-tuning

Mapping: $y = A^{ft}x$

$$A^{pre} = U\Sigma^{pre}V^T$$

$$A^{ft} = U\Sigma^{ft}V^T$$

Transfer Learning: Share U, V but different Σ

pre-training task and fine-tuning task share the same feature directions, but with different importance

2. Fine-tuning

Mapping: $y = A^{ft}x$

$$A^{pre} = U\Sigma^{pre}V^T$$

$$A^{ft} = U\Sigma^{ft}V^T$$

Transfer Learning: Share U, V but different Σ

pre-training task and fine-tuning task share the same feature directions, but with different importance

Fine-tuning loss

Pre-trained for t, fine-tuned for K step

Regulation -> use LR limit parameter movement

$$L_{ft}(\theta(t;k);\lambda) = \|\theta(t;k) - A^{ft}\|_F^2 + \lambda\|\theta(t;k) - \theta(t)\|_F^2$$

Close to fine-tuning task

Not too far away from pre-trained Initialization

2. Fine-tuning – Progressive Sensitivity Definition

$$\Delta_{pre}(t_n) = L_{pre}(\theta(t_n; K)) - L_{pre}(\theta(t_n; 0))$$

$\theta(t_n; 0)$: model before fine-tuning
$\theta(t_n; K)$: model after K step fune-tuning
Lpre ： Loss on pre-training tasks

Δpre(tn): degradation from fine-tuning to pre-training performance

If Δpre(tn) monotonically increasing, then progressive sensitivity exist

2. Fine-tuning – Catastrophic Overtraining

$$L_{pre}(\theta(t_n;K)) - L_{pre}(\theta(t_{n-1};K))$$

$$\approx \underbrace{-(\sigma_n^{pre})^2}_{\text{pre-training improvement}} + \underbrace{(\sigma_n^{ft} - \sigma_n^{pre})^2}_{\text{extra fine-tuning degradation}}$$

With regulation:

$$\left(\frac{\sigma_n^{ft} - \sigma_n^{pre}}{1+\lambda}\right)^2$$

2. Fine-tuning – Catastrophic Overtraining

$$L_{pre}(\theta(t_n;K)) - L_{pre}(\theta(t_{n-1};K))$$

$$\approx \underbrace{-(\sigma_n^{pre})^2}_{\text{pre-training improvement}} + \underbrace{(\sigma_n^{ft} - \sigma_n^{pre})^2}_{\text{extra fine-tuning degradation}}$$

With regulation:

$$\left(\frac{\sigma_n^{ft} - \sigma_n^{pre}}{1+\lambda}\right)^2$$

When fine-tuning damage increase > base model improvement

$$\left(\frac{\sigma_n^{ft} - \sigma_n^{pre}}{1+\lambda}\right)^2 > (\sigma_n^{pre})^2$$

Fine-tuned model will perform worse on pre-training tasks after step n (inflection point)

2. Fine-tuning – Catastrophic Overtraining

$$\left(\frac{\sigma_n^{ft} - \sigma_n^{pre}}{1+\lambda}\right)^2 > (\sigma_n^{pre})^2$$

Catastrophic overtraining is inevitable without regularization.

Let λ = 0. There exists an α0 > 0. such that if Apre and Aft are (α, r)-misaligned, for α > α0, then the pre-training loss after fine-tuning Lpre(θ(tn;K)) monotonically increases for n ≥ r.

Regularization can delay the degradation of pretraining performance at the cost of downstream performance.

Relationship to Prior Work

Prior Work	Previous Focus	This Paper's Contribution
Loss of Plasticity	RL / small-scale synthetic settings	Web-scale LLM pre-training; generic data hurts targeted fine-tuning
Catastrophic Forgetting	Sequential task training	Overtraining amplifies severity of forgetting after fine-tuning
Scaling Laws	Compute-optimal pre-training loss	Fine-tuning-optimal ≠ pre-training-optimal; new metric needed
Progressive Sharpening	Hessian spectral norm grows with training	Close related. Necessary but insufficient One on local property of loss function One on global degradation of model performance

Practical Implications

Inference-time scaling trends will exacerbate this problem

Current: scale pre-training then fine-tune — may be suboptimal

Efficient small models will be extra fragile

Questions

1. ARC (AI2 Reasoning Challenge)

•**Summary:** A benchmark consisting of grade-school science questions that require logical reasoning rather than simple keyword matching.

•**Example:** * *Question:* Which property of a substance can be measured using a graduated cylinder?

- *Options:* (A) Air pressure (B) Volume (C) Temperature (D) Weight
- *Answer:* (B)

2. HellaSwag

•**Summary:** It evaluates a model's ability to predict the most plausible continuation of a given everyday scenario.

•**Example:** * *Scenario:* A woman is sitting at a piano. She opens the lid and places her fingers on the keys.

- *Continuation:* (A) She starts to play a classical melody. (B) She begins to cook dinner in the kitchen. (C) The piano turns into a car and drives away.
- *Answer:* (A)

3. PIQA (Physical Interaction QA)

•**Summary:** It focuses on testing the model’s understanding of physical commonsense and how objects interact in the real world.

•**Example:** * *Question:* How do you keep a bouquet of flowers fresh for a longer time?

- *Options:* (A) Put them in a vase with water. (B) Put them in a dry drawer.
- *Answer:* (A)

4. Winogrande

•**Summary:** A benchmark designed to test pronoun resolution by requiring the model to identify which object a pronoun refers to based oncontext.

•**Example:** * *Sentence:* The trophy doesn't fit into the brown suitcase because **it** is too large.

- *Question:* What does "**it**" refer to?
- *Options:* (A) The trophy (B) The suitcase
- *Answer:* (A) (Note: If the sentence said "too small," the answer would be B).

5. SIQA (Social IQA)

•**Summary:** It assesses the model's understanding of social interactions and the emotional intentions behind human actions.

•**Example:** * *Scenario:* Tracy accidentally broke her friend's favorite mug and immediately offered to buy a new one.

- *Question:* Why did Tracy do this?
- *Options:* (A) To make her friend angry. (B) To make up for her mistake.
- *Answer:* (B)

AlpacaEval is an automatic evaluation benchmark that uses a powerful language model (like GPT-4) as a judge to compare and rank how helpful different LLMs are in open-ended conversations. / win-rate

VLM Score: In VLM (Vision-Language Model) evaluation, VLM Score measures how well the generated text description matches the input image.

MME: A comprehensive benchmark that evaluates vision-language models across both perception (e.g., object recognition, attribute identification) and cognition (e.g., relation understanding, reasoning) abilities.
GQA: A visual question answering dataset focused on real-world visual reasoning, where models must understand compositional questions about scene graphs and structures.
AI2D: A diagram-based question answering benchmark designed to test a model's ability to interpret and reason over scientific and educational diagrams.
POPE: A evaluation framework specifically designed to measure object hallucination in vision-language models by checking if they mistakenly identify non-existent objects.
TextVQA: A benchmark that requires models to look at images, read and recognize the embedded text (OCR), and reason over that text to answer questions.

BoolQ:
BoolQ is a yes/no question answering benchmark that evaluates a model's ability to understand natural questions and reason over short passages.

SCIQ:
SCIQ is a science question answering benchmark that tests a model's ability to answer multiple-choice science questions using commonsense and domain knowledge.