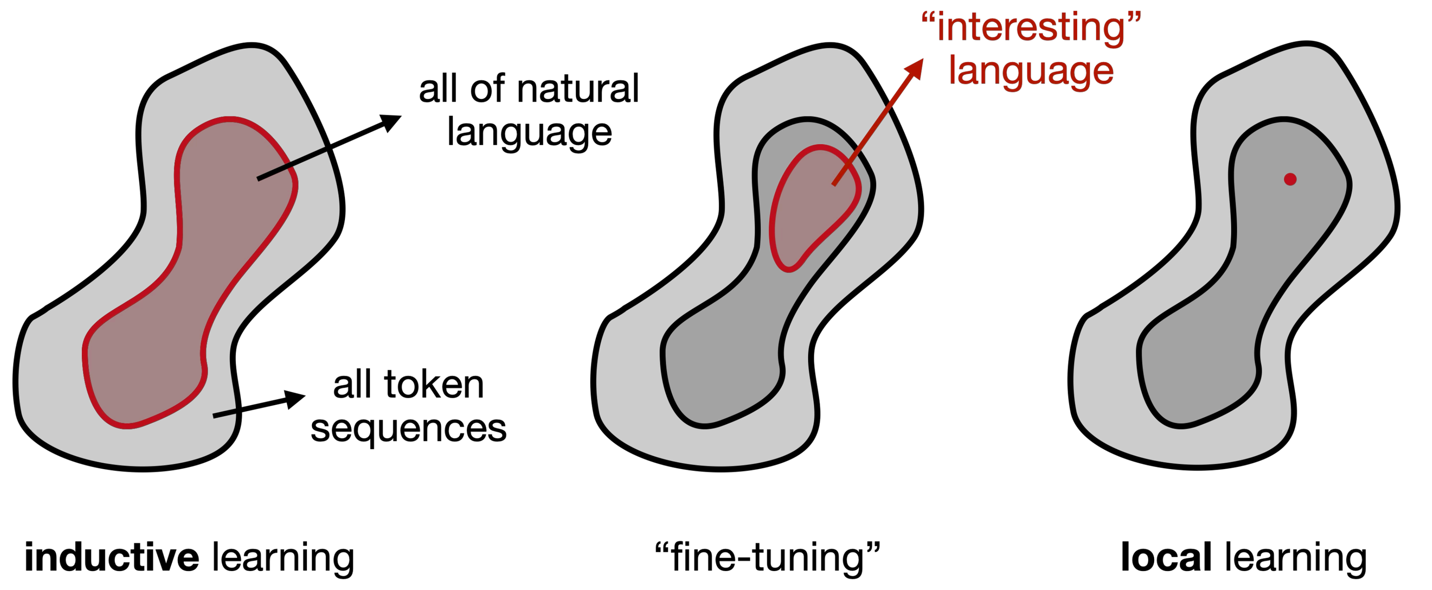
Efficiently Learning at Test-Time: Active Fine-Tuning of LLMs

Jonas Hübotter, Sascha Bongni, Ido Hakimi, Andreas Krause

ETH zürich

Motivation

- **Goal:** Learn a specific model, tailored to each prompt
- Requires automatic data selection (like with RAG)



How can we select data that effectively reduces uncertainty about the response to the prompt?

Insufficiency of Nearest Neighbor Retrieval

Nearest Neighbor selects redundant data!

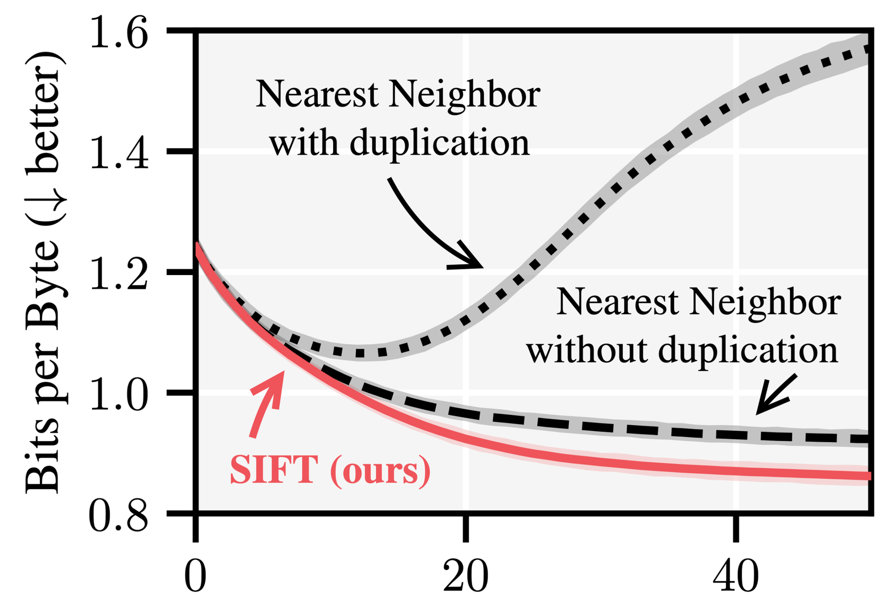
Prompt: What is the age of Michael Jordan and how many kids does he have?

Nearest Neighbor:

1. The age of Michael Jordan is 61 years.
2. Michael Jordan was born on February 17, 1963.

SIFT (ours):

1. The age of Michael Jordan is 61 years.
2. Michael Jordan has five children.



SIFT: Selecting Informative Data for Fine-Tuning

Idea: Select data that *maximally* reduces "uncertainty" about how to respond to the prompt

1st step: Estimate uncertainty

- *Surrogate model:* logit-linear model $s(f^\star(x))$ with $f^\star(x) = \boldsymbol{W}^\star \phi(x)$ [$\boldsymbol{W}^\star$ unknown, $\phi(\cdot)$ known]:

$$\underbrace{s^\star(x) = s(f^\star(x))}_{\text{"truth"}} \qquad \underbrace{s_n(x) = s(\boldsymbol{W}_n \phi(x))}_{\text{fine-tuned model on } n \text{ data points}}$$

- *Confidence sets:* $\underbrace{d_{\mathrm{TV}}(s_n(x), s^\star(x))}_{\text{error}} \leq \underbrace{\beta_n(\delta)}_{\text{scaling}} \underbrace{\sigma_n(x)}_{\textbf{key obj.}}$

[with probability $1 - \delta$]

$\rightsquigarrow$ $\sigma_n(x)$ measures **uncertainty** about response to x!

2nd step: Minimize "posterior" uncertainty

$$x_{n+1} = \operatorname*{argmin}_x \sigma_{X_n \cup \{x\}}(\underbrace{x^\star}_{\text{prompt}}) \qquad \text{with } k(x, x') = \boldsymbol{\phi}(x)^\top \boldsymbol{\phi}(x')$$

$$= \operatorname*{argmax}_x \underbrace{\begin{bmatrix} k(x^\star, x_1) \\ \vdots \\ k(x^\star, x_n) \\ k(x^\star, x) \end{bmatrix}^\top}_{\text{maximize relevance}} \left(\underbrace{\begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) & k(x_1, x) \\ \vdots & \ddots & \vdots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) & k(x_n, x) \\ k(x, x_1) & \cdots & k(x, x_n) & k(x, x) \end{bmatrix}}_{\text{minimize redundancy}} + \lambda I_{n+1} \right)^{-1} \begin{bmatrix} k(x^\star, x_1) \\ \vdots \\ k(x^\star, x_n) \\ k(x^\star, x) \end{bmatrix}$$

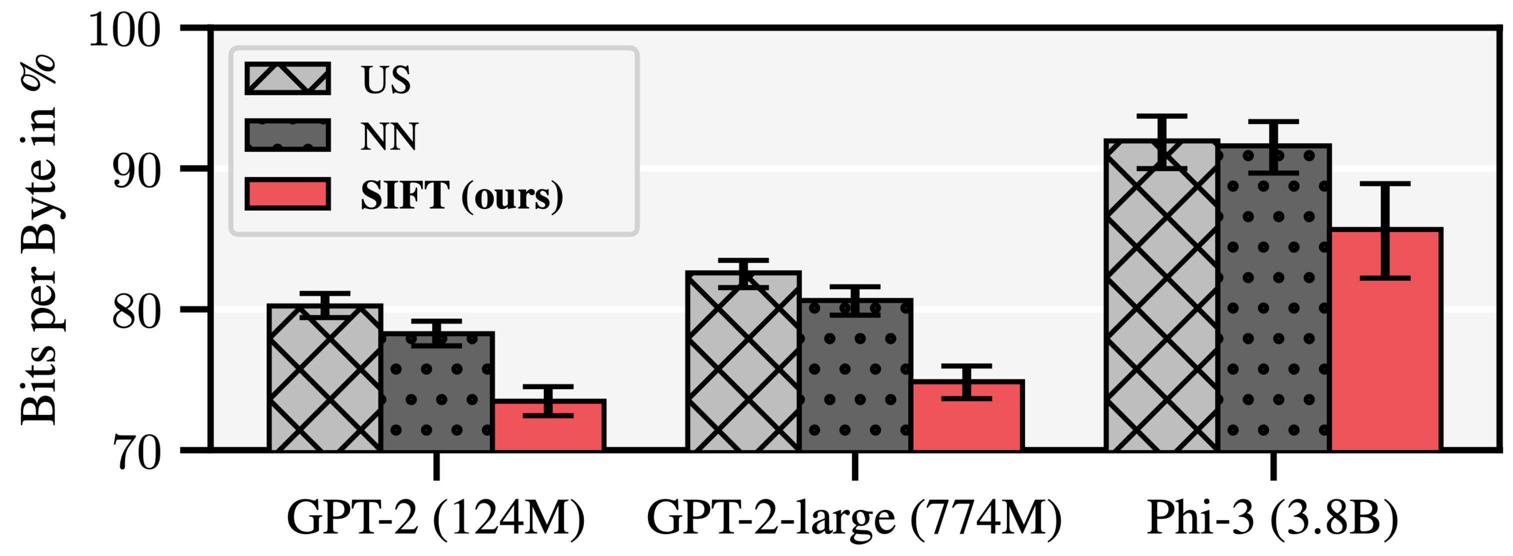
Theory: $\sigma_n^2(x) - \overbrace{\sigma_\infty^2(x)}^{\text{irreducible uncertainty}} \leq \frac{O(\lambda \log(n))}{\sqrt{n}}$

$\rightsquigarrow$ predictions can be only as good as the data and the learned abstractions!

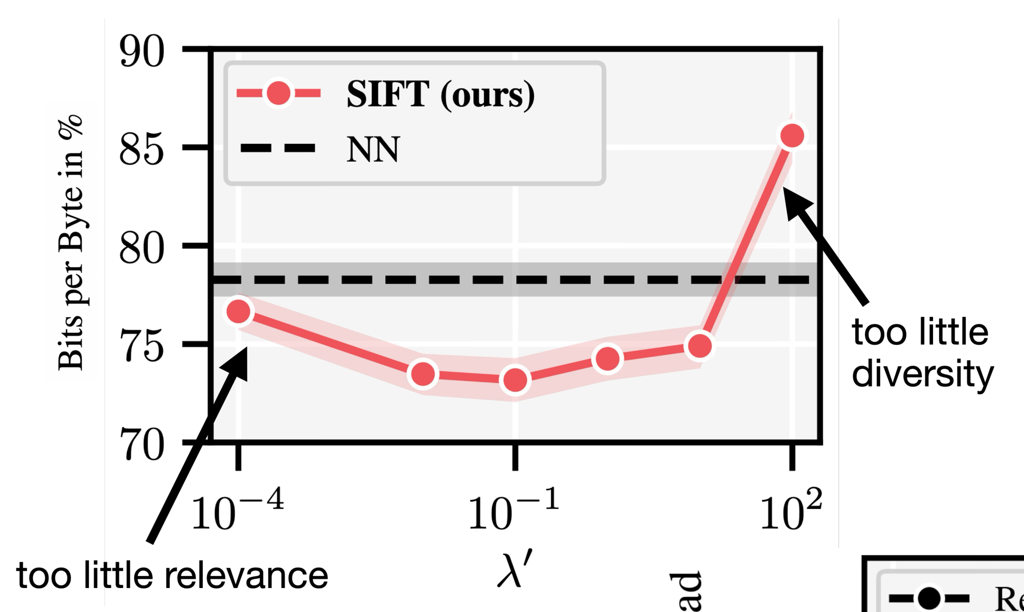
Test-Time Fine-Tuning with SIFT

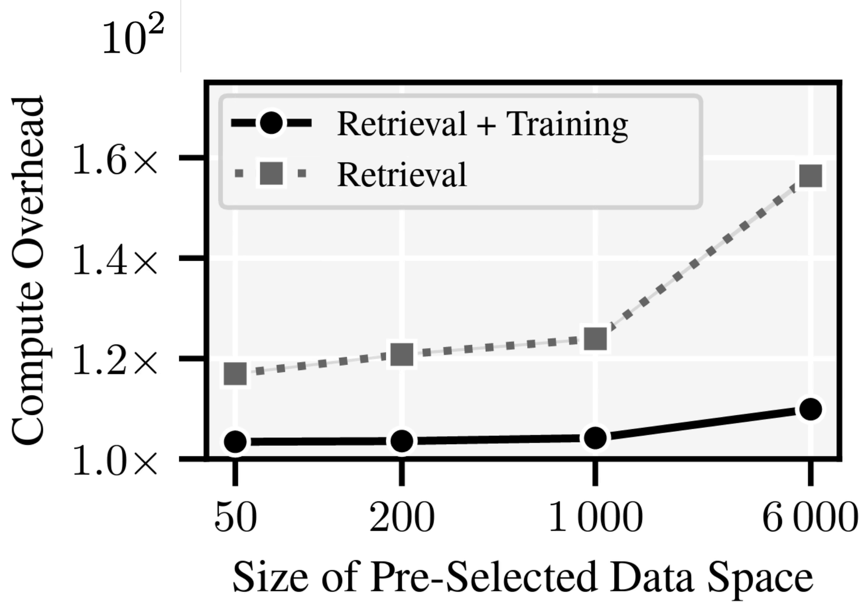
Taking a single gradient step on each selected data point

1. SIFT selects informative data!

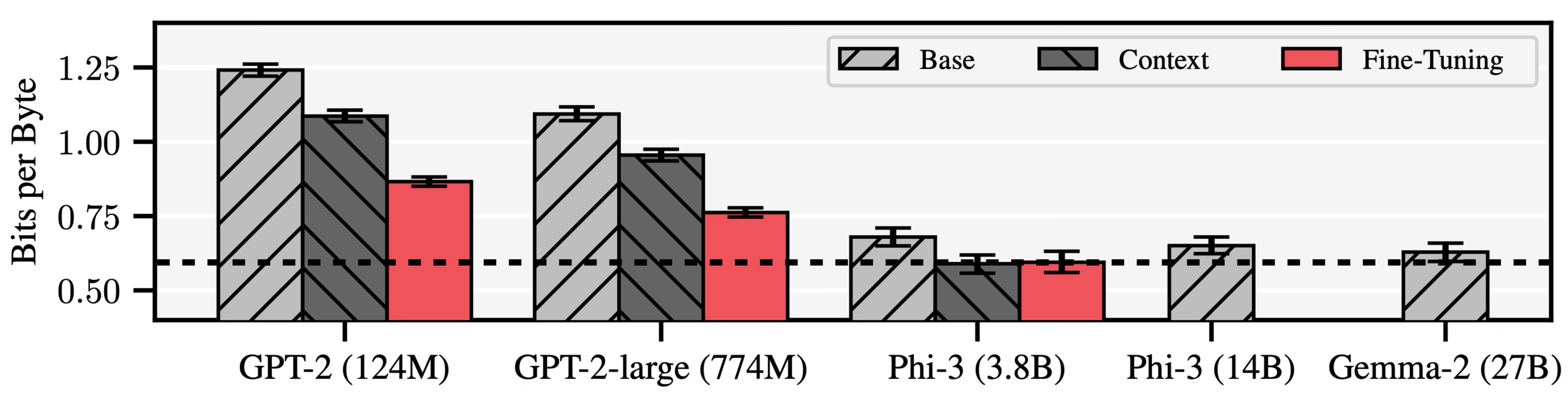


	US	NN	NN-F	SIFT	Δ
NIH Grants	93.1 (1.1)	84.9 (2.1)	91.6 (16.7)	**53.8** (8.9)	↓31.1
US Patents	85.6 (1.5)	80.3 (1.9)	108.8 (6.6)	**62.9** (3.5)	↓17.4
GitHub	45.6 (2.2)	42.1 (2.0)	53.2 (4.0)	**30.0** (2.2)	↓12.1
Enron Emails	**68.6** (9.8)	**64.4** (10.1)	91.6 (20.6)	**53.1** (11.4)	↓11.3
Wikipedia	67.5 (1.9)	**66.3** (2.0)	121.2 (3.5)	**62.7** (2.1)	↓3.6
Common Crawl	92.6 (0.4)	90.4 (0.5)	148.8 (1.5)	**87.5** (0.7)	↓2.9
PubMed Abstr.	88.9 (0.3)	87.2 (0.4)	162.6 (1.3)	**84.4** (0.6)	↓2.8
ArXiv	85.4 (1.2)	**85.0** (1.6)	166.8 (6.4)	**82.5** (1.4)	↓2.5
PubMed Central	**81.7** (2.6)	**81.7** (2.6)	155.6 (5.1)	**79.5** (2.6)	↓2.2
Stack Exchange	78.6 (0.7)	78.2 (0.7)	141.9 (1.5)	**76.7** (0.7)	↓1.5
Hacker News	**80.4** (2.5)	**79.2** (2.8)	133.1 (6.3)	**78.4** (2.8)	↓0.8
FreeLaw	**63.9** (4.1)	**64.1** (4.0)	122.4 (7.1)	**64.0** (4.1)	↑0.1
DeepMind Math	**69.4** (2.1)	**69.6** (2.1)	121.8 (3.1)	**69.7** (2.1)	↑0.3
All	80.2 (0.5)	78.3 (0.5)	133.3 (1.2)	**73.5** (0.6)	↓4.8





2. Test-time fine-tuning reduces next-token prediction error of SOTA models!



	Context	Fine-Tuning	Δ
GitHub	74.6 (2.5)	**28.6** (2.2)	↓56.0
DeepMind Math	100.2 (0.1)	**70.1** (2.1)	↓30.1
US Patents	87.4 (2.5)	**62.2** (3.6)	↓25.2
FreeLaw	87.2 (3.6)	**65.5** (4.2)	↓21.7

GPT-2

	Context	Fine-Tuning	Δ
GitHub	74.6 (2.5)	**31.0** (2.2)	↓43.6
DeepMind Math	100.2 (0.7)	**74.2** (2.3)	↓26.0
US Patents	87.4 (2.5)	**64.7** (3.8)	↓22.7
FreeLaw	87.2 (3.6)	**68.3** (4.2)	↓18.9

GPT-2-large

	Context	Fine-Tuning	Δ
DeepMind Math	100.8	75.3	↓25.5
GitHub	71.3	46.5	↓24.8
FreeLaw	78.2	67.2	↓11.0
ArXiv	101.0	94.3	↓6.4

Phi-3

Key Takeaways

- Test-Time Fine-Tuning is a promising approach to improve LLM performance at test-time
- SIFT selects better data for fine-tuning than Nearest Neighbor retrieval