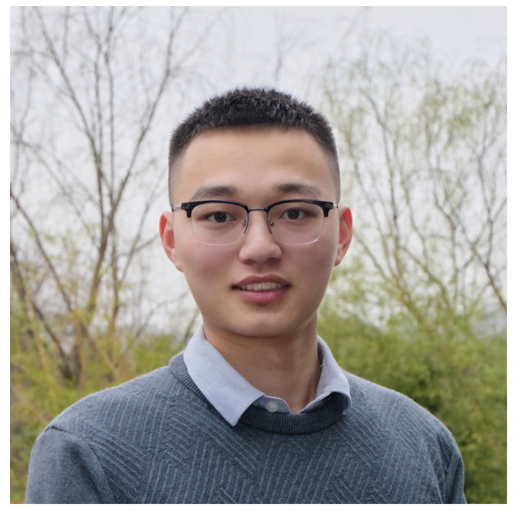
Observational Scaling Laws

& the Predictability of Language Model Performance



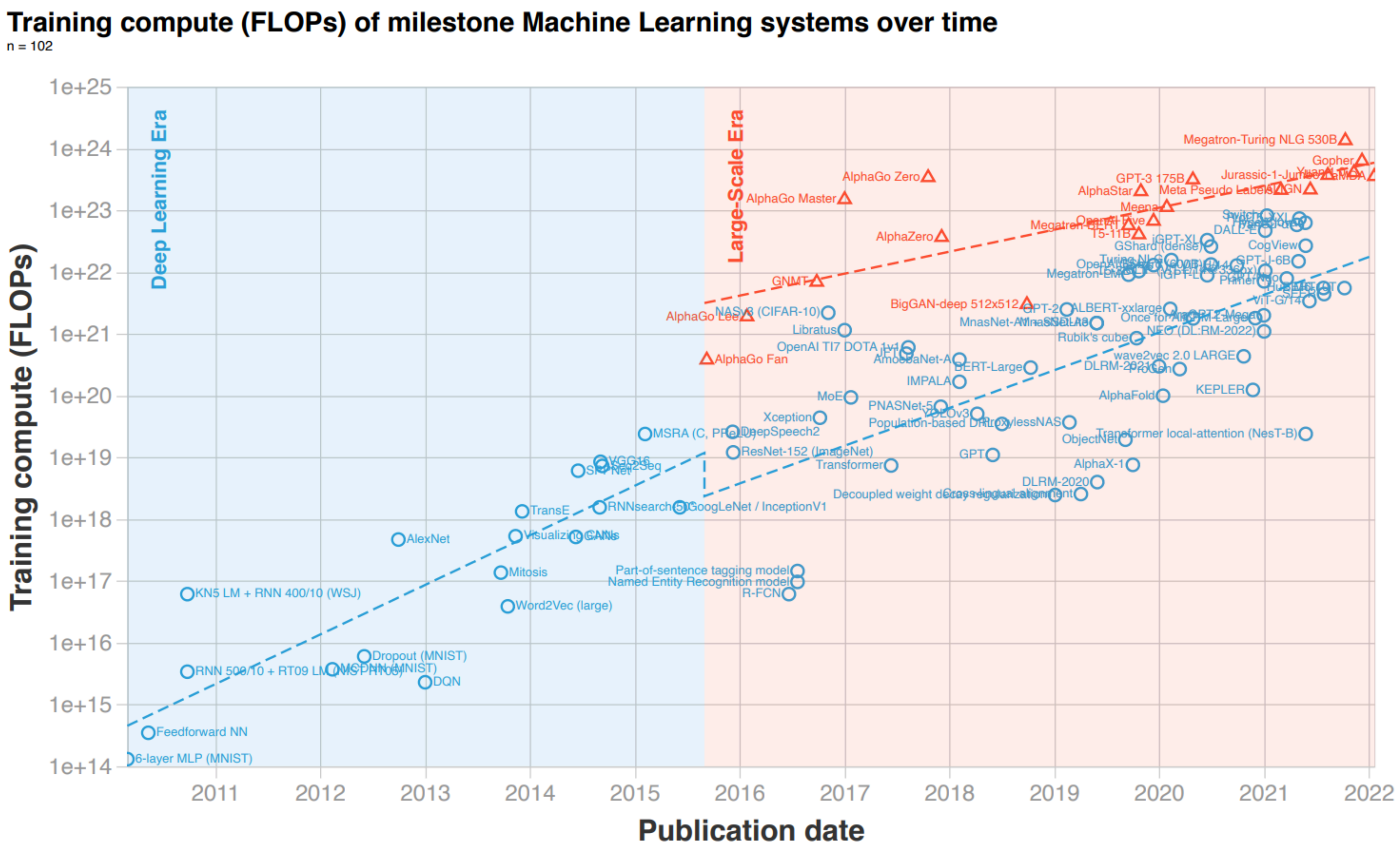
Yangjun Ruan

Stanford University

University of Toronto

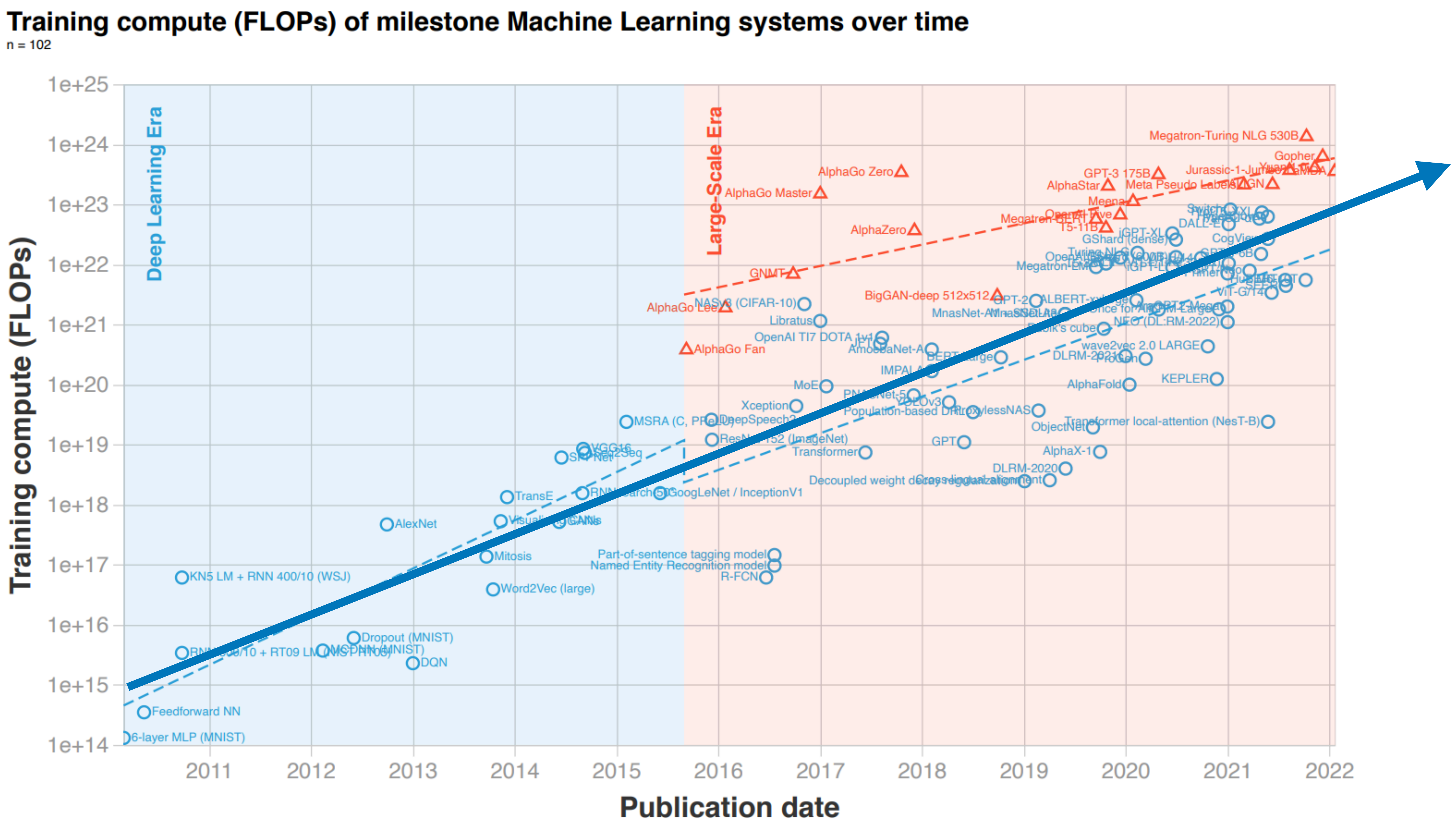
Vector Institute

Scaling Trend of AI Systems



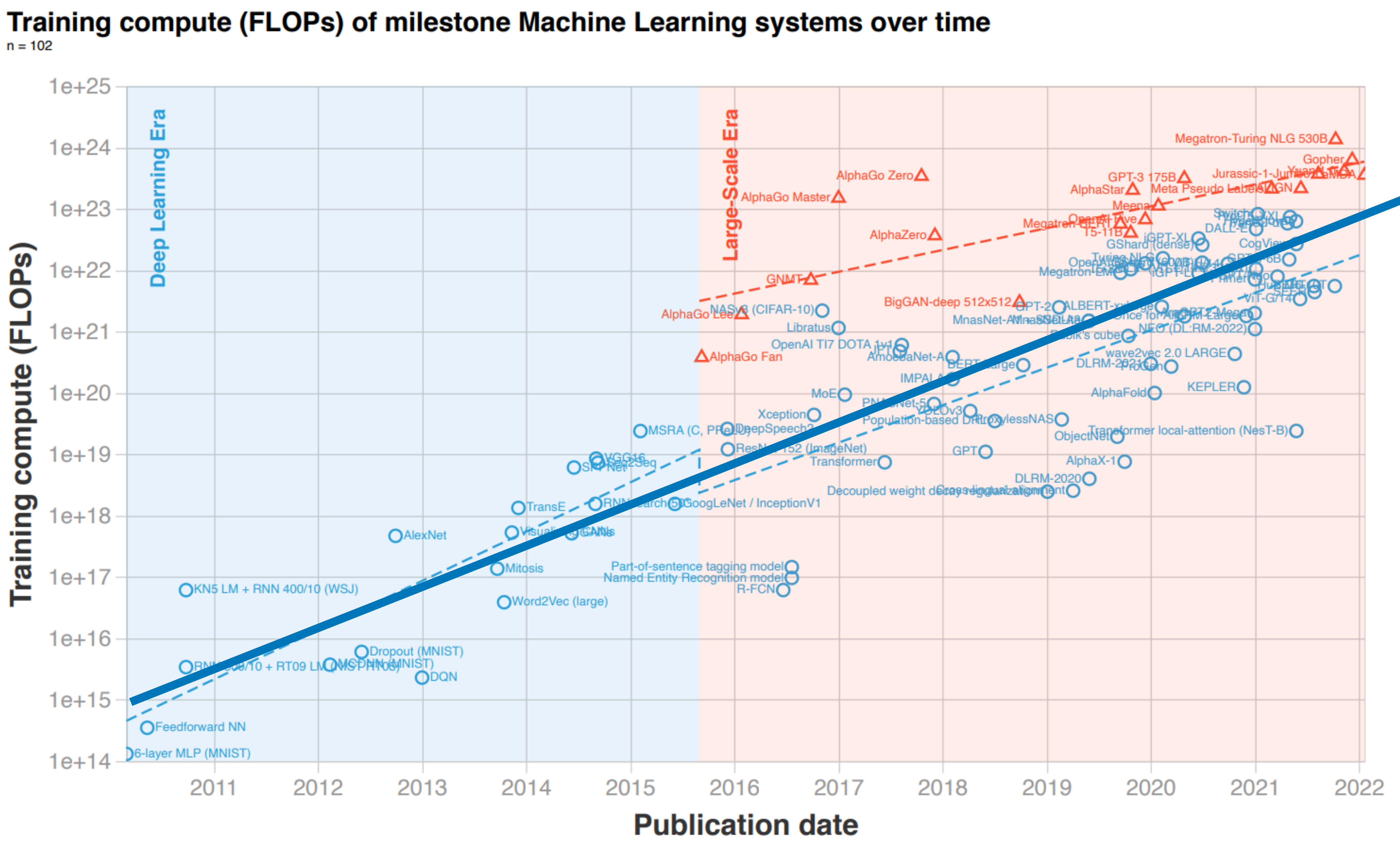
Sevilla et al., 2022. “Compute trends across three eras of machine learning”

Scaling Trend of AI Systems



Sevilla et al., 2022. “Compute trends across three eras of machine learning”

Scaling Trend of AI Systems



Sevilla et al., 2022. “Compute trends across three eras of machine learning”

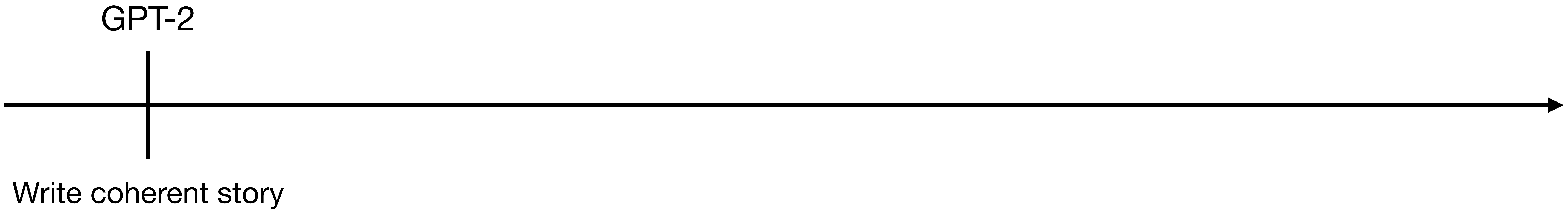
Zuckerberg's Meta Is Spending Billions to Buy 350,000 Nvidia H100 GPUs

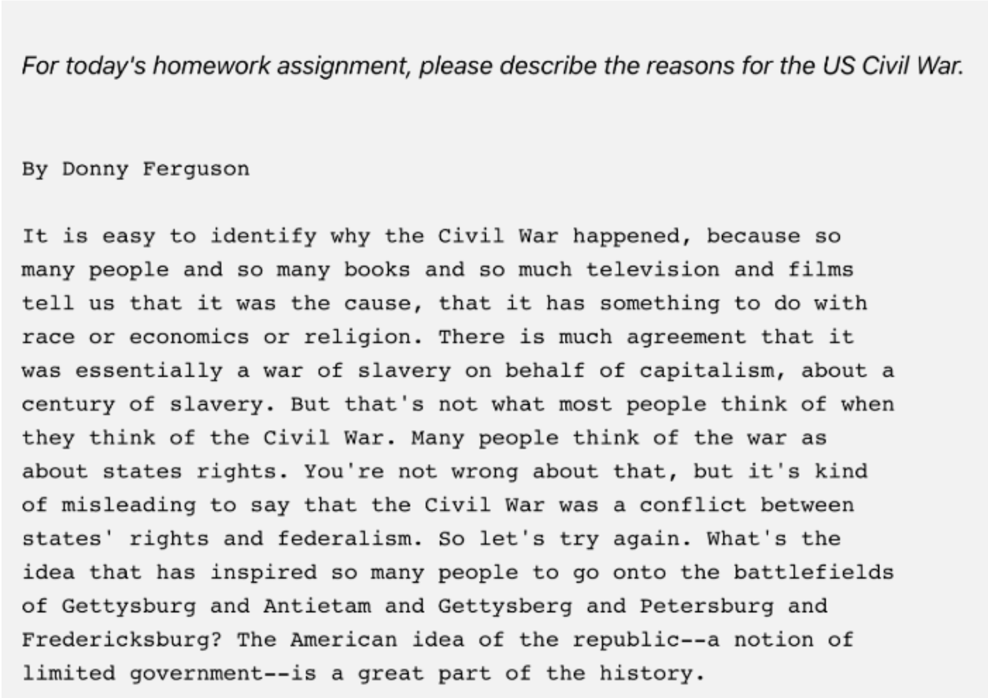
In total, Meta will have the compute power equivalent to 600,000 Nvidia H100 GPUs to help it develop next-generation AI, says CEO Mark Zuckerberg.

Elon Musk turns on xAI's new AI supercomputer: 100K liquid-cooled NVIDIA H100 AI GPUs at 4:20am

Elon Musk posts on X saying 'nice work by xAI and X team, NVIDIA and supporting companies getting Memphis Supercluster training started at 4:20am.

Scaling Unlocks New Capabilities





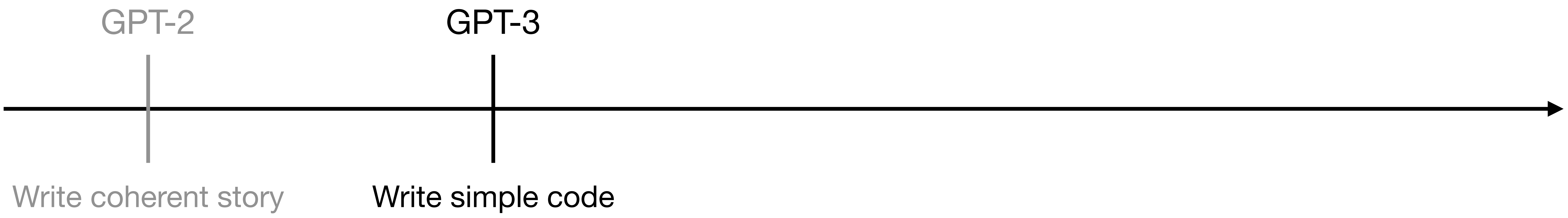
For today's homework assignment, please describe the reasons for the US Civil War.

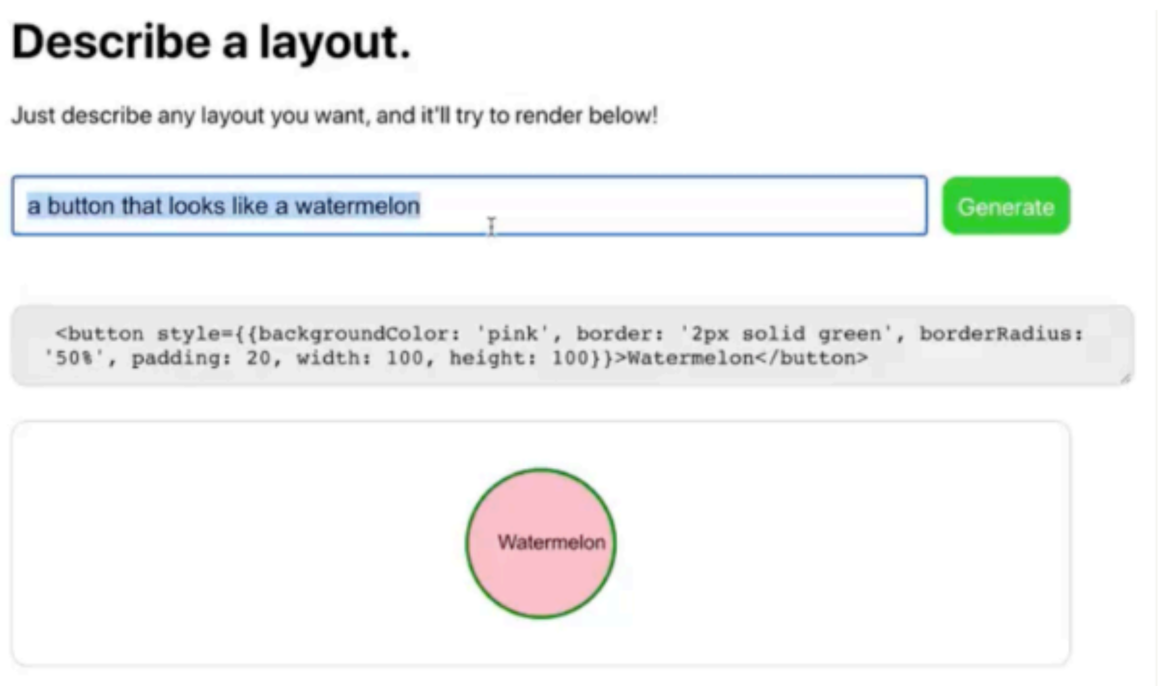
By Donny Ferguson

It is easy to identify why the Civil War happened, because so many people and so many books and so much television and films tell us that it was the cause, that it has something to do with race or economics or religion. There is much agreement that it was essentially a war of slavery on behalf of capitalism, about a century of slavery. But that's not what most people think of when they think of the Civil War. Many people think of the war as about states rights. You're not wrong about that, but it's kind of misleading to say that the Civil War was a conflict between states' rights and federalism. So let's try again. What's the idea that has inspired so many people to go onto the battlefields of Gettysburg and Antietam and Gettysberg and Petersburg and Fredericksburg? The American idea of the republic--a notion of limited government--is a great part of the history.

Aschenbrenner, 2024. “Situational Awareness”

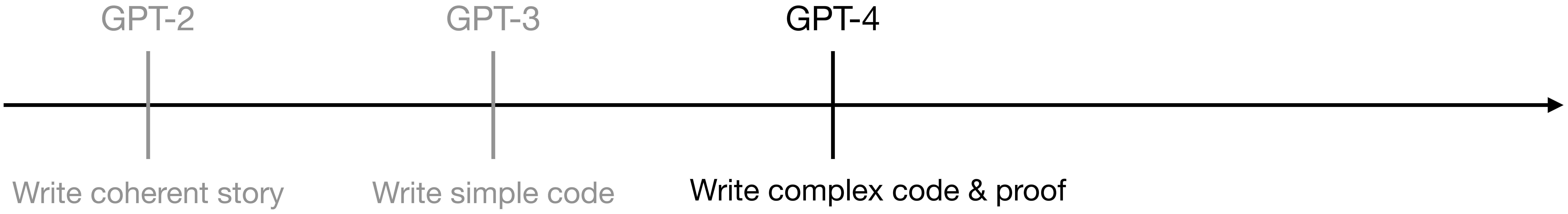
Scaling Unlocks New Capabilities





Aschenbrenner, 2024. “Situational Awareness”

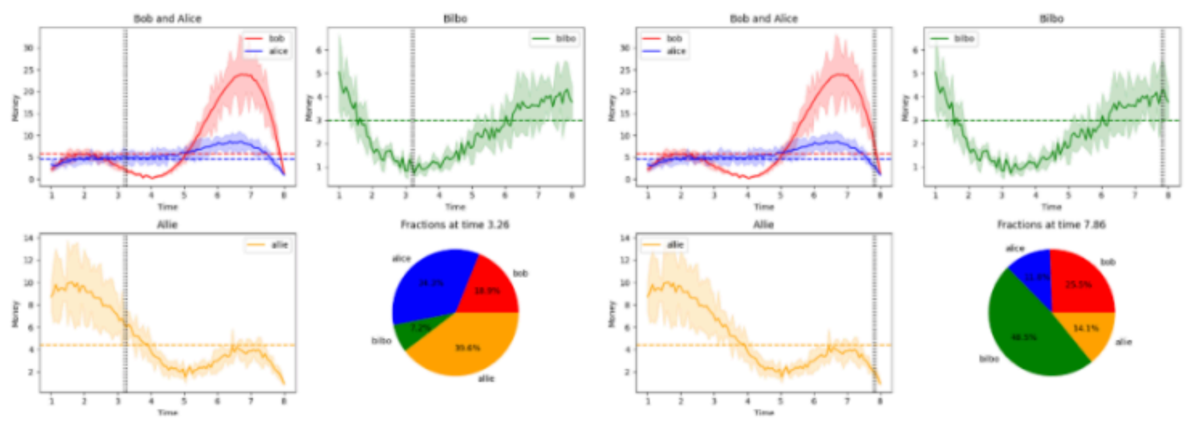
Scaling Unlocks New Capabilities



Prompt: Can you generate a pyplot for the following data: $x = [1, 3, 5, 6, 8]$, $y1 = [2, 3, 5, 18, 1]$, $y2 = [3, 5, 6, 8, 1]$, $y3 = [5, 1, 2, 3, 4]$, $y4 = [9, 7, 2, 3, 1]$. I want $y1, y2$ to be in the same plot, but $y3$ is in another plot next to that plot, $y4$ is in below. I want the legend of $y1$ to be "bob", $y2$ to be "alice", $y3$ to be "bilbo", $y4$ to be "allie". I want the x-axis to be labeled with "time" and y axis to be labeled with "money". I want to add a 10 %-40% random error bar to each curve, through all times (including non-integers). I want smoothed curves to show the plot, and smoothed error bar. Do not use linear interpolation, use smooth interpolation! I want to also add some small zig-zag to the smoothed curve to make it look more real. I want to put a baseline as the mean of each line. I want to put a pie chart below indicating the fraction of the four people in each time step. I also want an animation to show how the fractions are changing in the pie chart in **continuous time**. Interpolate the missing fractions! I also want vertical line animation in other three plots to match the pie chart. I want the fanciest plot. Please add as many fancy things as possible.

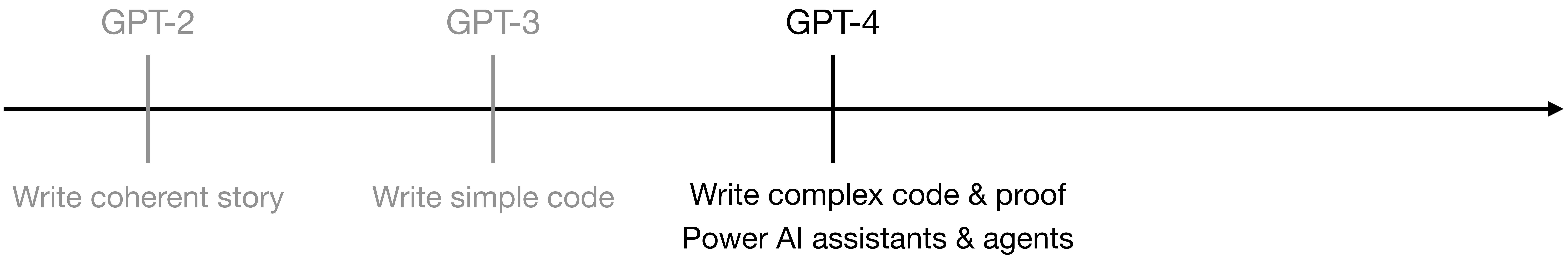
Here is a possible code to generate a pyplot for the following data, with some comments to explain the steps:

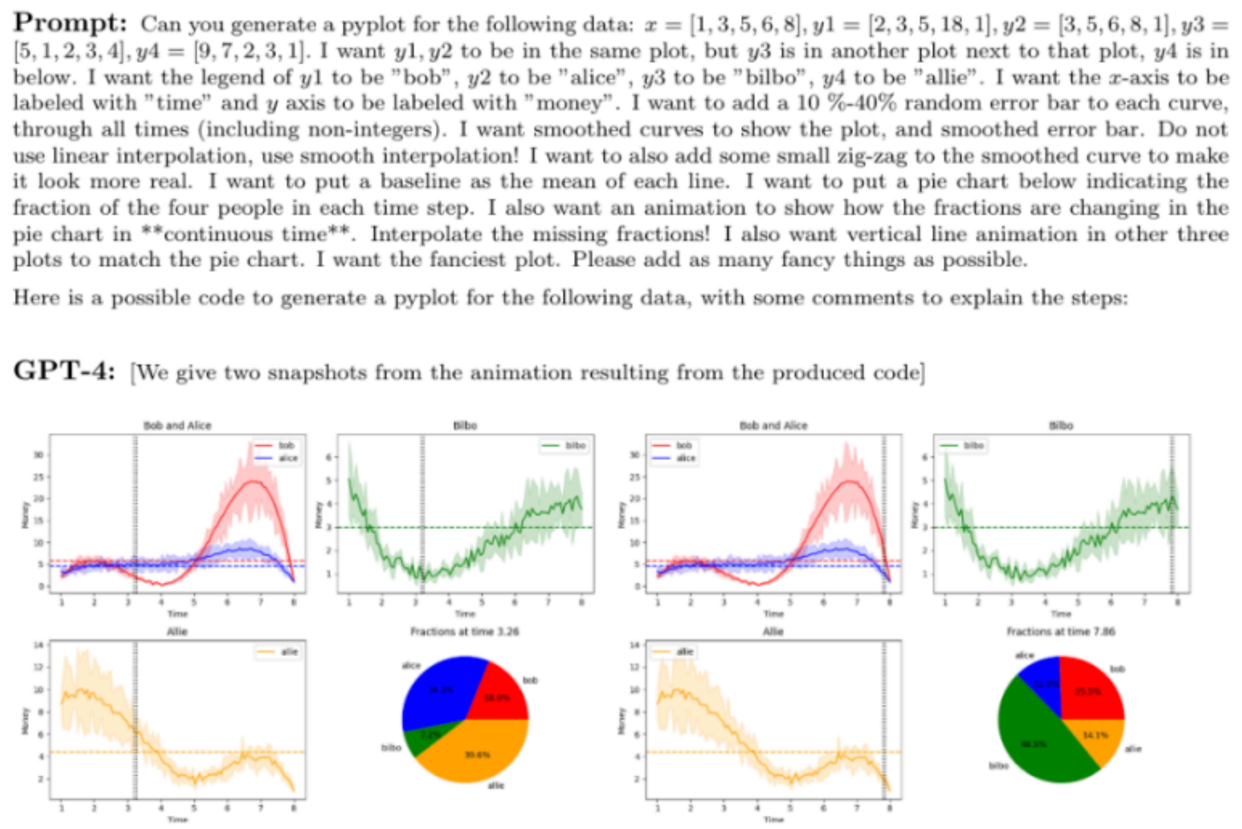
GPT-4: [We give two snapshots from the animation resulting from the produced code]



Aschenbrenner, 2024. "Situational Awareness"

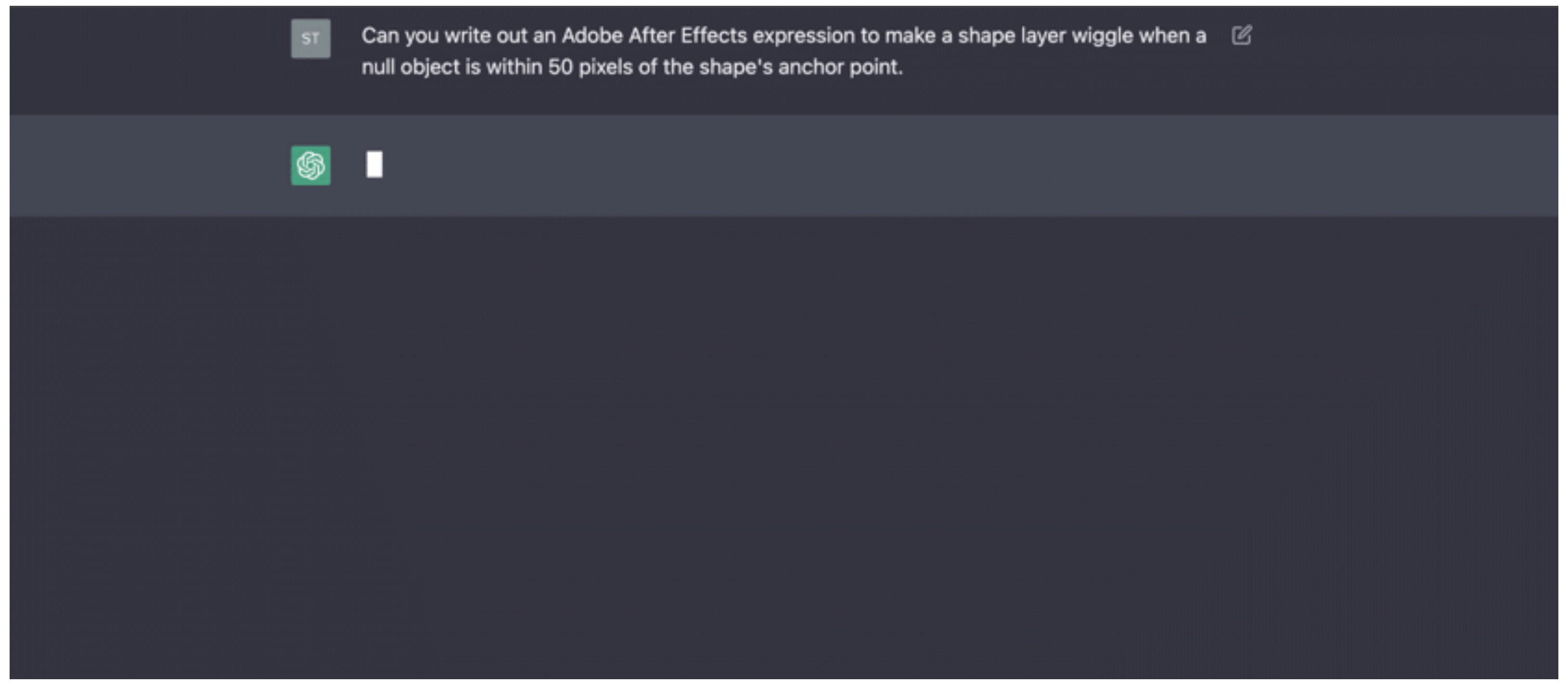
Scaling Unlocks New Capabilities




Prompt: Can you generate a pyplot for the following data: $x = [1, 3, 5, 6, 8]$, $y1 = [2, 3, 5, 18, 1]$, $y2 = [3, 5, 6, 8, 1]$, $y3 = [5, 1, 2, 3, 4]$, $y4 = [9, 7, 2, 3, 1]$. I want $y1, y2$ to be in the same plot, but $y3$ is in another plot next to that plot, $y4$ is in below. I want the legend of $y1$ to be "bob", $y2$ to be "alice", $y3$ to be "bilbo", $y4$ to be "allie". I want the x-axis to be labeled with "time" and y axis to be labeled with "money". I want to add a 10 %-40% random error bar to each curve, through all times (including non-integers). I want smoothed curves to show the plot, and smoothed error bar. Do not use linear interpolation, use smooth interpolation! I want to also add some small zig-zag to the smoothed curve to make it look more real. I want to put a baseline as the mean of each line. I want to put a pie chart below indicating the fraction of the four people in each time step. I also want an animation to show how the fractions are changing in the pie chart in **continuous time**. Interpolate the missing fractions! I also want vertical line animation in other three plots to match the pie chart. I want the fanciest plot. Please add as many fancy things as possible.

Here is a possible code to generate a pyplot for the following data, with some comments to explain the steps:

GPT-4: [We give two snapshots from the animation resulting from the produced code]



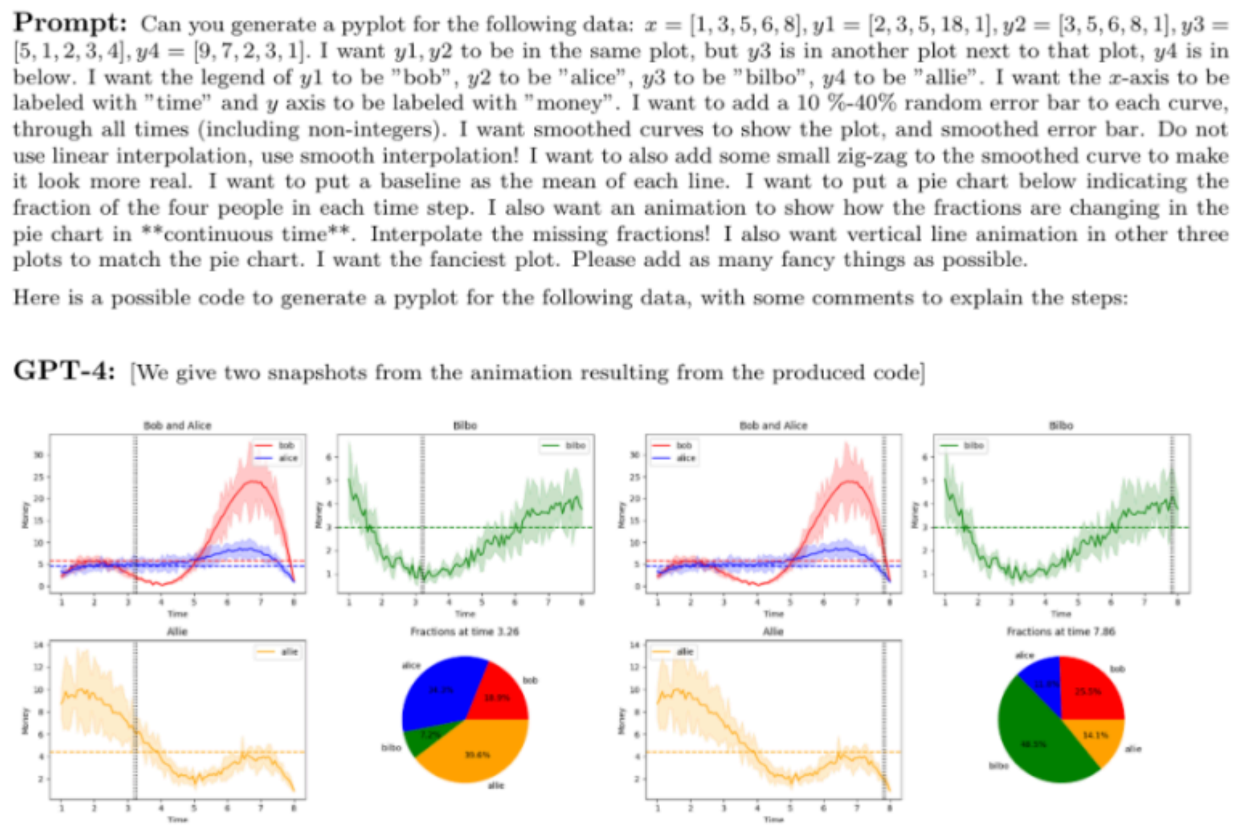
Aschenbrenner, 2024. “Situational Awareness”

Scaling Unlocks New Capabilities

GPT-2 — Write coherent story

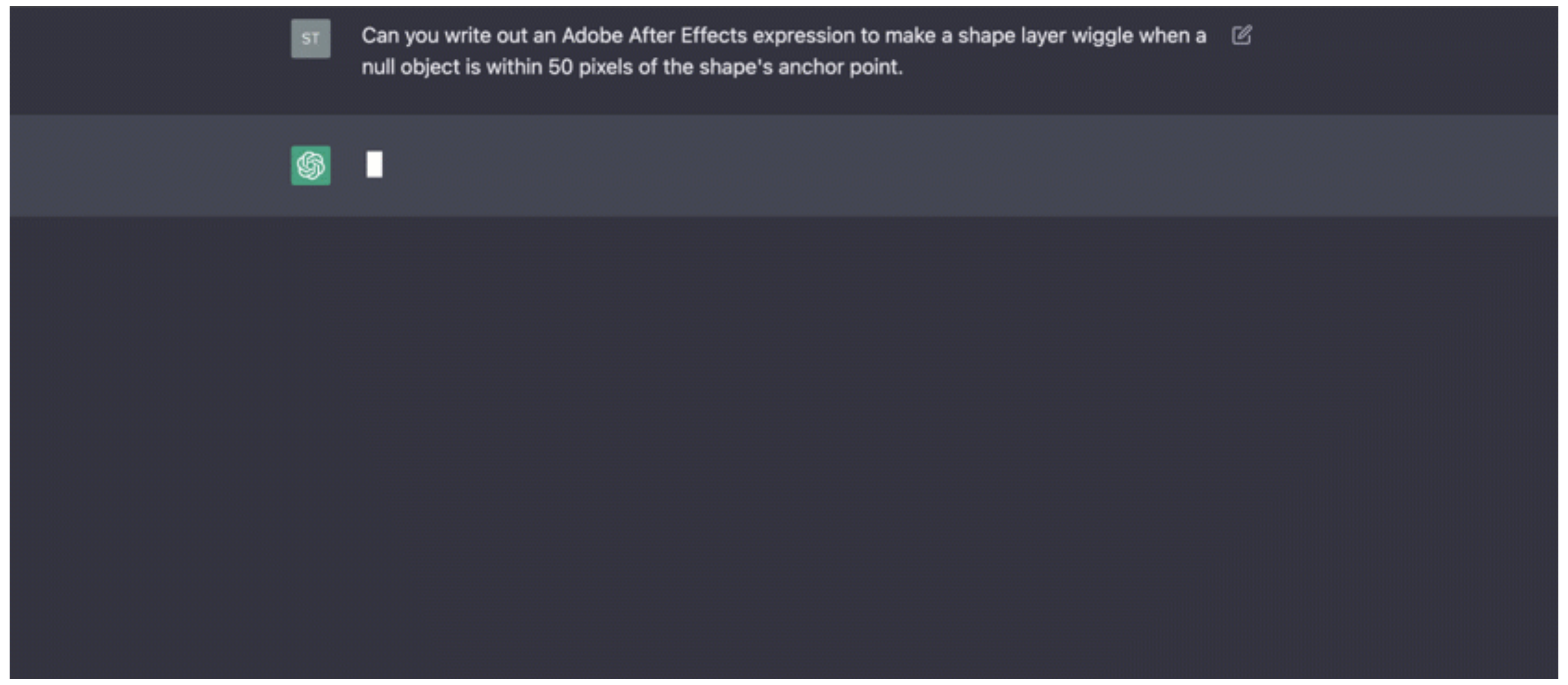
GPT-3 — Write simple code

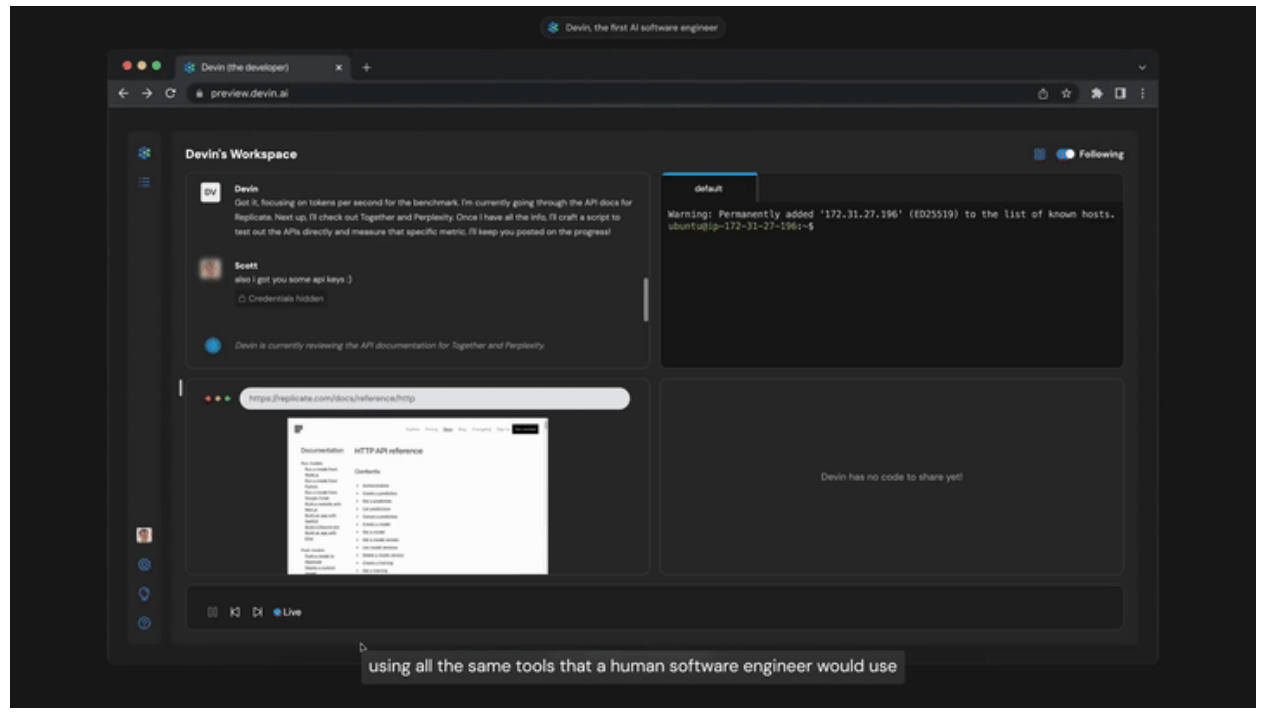
GPT-4 — Write complex code & proof
Power AI assistants & agents


Prompt: Can you generate a pyplot for the following data: $x = [1, 3, 5, 6, 8], y1 = [2, 3, 5, 18, 1], y2 = [3, 5, 6, 8, 1], y3 = [5, 1, 2, 3, 4], y4 = [9, 7, 2, 3, 1]$. I want $y1, y2$ to be in the same plot, but $y3$ is in another plot next to that plot, $y4$ is in below. I want the legend of $y1$ to be "bob", $y2$ to be "alice", $y3$ to be "bilbo", $y4$ to be "allie". I want the x-axis to be labeled with "time" and y axis to be labeled with "money". I want to add a 10 %-40% random error bar to each curve, through all times (including non-integers). I want smoothed curves to show the plot, and smoothed error bar. Do not use linear interpolation, use smooth interpolation! I want to also add some small zig-zag to the smoothed curve to make it look more real. I want to put a baseline as the mean of each line. I want to put a pie chart below indicating the fraction of the four people in each time step. I also want an animation to show how the fractions are changing in the pie chart in **continuous time**. Interpolate the missing fractions! I also want vertical line animation in other three plots to match the pie chart. I want the fanciest plot. Please add as many fancy things as possible.

Here is a possible code to generate a pyplot for the following data, with some comments to explain the steps:

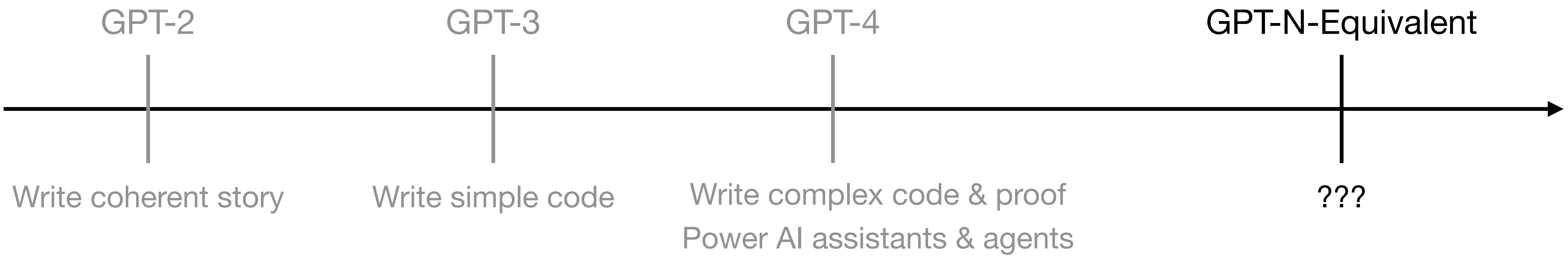
GPT-4: [We give two snapshots from the animation resulting from the produced code]





Aschenbrenner, 2024. "Situational Awareness"

Scaling Unlocks New Capabilities



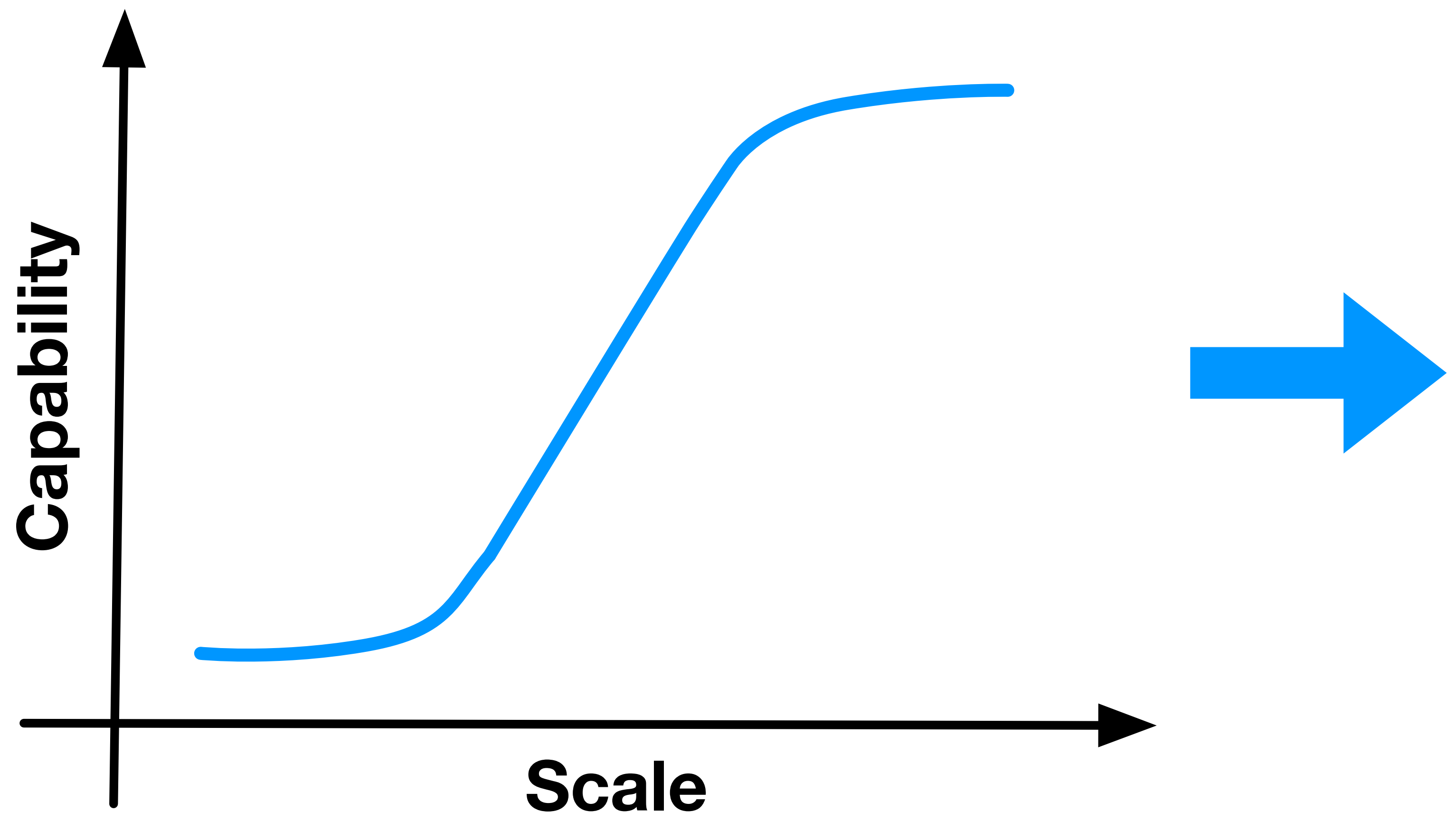
Aschenbrenner, 2024. “Situational Awareness”

Understanding LM Scaling is Critical

Do complex downstream (e.g., agentic) capabilities scale predictably?

Understanding LM Scaling is Critical

Do complex downstream (e.g., agentic) capabilities scale predictably?



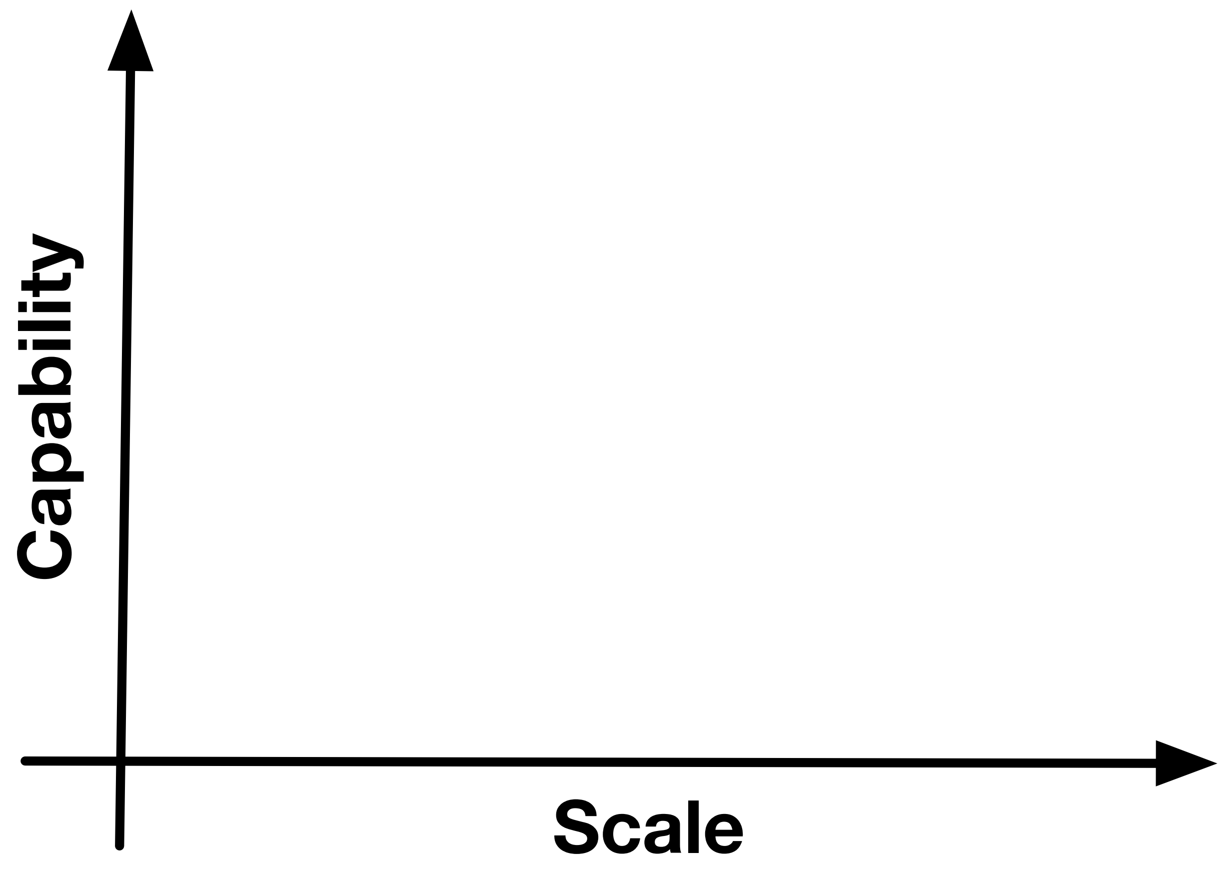
Smooth, predictable scaling

✔ forecasting

✔ algorithmic dev at small scale

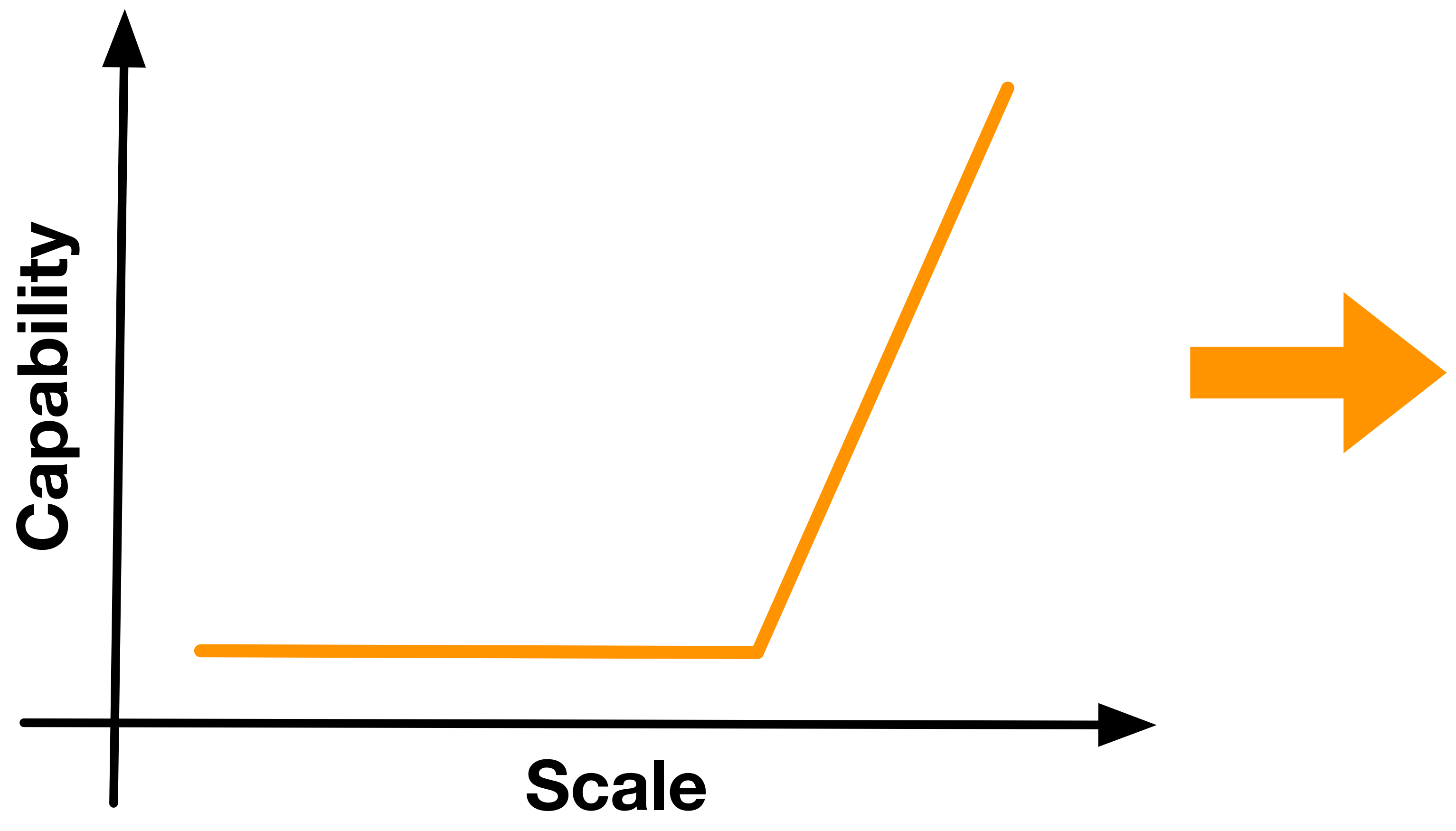
Understanding LM Scaling is Critical

Do complex downstream (e.g., agentic) capabilities scale predictably?



Understanding LM Scaling is Critical

Do complex downstream (e.g., agentic) capabilities scale predictably?



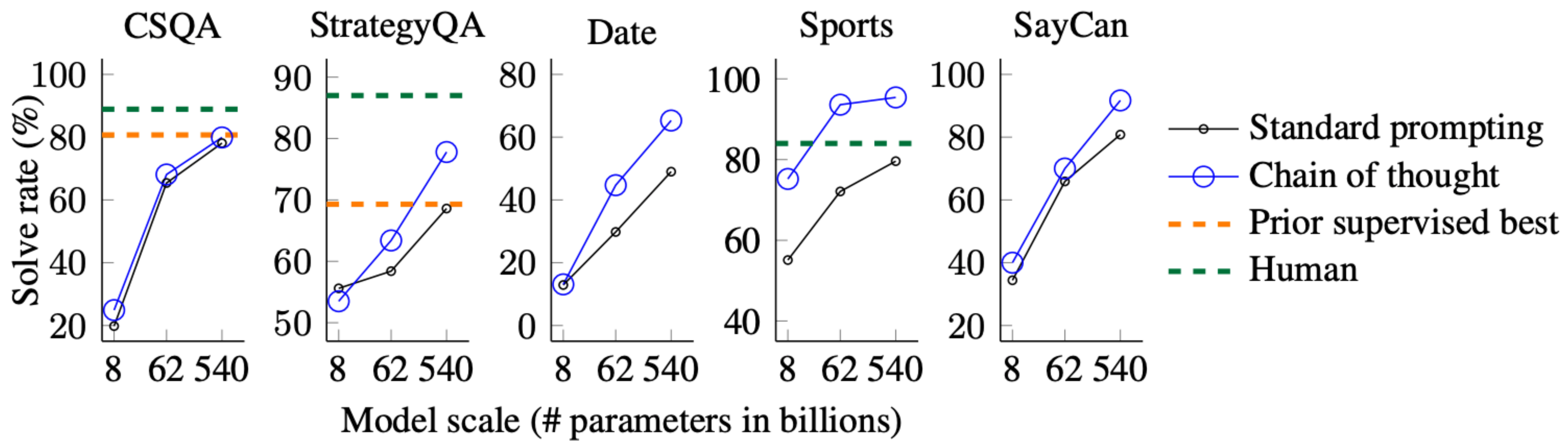
Non-smooth, emergent behaviour

✘ unpredictability

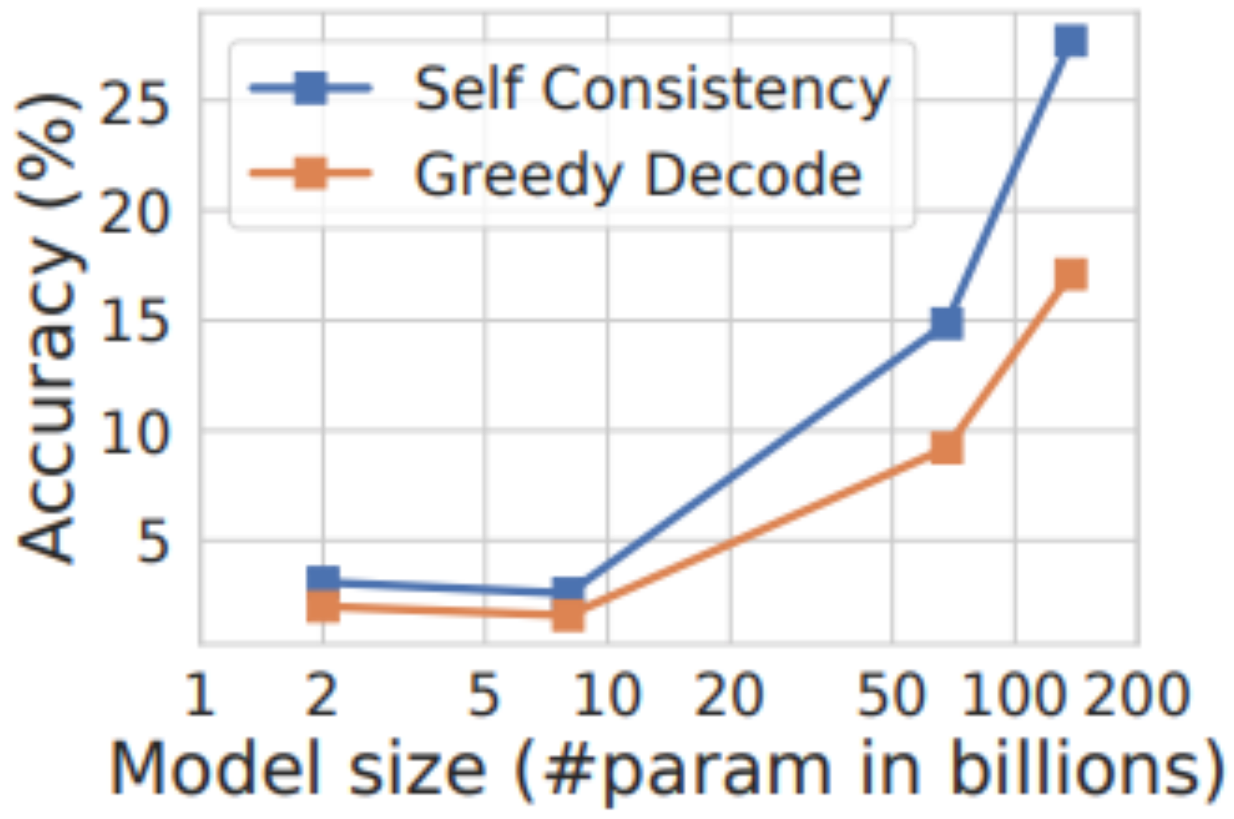
✘ safety concerns

Understanding LM Scaling is Critical

Do our proposed algorithmic interventions stand the test of future scale?



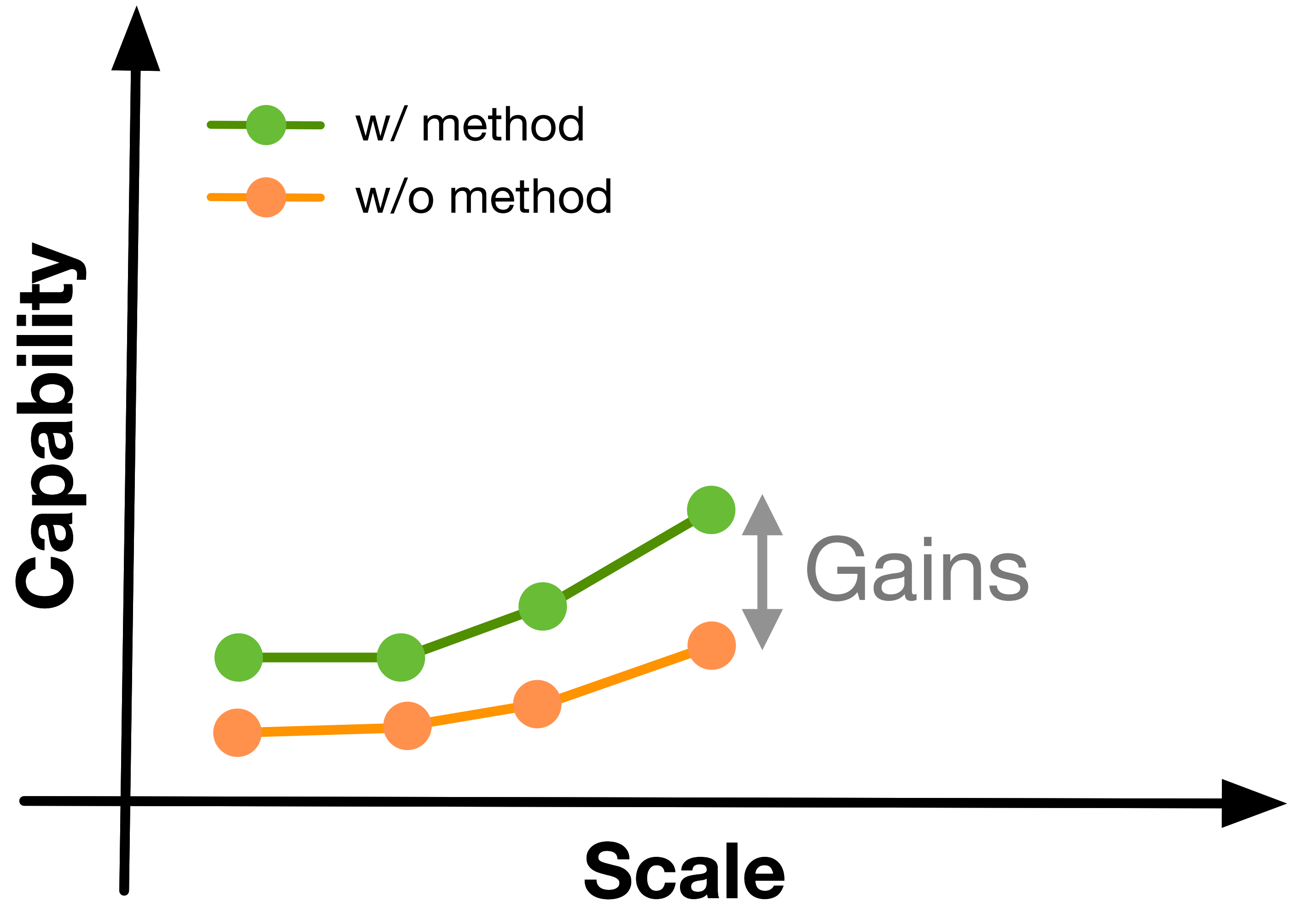
Wei et al., 2022. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”



Wang et al., 2023. “Self-Consistency Improves Chain of Thought Reasoning in Language Models”

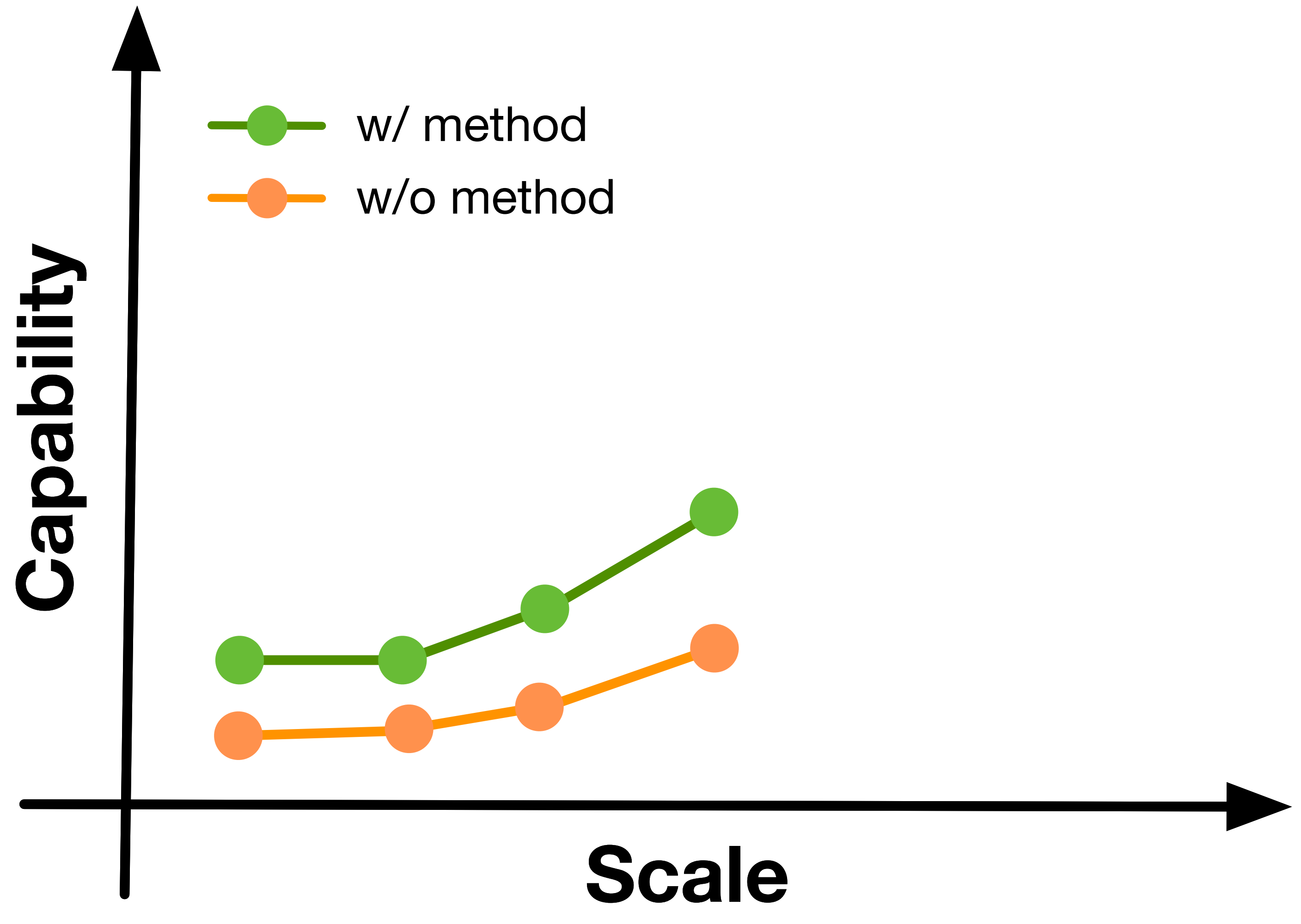
Understanding LM Scaling is Critical

Do our proposed algorithmic interventions stand the test of future scale?



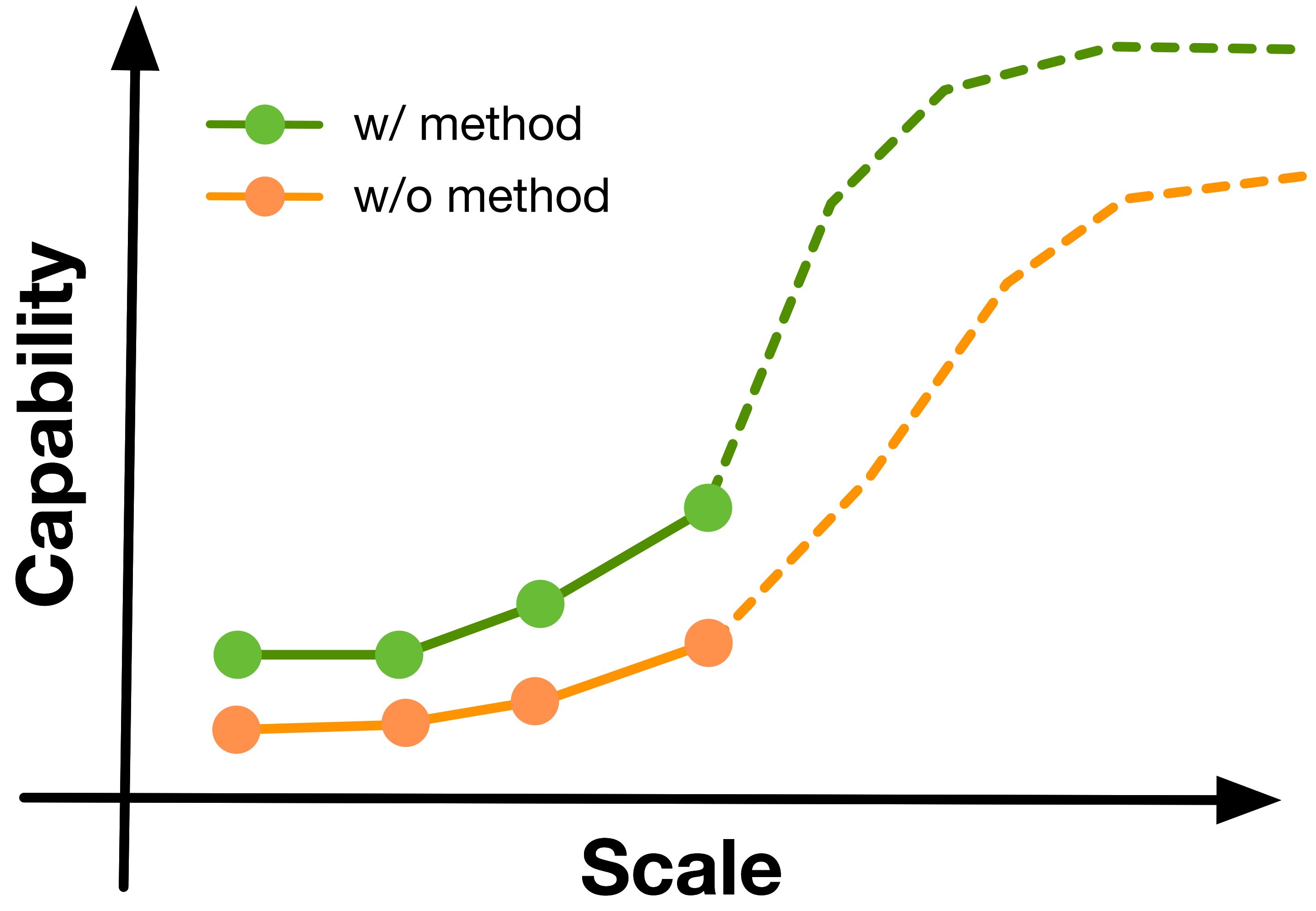
Understanding LM Scaling is Critical

Do our proposed algorithmic interventions stand the test of future scale?



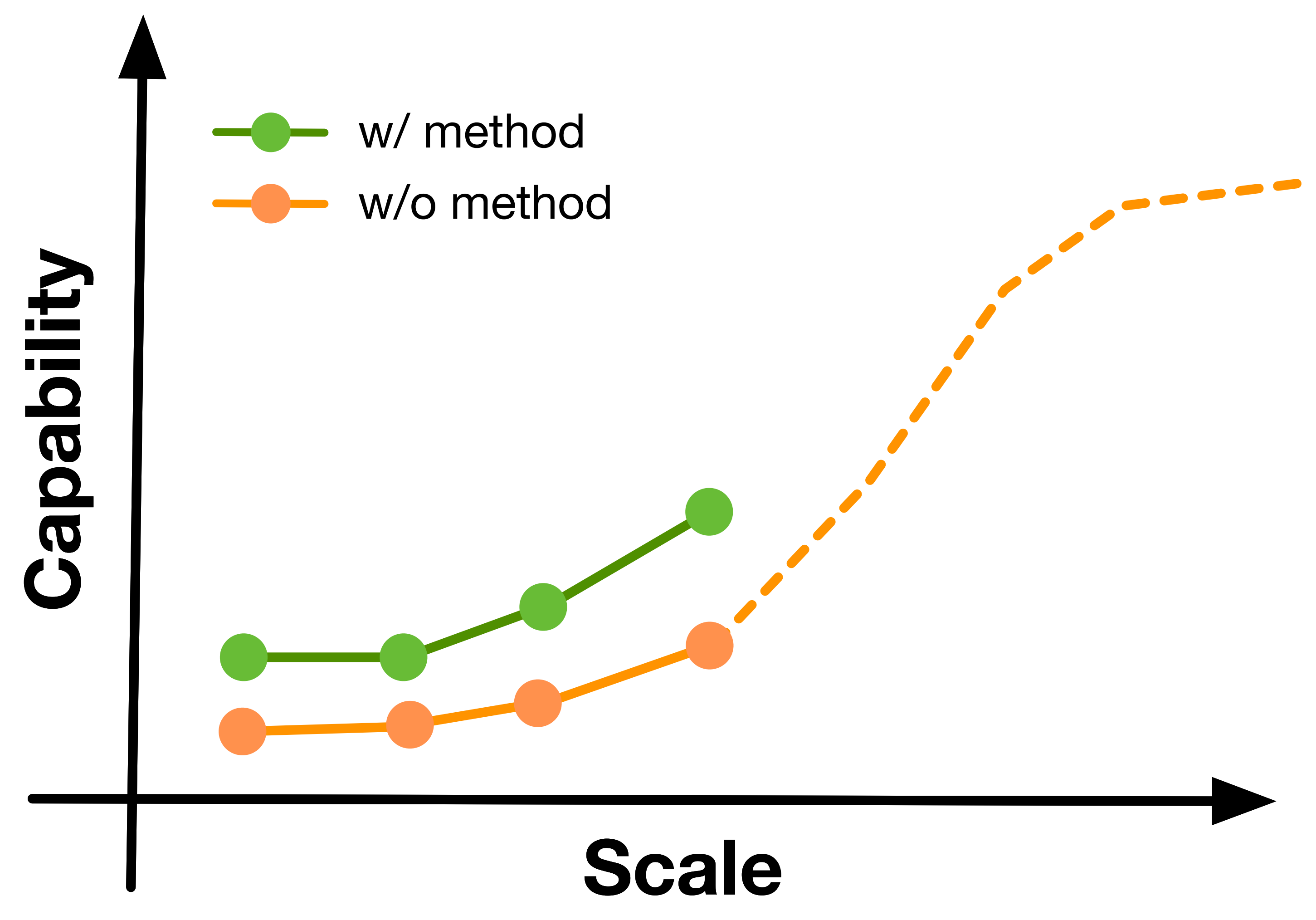
Understanding LM Scaling is Critical

Do our proposed algorithmic interventions stand the test of future scale?



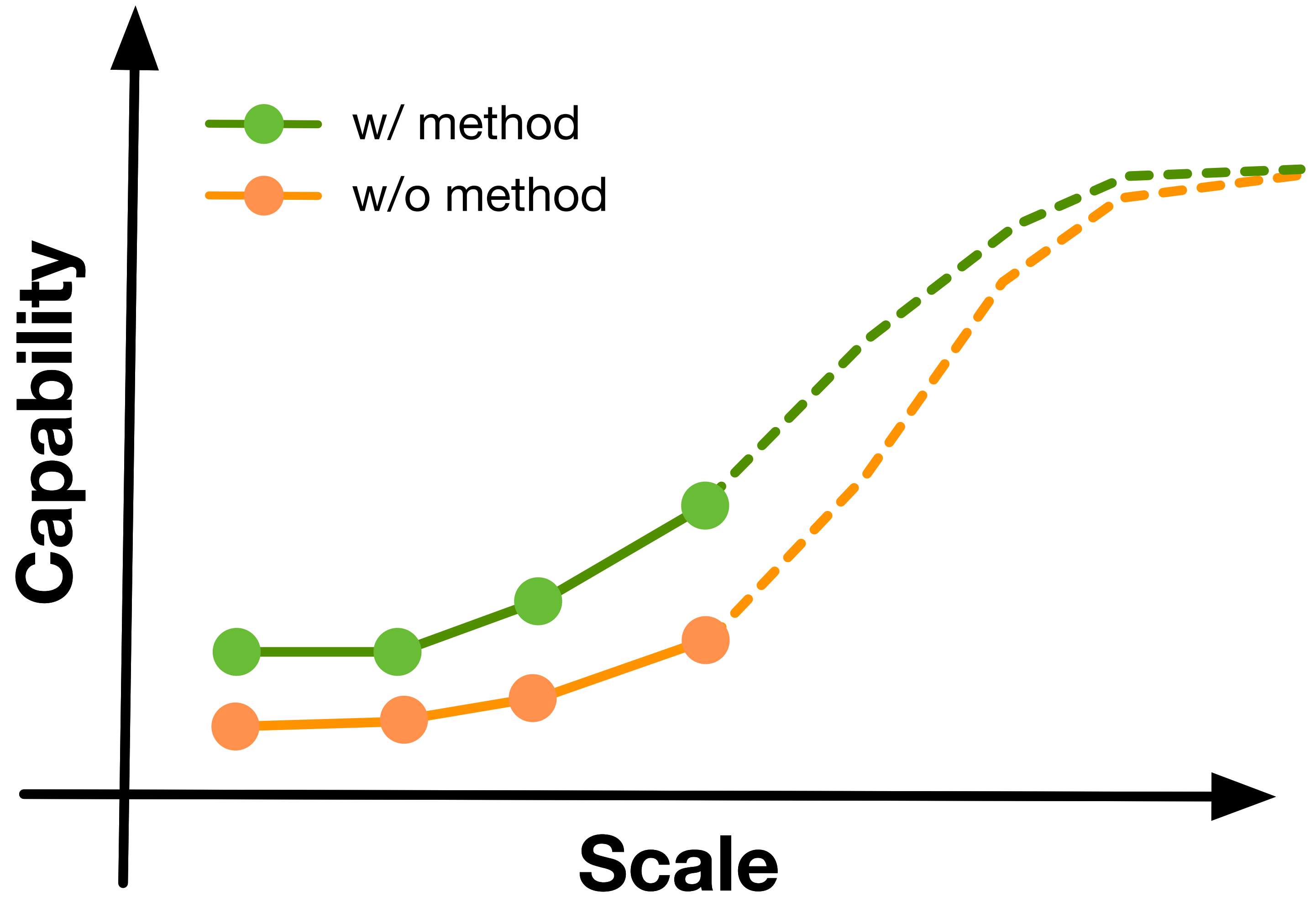
Understanding LM Scaling is Critical

Do our proposed algorithmic interventions stand the test of future scale?



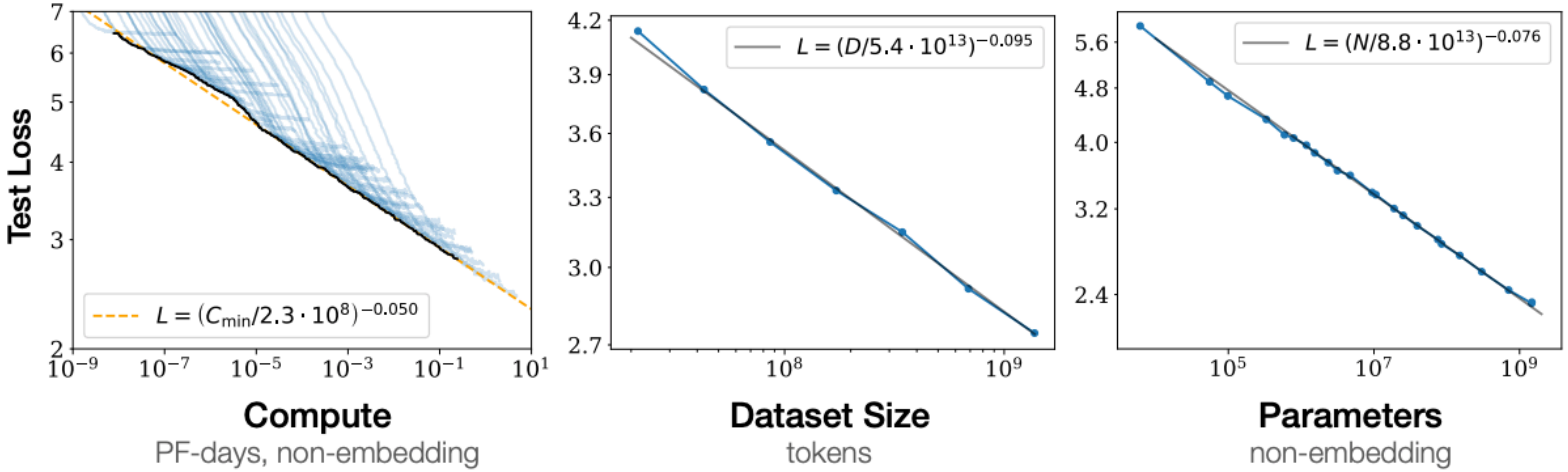
Understanding LM Scaling is Critical

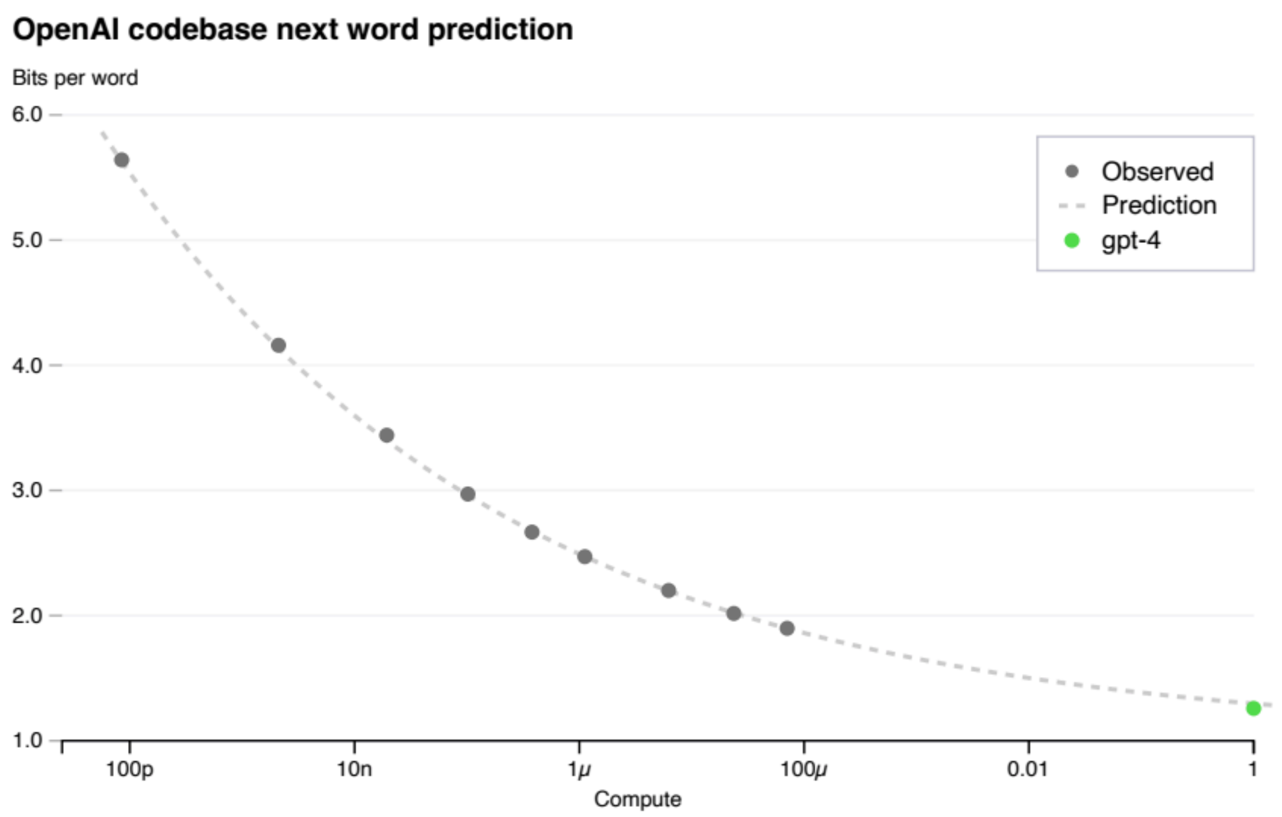
Do our proposed algorithmic interventions stand the test of future scale?



Scaling Laws are the Tools

Scaling laws demonstrate a predictable power-law relationship between LM's performance (e.g., pretraining loss) and compute measures





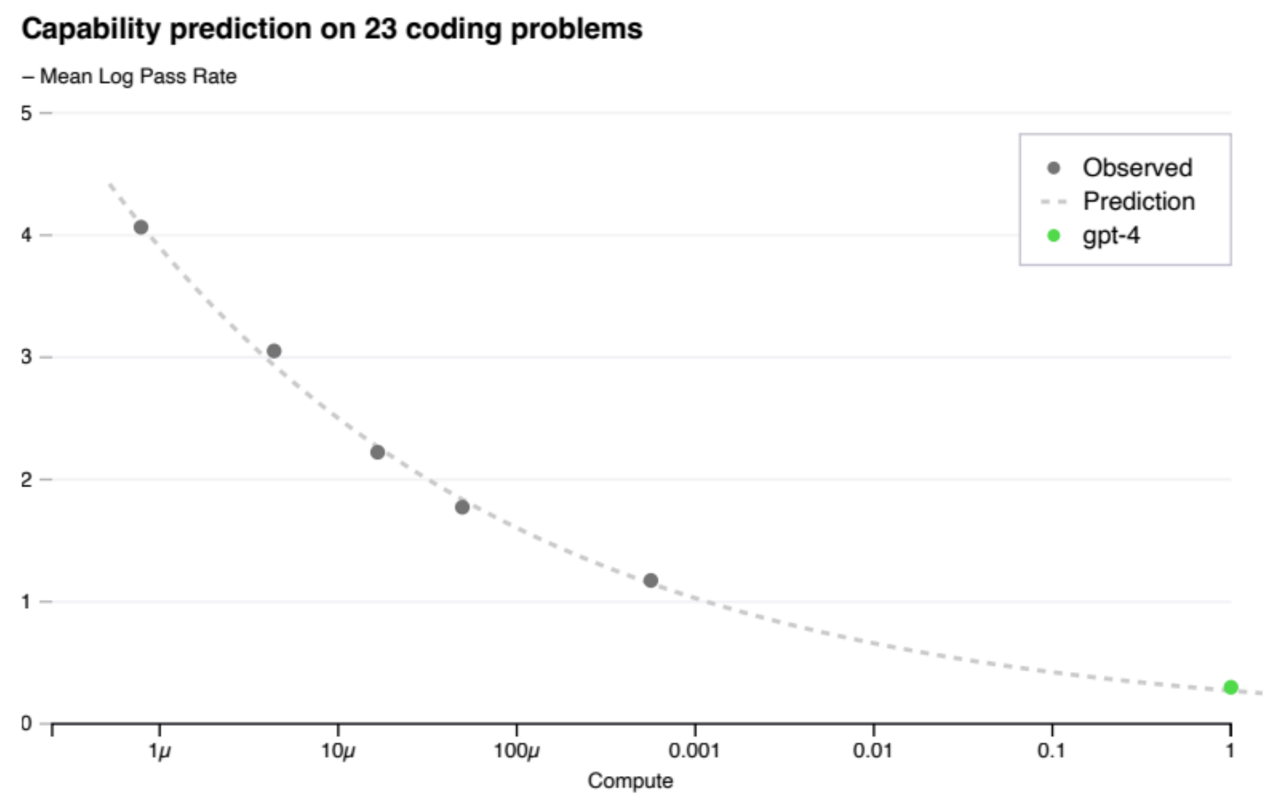
Kaplan et al., 2020. "Scaling Laws for Neural Language Models"

OpenAI, 2023. "GPT-4 Technical Report"

Scaling Laws are the Tools

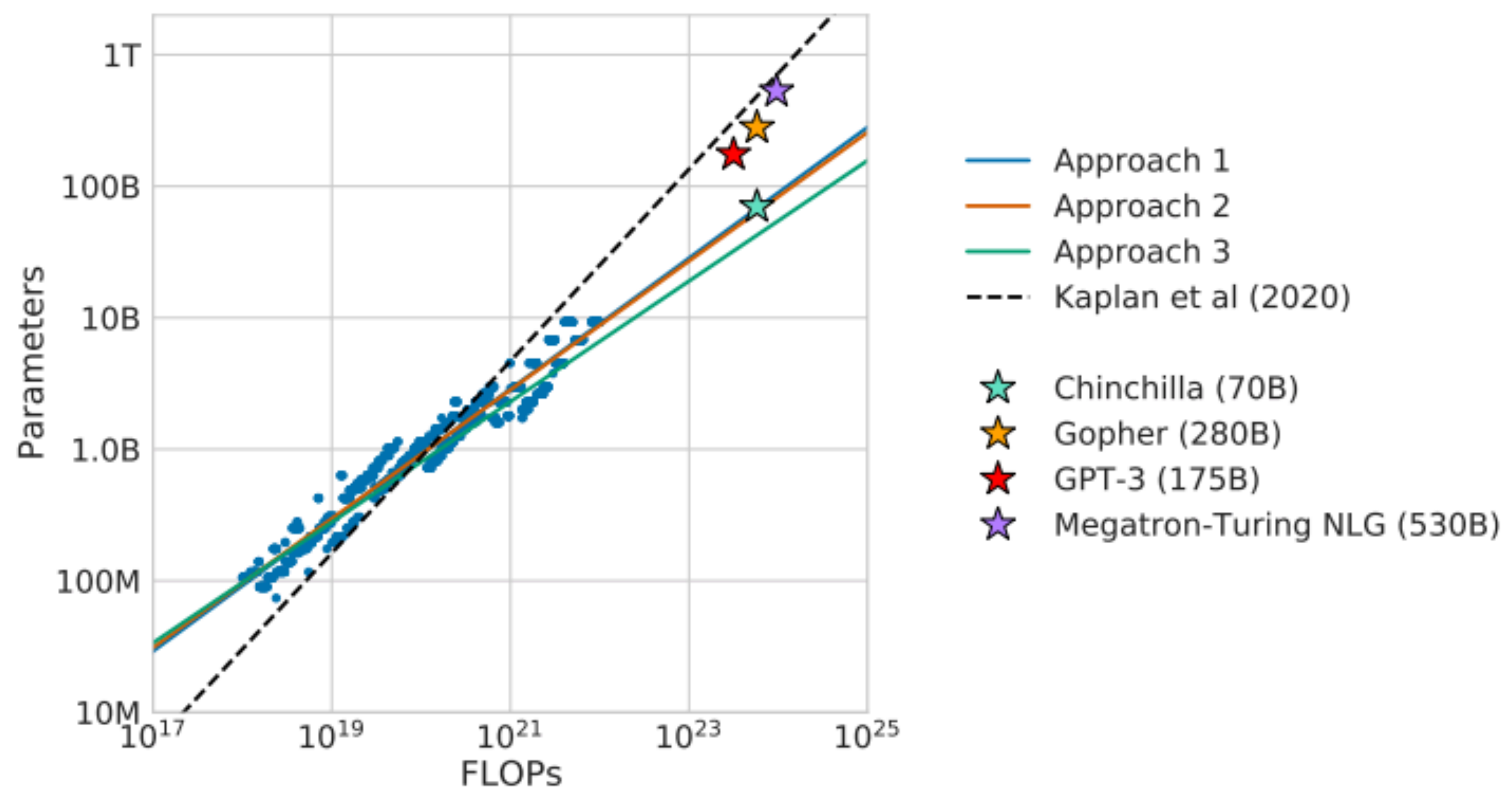
Compute scaling laws have been used in a broad range of applications

Capability prediction



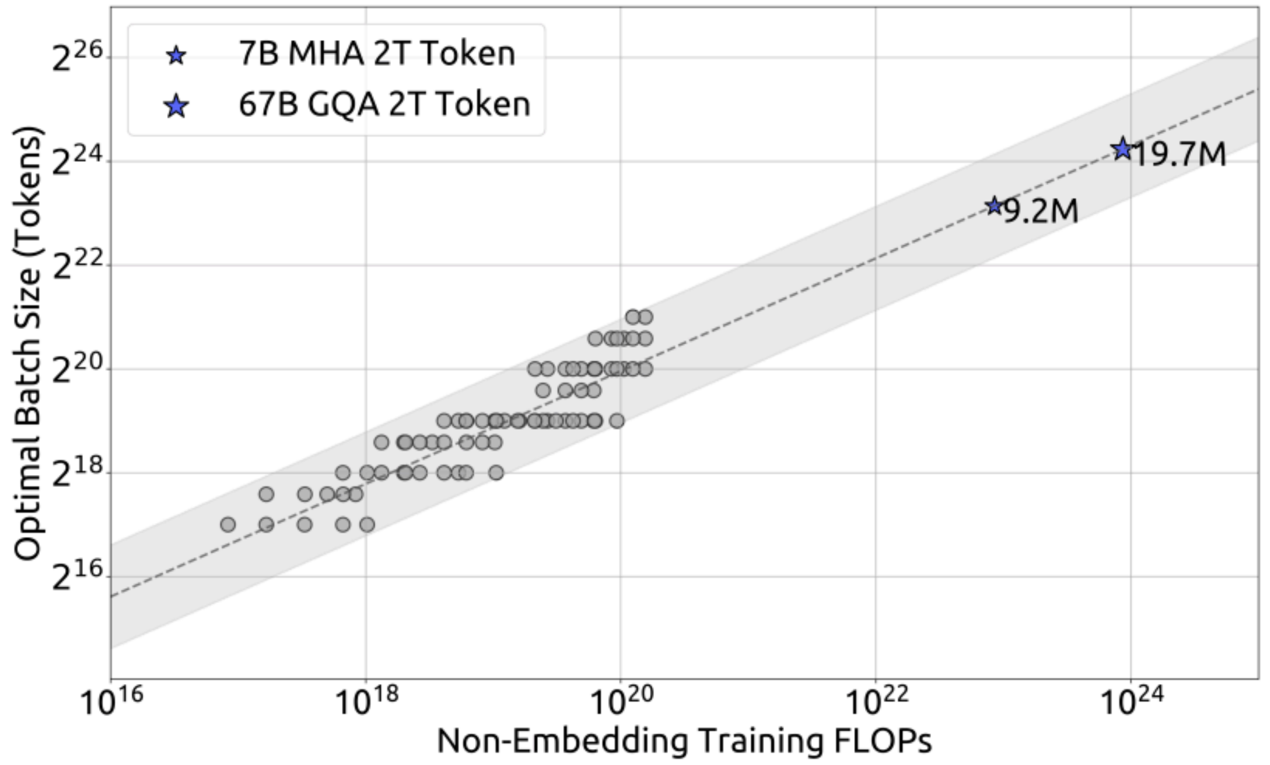
OpenAI, 2023. “GPT-4 Technical Report”

Resource allocation



Hoffmann et al., 2022. “Training Compute-Optimal Large Language Models”

Hyperparameter tuning



Bi et al., 2024. “DeepSeek LLM: Scaling Open-Source Language Models with Longtermism”

But compute scaling analyses remain uncommon in benchmarking or algorithmic studies…

Why?

Limitations of Compute Scaling Analyses

Substantial Cost

Fitting reliable scaling laws requires training a large family of models across scales

We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are significantly under-trained, a consequence of the recent focus on scaling language models whilst keeping the amount of training data constant. By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, we find that for compute-optimal training, the model size and the number of training tokens should be scaled equally: for every doubling of model size the number

Hoffmann et al., 2022. “Training Compute-Optimal Large Language Models”

Limitations of Compute Scaling Analyses

Substantial Cost

Fitting reliable scaling laws requires training a large family of models across scales

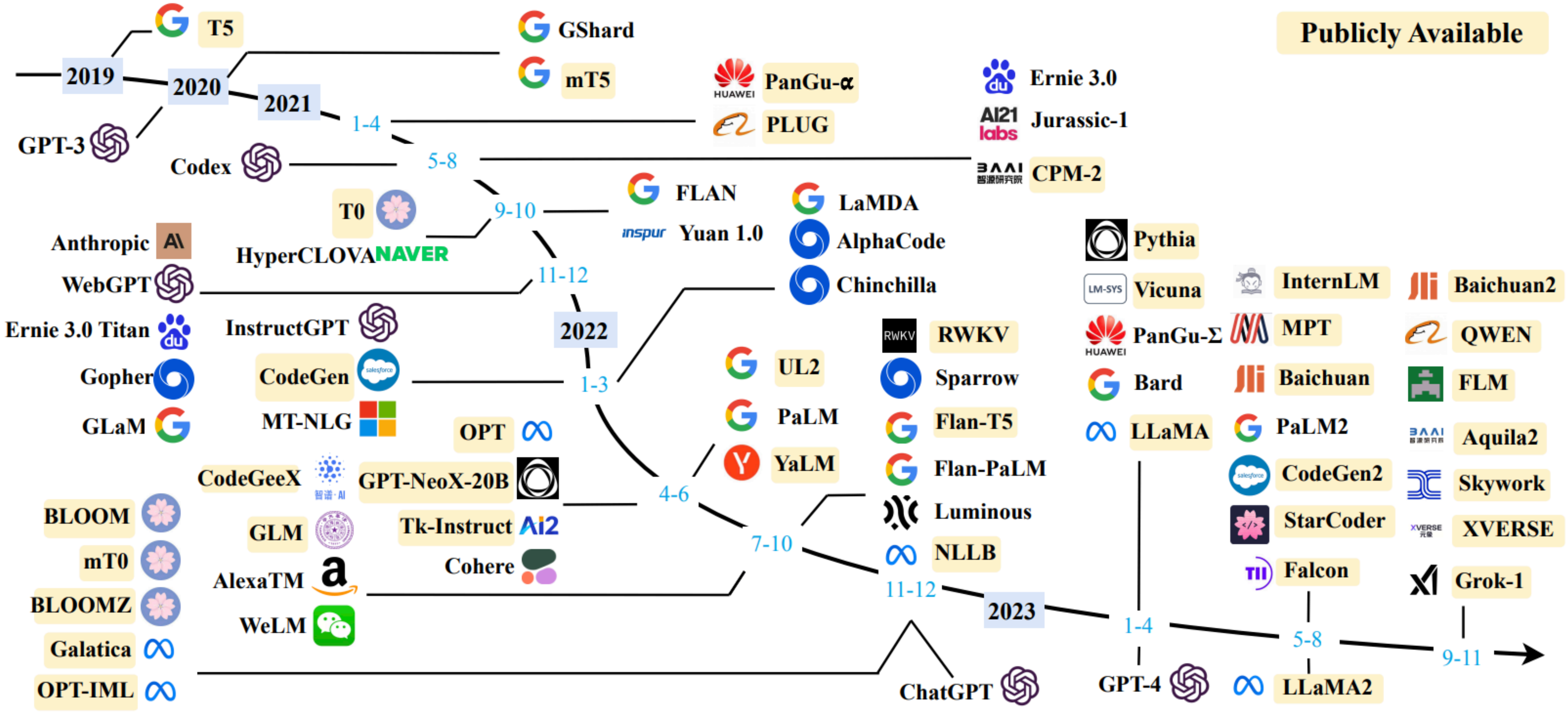
We investigate the optimal model size and number of tokens for training a transformer language model under a given compute budget. We find that current large language models are significantly under-trained, a consequence of the recent focus on scaling language models whilst keeping the amount of training data constant. By training over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens, we find that for compute-optimal training, the model size and the number of training tokens should be scaled equally: for every doubling of model size the number

Hoffmann et al., 2022. “Training Compute-Optimal Large Language Models”

Limitations of Compute Scaling Analyses

Substantial Cost

What if we use existing, public models?

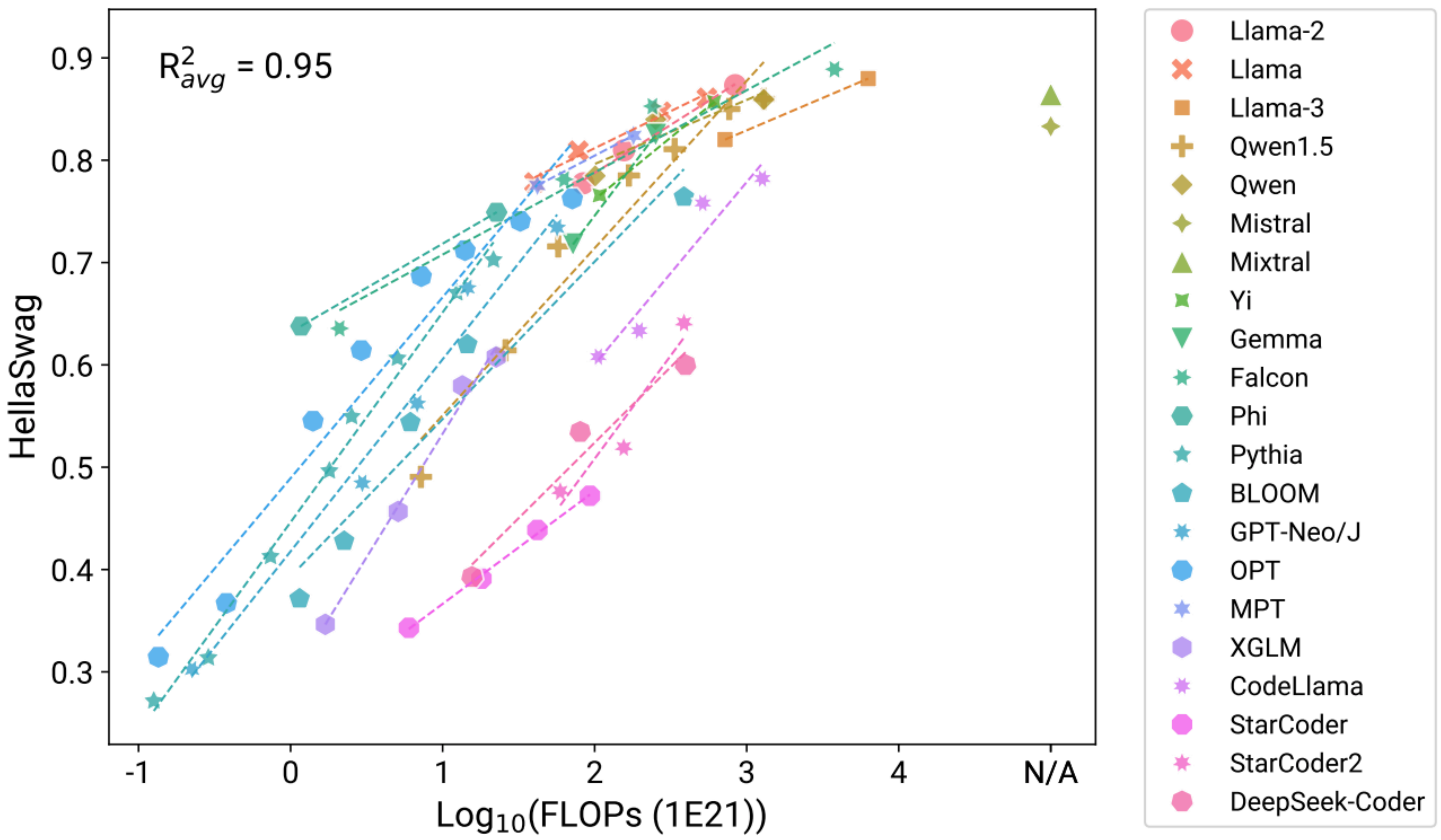


Zhao et al., 2023. “A Survey of Large Language Models”

Limitations of Compute Scaling Analyses

Restricted Coverage

Different model families (trained with heterogenous recipes) demonstrate varying compute efficiencies



Limitations of Compute Scaling Analyses

Restricted Coverage

Compute scaling laws need to be established with a carefully controlled training recipe (e.g., model arch., data dist.)

Approach	Coeff. a where $N_{\text{opt}}(M_{\text{opt}}) \propto C^a$	Coeff. b where $D_{\text{opt}} \propto C^b$
OpenAI (OpenWebText2)	0.73	0.27
Chinchilla (MassiveText)	0.49	0.51
Ours (Early Data)	0.450	0.550
Ours (Current Data)	0.524	0.476
Ours (OpenWebText2)	0.578	0.422

Bi et al., 2024. “DeepSeek LLM Scaling Open-Source Language Models with Longtermism”

Inspiration

There are a lot of standard, unified evaluation benchmarks that measure various base capabilities of LMs

🤗 Open LLM Leaderboard

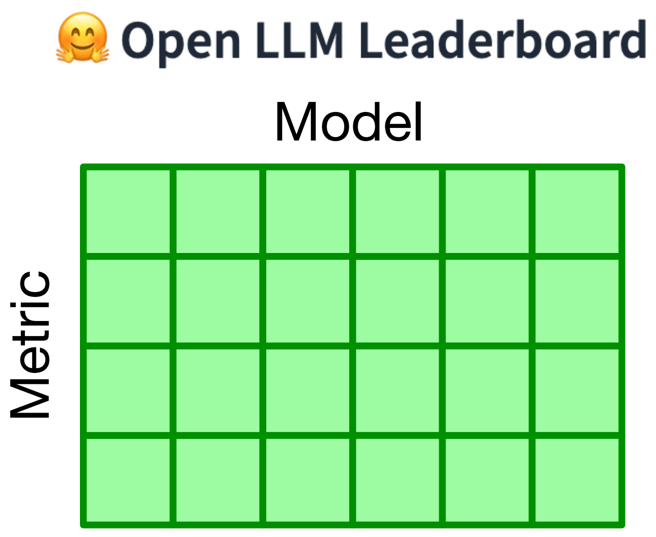
Model	Revision	Average	ARC (25-shot)	HellaSwag (10-shot)	MMLU (5-shot)	T
llama-65b	main	58.3	57.8	84.2	48.8	4
llama-30b	main	56.9	57.1	82.6	45.7	4
stable-vicuna-13b	main	52.4	48.1	76.4	38.8	4
llama-13b	main	51.8	50.8	78.9	37.7	3
alpaca-13b	main	51.7	51.9	77.6	37.6	3
llama-7b	main	47.6	46.6	75.6	34.2	3
EleutherAI/gpt-neox-20b	main	45.9	45.2	73.4	33.3	3
togethercomputer/RedPajama-INCITE-Base-7B-v0.1	main	45.7	44.4	71.3	34	3
togethercomputer/RedPajama-INCITE-Base-3B-v1	main	42.2	40.2	64.7	30.6	3
Salesforce/codegen-16B-multi	main	39.2	33.6	51.2	28.9	4
facebook/opt-1.3b	main	37.7	29.6	54.6	27.7	3
facebook/opt-350m	main	32.2	23.6	36.7	27.3	4
facebook/opt-125m	main	31.2	23.1	31.5	27.4	4
gpt2	main	30.4	21.9	31.6	27.5	4

Observational Scaling Laws

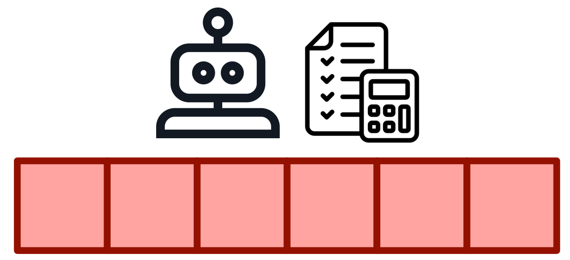
Idea: use observable, base capability measures as the surrogate, unified “scale”

Observational Scaling Laws

Idea: use observable, base capability measures as the surrogate, unified "scale"



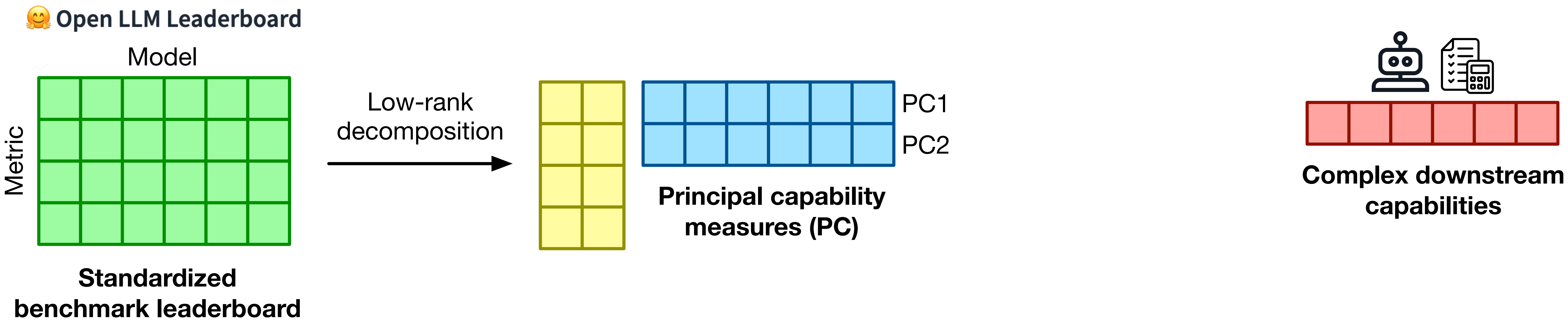
Standardized benchmark leaderboard



Complex downstream capabilities

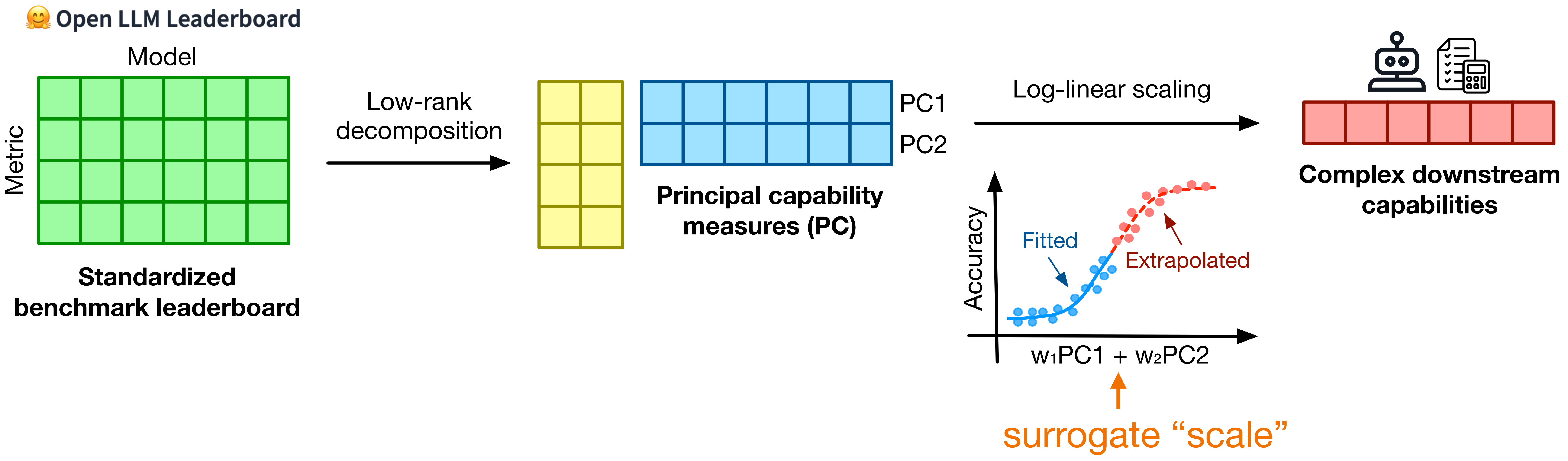
Observational Scaling Laws

Idea: use observable, base capability measures as the surrogate, unified "scale"



Observational Scaling Laws

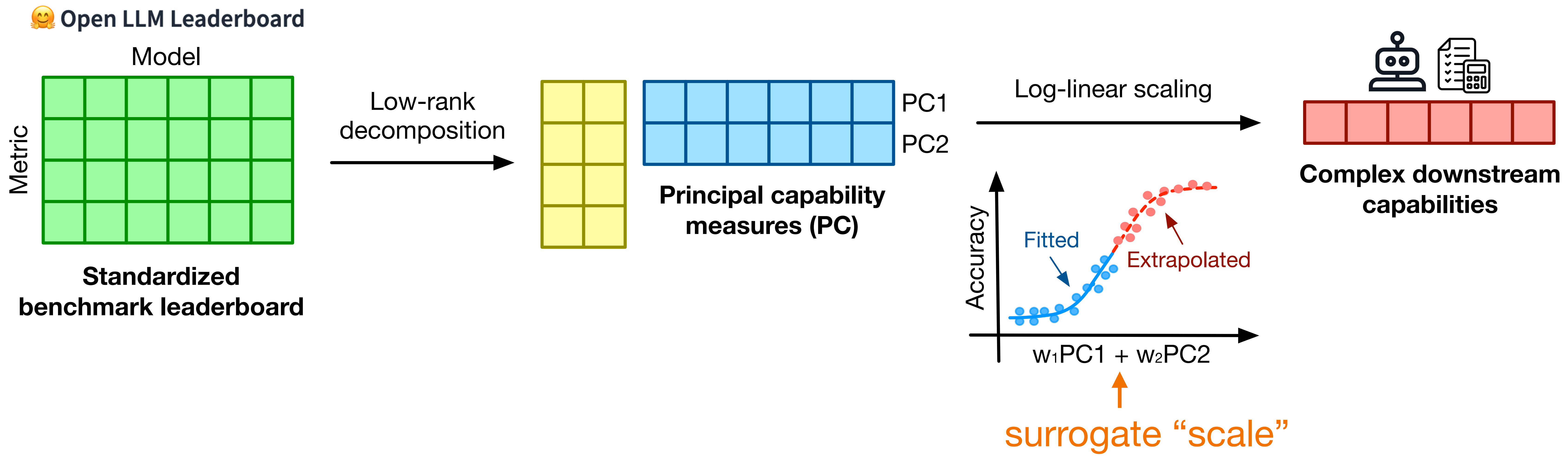
Idea: use observable, base capability measures as the surrogate, unified “scale”



Observational Scaling Laws

Idea: use observable, base capability measures as the surrogate, unified “scale”

✔ Low cost: no training required

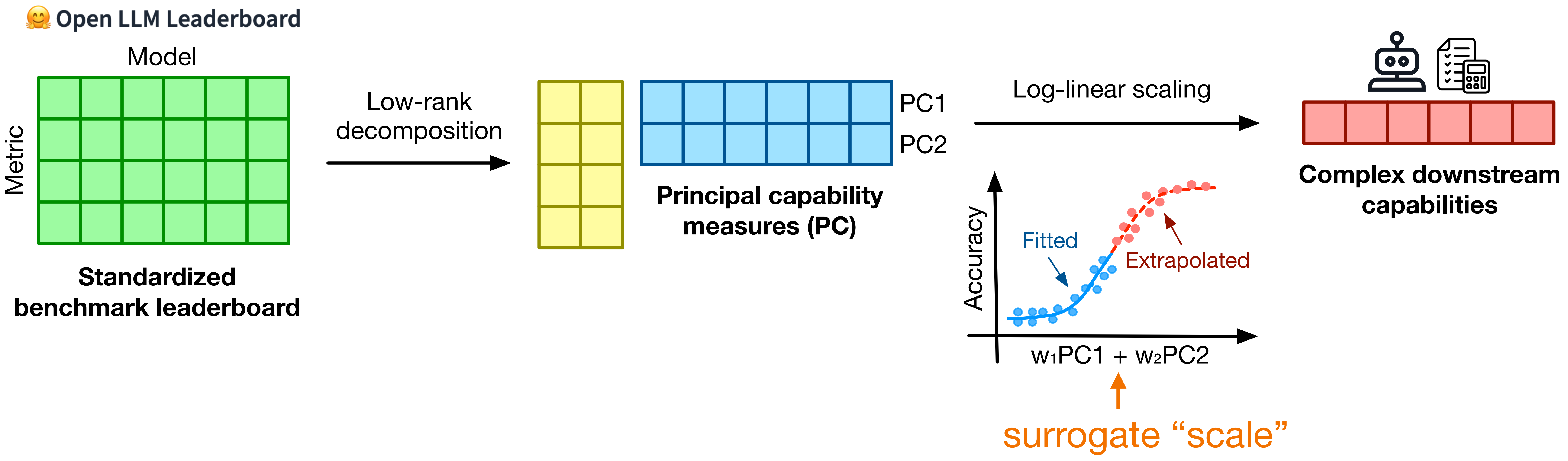


Observational Scaling Laws

Idea: use observable, base capability measures as the surrogate, unified “scale”

✔ Low cost: no training required

✔ High resolution: leveraging public models



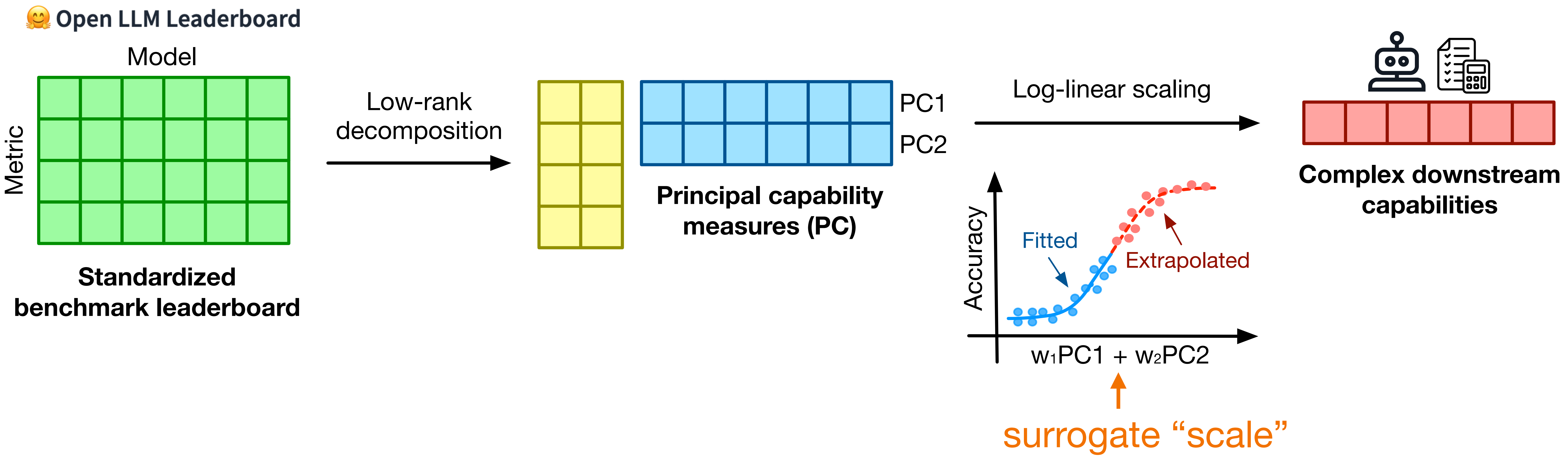
Observational Scaling Laws

Idea: use observable, base capability measures as the surrogate, unified “scale”

✔ Low cost: no training required

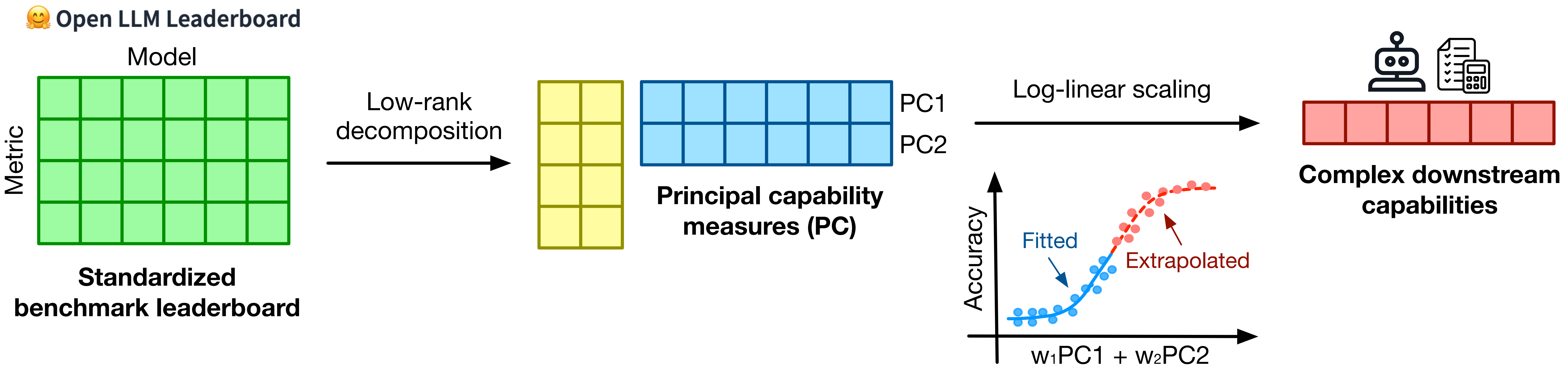
✔ High resolution: leveraging public models

✔ Broad coverage: covering different families



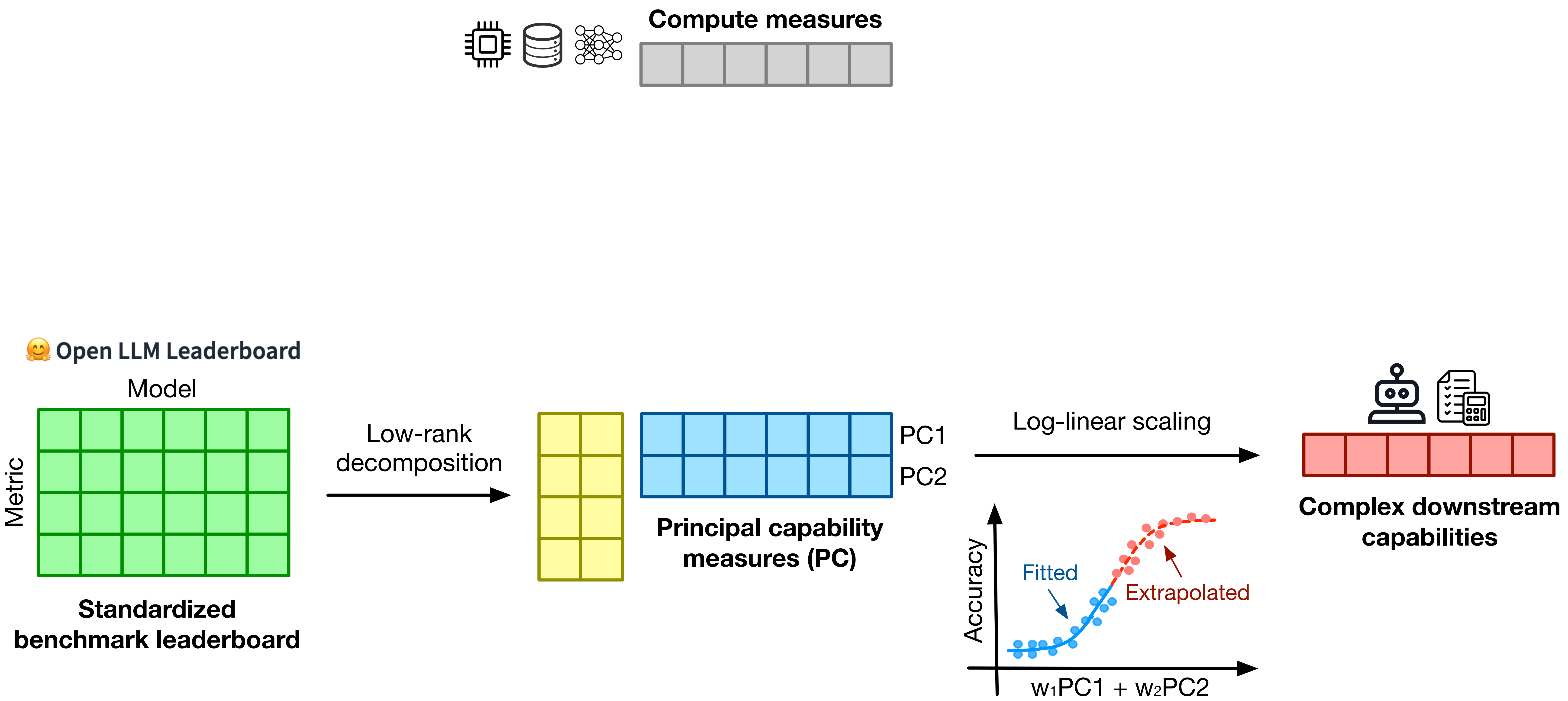
Observational Scaling Laws

Observational scaling laws generalize compute scaling laws



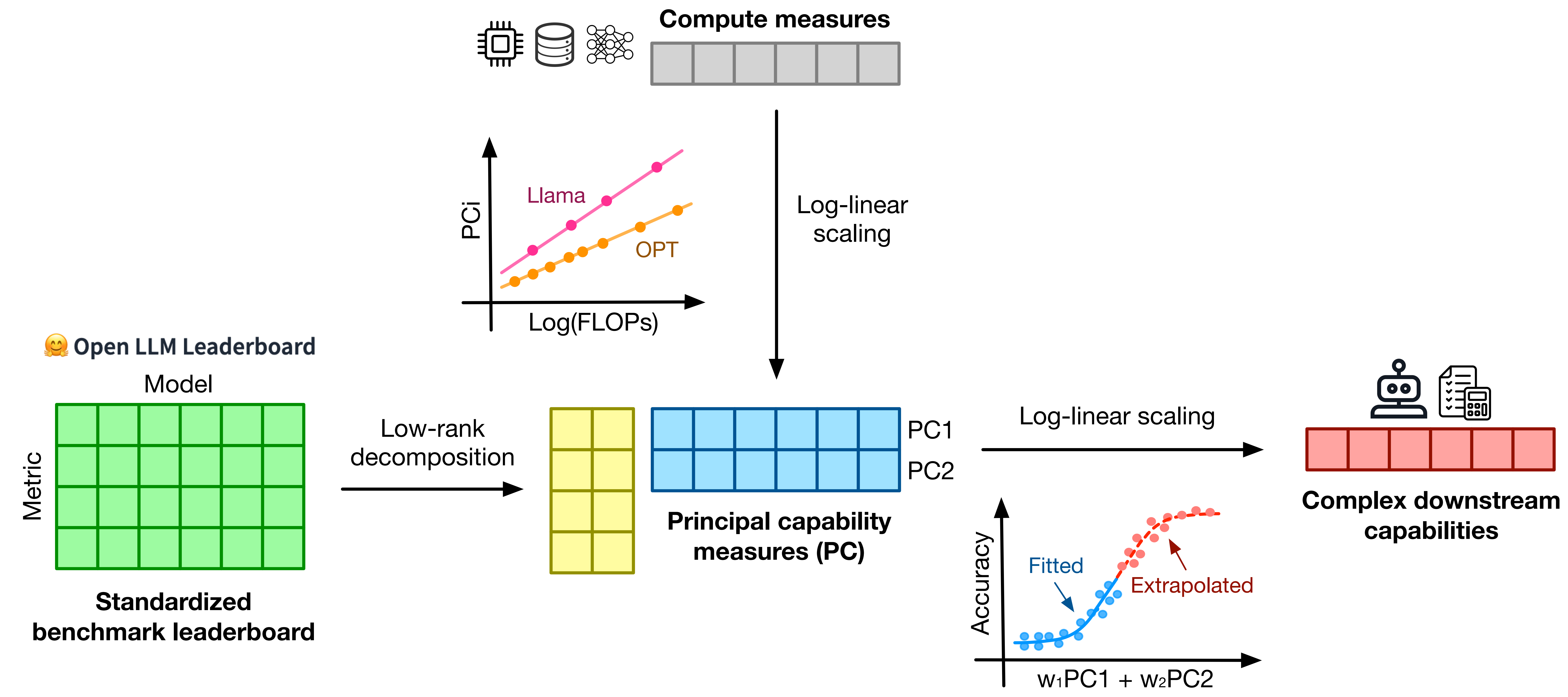
Observational Scaling Laws

Observational scaling laws generalize compute scaling laws



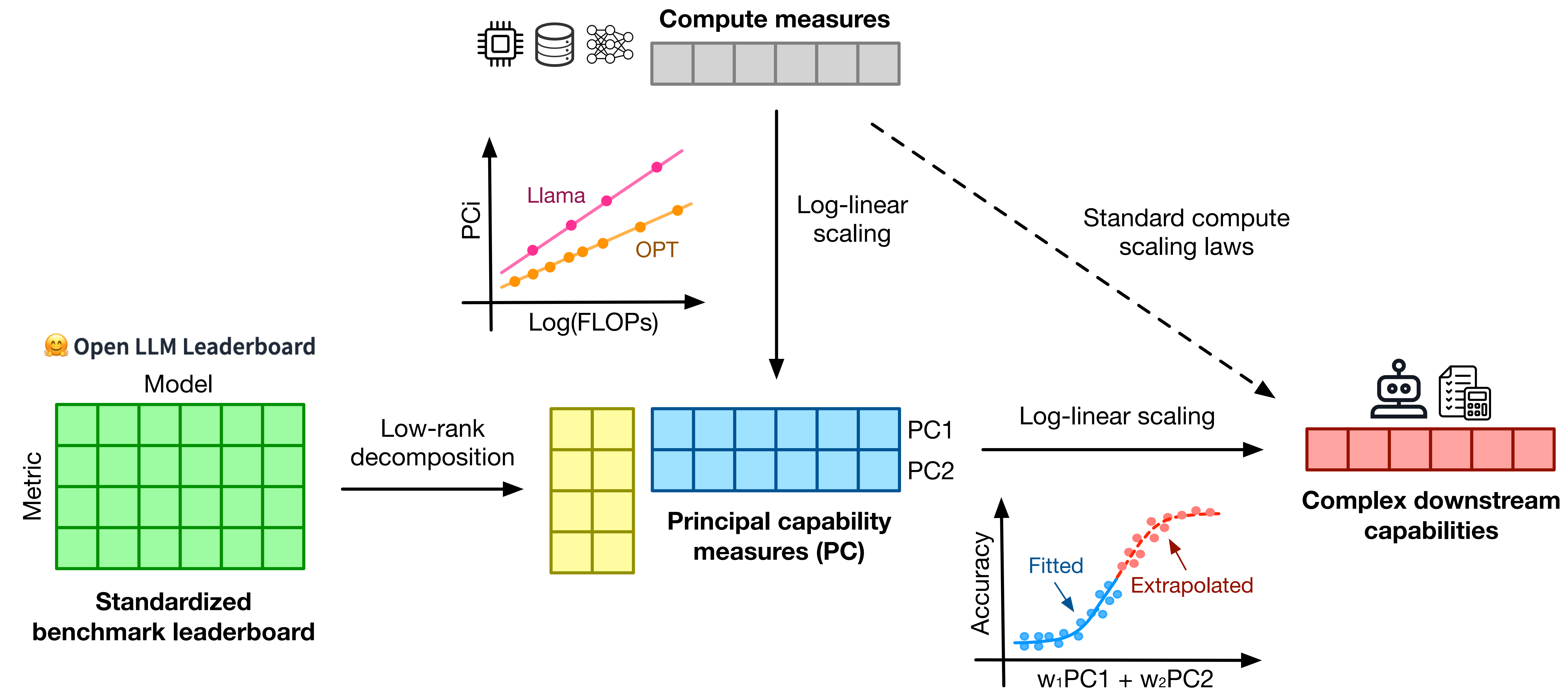
Observational Scaling Laws

Observational scaling laws generalize compute scaling laws



Observational Scaling Laws

Observational scaling laws generalize compute scaling laws



Extracting Principal Capability Measures

Extracting Principal Capability Measures

100+ Public, Heterogenous Pretrained Models

- **Standard**: Llama, Gemma, …
- **Code**: CodeLlama, StarCoder, …
- **Multilingual**: BLOOM, XGLM, …
- **Synthetic**: Phi
- **MoE**: Mixtral, DeepSeek-V2, …
- **Mamba-Hybrid**: Jamba

Extracting Principal Capability Measures

100+ Public, Heterogenous Pretrained Models

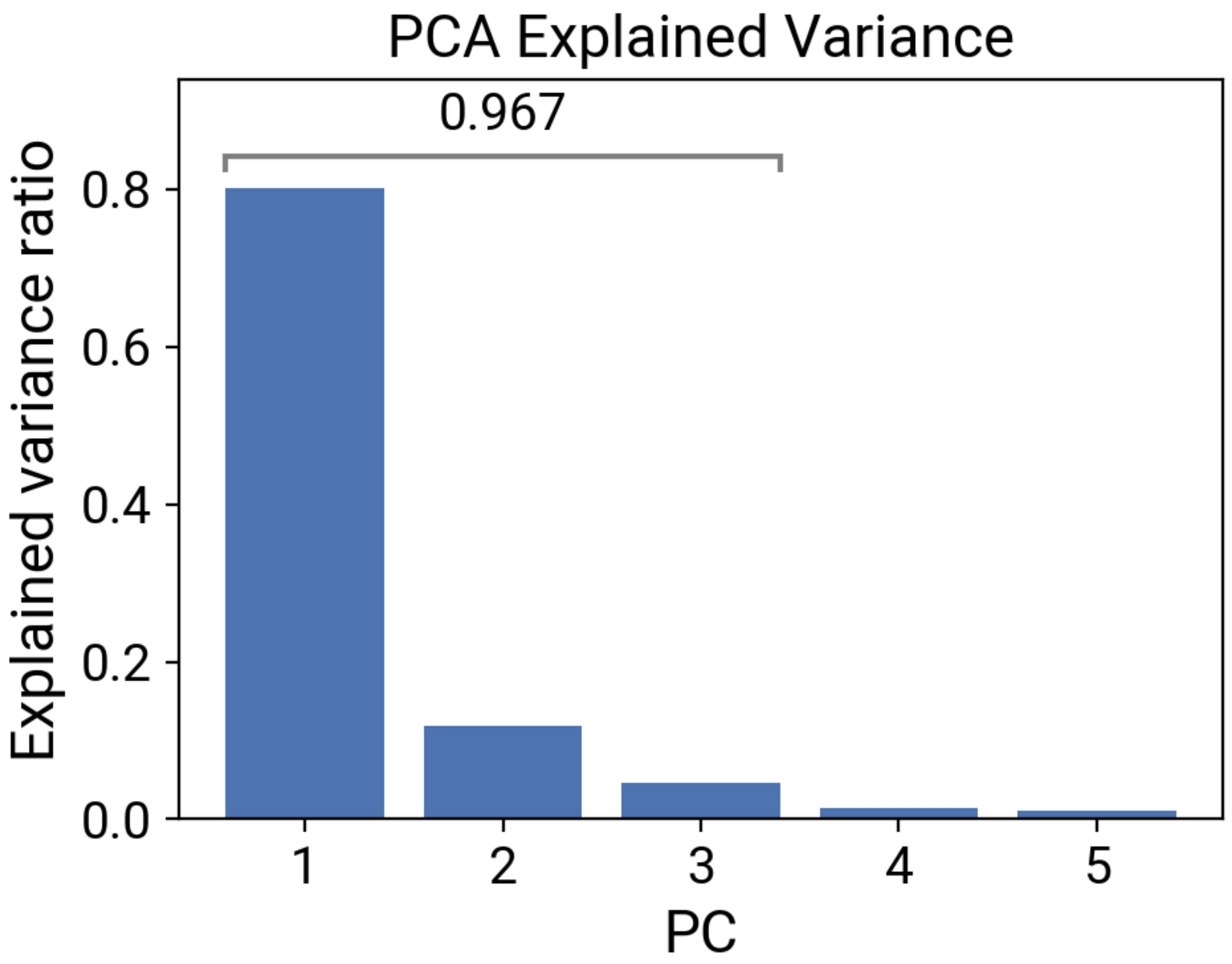
- **Standard:** Llama, Gemma, …
- **Code:** CodeLlama, StarCoder, …
- **Multilingual:** BLOOM, XGLM, …
- **Synthetic:** Phi
- **MoE:** Mixtral, DeepSeek-V2, …
- **Mamba-Hybrid:** Jamba

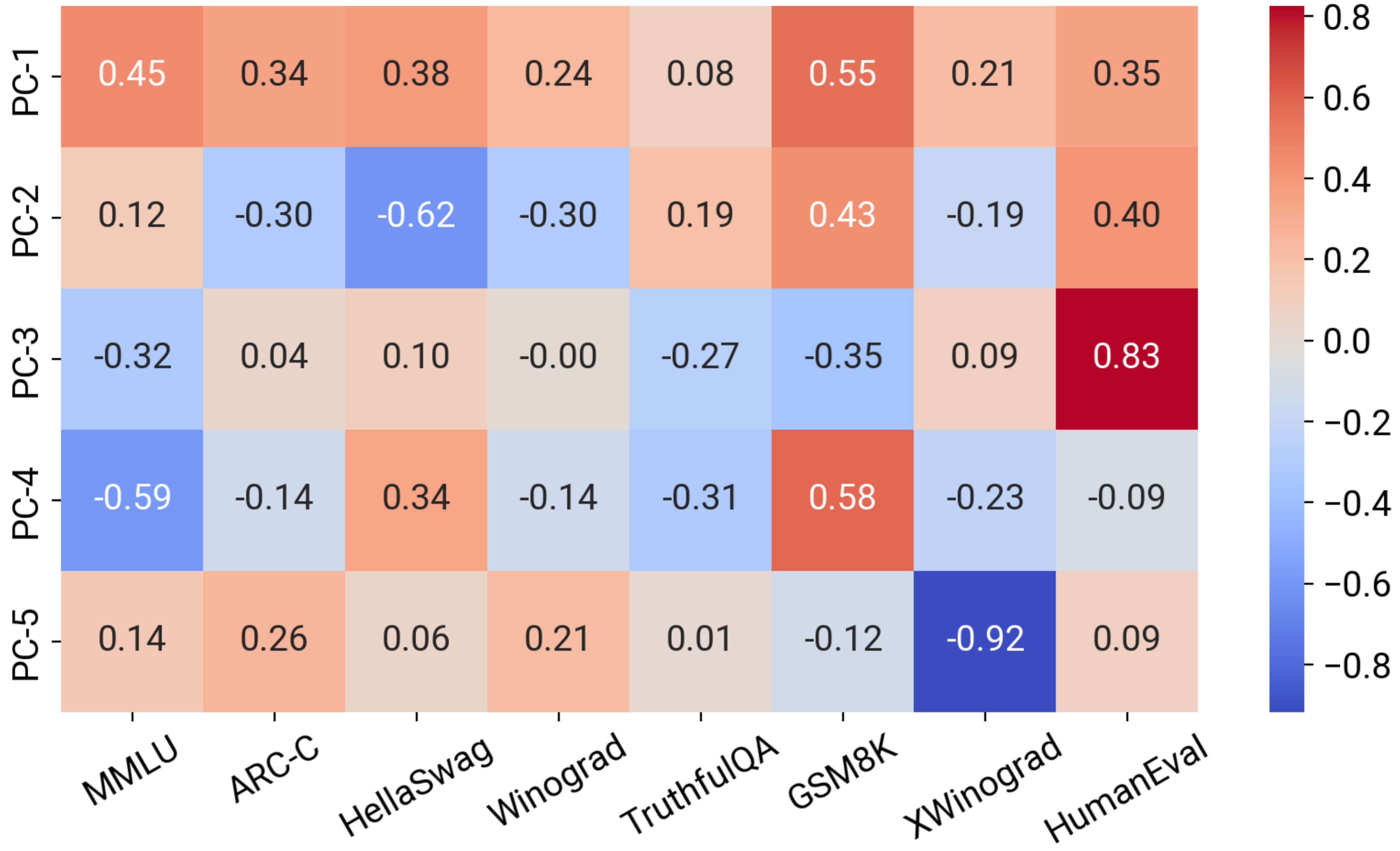
Diverse Metrics from Standardized Benchmarks

- **Aggregated:** MMLU
- **Commonsense:** ARC-C, HellaSwag, Winogrande
- **Math:** GSM8K
- **Code:** HumanEval
- **Truthfulness:** TruthfulQA
- **Multilinguality:** XWinograd

Extracting Principal Capability Measures

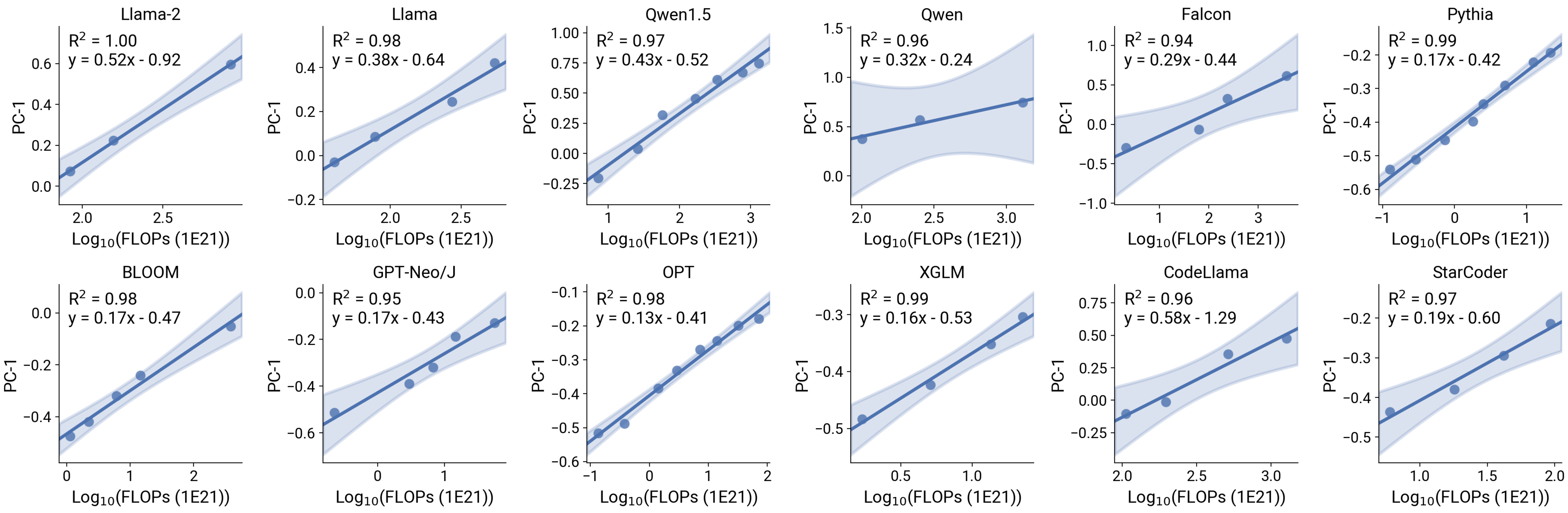
PC measures are **low-dimensional** and **interpretable** (to some extent)





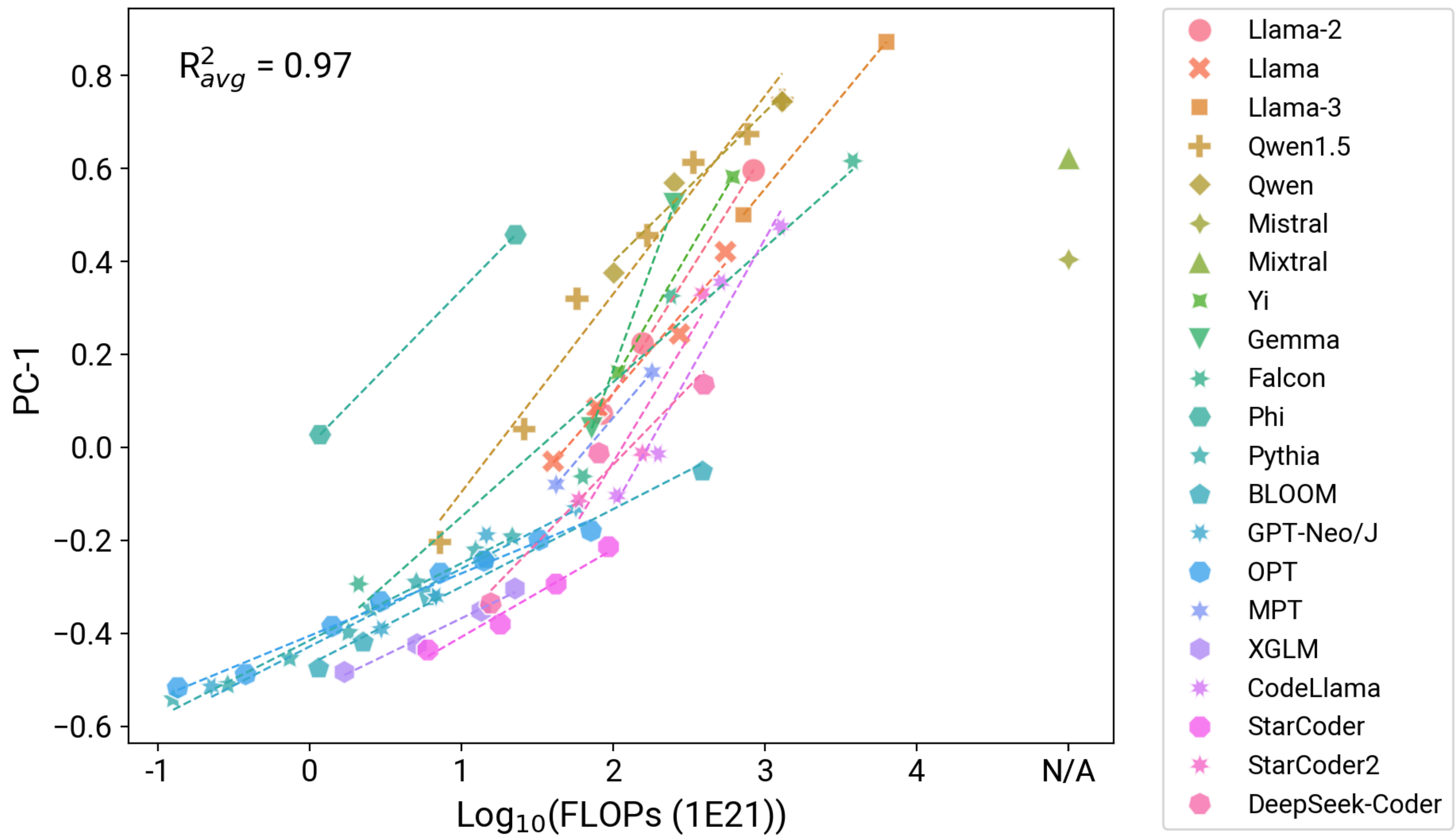
PC Measures as Surrogate Scale

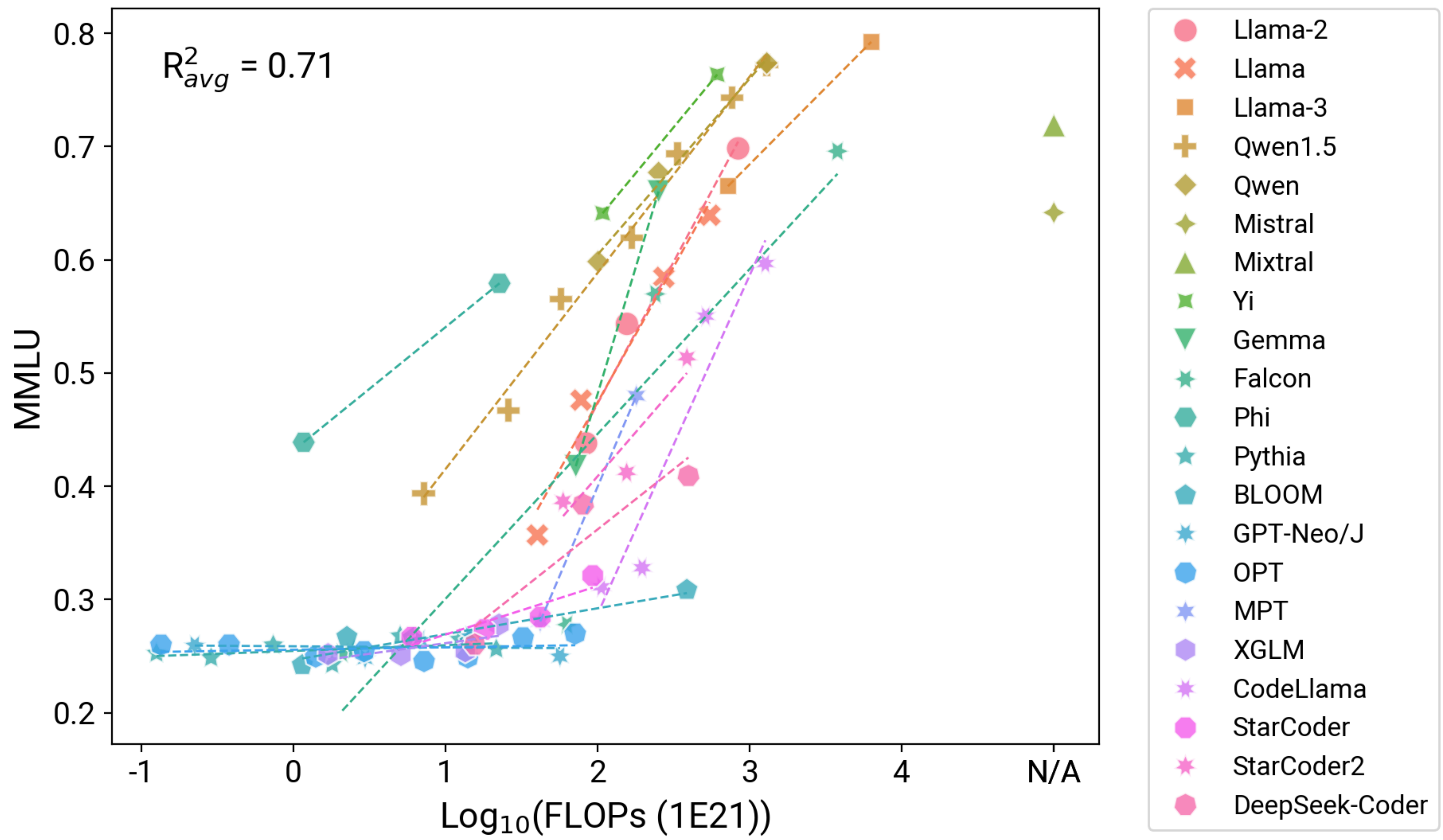
PC measures **linearly correlate with log-compute** within each model family



PC Measures as Surrogate Scale

PC measures provide a **smooth** and **unified** capability measure for models from heterogeneous sources





Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

✔ Complex model capabilities (e.g., agentic or “emergent” behaviours)

Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

- ✔ Complex model capabilities (e.g., agentic or “emergent” behaviours)
- ✔ Post-training techniques

Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

✔ Complex model capabilities (e.g., agentic or “emergent” behaviours)

✔ Post-training techniques

✘ Pretraining algorithmic dev

Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

✔ Complex model capabilities (e.g., agentic or “emergent” behaviours)

✔ Post-training techniques

✘ Pretraining algorithmic dev

Validation: measure how well fitted scaling laws extrapolate from smaller-scale, weaker models to larger-scale, stronger models

Observational Scaling Analyses

Observational scaling laws are applicable to many types of scaling analyses

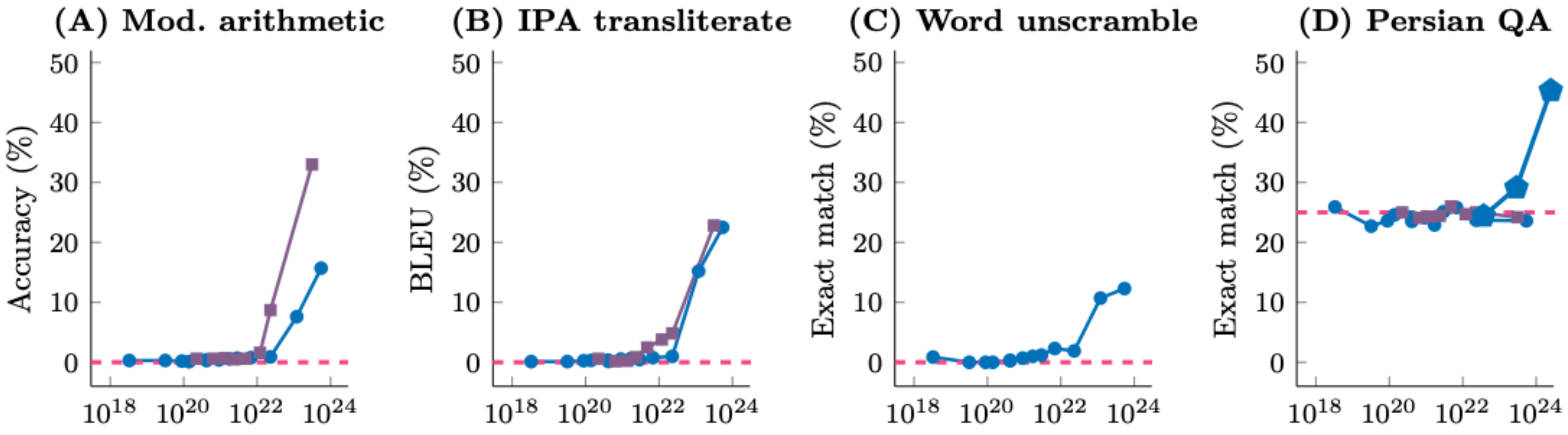
- ✔ Complex model capabilities (e.g., agentic or “emergent” behaviours)
- ✔ Post-training techniques
- ✘ Pretraining algorithmic dev

Validation: measure how well fitted scaling laws extrapolate from smaller-scale, weaker models to larger-scale, stronger models

Preregistration: test on newly released models after the paper release (05/2024)

Predictability of “Emergent” Capabilities

There have been ongoing debates about whether “emergent” capabilities are truly discontinuous or inherently smooth

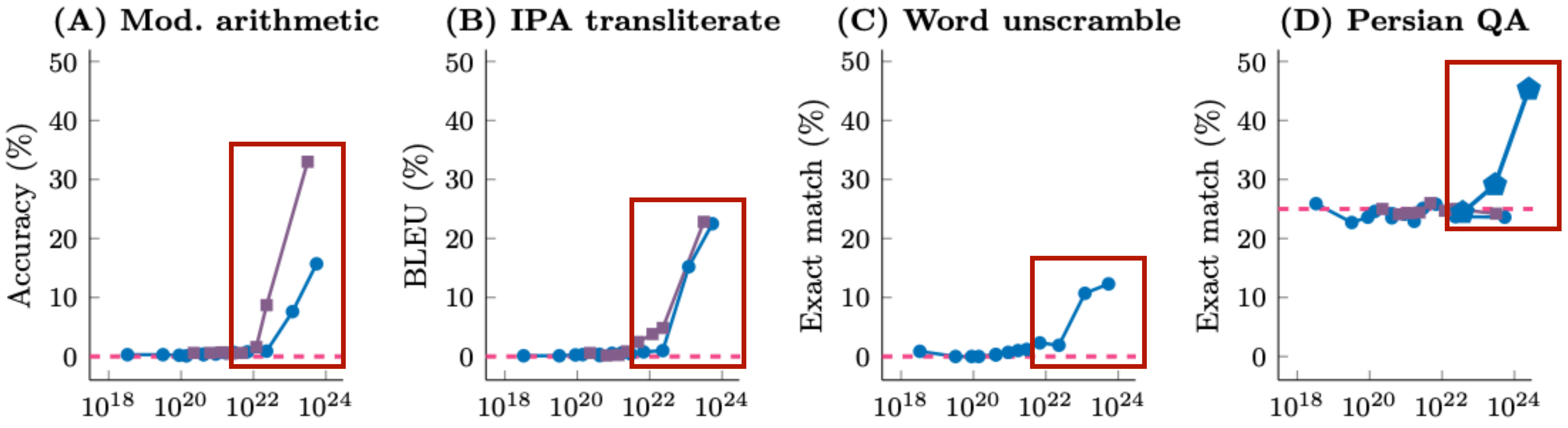


Wei et al., 2022. “Emergent Abilities of Large Language Models”

Predictability of "Emergent" Capabilities

There have been ongoing debates about whether "emergent" capabilities are truly discontinuous or inherently smooth

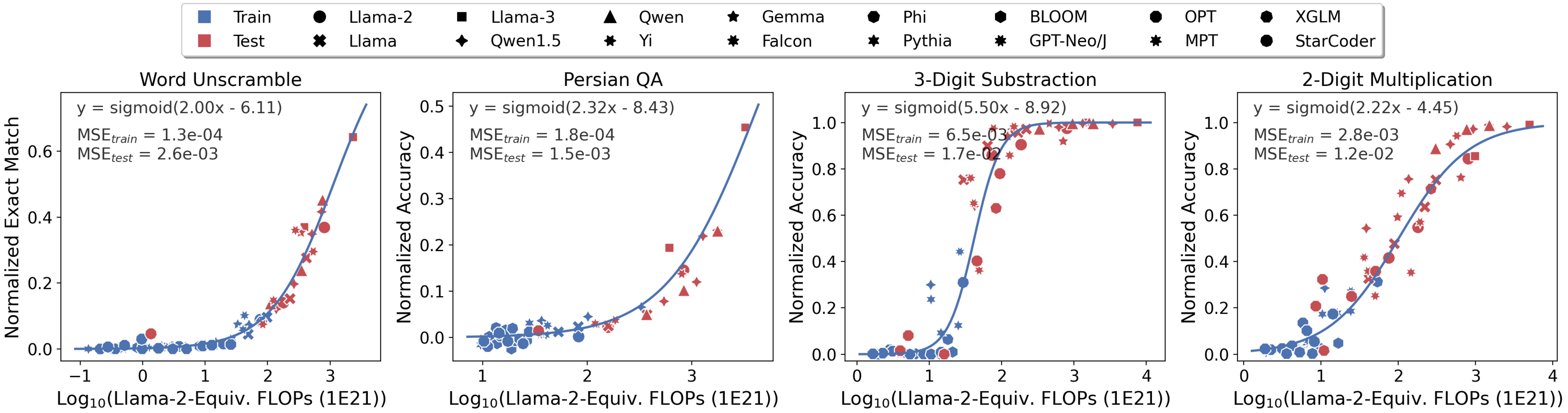
"Emergence" could be an artifact of low-resolution data points?



Wei et al., 2022. "Emergent Abilities of Large Language Models"

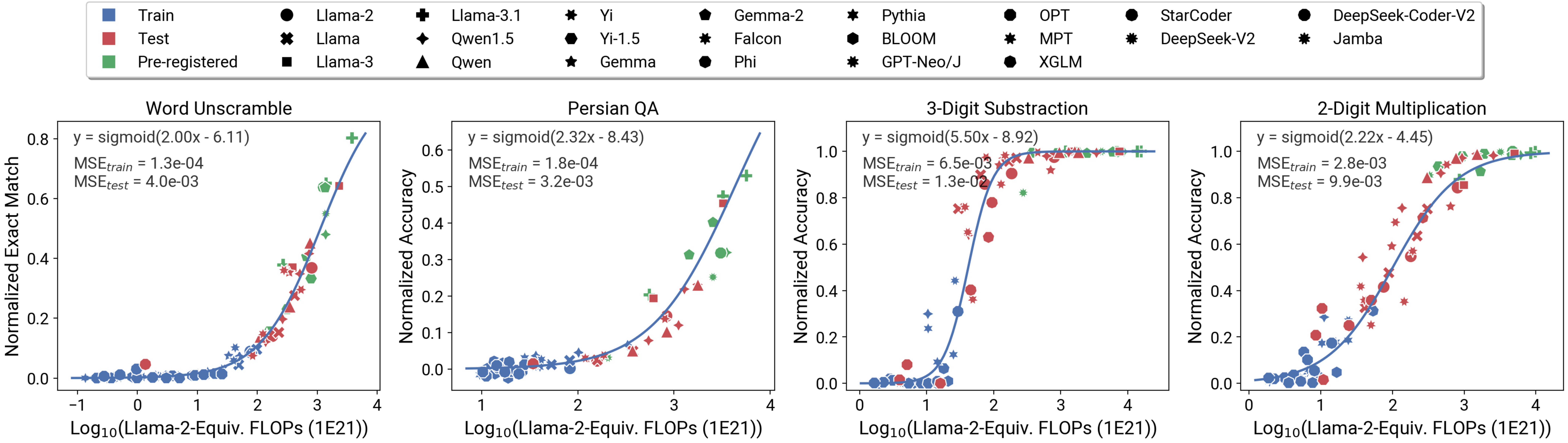
Predictability of "Emergent" Capabilities

Emergent capabilities can be accurately predicted with obs. scaling laws



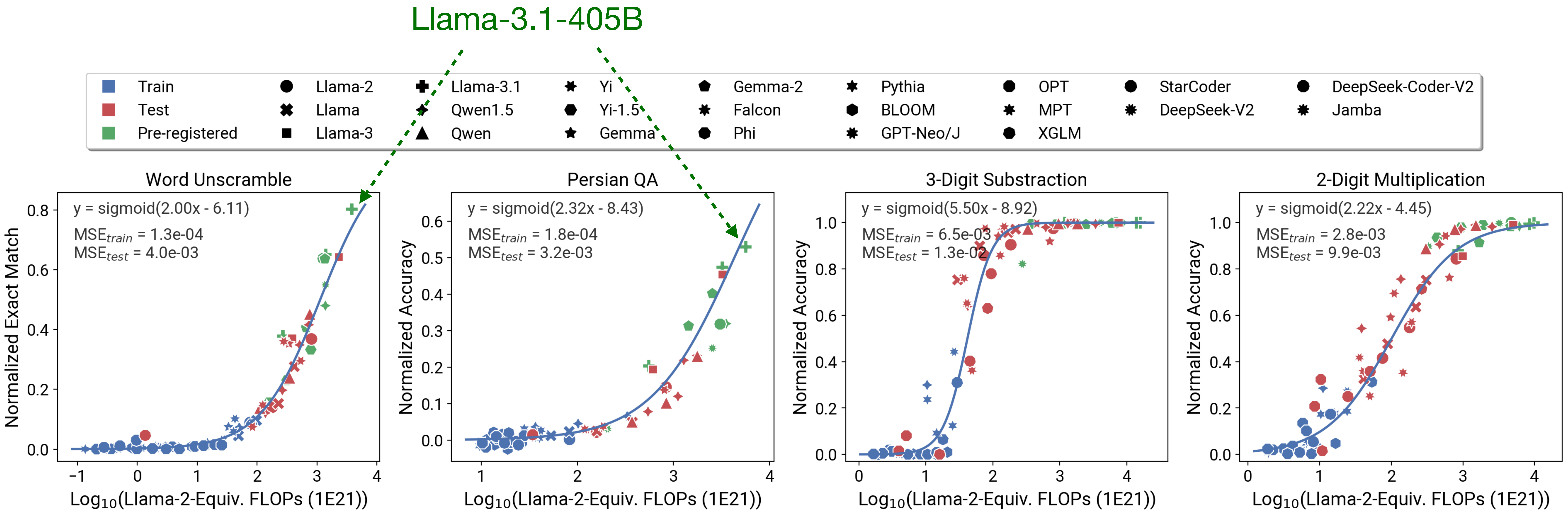
Predictability of “Emergent” Capabilities

Emergent capabilities can be accurately predicted with obs. scaling laws



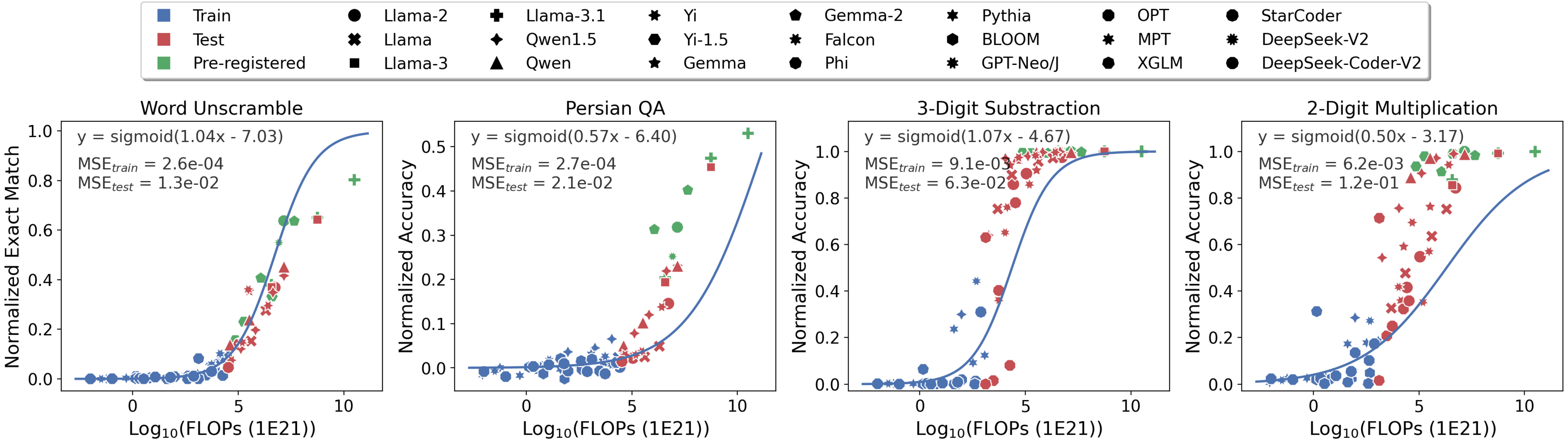
Predictability of “Emergent” Capabilities

Emergent capabilities can be accurately predicted with obs. scaling laws



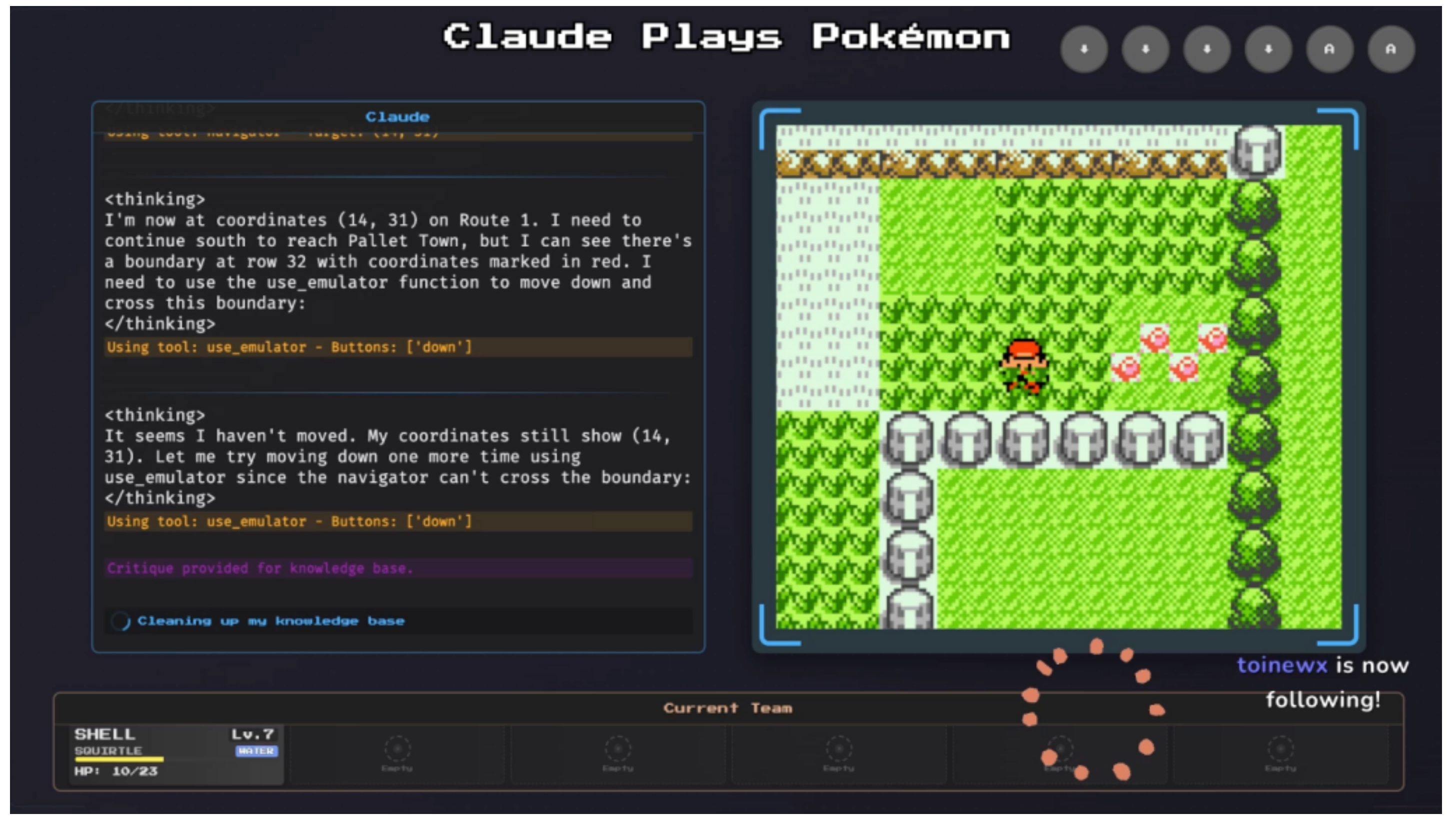
Predictability of "Emergent" Capabilities

Compute scaling laws provide poor extrapolations



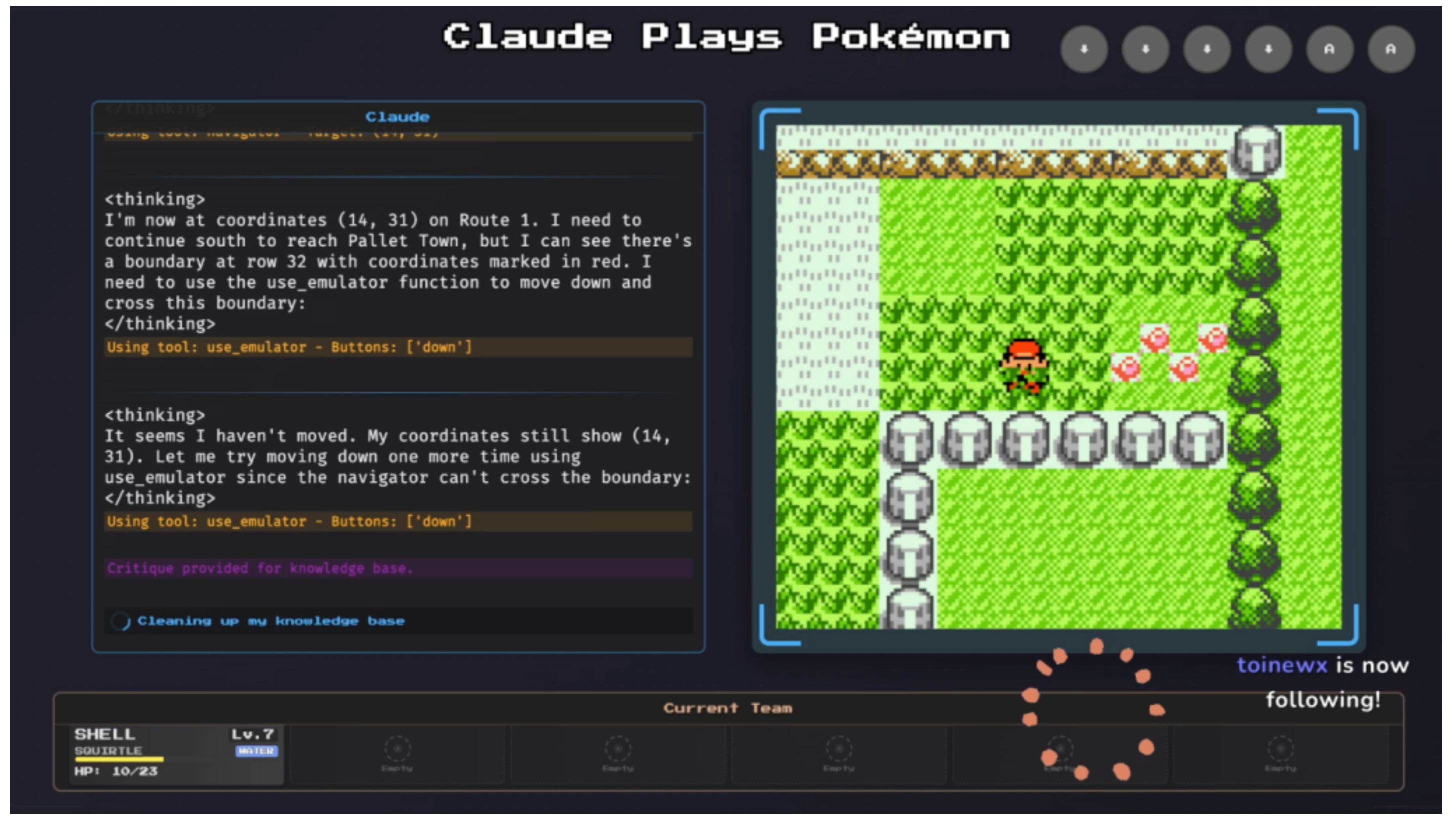
Predictability of Agentic Capabilities

There has been lots of excitement about developing autonomous agent



Predictability of Agentic Capabilities

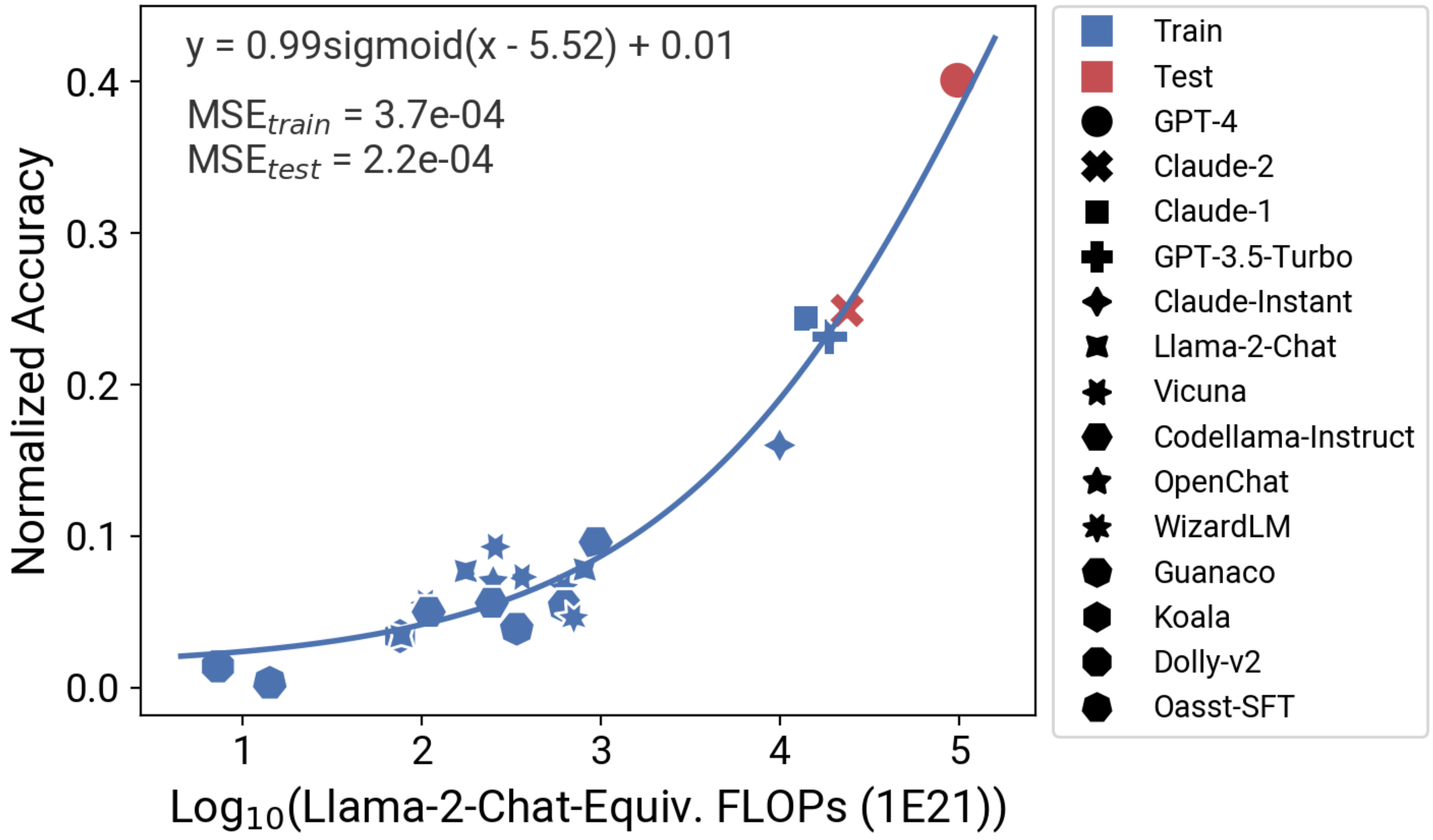
There has been lots of excitement about developing autonomous agent



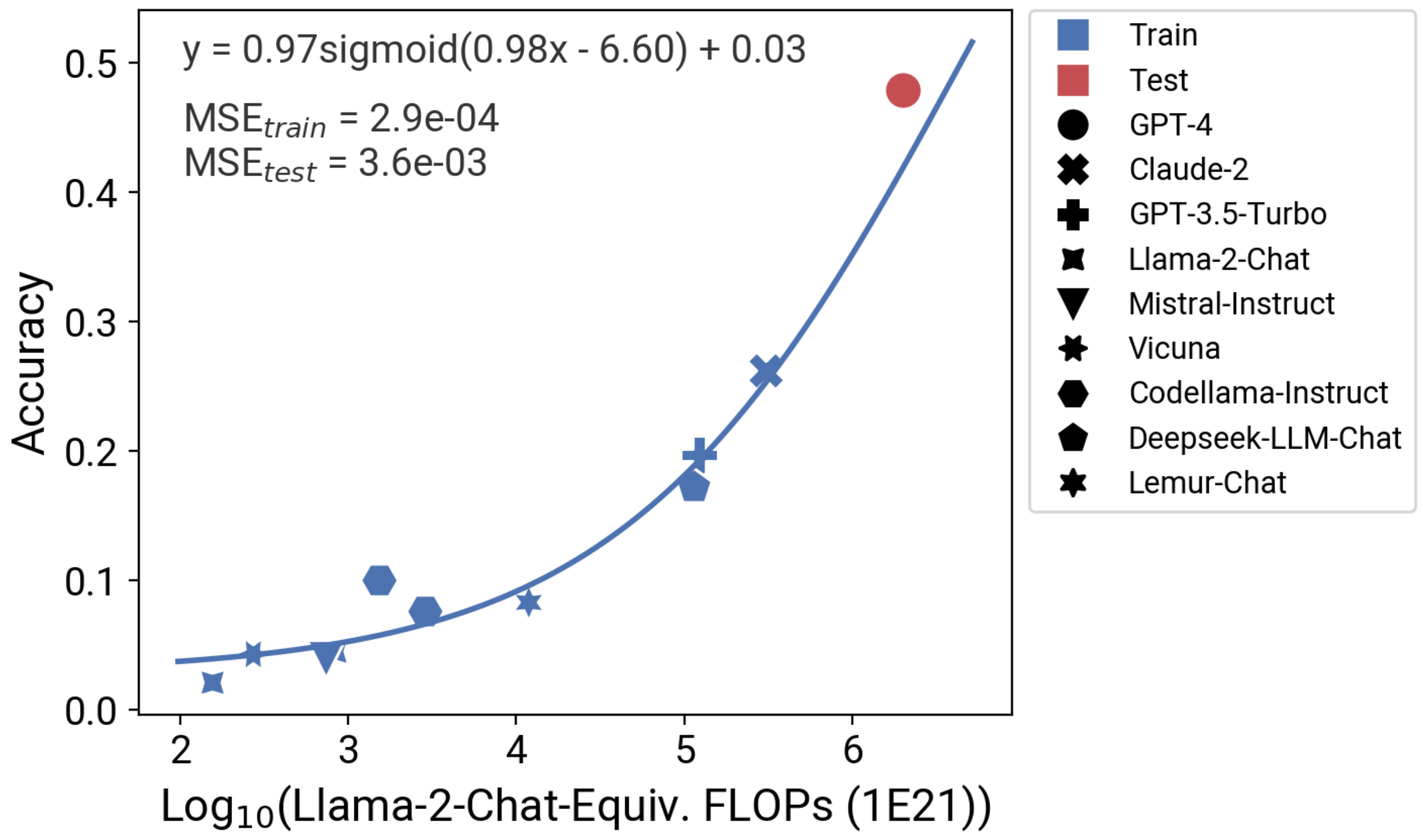
How do LMs' agentic capabilities scale?

Predictability of Agentic Capabilities

Agentic capabilities can be predicted with LMs' simple benchmark metrics



AgentBench [Liu et al., 2023]

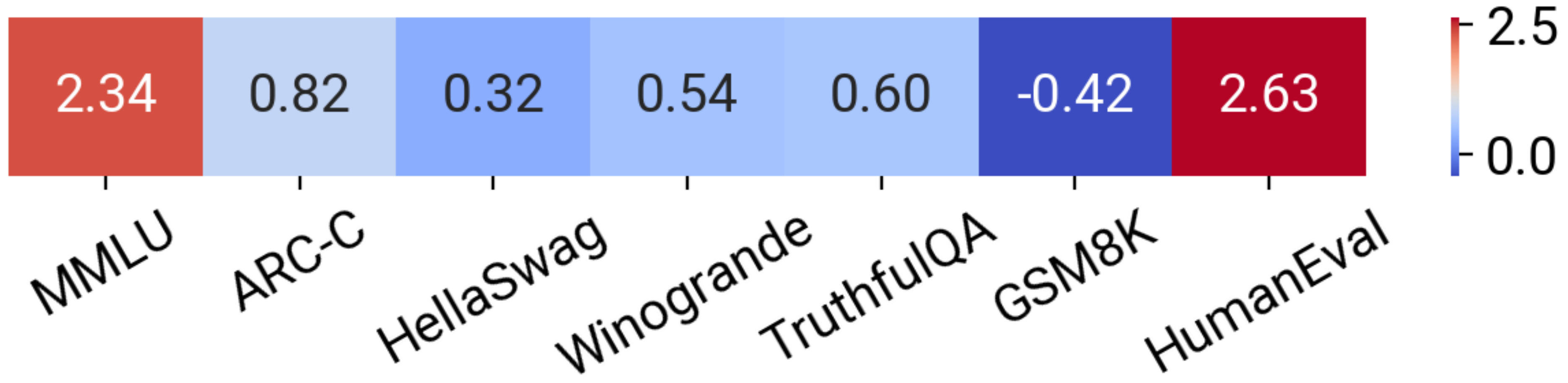


AgentBoard [Ma et al., 2024]

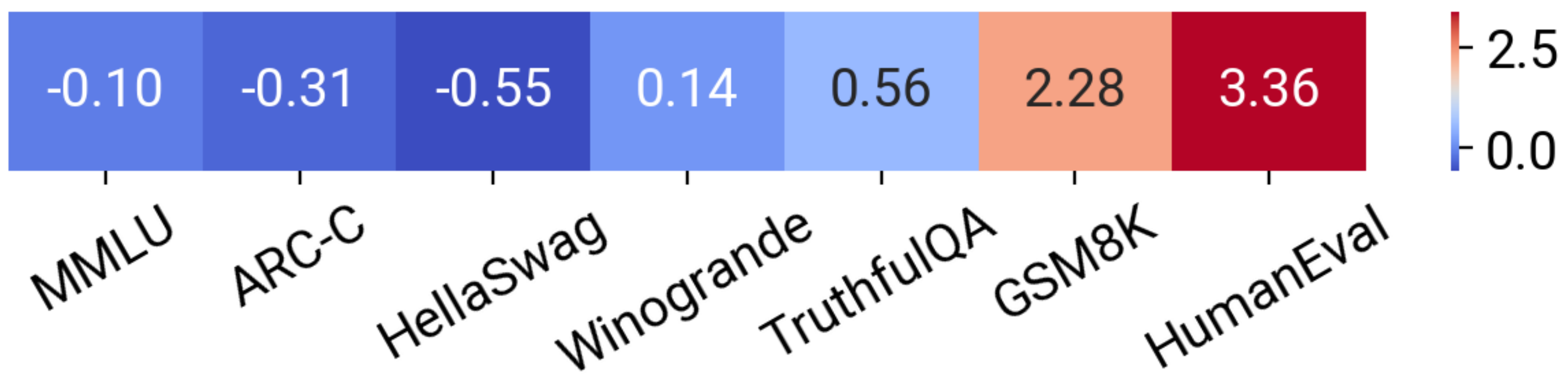
Predictability of Agentic Capabilities

Programming capabilities are essential

AgentBench

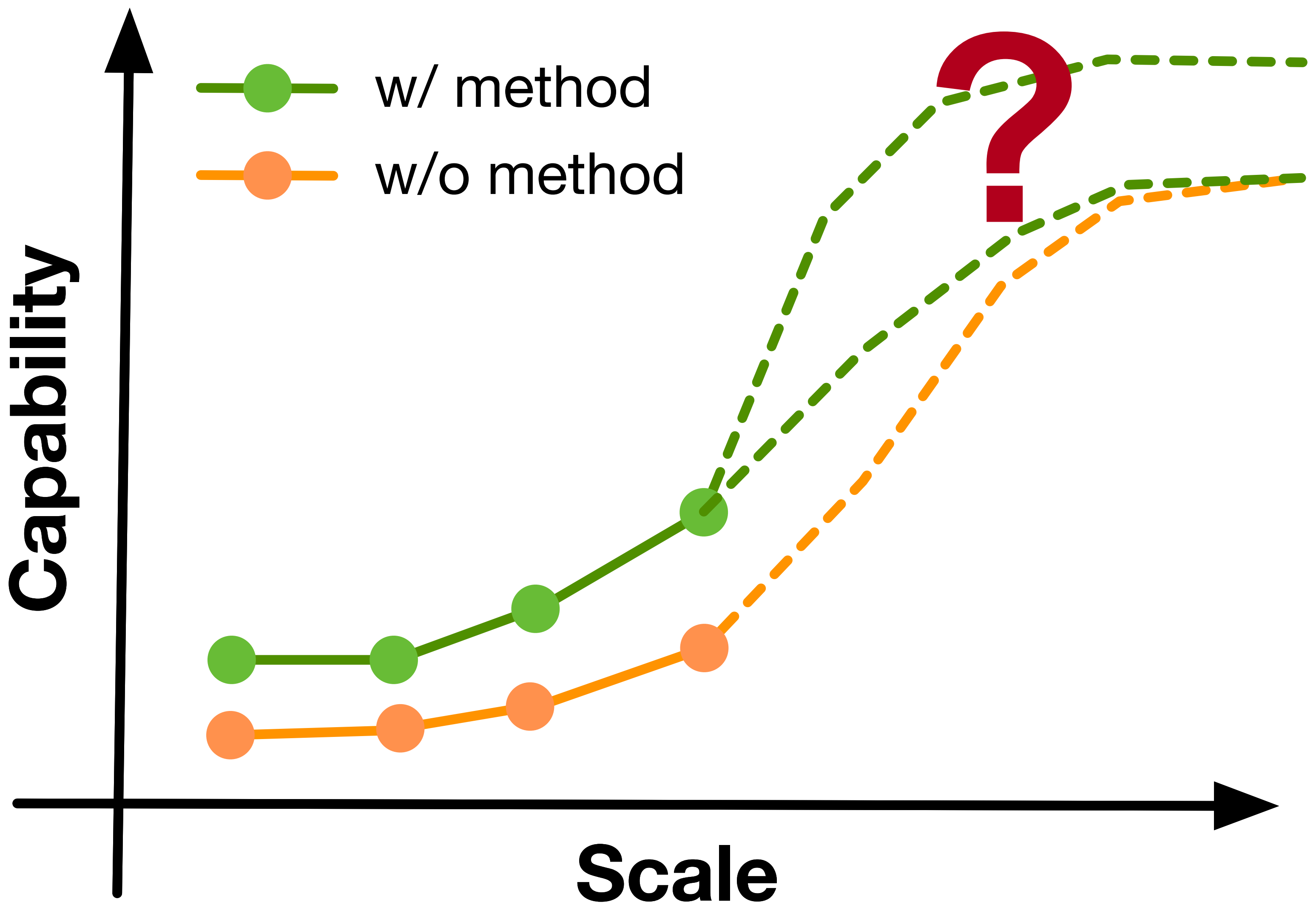


AgentBoard



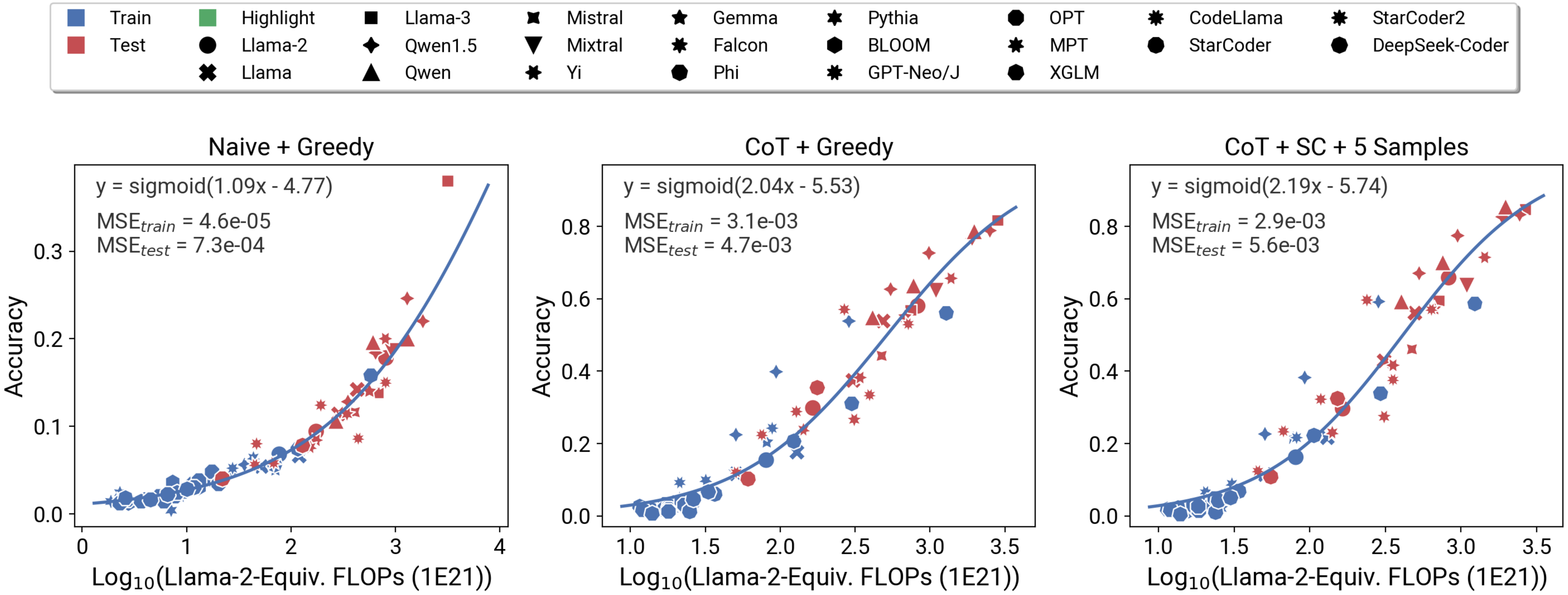
Predicting the Impact of Post-Training Techniques

Effective post-training techniques should persist gains across scales



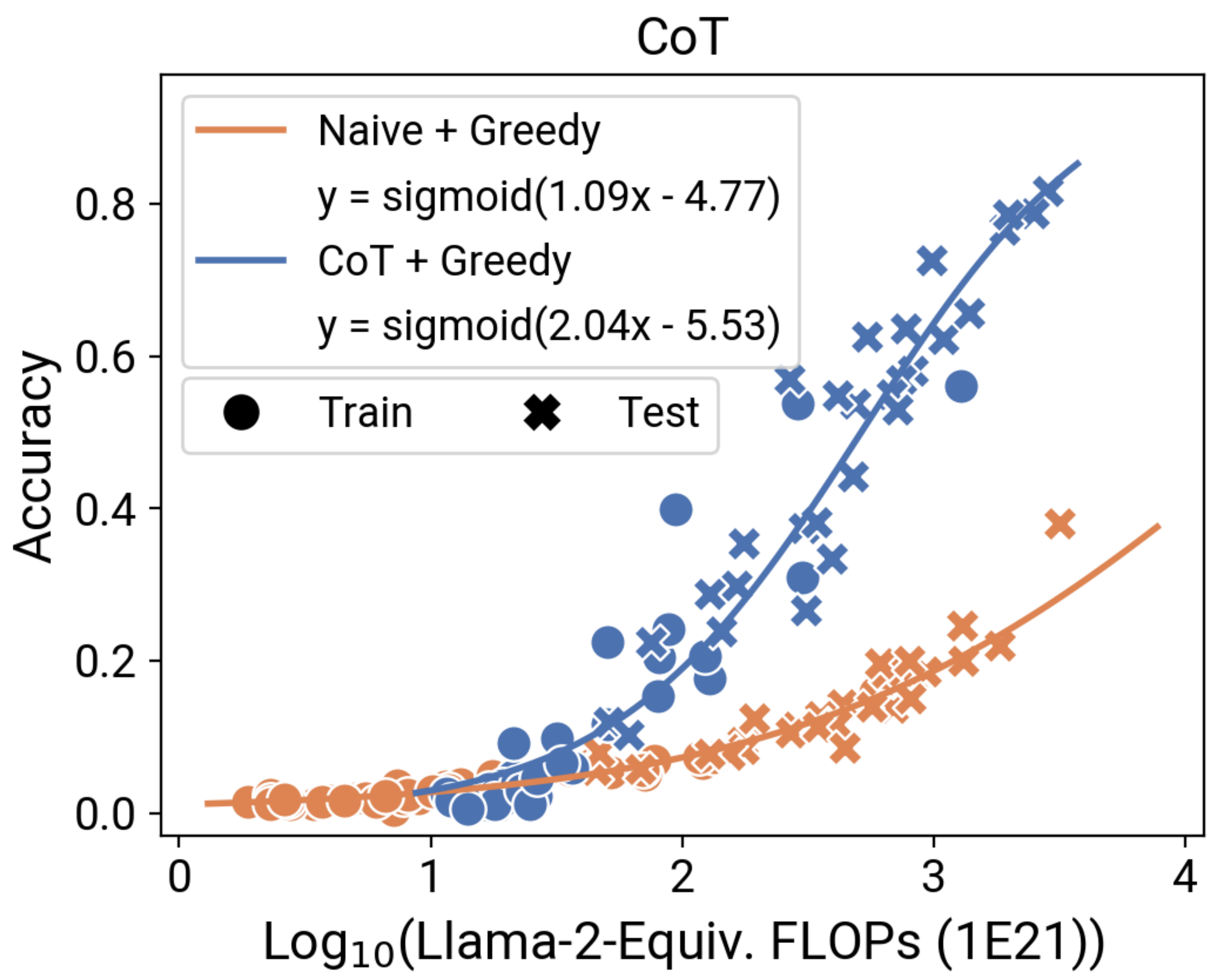
Predicting the Impact of Post-Training Techniques

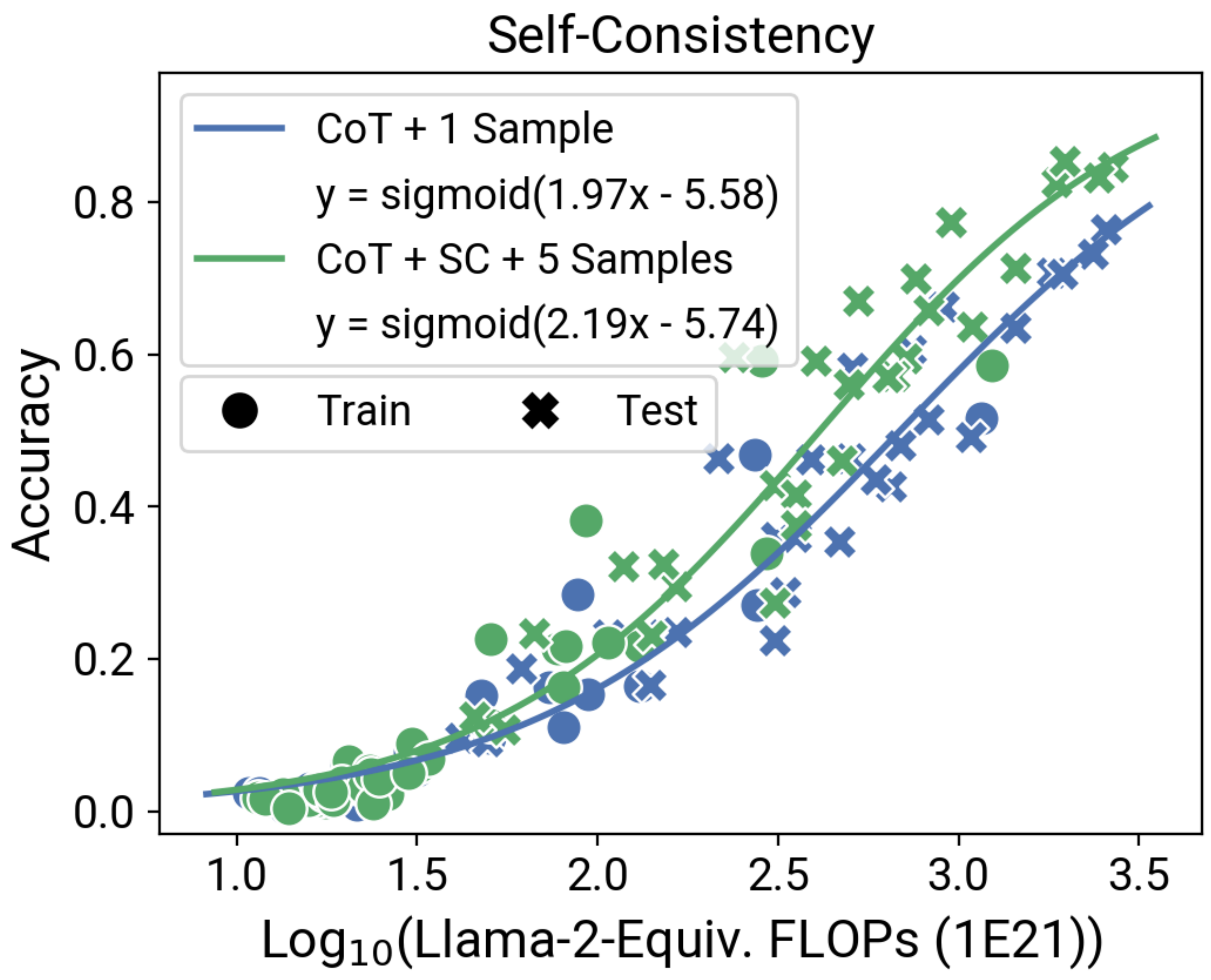
LMs' performance with post-training methods are predictable



Predicting the Impact of Post-Training Techniques

Different techniques demonstrate different scaling properties





Takeaways

- LM capabilities are highly correlated and low-dimensional
- Observational scaling laws offer a lower-cost, higher-resolution, broader-coverage alternative for complex capability and post-training analyses
- Many downstream LM capabilities—including seemingly emergent ones—may be smoothly predictable

Future Directions

- Reasoning models
  - Are obs. scaling laws still applicable?
  - Can we predict the gains of RL training from various base LMs with obs. scaling?
- Complex downstream capability analyses
  - More reliable capability forecasts with obs. scaling (e.g., Pimpale et al., 2025)?
  - Simpler optimization surrogate from fitted obs. scaling predictions?

Thank you!