Efficiently learning at test-time with LLMs via transductive active learning

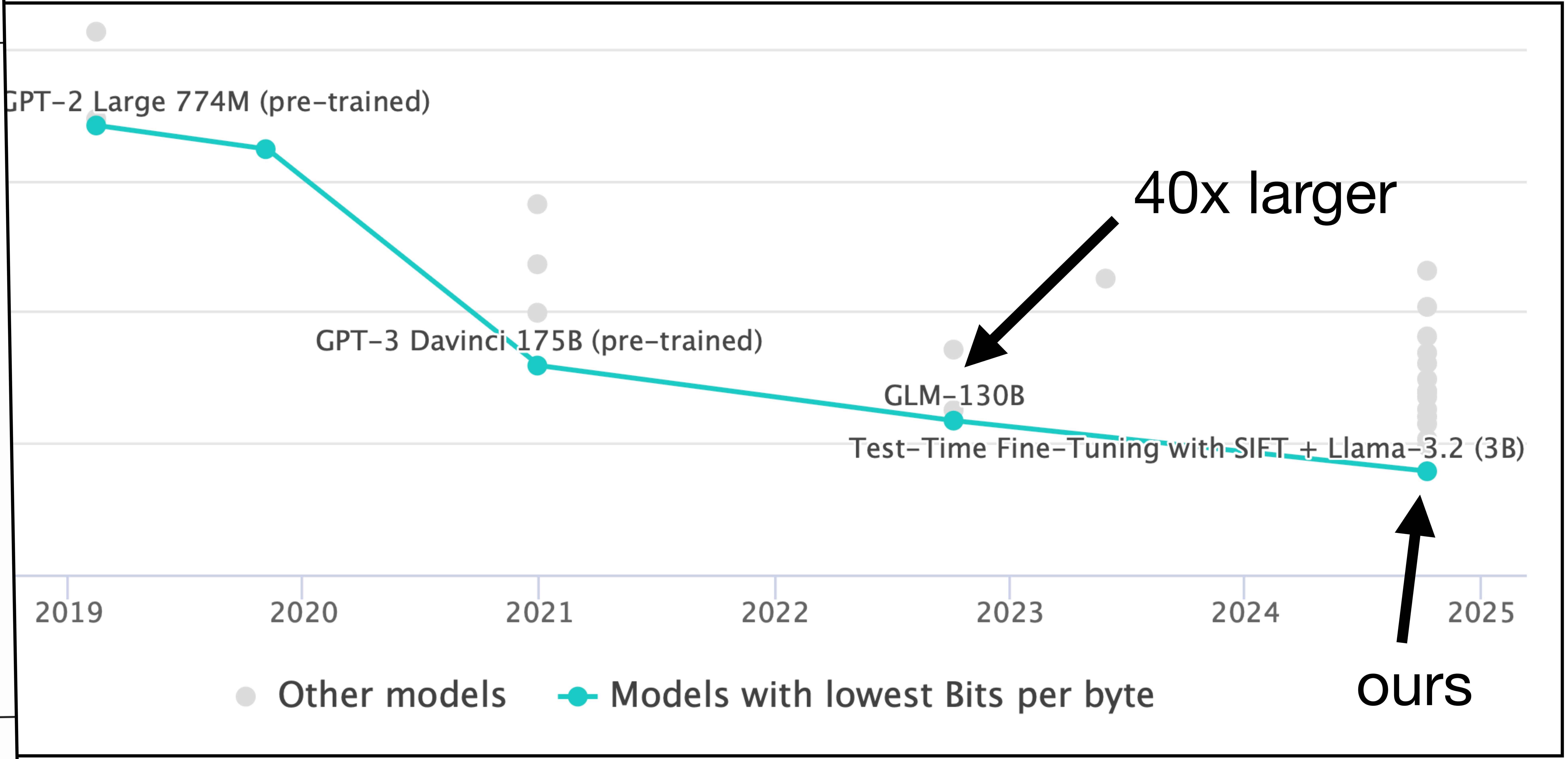
Jonas Hübotter

ETH zürich

Pile benchmark

Pile dataset
NIH Grants
US Patents
GitHub
Enron Emails
Common Crawl
ArXiv
Wikipedia
PubMed Abstr.
Hacker News
Stack Exchange
PubMed Central
DeepMind Math
FreeLaw
All



https://paperswithcode.com/sota/language-modelling-on-the-pile

Train **Test**

Local learning (at test-time)

known!

Test instance $\boldsymbol{x}^\star$

Training data

$\mathcal{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^n$

Learnt model

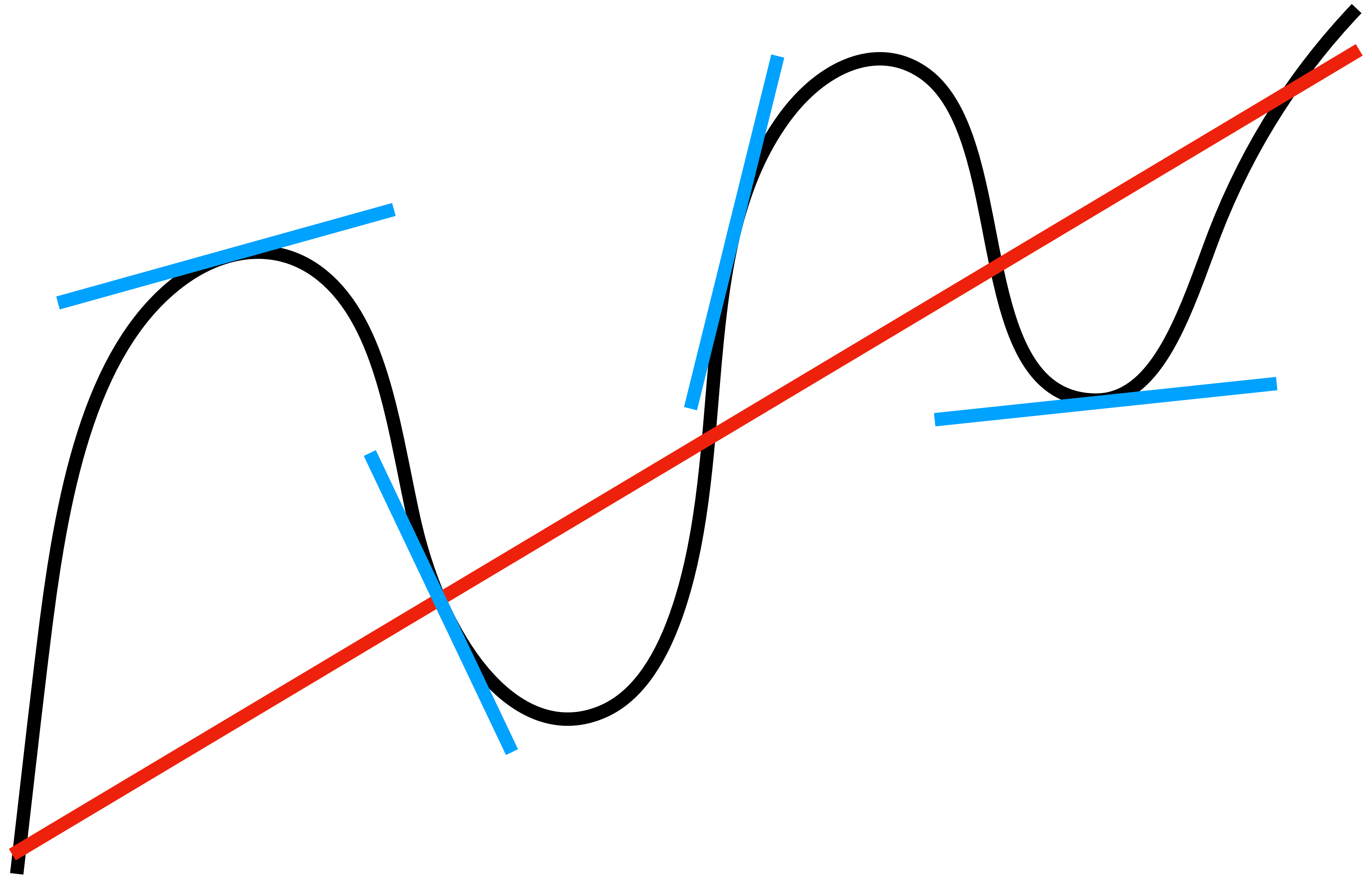
$f : \mathcal{X} \to \mathcal{Y}$

Prediction

$f(\boldsymbol{x}^\star)$

A story of curve fitting

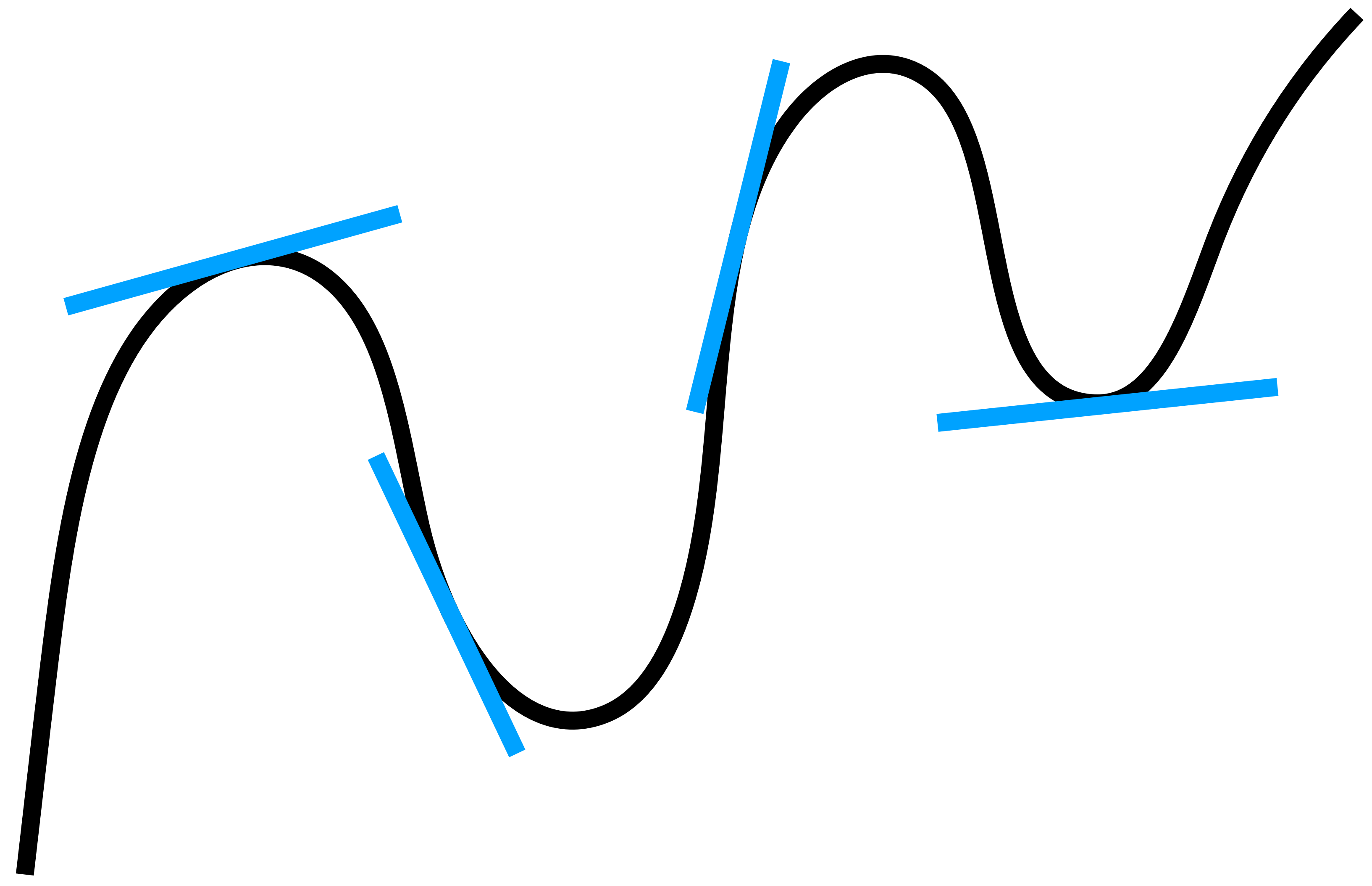
A story of curve fitting



Remedies:

- Parametric models
  polynomial regression
  neural networks
- Non-parametric models
  kernel (ridge) regression
  k-nearest neighbor
- **Local** models
  local linear regression
  ...

A story of curve fitting



Local models have two components:

- *Parametric* “controller”

  linear regression

  …

- *Non-parametric* “memory”

  k-nearest neighbor

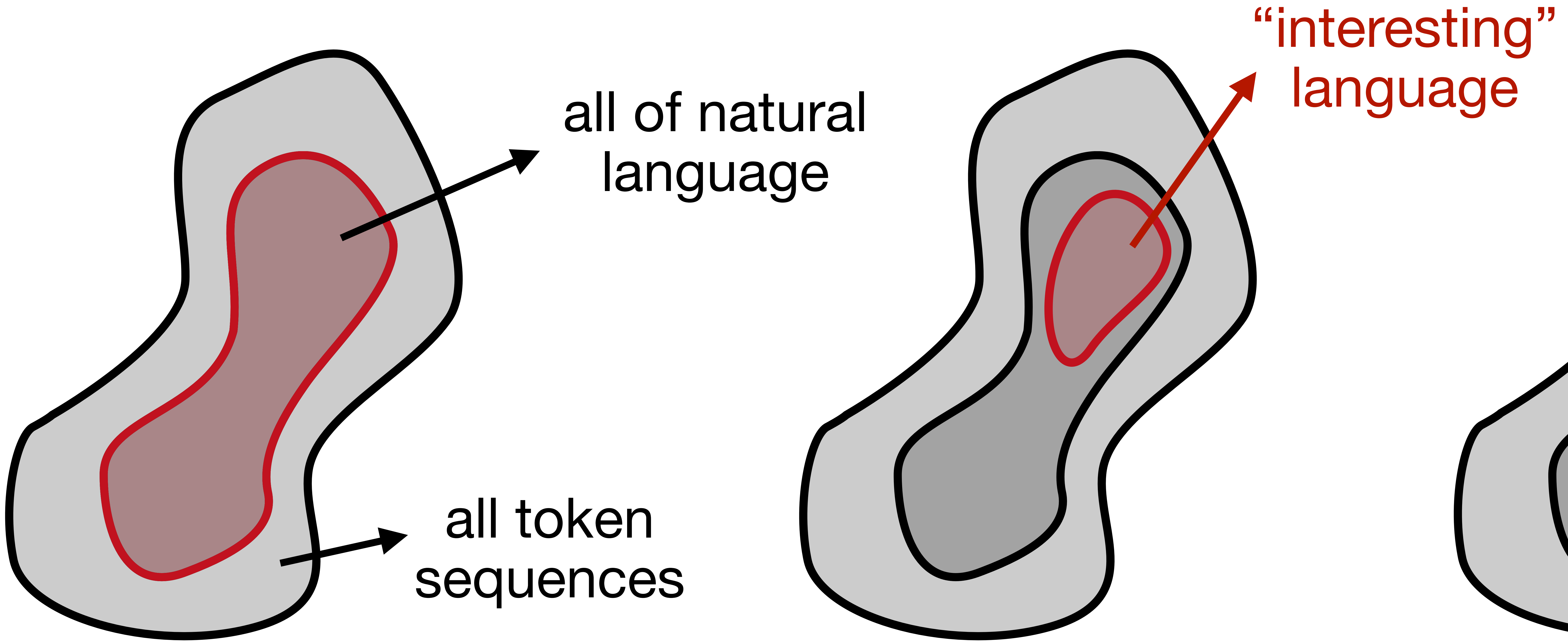
  …

→ a small model class can fit a rich function class!

→ one local model needs only little data!

→ too good to be true?

Local learning in a picture



inductive learning "fine-tuning" **local** learning

History

Fix

Hodges

Cover

Hart

- **since 1950s:** k-nearest neighbors
- **since 1960s:** kernel regression (Nadaraya & Watson)
- **since 1970s:** local (linear) learning (Cleveland & Devlin)
- **since 1980s:** transductive learning (Vapnik)

> "When solving a problem of interest, do not solve a more general problem as an intermediate step. Try to get the answer that you really need but not a more general one."

- **in 1990s:** local fine-tuning (Vapnik & Bottou)

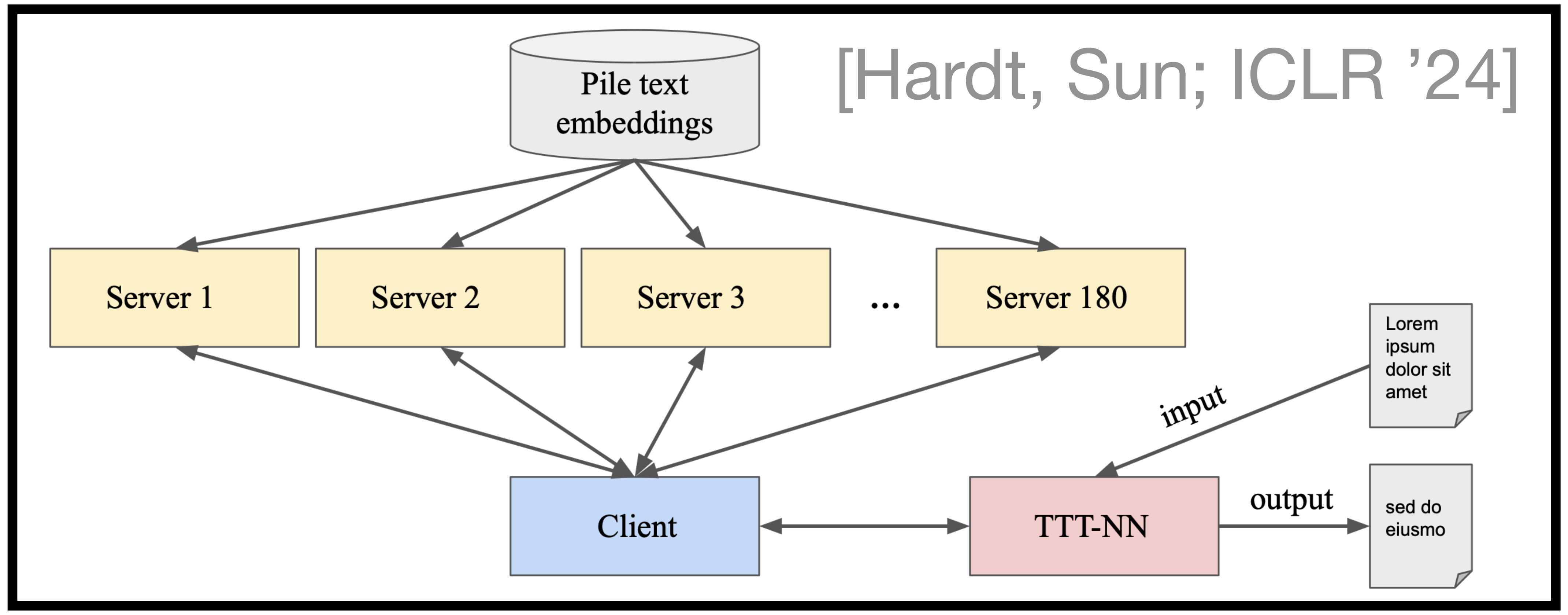
  CNNs on MNIST

History

- **since 2020s:** (few-shot) in-context learning (GPT-3)

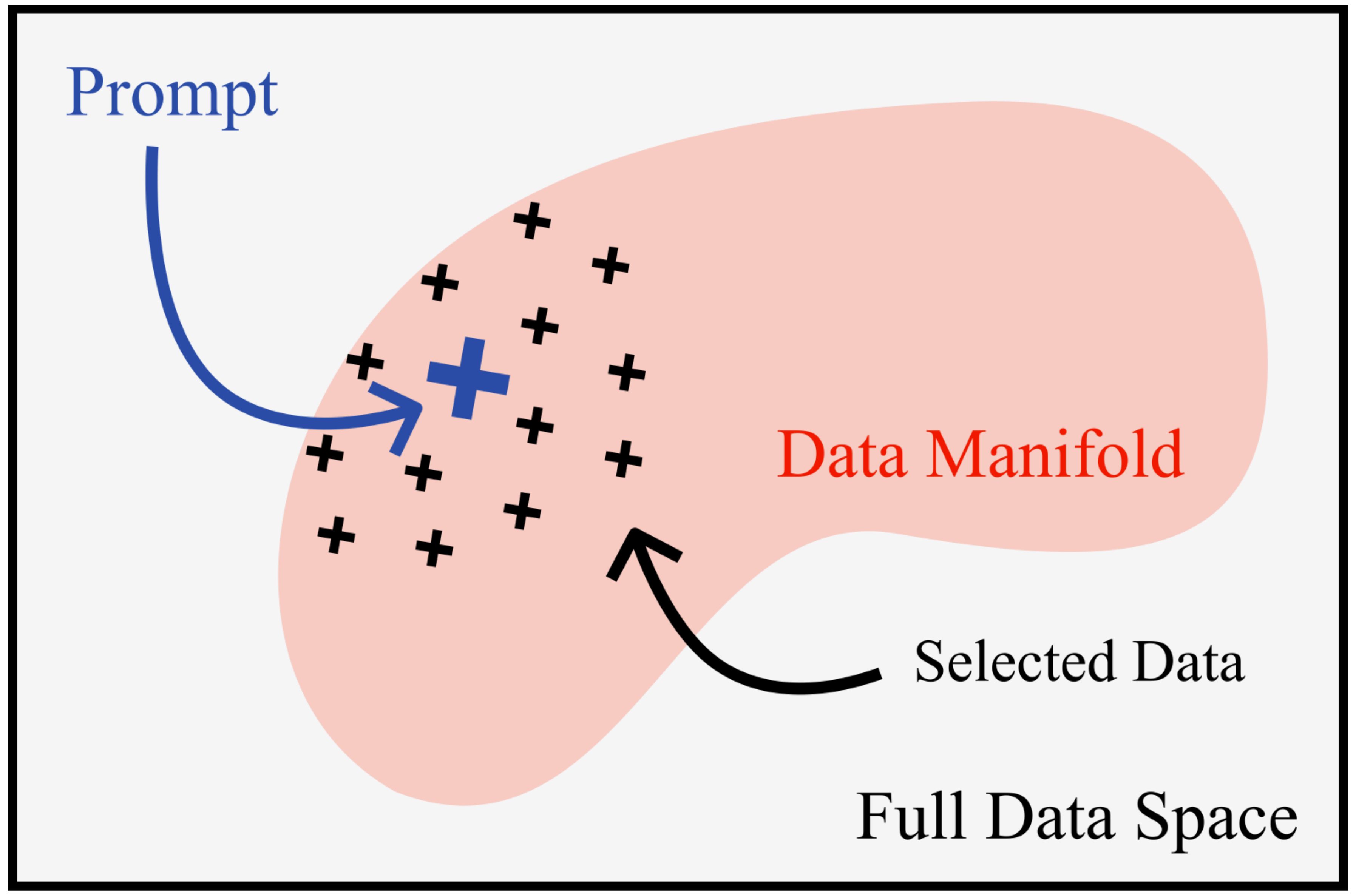
  parametric controller: LLM
  non-parametric memory: context (+ retrieval from database)

- **recently:** local fine-tuning (again!) with GPT-2 (Hardt & Sun)



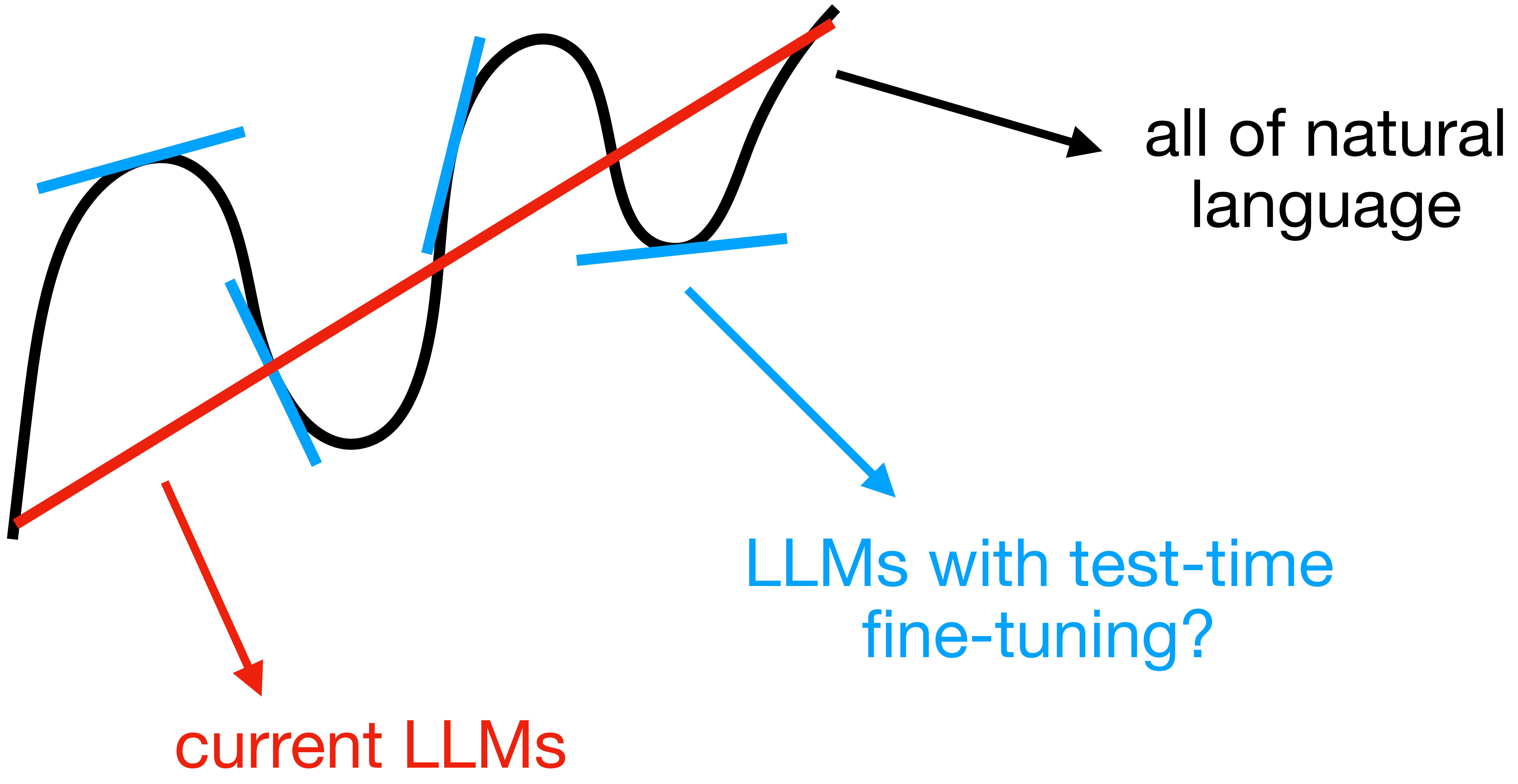
Test-time fine-tuning

Summary

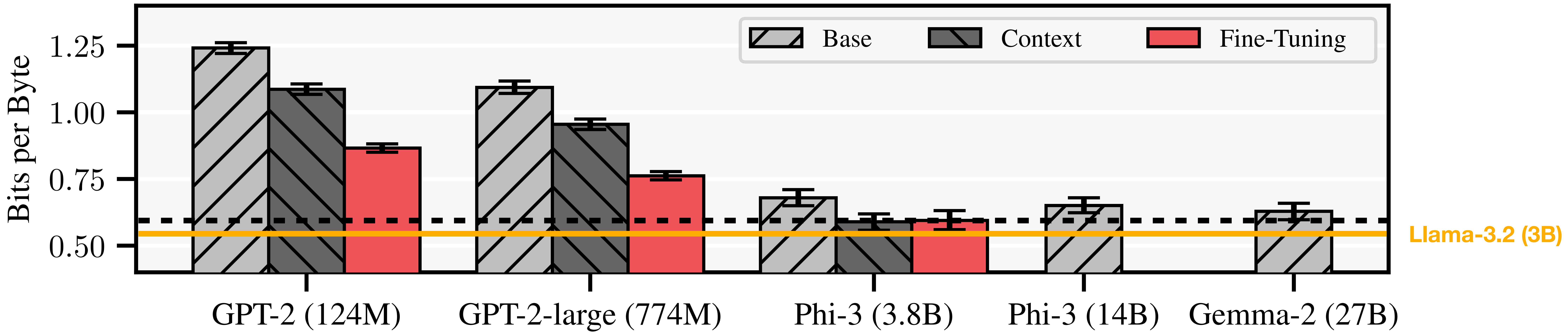


1. take pre-trained model f
2. given input x, find local data D_x from memory
3. fine-tune model f on local data D_x to get **local model** f_x
4. predict $f_x(x)$

Hypothesis for LLMs



Does local learning work with LLMs?



	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 challenge: which data to select?

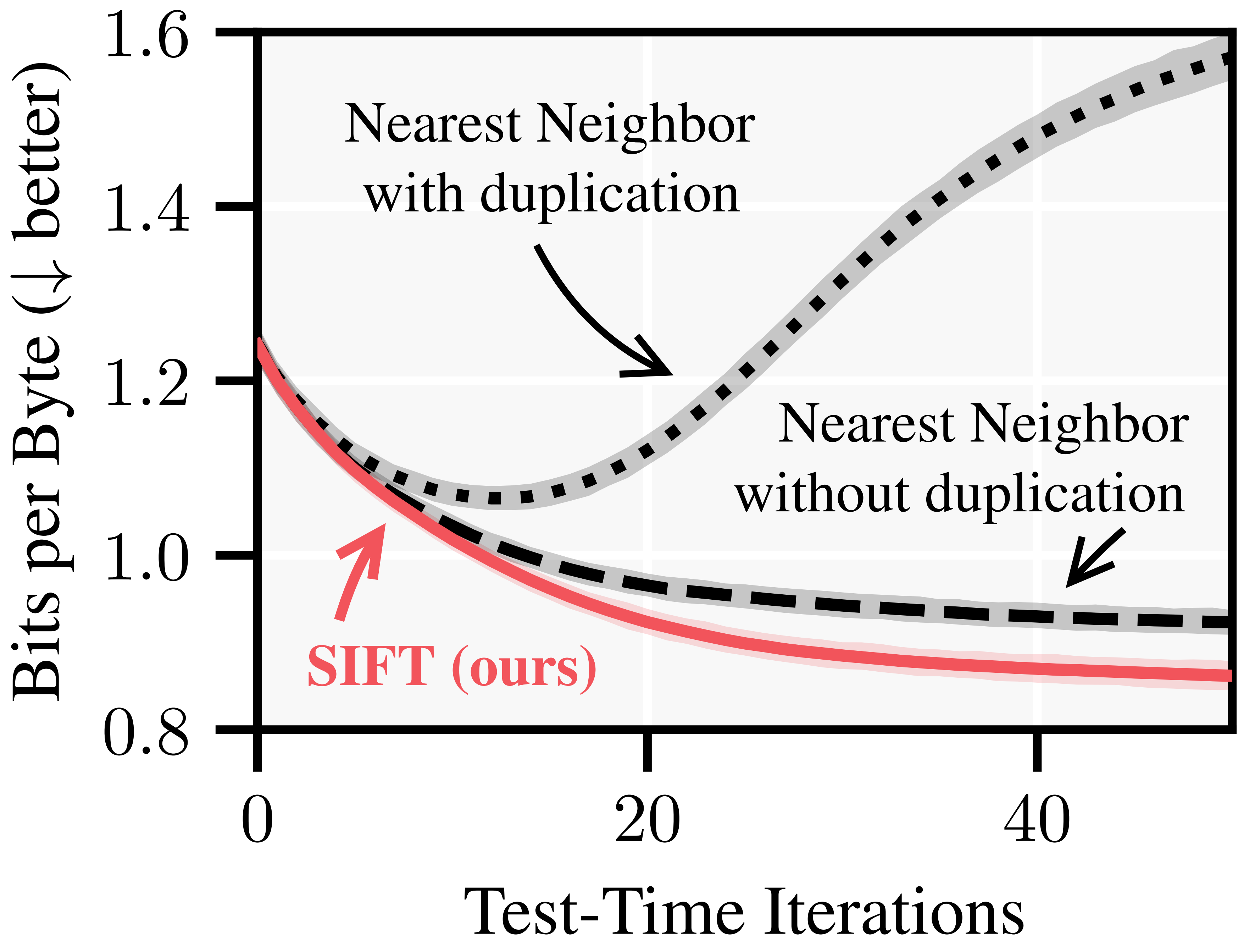
Prompt: What is the age of Michael Jordan and how many kids does he have?

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: **S**electing **I**nformative data for **F**ine-**T**uning

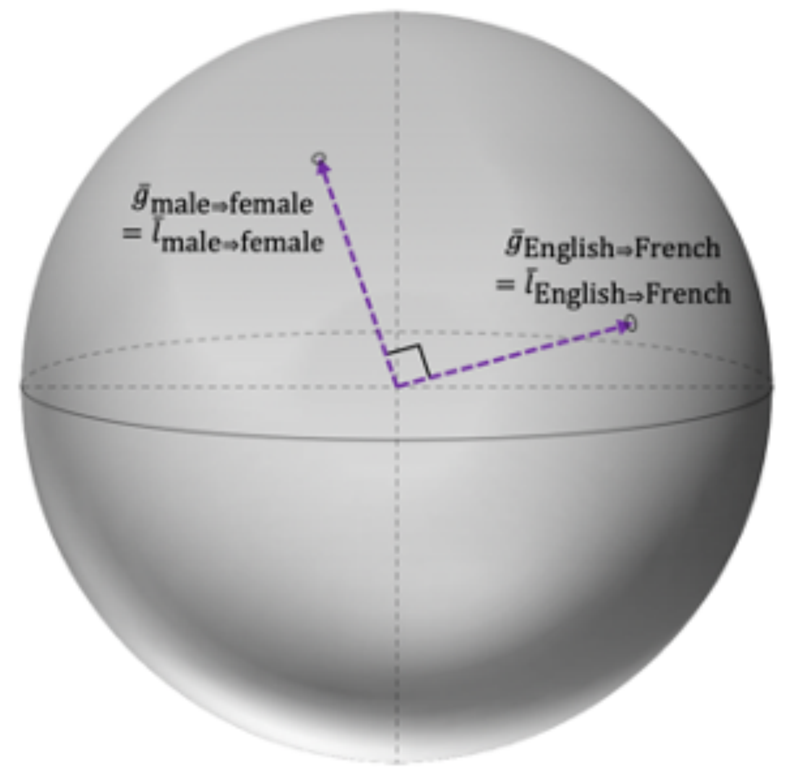
Principle:

Select data that maximally reduces "uncertainty" about how to respond to the prompt.

1. Estimate uncertainty
2. Minimize uncertainty

(**H**, Bongni, Hakimi, Krause; ICLR '25)

1) Estimating uncertainty



- Making this tractable…

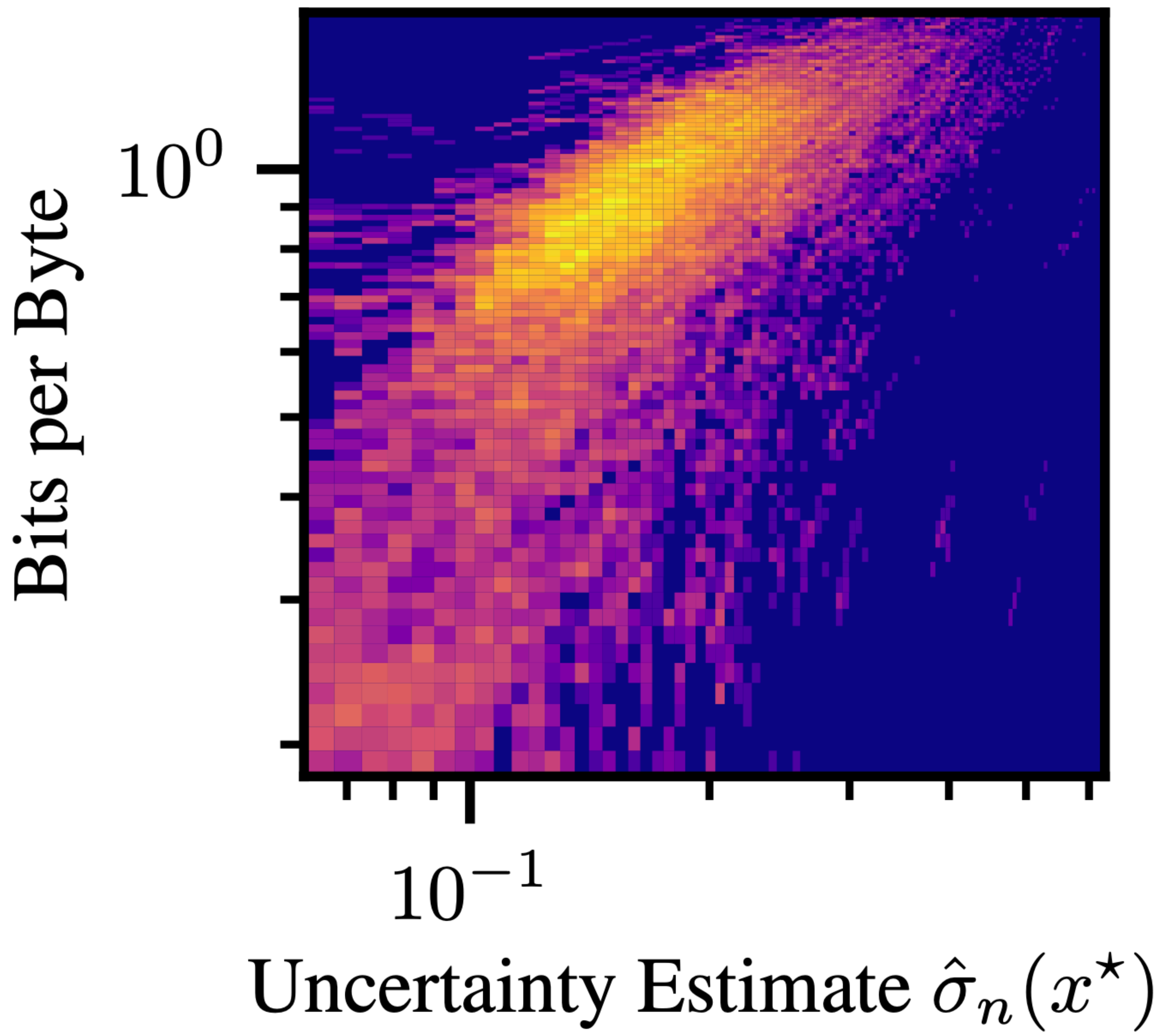
  Surrogate model: approximate model f as logit-linear model in a known representation space

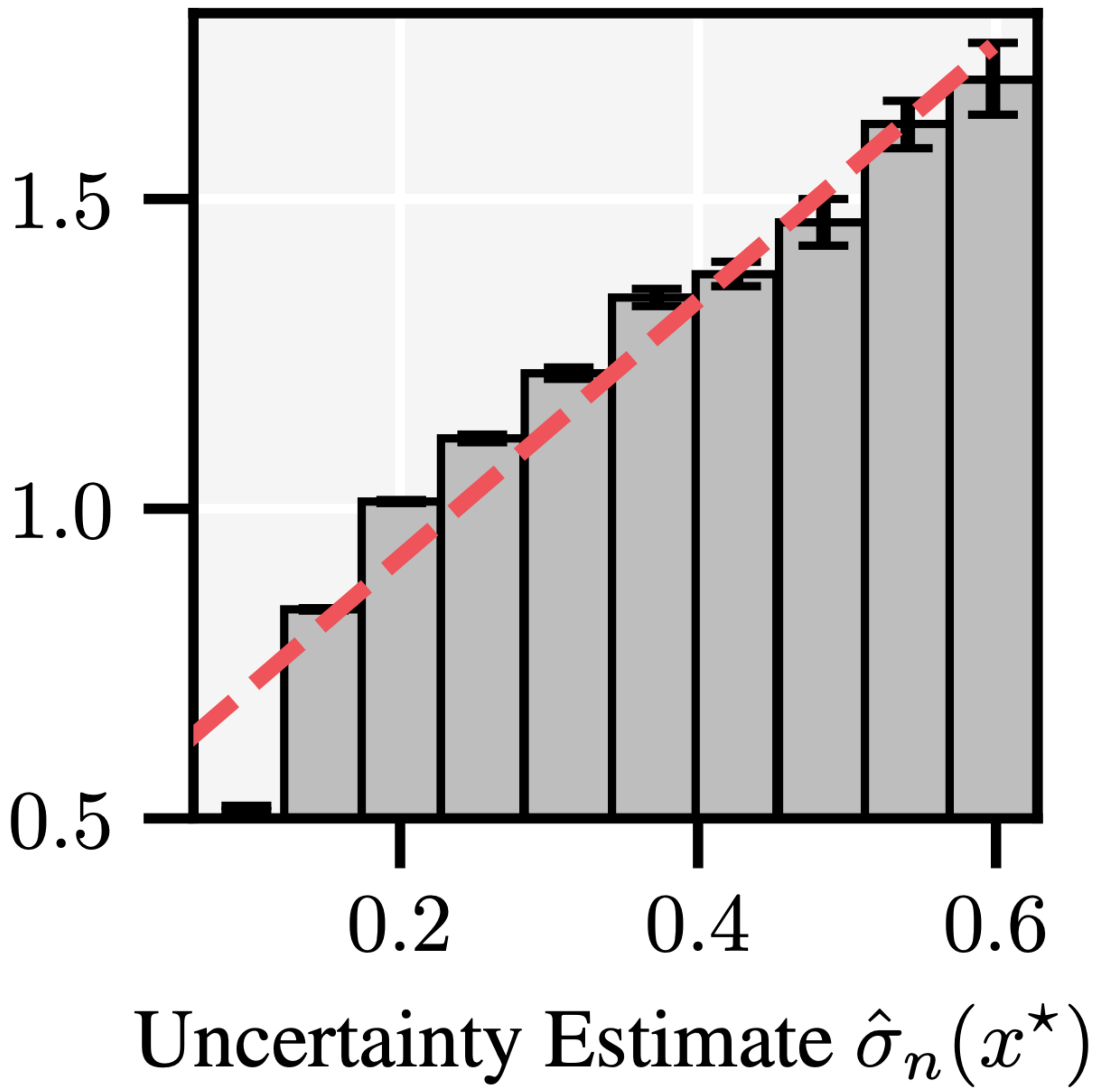
  → linear representation hypothesis (e.g., Park et al; ICML ’24)

- **Error bound:** $\underbrace{d_{\mathrm{TV}}(f_n(x), f^\star(x))}_{\text{error}} \leq \underbrace{\beta(\delta)}_{\text{scaling}} \underbrace{\sigma_n(x)}_{\text{uncertainty}}$ (with prob. $1-\delta$)

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

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





2) Minimizing uncertainty

- SIFT: minimize uncertainty about response to input $x^\star$: $D_{x^\star} = X_n \cup \{x_{n+1}\}$

$$x_{n+1} = \operatorname*{argmin}_x \; \sigma_{X_n \cup \{x\}}(x^\star)$$

prompt

$$= \operatorname*{argmax}_x \begin{bmatrix} k(x^\star, x_1) \\ \vdots \\ k(x^\star, x_n) \\ k(x^\star, x) \end{bmatrix}^\top \left(\begin{bmatrix} k(x_1, x_1) & \cdots & k(x_1, x_n) & k(x_1, x) \\ \vdots & \ddots & \vdots & \vdots \\ k(x_n, x_1) & \cdots & k(x_n, x_n) & k(x_n, x) \\ k(x, x_1) & \cdots & k(x, x_n) & k(x, x) \end{bmatrix} + \frac{1}{\eta} I_{n+1} \right)^{-1} \begin{bmatrix} k(x^\star, x_1) \\ \vdots \\ k(x^\star, x_n) \\ k(x^\star, x) \end{bmatrix}$$

with $k(x, x') = \boldsymbol{\phi}(x)^\top \boldsymbol{\phi}(x')$

maximize relevance

minimize redundancy

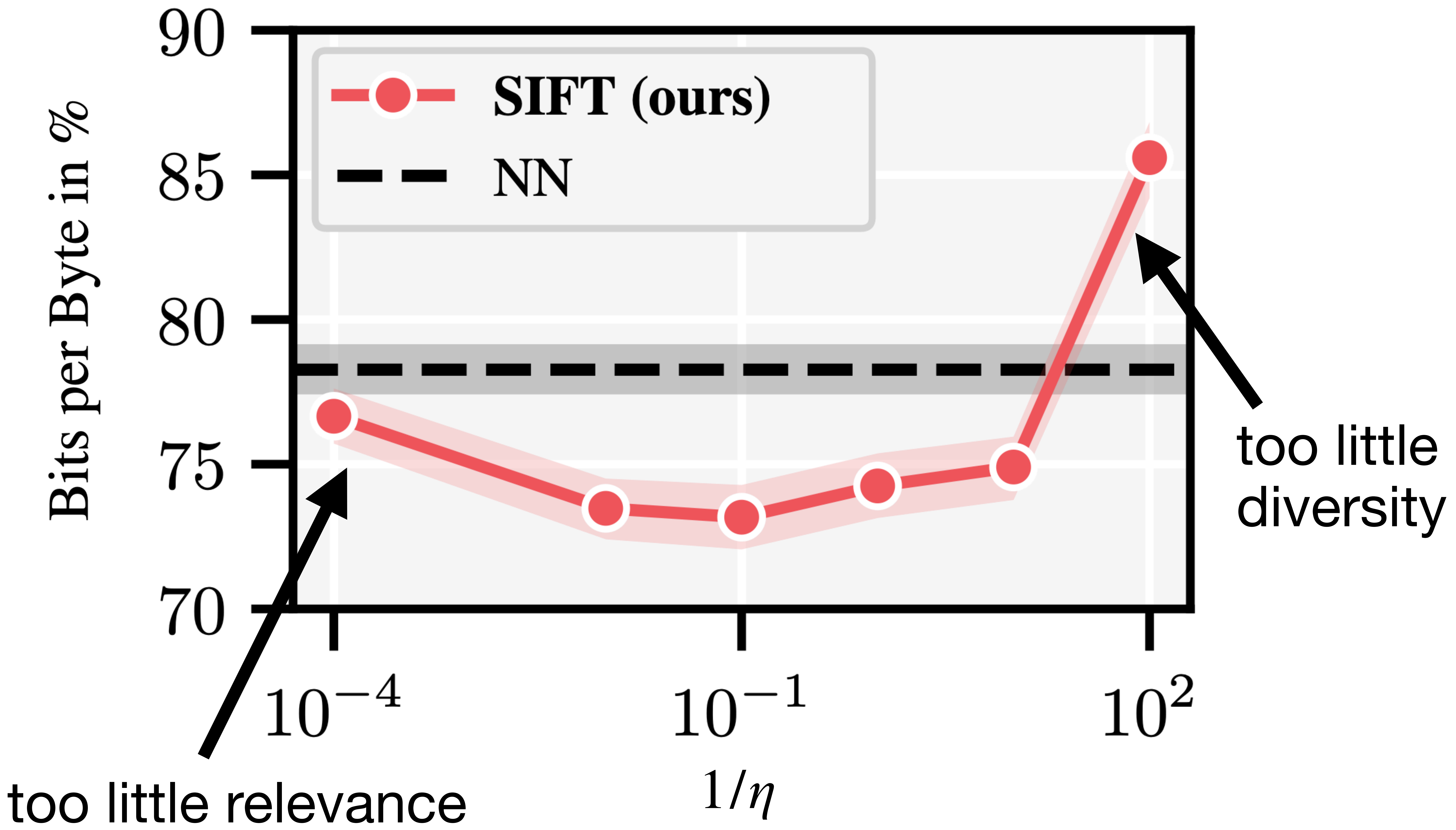
- convergence of uncertainty is guaranteed!

$$\sigma_n(x^\star) \to \sigma_\infty(x^\star)$$

irreducible uncertainty

Not possible with nearest neighbor retrieval!

→ predictions can be only as good as the data and the learned abstractions!


SIFT (ours)
NN
90
85
80
75
70
Bits per Byte in %
10^{-4}
10^{-1}
10^{2}
$1/\eta$
too little relevance
too little diversity

A probabilistic interpretation of SIFT

probabilistic model
with **belief** about f
("controller")

search for x

response $y(x)$

"memory"

Tractable Probabilistic Model

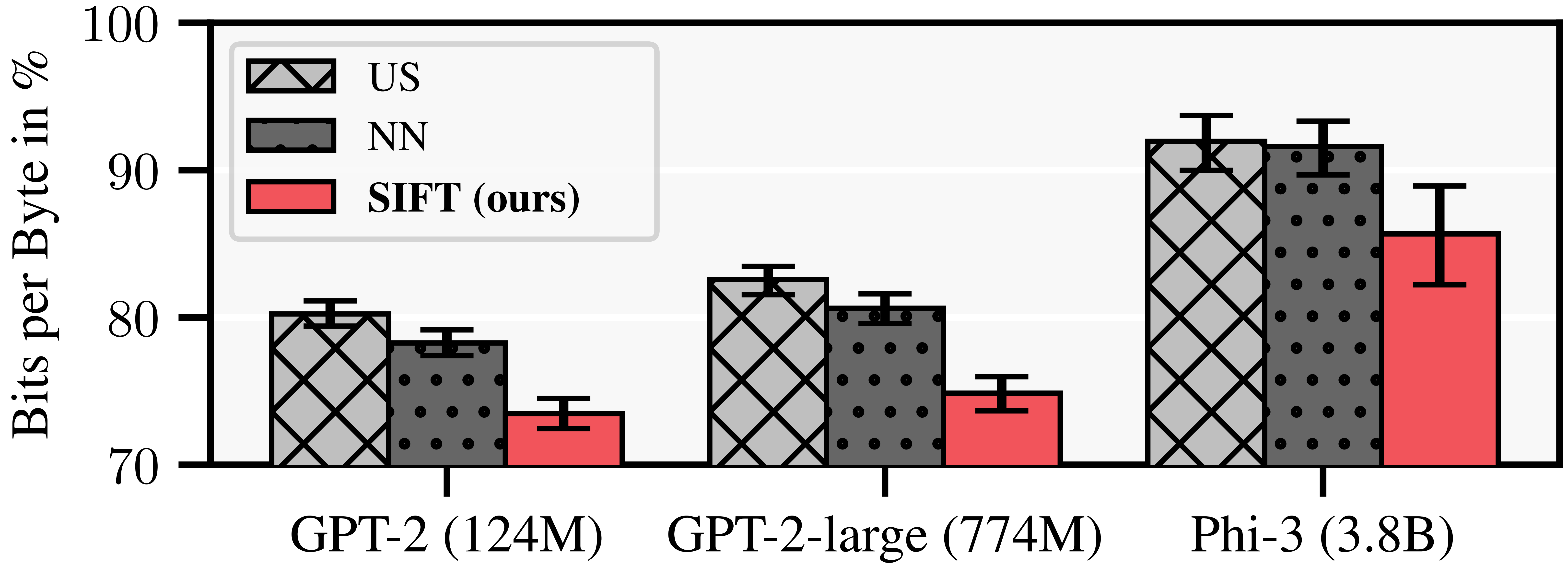
$$y(x) = f(x) + \varepsilon(x)$$

$$f \sim \mathcal{GP}(\mu, k)$$

$$\varepsilon(x) \overset{iid}{\sim} \mathcal{N}(0, \sqrt{\lambda})$$

$$\begin{aligned} x_{n+1} &= \arg\min_x \overbrace{\mathrm{Var}(f(x^\star) \mid y_{1:n}, y(x))}^{\textbf{posterior}\text{ variance } \sigma_n^2(x^\star)} \\ &= \arg\max_x \mathrm{I}(f(x^\star); y(x) \mid y_{1:n}) \\ &= \arg\max_x \underbrace{\mathrm{I}(f(x^\star); y(x))}_{\text{relevance}} - \underbrace{\mathrm{I}(f(x^\star); y(x); y_{1:n})}_{\text{redundancy}} \end{aligned}$$

Evaluation: language modeling on the Pile

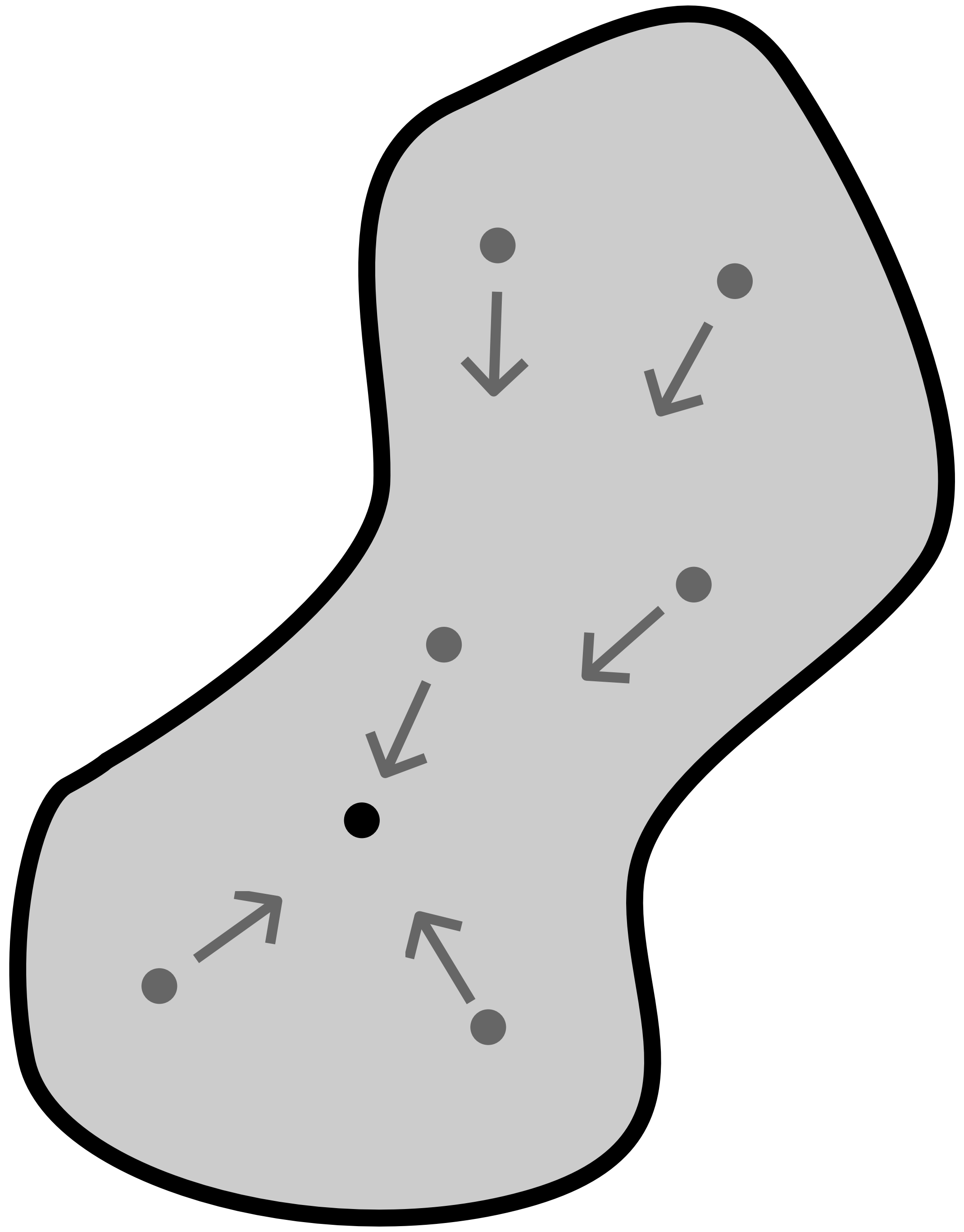


Observations

- larger relative gains with stronger base models
- larger relative gains with larger “memory”

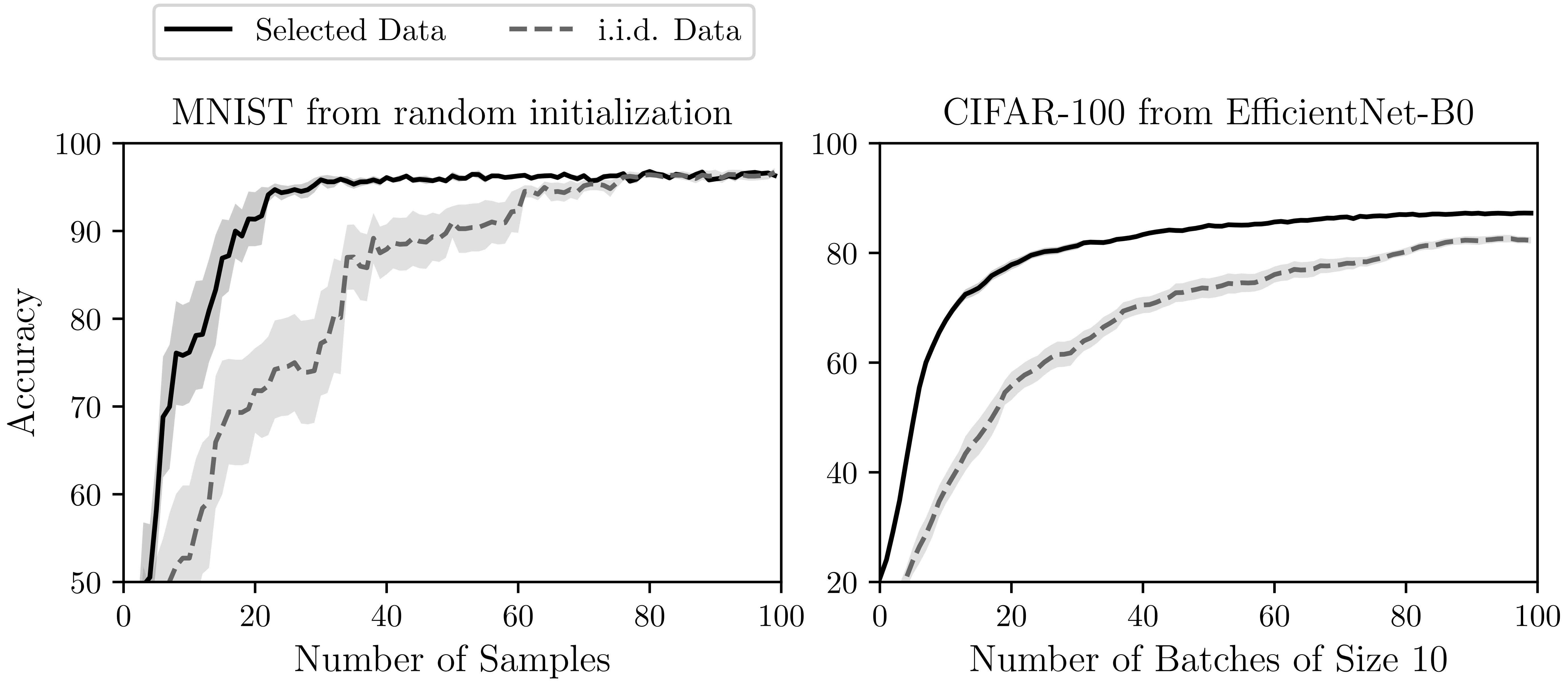
	US	NN	NN-F	SIFT	Δ
NIH Grants	93.1 (1.1)	84.9 (2.1)	91.6 (16.7)	**53.8** (8.9)	↓31.1
US Patents	85.6 (1.5)	80.3 (1.9)	108.8 (6.6)	**62.9** (3.5)	↓17.4
GitHub	45.6 (2.2)	42.1 (2.0)	53.2 (4.0)	**30.0** (2.2)	↓12.1
Enron Emails	**68.6** (9.8)	**64.4** (10.1)	91.6 (20.6)	**53.1** (11.4)	↓11.3
Wikipedia	67.5 (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

Can we learn representations over time?



Strong representations can be bootstrapped!

(**H**, Sukhija, Treven, As, Krause; NeurIPS '24)



Conclusion

Local models
solve one problem at a time

Inductive models (most current SOTA models)
attempt to solve all possible problems at once

→ local learning allows allocating compute where it is “interesting”!

Jonas Hübotter
jonas.huebotter@inf.ethz.ch

- **Transductive Active Learning: Theory and Applications**
  NeurIPS ’24

- **Efficiently Learning at Test-Time: Active Fine-Tuning of LLMs**
  ICLR ’25