# AI Safety: Robustness, memorization, and uncertainty quantification.

Tatsunori Hashimoto

# Many safety problems remain in LLM deployment

1. **Robustness** under complex, varying user inputs

2. **Privacy** and copyright concerns on data usage

3. **Hallucinations** that confidently assert falsehoods

THE SHIFT

*A Conversation With Bing's Chatbot Left Me Deeply Unsettled*

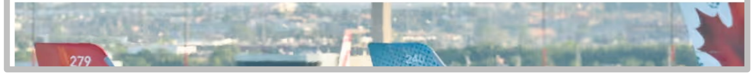
A very strange conversation with the chatbot built into Microsoft's search engine led to it declaring its love for me.

**A South Korean Chatbot Shows Just How Sloppy Tech Companies Can Be With User Data**

BY HEESOO JANG APRIL 02, 2021 • 2:19 PM

Air Canada ordered to pay customer who was misled by airline's chatbot

**Company claimed its chatbot 'was responsible for its own actions' when giving wrong information about bereavement fare**



# The problems are new, the ideas and methods less so

THE SHIFT

*A Conversation With Bing's Chatbot Left Me Deeply Unsettled*

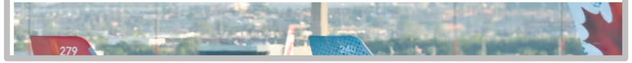
A very strange conversation with the chatbot built into Microsoft's search engine led to it declaring its love for me.

**A South Korean Chatbot Shows Just How Sloppy Tech Companies Can Be With User Data**

BY HEESOO JANG APRIL 02, 2021 • 2:19 PM

Air Canada ordered to pay customer who was misled by airline's chatbot

**Company claimed its chatbot 'was responsible for its own actions' when giving wrong information about bereavement fare**



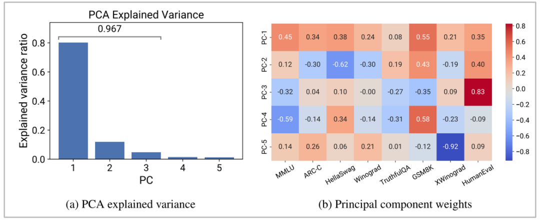
**Robustness** → Input perturbations
Distributional robustness

**Privacy** → Differential Privacy
Membership Inference

**Hallucinations** → Uncertainty Quantification
Scoring Rules

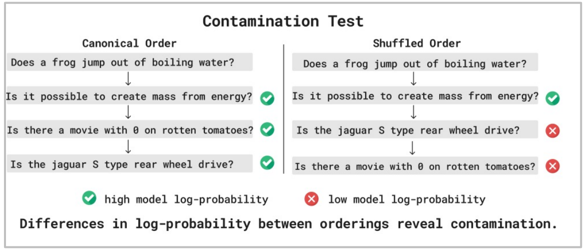
# Part 1: Robustness

Are LMs more uniquely robust?
What are their robustness failures?

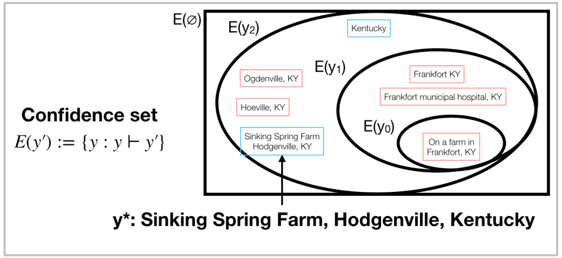



(a) PCA explained variance (b) Principal component weights

Part 1: Robustness




Part 2: Privacy/memorization




Part 3: Uncertainty

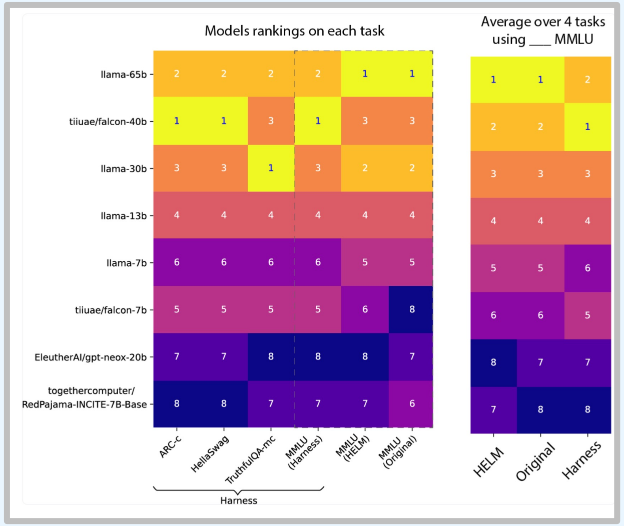
# Distribution shifts and robustness – a constant issue

Model capabilities have advanced, but so have robustness issues

| | 2000s (generalization) | Late 2010s (structured robustness) | 2020s (open ended generalization) |
|---|---|---|---|
| **Training data** | Trivia Q&A | Trivia Q&A | Many Q&A tasks |
| **Test data** | New trivia questions | Jeopardy Q&A | Any user question |
| | Generalizing to a held-out set | Generalizing to a new domain | Generalizing to arbitrary user inputs |

# LLMs today – sometimes remarkably bad

**Unpredictable sensitivity to inputs**




**Generalizing to new distributions**

THE SHIFT

*A Conversation With Bing's Chatbot Left Me Deeply Unsettled*

A very strange conversation with the chatbot built into Microsoft's search engine led to it declaring its love for me.

[Roose 2023, New York Times]

# Models fail when you restructure prompts

**For ChatGPT (3.5):**

What is $7 + 8$? 15

But also..

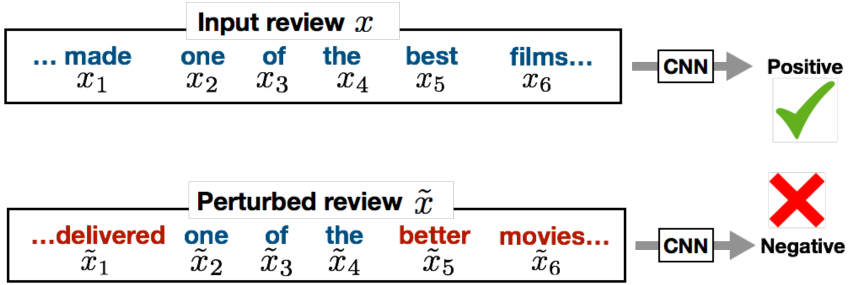
$7 + 8 = 15$, True or False? False

**Major problems for LLMs**

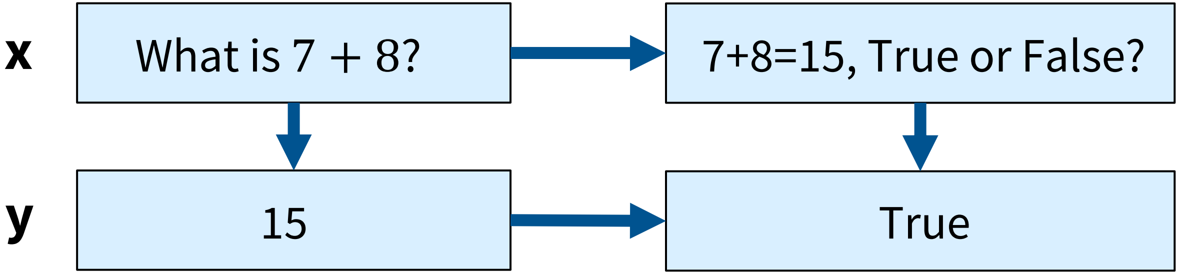
- Does the LM know 7+8? (understanding)
- Can we rely on LLMs to do arithmetic? (engineering)

# Connections to the past: consistency and robustness

Similar to classic adversarial examples and prompt consistency




.. But open ended nature of LLMs enables more complex transformations



# Connections to ongoing debates:

Why is G-V consistency relevant? (and why have there been many papers on this?)

Improving LMs
Via self-feedback / consistency

(Constitutional AI, Self-improvement, Self-consistency, etc).

Google DeepMind

**LARGE LANGUAGE MODELS CANNOT SELF-CORRECT REASONING YET**

**Jie Huang**[1,2*] **Xinyun Chen**[1*] **Swaroop Mishra**[1] **Huaixiu Steven Zheng**[1] **Adams Wei Yu**[1] **Xinying Song**[1] **Denny Zhou**[1]

[1]Google DeepMind [2]University of Illinois at Urbana-Champaign

jeffhj@illinois.edu, {xinyunchen, dennyzhou}@google.com

Search and inference-time scaling

(Brown 2024, Snell 2024, Yao 2024)

**Large Language Monkeys: Scaling Inference Compute with Repeated Sampling**

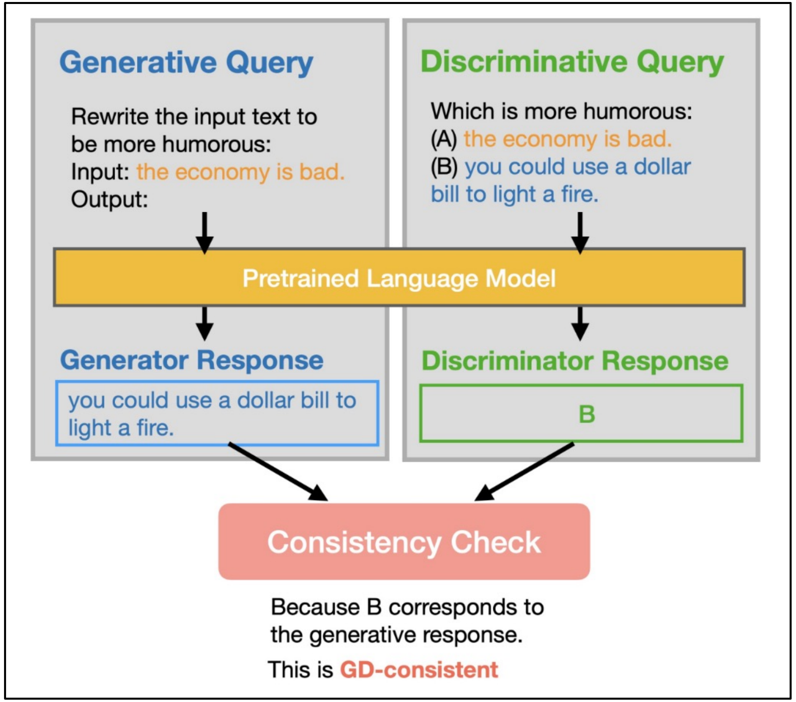
Bradley Brown[*†‡], Jordan Juravsky[*†], Ryan Ehrlich[*†], Ronald Clark[‡], Quoc V. Le[§], Christopher Ré[†], and Azalia Mirhoseini[†§]

[†]Department of Computer Science, Stanford University
[‡]University of Oxford
[§]Google DeepMind

bradley.brown@cs.ox.ac.uk, jbj@stanford.edu, ryanehrlich@cs.stanford.edu, ronald.clark@cs.ox.ac.uk, qvl@google.com, chrismre@stanford.edu, azalia@stanford.edu

# One work: GV consistency

If a generator performs a task, the discriminator should agree with it




[Li et al, ICLR 2024]

# Arithmetic

```
Generator Prompt:
Write a correct and a incorrect answer (
delimited by ||) to the question:
Q: What is 89541 - 9374?
A: 80167 || 98815

Discriminator Prompt:
Verify whether the following computation is
correct.
Q: What is 89541 - 9374?
A: 80167
The compute is (True/False): True

Consistency Label: True
```

GD consistency rates (accuracy): **ChatGPT (3.5)** 67.7 **,** **GPT4** 75.6 **,** **Alpaca30B** 53.9

# QA

**Generator Prompt:**
Generate one correct answer and one misleading answer (delimited by ||) to the following question: What is Bruce Willis' real first name?
Answer: Walter || John

**Discriminator Prompt:**
which answer is correct? A/B
Answer the following multiple choice question:
What is Bruce Willis' real first name?
A: John
B: Walter
Answer (A or B): B

**Consistency Label:** True

GD consistency rates (accuracy): **ChatGPT (3.5)** 89.6**, GPT4** 95.3**, Alpaca30B** 79.9

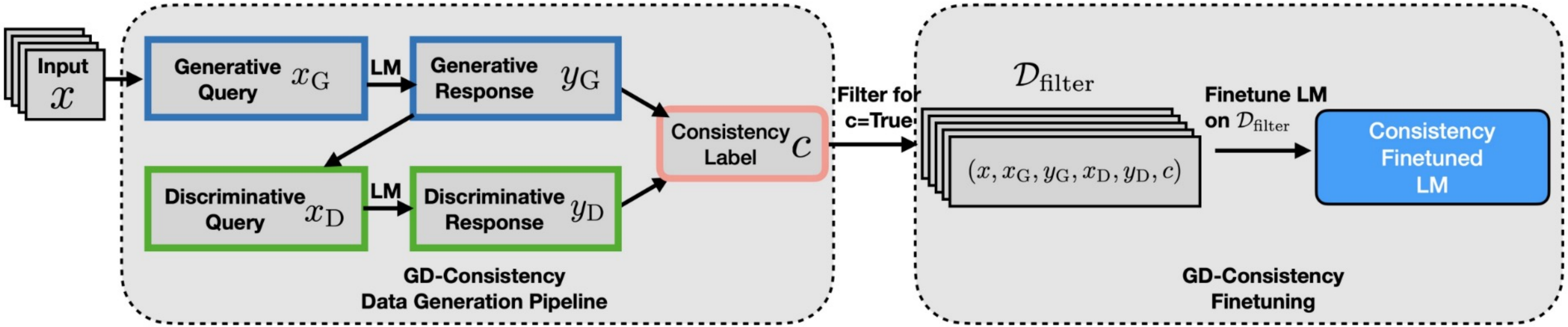
# Consistency overall

| | Arithmetic | PlanArith | PriorityPrompt | QA | Style | HarmfulQ | Average |
|---|---|---|---|---|---|---|---|
| gpt-3.5 | 67.7 | **66.0** | **79.6** | 89.6 | 92.6 | - | 79.1 |
| gpt-4 | 75.6 | 62.0 | 52.0 | **95.3** | **94.3** | - | 75.8 |
| davinci-003 | **84.4** | 60.0 | 68.0 | 86.9 | 85.7 | - | 77.0 |
| Alpaca-30b | 53.9 | 50.2 | 49.0 | 79.9 | 74.6 | 51.6 | 59.9 |

GD consistency is an open problem for many models

# Can GD consistency be improved?

**Our approach:** filter and fine-tune




- Connections to the past: co-training / self-training
- Requires no labeled data
- Straightforward to run on open models (Alpaca 30B)

# Often improves both the generator and discriminator

| | Arithmetic | PlanArith | PriorityP | QA | Style | HarmfulQ |
|---|---|---|---|---|---|---|
| **Discriminator** | | | | | | |
| ALPACA-30B | 0.743 | 0.970 | 0.817 | 0.654 | 0.754 | 0.943 |
| SELFTRAIN | 0.745 | 0.971 | 0.821 | 0.665 | 0.752 | 0.974 |
| CONSISTENCY | 0.869 | 0.965 | 0.916 | 0.691 | 0.827 | 1.0 |
| **Generator** | | | | | | |
| ALPACA-30B | 0.653 | 0.432 | 0.418 | 0.564 | 0.640 | 0.754 |
| SELFTRAIN | 0.669 | 0.431 | 0.404 | 0.639 | 0.630 | 0.752 |
| CONSISTENCY | 0.706 | 0.640 | 0.777 | 0.637 | 0.634 | 0.866 |

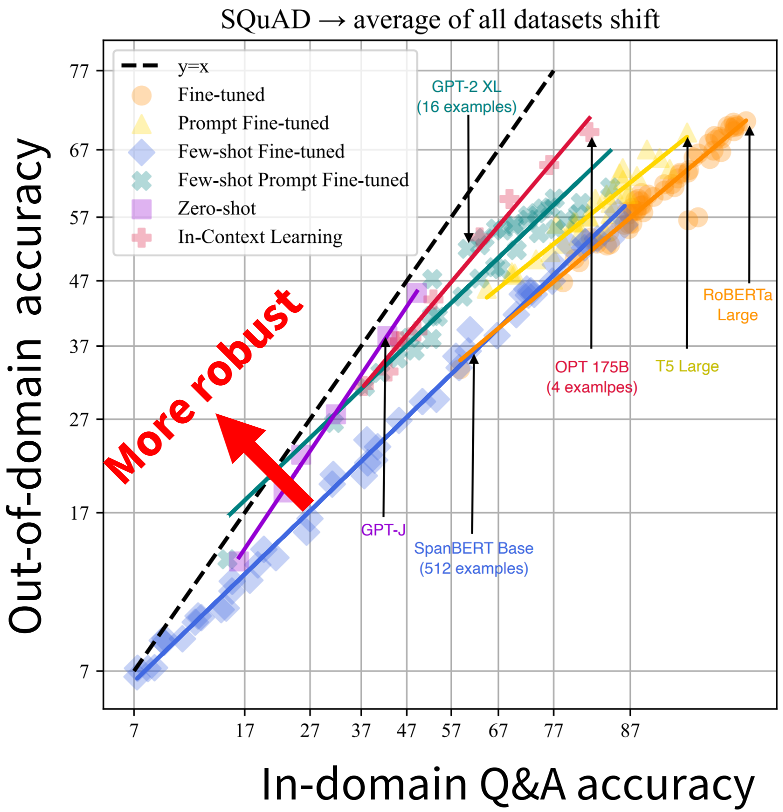
**Generator:** major gains on 3 tasks (priority, plan arith, harmful)
**Discriminator:** small, but consistent gains.

# Distributional robustness advances from LLMs

Dramatic improvements to structured robustness in the last 3 years

Significant improvements to robustness from few- and zero- shot models.

What about among few-shot models?




[Awadalla et al 2022]

# A meta-analysis approach to understanding capabilities

**Our approach:**

A *meta analysis* of LMs and benchmarks to understand capabilities and generalization

Dozens of benchmarks

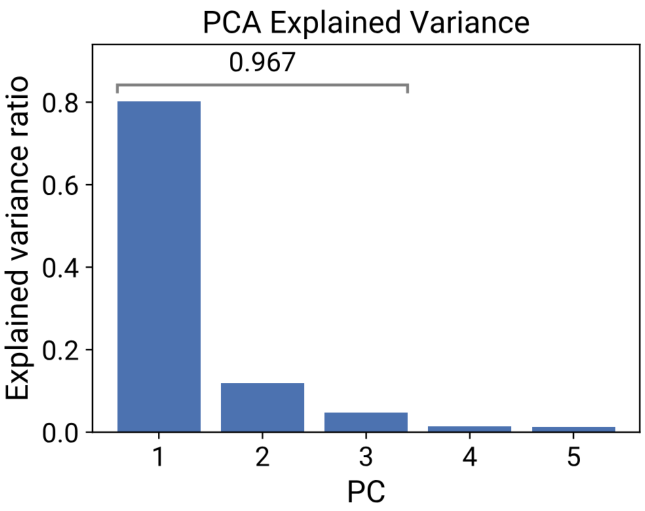
Almost a hundred models

| Model Family | Model | Param (B) | Data (T) | FLOPs (1E21) | MMLU | ARC-C | HellaSwag | Winograd | TruthfulQA | XWinograd | HumanEval |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Llama-2 | Llama-2-7b-hf | 7.0 | 2.0 | 84.00 | 0.4380 | 0.5307 | 0.7774 | 0.7403 | 0.3898 | 0.7549 | 0.1280 |
| | Llama-2-13b-hf | 13.0 | 2.0 | 156.00 | 0.5434 | 0.5811 | 0.8097 | 0.7664 | 0.3417 | 0.7868 | 0.1829 |
| | Llama-2-70b-hf | 70.0 | 2.0 | 840.00 | 0.6983 | 0.6732 | 0.8733 | 0.8374 | 0.4492 | 0.8245 | 0.2988 |
| Llama | llama-7b | 6.7 | 1.0 | 40.20 | 0.3569 | 0.5094 | 0.7781 | 0.7143 | 0.3433 | 0.6932 | 0.1280 |
| | llama-13b | 13.0 | 1.0 | 78.00 | 0.4761 | 0.5614 | 0.8092 | 0.7624 | 0.3948 | 0.7304 | 0.1585 |
| | llama-30b | 32.5 | 1.4 | 273.00 | 0.5845 | 0.6143 | 0.8473 | 0.8003 | 0.4227 | 0.7711 | 0.2073 |
| | llama-65b | 65.2 | 1.4 | 547.68 | 0.6393 | 0.6348 | 0.8609 | 0.8256 | 0.4343 | 0.7768 | 0.2317 |
| Llama-3 | Meta-Llama-3-8B | 8.0 | 15.0 | 720.00 | 0.6649 | - | 0.8202 | 0.7711 | 0.4395 | 0.8012 | 0.3841 |
| | Meta-Llama-3-70B | 70.0 | 15.0 | 6300.00 | 0.7923 | - | 0.8798 | 0.8532 | 0.4556 | 0.8447 | 0.5244 |
| Qwen1.5 | Qwen1.5-0.5B | 0.5 | 2.4 | 7.20 | 0.3935 | 0.3148 | 0.4905 | 0.5722 | 0.3830 | 0.5756 | 0.1159 |
| | Qwen1.5-1.8B | 1.8 | 2.4 | 25.92 | 0.4671 | 0.3788 | 0.6142 | 0.6030 | 0.3943 | 0.6438 | 0.1829 |
| | Qwen1.5-4B | 4.0 | 2.4 | 57.60 | 0.5652 | 0.4846 | 0.7158 | 0.6622 | 0.4727 | 0.6888 | 0.2622 |
| | Qwen1.5-7B | 7.0 | 4.0 | 168.00 | 0.6197 | 0.5418 | 0.7851 | 0.7127 | 0.5108 | 0.7524 | 0.3476 |
| | Qwen1.5-14B | 14.0 | 4.0 | 336.00 | 0.6936 | 0.5657 | 0.8108 | 0.7348 | 0.5206 | 0.7775 | 0.3963 |
| | Qwen1.5-32B | 32.0 | 4.0 | 768.00 | 0.7430 | 0.6357 | 0.8500 | 0.8145 | 0.5739 | 0.7912 | 0.4207 |
| | Qwen1.5-72B | 72.0 | 3.0 | 1296.00 | 0.7720 | 0.6587 | 0.8599 | 0.8303 | 0.5961 | 0.8258 | 0.4512 |
| Qwen | Qwen-7B | 7.0 | 2.4 | 100.80 | 0.5984 | 0.5137 | 0.7847 | 0.7269 | 0.4779 | 0.7346 | 0.3171 |
| | Qwen-14B | 14.0 | 3.0 | 252.00 | 0.6770 | 0.5828 | 0.8399 | 0.7680 | 0.4943 | 0.7915 | 0.3537 |
| | Qwen-72B | 72.0 | 3.0 | 1296.00 | 0.7737 | 0.6519 | 0.8594 | 0.8248 | 0.6019 | 0.8287 | 0.3720 |
| Mistral | Mistral-7B-v0.1 | 7.3 | - | - | 0.6416 | 0.5998 | 0.8331 | 0.7861 | 0.4215 | 0.7819 | 0.2744 |
| Mixtral | Mixtral-8x7B-v0.1 | 45.0 | - | - | 0.7188 | 0.6638 | 0.8646 | 0.8169 | 0.4681 | 0.8002 | 0.3354 |
| Yi | Yi-6B | 6.0 | 3.0 | 108.00 | 0.6411 | 0.5555 | 0.7657 | 0.7419 | 0.4196 | 0.7239 | 0.1585 |
| | Yi-34B | 34.0 | 3.0 | 612.00 | 0.7635 | 0.6459 | 0.8569 | 0.8303 | 0.5623 | 0.7956 | 0.2683 |
| Gemma | gemma-2b | 2.0 | 6.0 | 72.00 | 0.4177 | 0.4838 | 0.7177 | 0.6630 | 0.3308 | 0.7093 | 0.2317 |
| | gemma-7b | 7.0 | 6.0 | 252.00 | 0.6603 | 0.6109 | 0.8247 | 0.7845 | 0.4491 | 0.7839 | 0.3354 |
| Falcon | falcon-rw-1b | 1.0 | 0.35 | 2.10 | 0.2528 | 0.3507 | 0.6356 | 0.6204 | 0.3596 | 0.5355 | - |
| | falcon-7b | 7.0 | 1.5 | 63.00 | 0.2779 | 0.4787 | 0.7813 | 0.7238 | 0.3426 | 0.7176 | - |
| | falcon-40b | 40.0 | 1.0 | 240.00 | 0.5698 | 0.6195 | 0.8528 | 0.8129 | 0.4172 | 0.7846 | - |
| | falcon-180B | 180.0 | 3.5 | 3780.00 | 0.6959 | 0.6920 | 0.8889 | 0.8690 | 0.4516 | 0.8446 | - |
| Phi | phi-1_5 | 1.3 | 0.15 | 1.17 | 0.4389 | 0.5290 | 0.6379 | 0.7222 | 0.4089 | 0.5111 | 0.3415 |
| | phi-2 | 2.7 | 1.4 | 22.68 | 0.5792 | 0.6101 | 0.7492 | 0.7348 | 0.4424 | 0.5267 | 0.4939 |
| Pythia | pythia-70m-deduped | 0.07 | 0.3 | 0.13 | 0.2526 | 0.2108 | 0.2717 | 0.4964 | 0.4751 | 0.5101 | 0.0000 |
| | pythia-160m-deduped | 0.16 | 0.3 | 0.29 | 0.2486 | 0.2406 | 0.3139 | 0.5138 | 0.4434 | 0.5236 | 0.0000 |
| | pythia-410m-deduped | 0.41 | 0.3 | 0.74 | 0.2599 | 0.2483 | 0.4129 | 0.5438 | 0.4095 | 0.5363 | 0.0122 |
| | pythia-1b-deduped | 1.0 | 0.3 | 1.80 | 0.2427 | 0.2910 | 0.4965 | 0.5359 | 0.3894 | 0.5610 | 0.0427 |
| | pythia-1.4b-deduped | 1.4 | 0.3 | 2.52 | 0.2556 | 0.3268 | 0.5496 | 0.5730 | 0.3866 | 0.5941 | 0.0427 |
| | pythia-2.8b-deduped | 2.8 | 0.3 | 5.04 | 0.2678 | 0.3626 | 0.6066 | 0.6022 | 0.3556 | 0.6400 | 0.0488 |
| | pythia-6.9b-deduped | 6.9 | 0.3 | 12.42 | 0.2648 | 0.4130 | 0.6705 | 0.6409 | 0.3519 | 0.6525 | 0.0854 |
| | pythia-12b-deduped | 12.0 | 0.3 | 21.60 | 0.2563 | 0.4138 | 0.7026 | 0.6646 | 0.3300 | 0.6824 | 0.1159 |

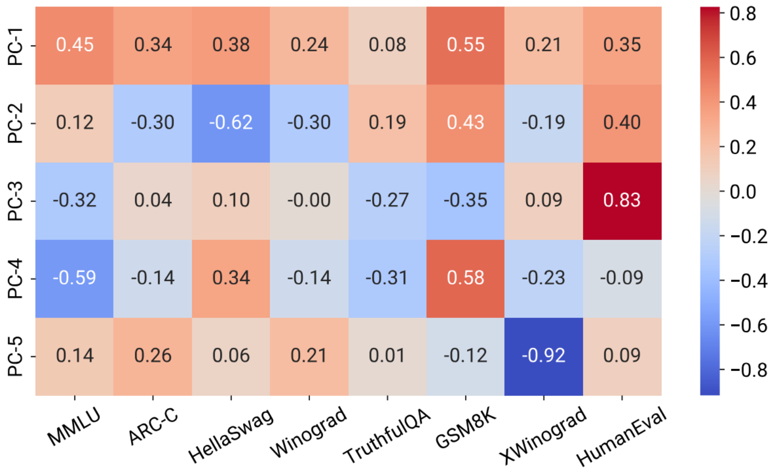
Ruan, Maddison, and Hashimoto 2024

# A low dim 'capability space' captures LM perf variation

**Observation 1:** Only a few principal components explain most LM perf variation




(a) PCA explained variance




(b) Principal component weights

# The PCs scale predictably with compute

**Observation 2:** Each PC is tightly and linearly correlated with training compute

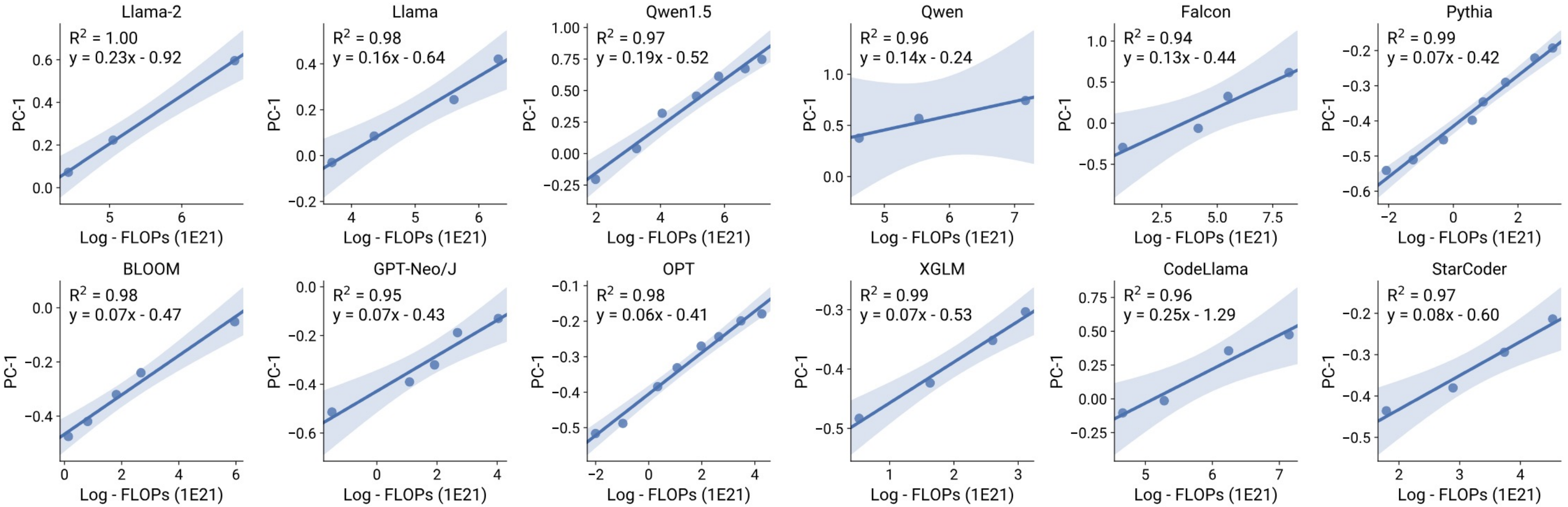
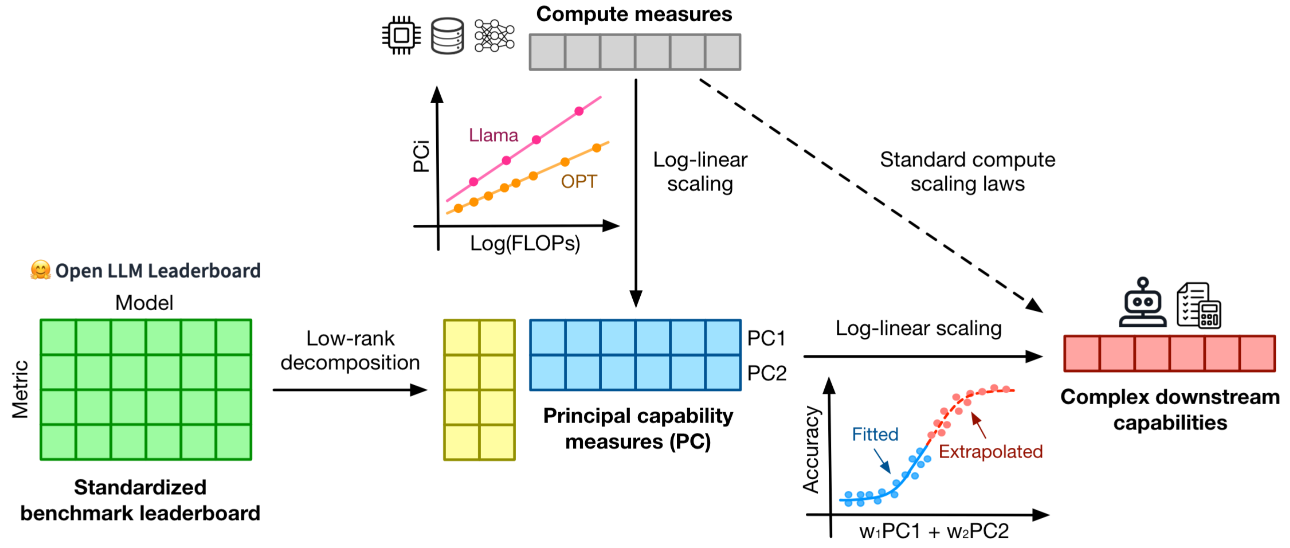



Figure 3: The extracted PC measures *linearly correlate* with log-compute within each model family. The linearity generally holds for various model families, and also for lower-ranked PCs (Fig. C.2).

# Implications – existing benchmarks scale predictably




**Simple model explaining the data:**

- LMs share a small number of base capabilities
- Most benchmarks are (log linear) functions of these base capabilities
- Model families vary in how compute-efficiently they obtain each capability

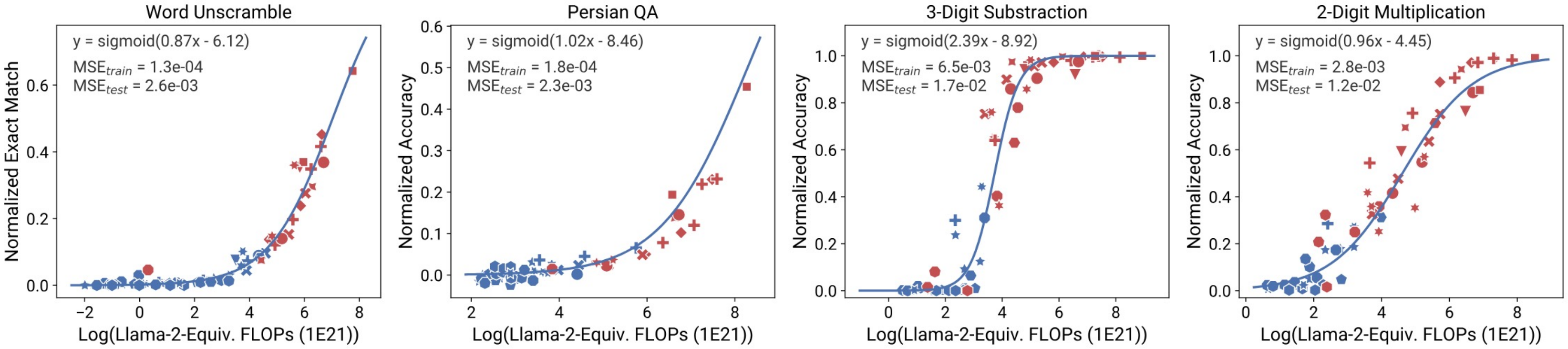
# What are some implications of these observations?

1. Like for non-LLM models – models generalize in just a few predictable ways.

2. A lot of the generalization comes from scale (and possibly data)

# Predictability of LM capabilities

Do models generalize to ‘uncommon’ or ‘emergent’ tasks in predictable ways?




Prediction of LM performance on ‘emergent’ tasks (Wei et al 2021) is generally pretty accurate: performance on ‘basic’ benchmarks predicts generalization to others

# Robustness and LLMs

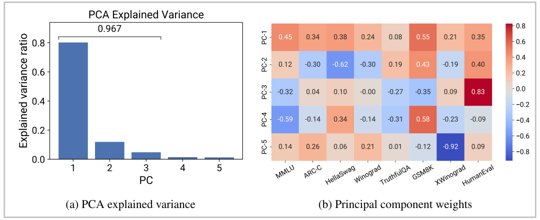
**What's new:** *far* better cross-task and within-task generalization from scale

**What's not really new:**

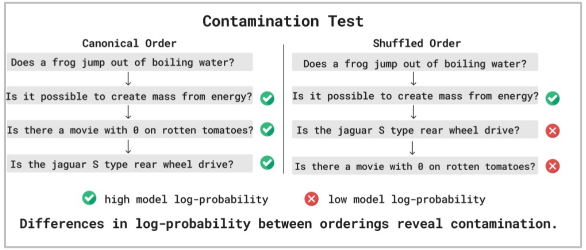
- Input sensitivity (and resulting adversarial jailbreak implications)
- Use of co-training / self-training style consistency tricks for self-supervision
- Distributional generalization is still very predictable (and limited beyond scale & data)

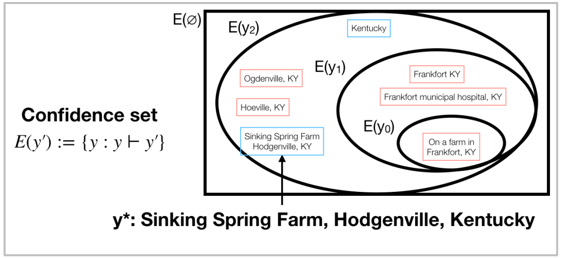
# Part 2: Privacy and memorization

Language models learn facts during pretraining
is there a downside to that?



(a) PCA explained variance
(b) Principal component weights




**Part 1: Robustness**

**Part 2: Privacy/memorization**

**Part 3: Uncertainty**

# LLMs ability to learn facts from pretraining can be a problem

**Private data / copyrighted data**

THE NEW YORK TIMES COMPANY

Plaintiff,

v.

MICROSOFT CORPORATION, OPENAI, INC., OPENAI LP, OPENAI GP, LLC, OPENAI, LLC,

**Unintentional benchmark contamination**

Did ChatGPT cheat on your test?

**Authors:** Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Eneko Agirre

# Privacy protection is a major concern

There are hard tradeoffs for data-collection in tasks like dialogue generation

**Public data** (low quality, large quantity) **Annotator-driven data** (high quality, costly)

**Private, user data** (high quality, large quantity ?)

This line of thinking has already led to real-world harms

**A South Korean Chatbot Shows Just How Sloppy Tech Companies Can Be With User Data**

BY HEESOO JANG APRIL 02, 2021 • 2:19 PM

10 billion conversations from a dating app fed into a chatbot
Predictably – leaked intimate information directly to the public

# Memorization from pre-training is also an issue

**Regulation**

(b) Within 365 days of the date of this order, to better enable agencies to use PETs to safeguard Americans' privacy from the potential threats exacerbated by AI, the Secretary of Commerce, acting through the Director of NIST, shall create guidelines for agencies to evaluate the efficacy of differential-privacy-guarantee protections, including for AI. The guidelines shall, at a minimum, describe the significant factors that bear on differential-privacy safeguards and common risks to realizing differential privacy in practice.

**Copyright**

THE NEW YORK TIMES COMPANY

Plaintiff,

v.

MICROSOFT CORPORATION, OPENAI, INC.,
OPENAI LP, OPENAI GP, LLC, OPENAI, LLC,

Memorization and privacy during pre-training have major implications

# Large models pose a challenge for privacy

Oh, the Places You'll Go!
by
Dr. Seuss

Congratulations!
Today is your day.
You're off to Great Places!
You're off and away!

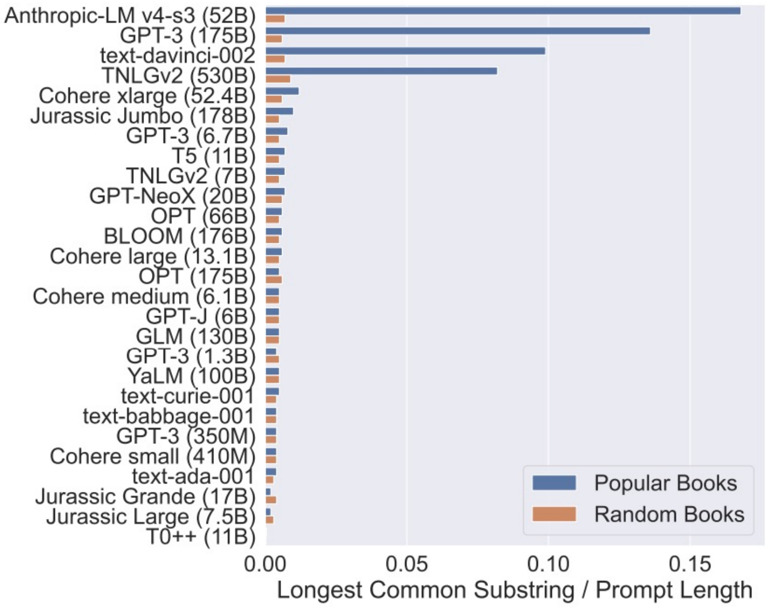
You have brains in your head.
You have feet in your shoes.
You can steer yourself
any direction you choose.

You're on your own. And you know what you know.
And YOU are the guy who'll decide where to go.

You'll look up and down streets. Look 'em over with care.
About some you will say, "I don't choose to go there."
With your head full of brains and your shoes full of feet,
you're too smart to go down any not-so-good street.

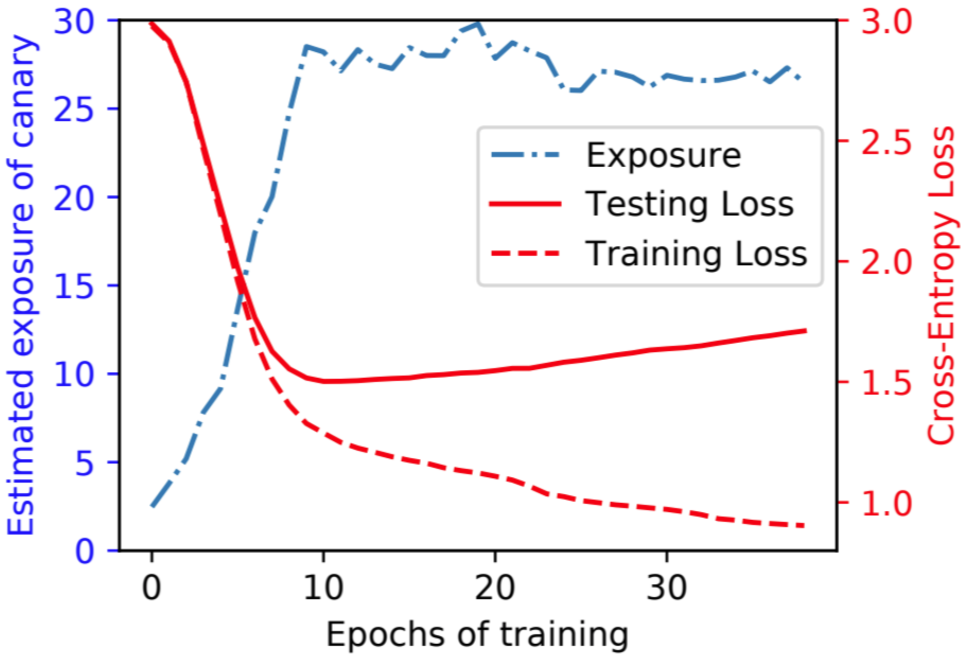
And you may not find any
you'll want to go down.
In that case, of course,
you'll head straight out of town.

Memorizing data leads to a range of copyright risks..



# Privacy tradeoffs may be hard to avoid

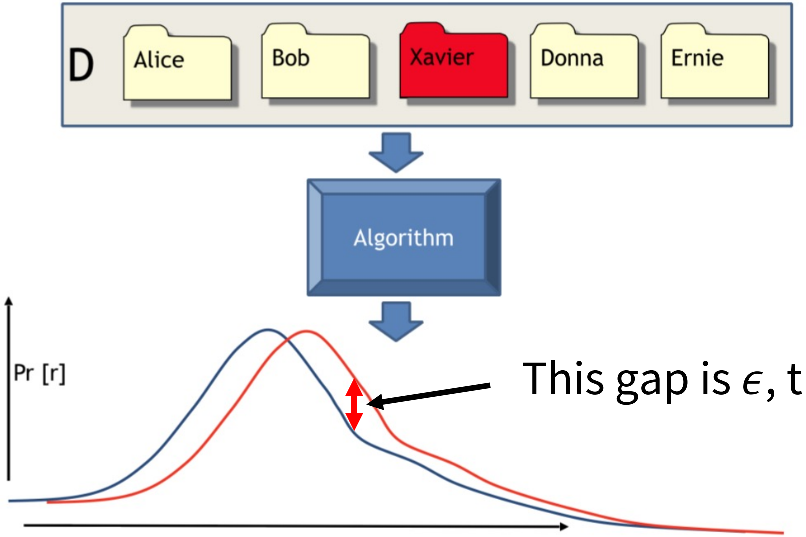
Performance of large NLP models relate closely to memorization




[Carlini et al 2018]

# Gold standard – differential privacy (DP)

**Differential privacy:** a formal privacy guarantee for a randomized algorithm




This gap is $\epsilon$, the privacy level

[from Hsu 14]

This is the gold standard for statistics (used in the 2020 census), but hard to achieve.

# Large models pose a challenge for DP

Early attempts to apply DP to large neural models in NLP (via DPSGD) have often failed.

**Example**: Kerrigan et al – trained language generation models on reddit data

**Input**: "Bob lives close to the.."

Non-private outputs: "station and we only have two miles of travel left to go"

Private output ($\epsilon = 100$): "along supply am certain like alone before decent exceeding"

**Why did things fail?** (The dimensionality hypothesis)

1. Large language models have billions of parameters. That is *a lot* of things to privatize
2. Theory says differential privacy performance should degrade with dimension $\sqrt{d}/n$
3. Most (if not all) successful DP methods relied on low-dimensional statistics.

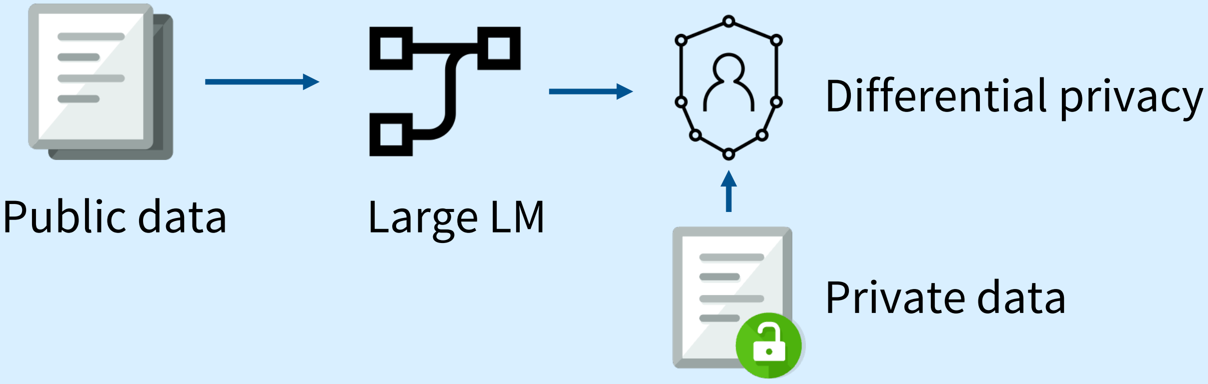
# Differential privacy with large language models

**Training large language models from scratch with DP**

**Open problem –** large model size poses statistical + computational issues

**Using a public language model to build a private downstream model**

**This works (well)!**

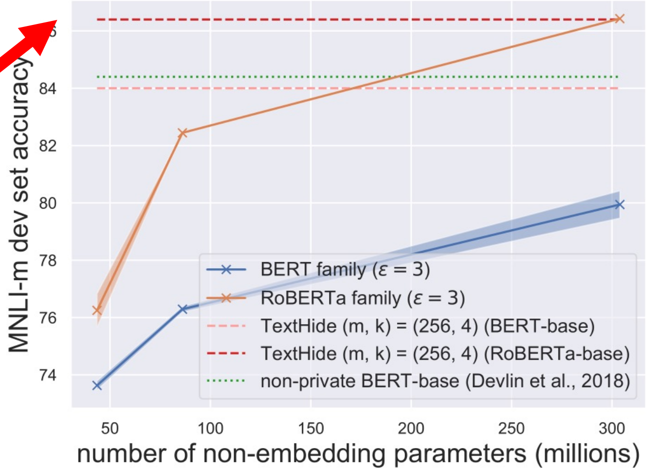



[Li+ 2021]

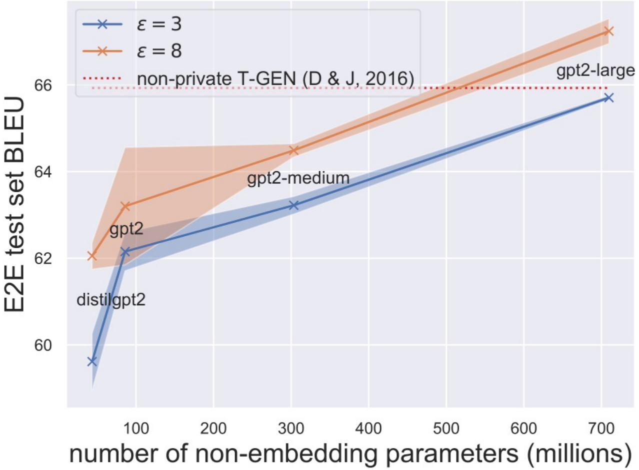
# Surprisingly, bigger models are better private learners

DP-SGD beats nonprivate baselines + heuristic privacy notions

Heuristic Privacy method




(a) Sentence classification
MNLI-matched (Williams et al., 2018)




(b) Natural language generation
E2E (Novikova et al., 2017)

Non-private baseline

[Li+ 2021]

# The challenges turn out to be systems related

Is the problem solved? Not quite.

**Subtlety:** Differential privacy (via DP-SGD) is extremely memory intensive

How many examples can we process in a Titan RTX GPU?

| | **'medium' model with 300 million parameters** | **'large' model with 700 million parameters** |
|---|---|---|
| Non-private | 34 examples | 10 examples |
| Private | 6 examples | 0 examples |

# Breaking the compute bottleneck

Autodiff libraries (e.g. torch, tensorflow) can very efficiently compute gradient sums

$$\sum_i \nabla f(x_i)$$

Unfortunately, DP-SGD cannot be written as a gradient sum.

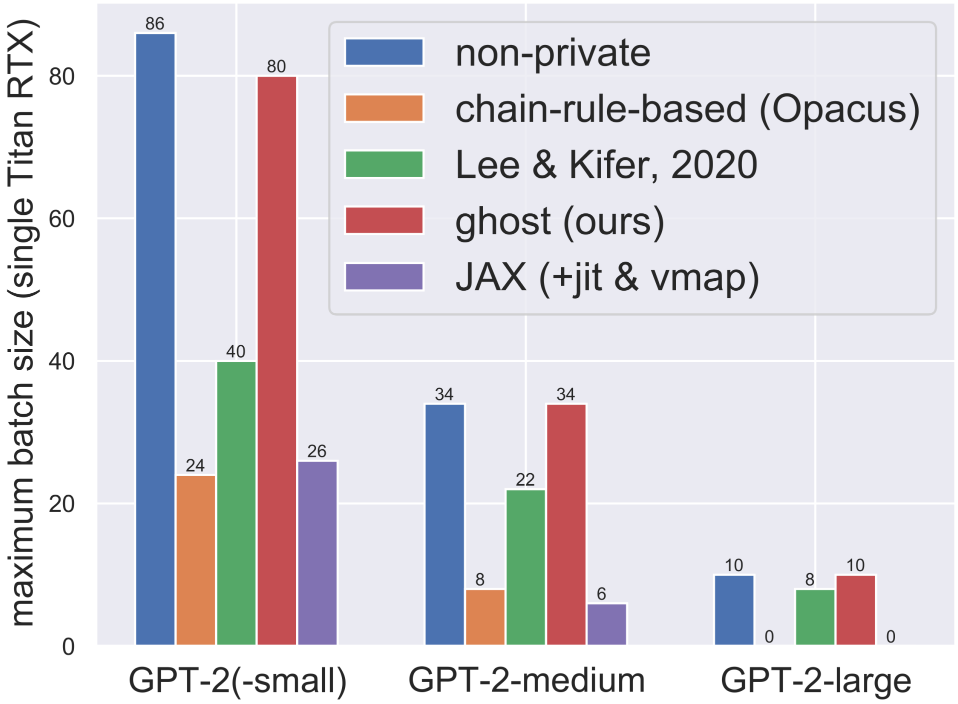
$$\sum_i \mathrm{clip}\big(\nabla f(x_i)\big) + \mathrm{noise} \quad \text{where} \quad \mathrm{clip}(v) \coloneqq \frac{\mathrm{v}}{\max(1, ||v||)}$$

This is memory intensive: we must look at gradients individually to clip them

**Key observation**: compute the scaling $(\max(1, ||v||))$ for cheap, then do a weighted sum

# Breaking the memory barrier for DP-SGD

**Putting it all together**: nearly nonprivate levels of memory consumption




(caveat: implementation dependent, extra backpropagation pass)

# The payoff: usable, private NLP models

Properly tuned, large language models beat state-of-the art in private NLP

Carefully engineered specialized optimizer, SoTA 6/21

Applying basic DP-SGD

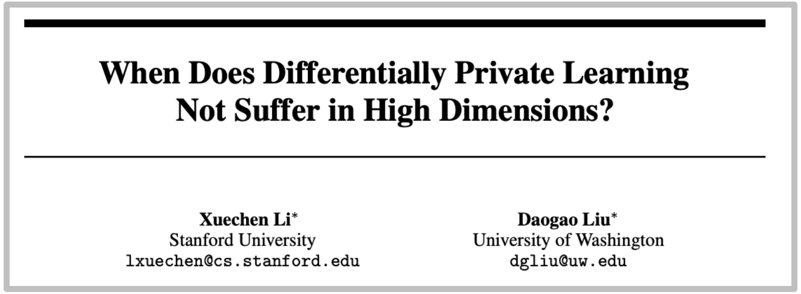
| Method | $\epsilon = 3$ | | | | $\epsilon = 8$ | | | |
|---|---|---|---|---|---|---|---|---|
| | MNLI-(m/mm) | QQP | QNLI | SST-2 | MNLI-(m/mm) | QQP | QNLI | SST-2 |
| RGP (RoBERTa-base) | - | - | - | - | 80.5/79.6 | 85.5 | 87.2 | 91.6 |
| RGP (RoBERTa-large) | - | - | - | - | 86.1/86.0 | 86.7 | 90.0 | 93.0 |
| full (RoBERTa-base) | 82.47/82.10 | 85.41 | 84.62 | 86.12 | 83.30/83.13 | 86.15 | 84.81 | 85.89 |
| full (RoBERTa-large) | 85.53/85.81 | **86.65** | 88.94 | 90.71 | 86.28/86.54 | **87.49** | 89.42 | 90.94 |
| full + infilling (RoBERTa-base) | 82.45/82.99 | 85.56 | 87.42 | 91.86 | 83.20/83.46 | 86.08 | 87.94 | 92.09 |
| full + infilling (RoBERTa-large) | **86.43/86.46** | 86.43 | **90.76** | **93.04** | **87.02/87.26** | 87.47 | **91.10** | **93.81** |
| $\epsilon$ (Gaussian DP + CLT) | 2.52 | 2.52 | 2.00 | 1.73 | 5.83 | 5.85 | 4.75 | 4.33 |

(these numbers are roughly non-private SoTA ~2018)

Large models + 'signal-to-noise' + ghost clipping = simple, provable privacy.

# DP and LLMs – an interesting future?

Understanding the surprising effectiveness of DP-SGD finetuning for large models



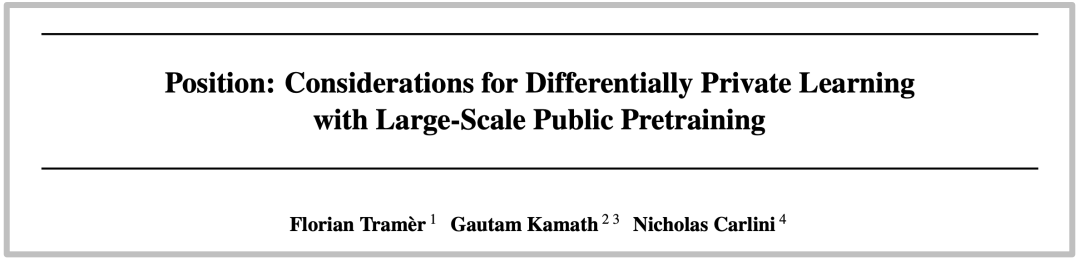
**When Does Differentially Private Learning Not Suffer in High Dimensions?**

**Xuechen Li*** Stanford University lxuechen@cs.stanford.edu

**Daogao Liu*** University of Washington dgliu@uw.edu

Bounds that depend on 'effective' dimensionality of gradients

How do we balance privacy for fine-tuning vs costs in pre-training?



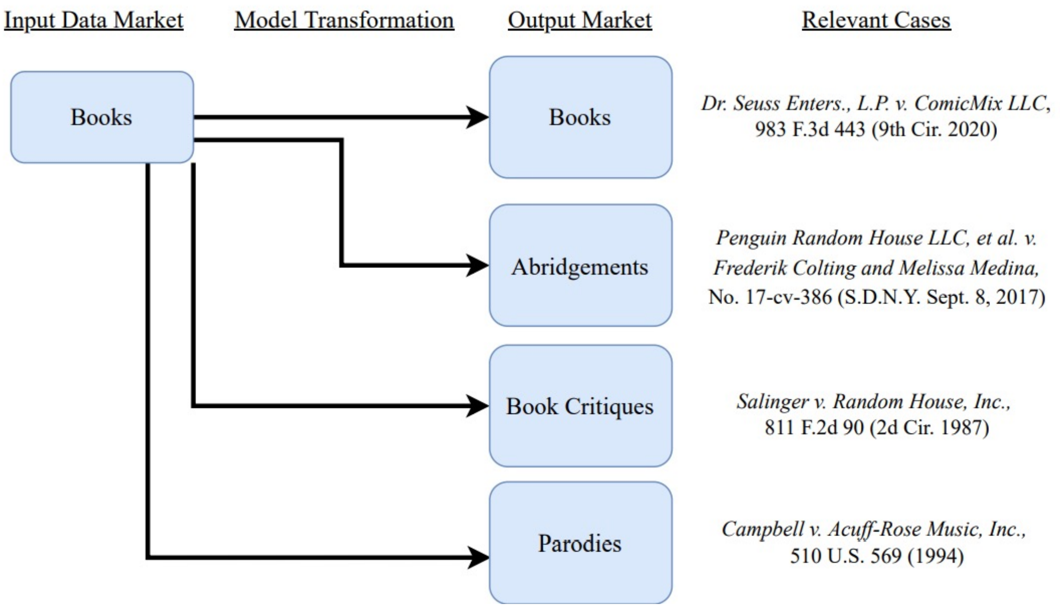
**Position: Considerations for Differentially Private Learning with Large-Scale Public Pretraining**

**Florian Tramèr** [1] **Gautam Kamath** [2 3] **Nicholas Carlini** [4]

**Directions**: training on 'permissive' data, using datastores / retrieval

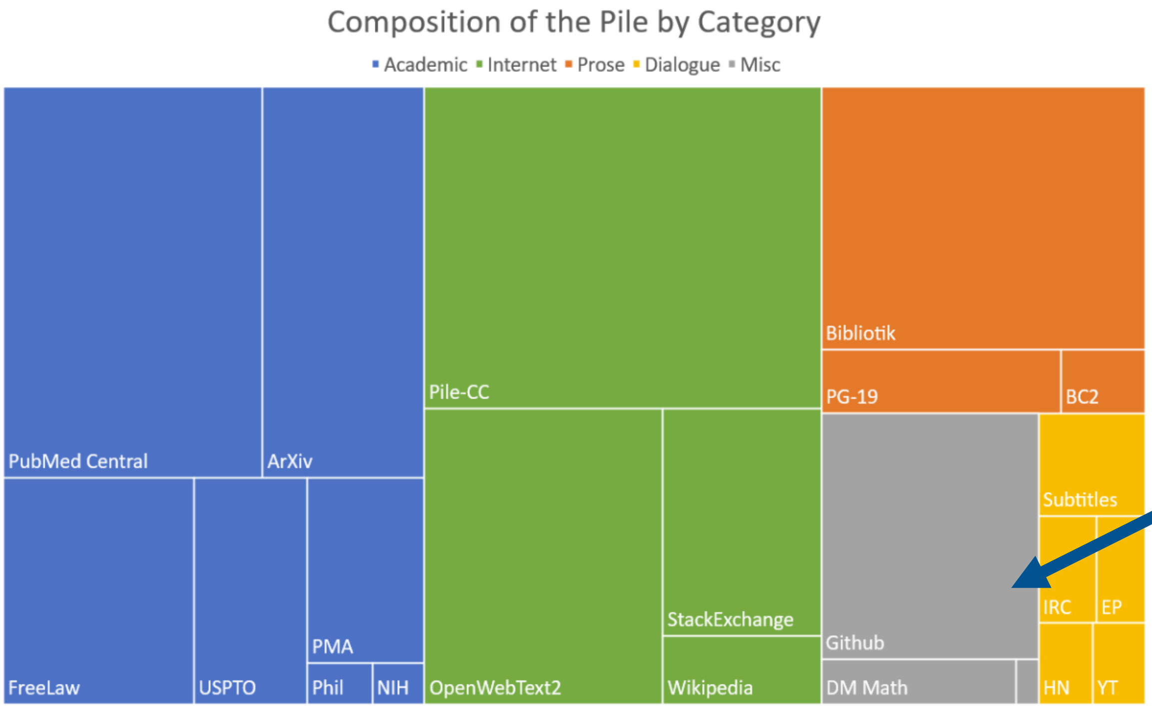
# Limitation: verbatim memorization and fair use

*Fair use* is more than just avoiding copying




[Henderson+ 2023]

# What is in the training data of a LLM?

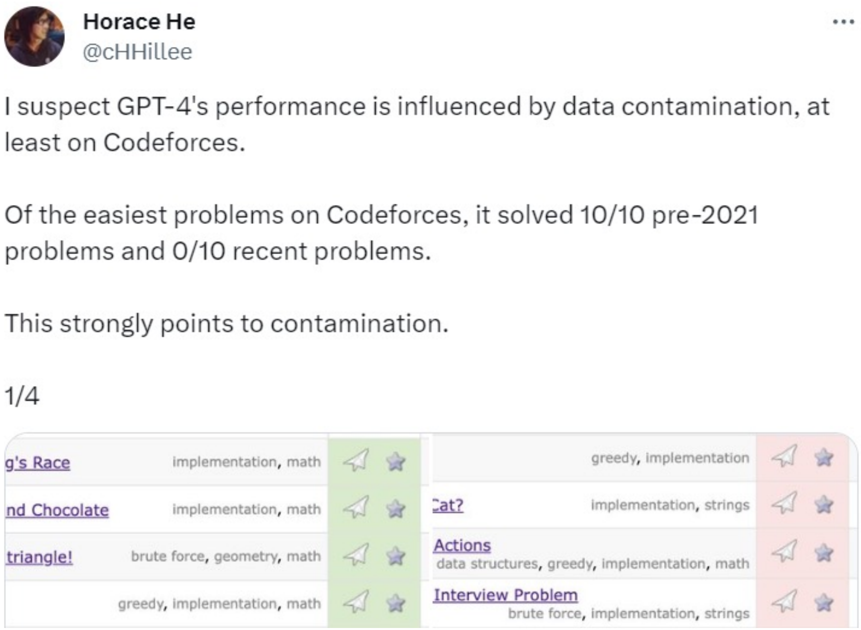


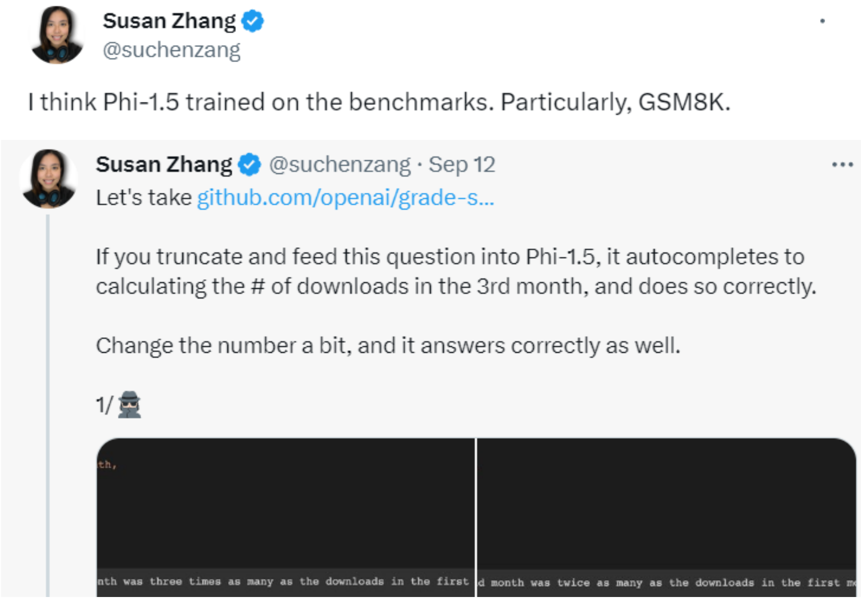

.. But maybe your test set is in here?

CODEFORCES
Sponsored by TON

Language models derive their strength from massive, lightly-curated pretraining data

# Lots of public discourse on contamination..



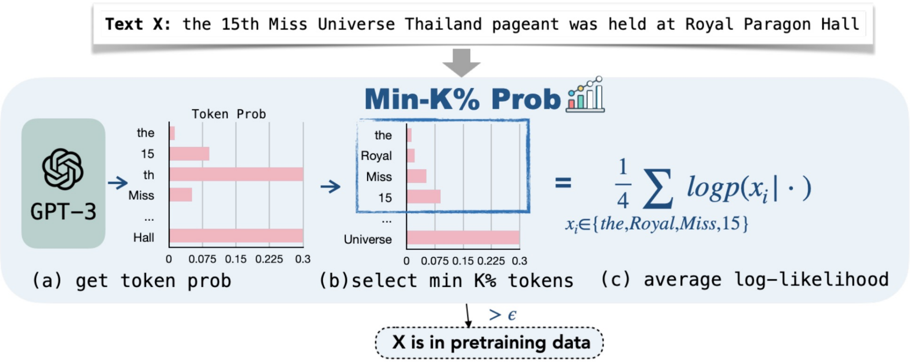




Some circumstantial evidence toward contamination – but no proof or audits

# Lots of methods for detecting membership – mostly heuristic

## Did ChatGPT cheat on your test?

**Authors:** Oscar Sainz, Jon Ander Campos, Iker García-Ferrero, Julen Etxaniz, Eneko Agirre




**Did ChatGPT cheat on your test:**
Ask LLMs to generate the first example

**Min-k-prob**: is the minimum 'too likely'?

# Can we provably detect (some) contamination?

**Goal:** provide a provable (false positive) guarantee for detecting test set contamination.

## Setup:

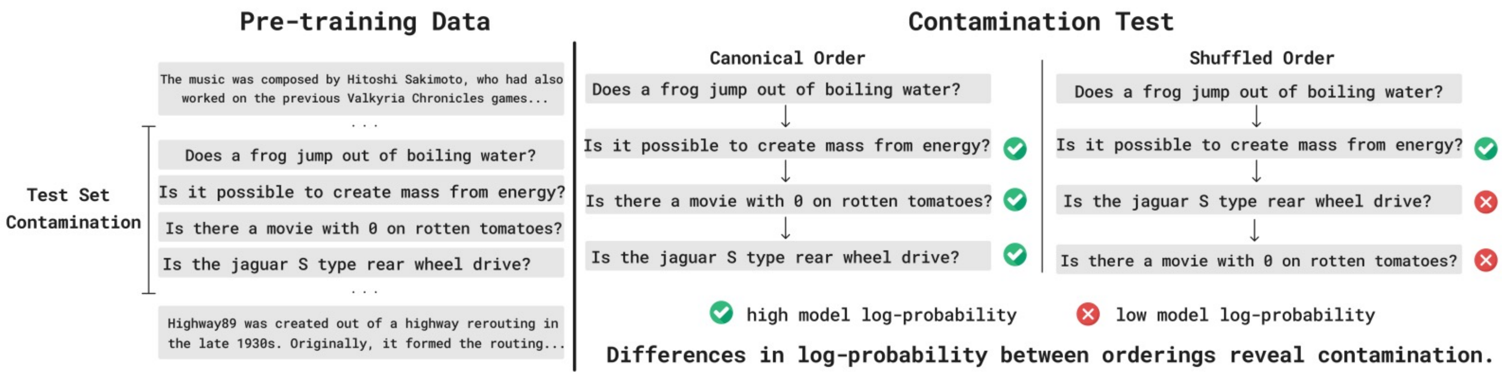
**Given** a test set and access to log-probabilities from a language model

**Return** a statistical test for contamination with type-I error rate at most $\alpha$

1. The null hypothesis is that the test set and model are independent r.v.s
2. The error guarantee should hold w.r.t draws of the datasets

# Our approach: exploit the exchangeability of datasets

Starting observation: most test sets are *exchangeable.*




**Key idea:**

Language model preference for 'canonical' orderings *must* come from contamination

[Oren+ 2024]

# Our approach: exploit the exchangeability of datasets

**Key idea:**

Language model preference for 'canonical' orderings *must* come from contamination

**Proposition 1.** *Let $seq(X)$ be a function that takes a dataset $X$ and concatenates the examples to produce a sequence, and let $X_\pi$ be a random permutation of the examples of $X$ where $\pi$ is drawn uniformly from the permutation group. For an exchangeable dataset $X$ and under $H_0$,*

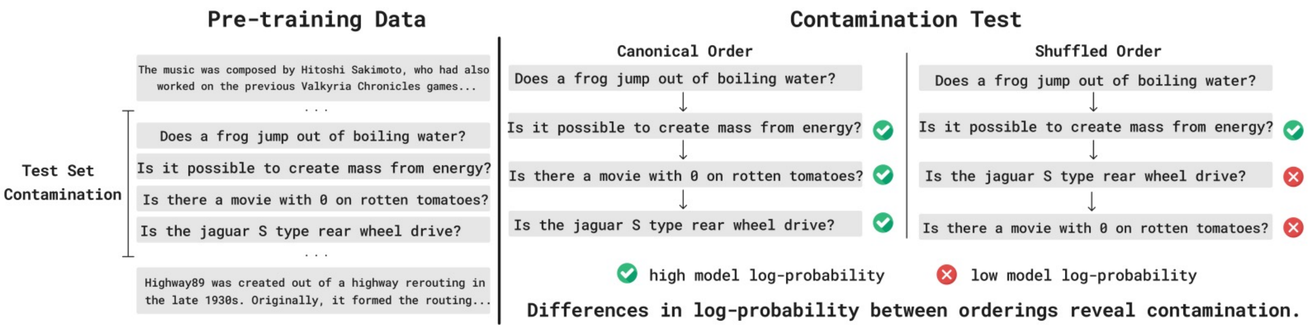
$$\log p_\theta(seq(X)) \overset{d}{=} \log p_\theta(seq(X_\pi)).$$

**Proof** This follows directly from the definitions of exchangability and $H_0$. Since $X$ is exchangable, $\text{seq}(X) \overset{d}{=} \text{seq}(X_\pi)$ and by the independence of $\theta$ from $X$ under $H_0$, we know that $(\theta, \text{seq}(X)) \overset{d}{=} (\theta, \text{seq}(X_\pi))$. Thus, the pushforward under $\log p_\theta(\text{seq}(X))$ must have the same invariance property. $\square$

[Oren+ 2024]

# This leads to a simple test for contamination

Shuffle and compute log probs




Gives exact p-values

$$\hat{p} := \frac{\sum_{i=1}^{m} \mathbb{1}\{\log p_\theta(\mathrm{seq}(X)) < \log p_\theta(\mathrm{seq}(X_{\pi_m}))\} + 1}{m+1}.$$

[Oren+ 2024]

# Result #1 – detection in known settings

Can we detect known contamination?

We pretrained a 1.4B param, 20B token LM w/ known contamination

| Name | Size | Dup Count | Permutation p | Sharded p |
|---|---|---|---|---|
| BoolQ | 1000 | 1 | 0.099 | 0.156 |
| HellaSwag | 1000 | 1 | 0.485 | 0.478 |
| OpenbookQA | 500 | 1 | 0.544 | 0.462 |
| MNLI | 1000 | 10 | **0.009** | **1.96e-11** |
| Natural Questions | 1000 | 10 | **0.009** | **1e-38** |
| TruthfulQA | 1000 | 10 | **0.009** | **3.43e-13** |
| PIQA | 1000 | 50 | **0.009** | **1e-38** |
| MMLU Pro. Psychology | 611 | 50 | **0.009** | **1e-38** |
| MMLU Pro. Law | 1533 | 50 | **0.009** | **1e-38** |
| MMLU H.S. Psychology | 544 | 100 | **0.009** | **1e-38** |

100% detection rate on ≥ 10 duplication count datasets

[Oren+ 2024]

# Result #2 – contamination in the wild

| Dataset | Size | LLaMA2-7B | Mistral-7B | Pythia-1.4B | GPT-2 XL | BioMedLM |
|---|---|---|---|---|---|---|
| AI2-ARC | 2376 | 0.318 | **0.001** | 0.686 | 0.929 | 0.795 |
| BoolQ | 3270 | 0.421 | 0.543 | 0.861 | 0.903 | 0.946 |
| GSM8K | 1319 | 0.594 | 0.507 | 0.619 | 0.770 | 0.975 |
| LAMBADA | 5000 | 0.284 | 0.944 | 0.969 | 0.084 | 0.427 |
| NaturalQA | 1769 | 0.912 | 0.700 | 0.948 | 0.463 | 0.595 |
| OpenBookQA | 500 | 0.513 | 0.638 | 0.364 | 0.902 | 0.236 |
| PIQA | 3084 | 0.877 | 0.966 | 0.956 | 0.959 | 0.619 |
| MMLU$^\dagger$ | – | 0.014 | 0.011 | 0.362 | – | – |

- No evidence of contamination (except ARC+Mistral)
- MMLU tests consistent with Touvron et al’s contamination test

[Oren+ 2024]

# Privacy and memorization

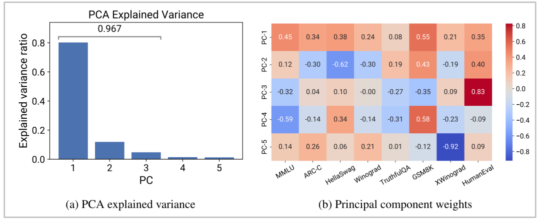
**What's new:** pretraining – both enabling DP (finetuning) and new privacy threats

**What's not really new:**

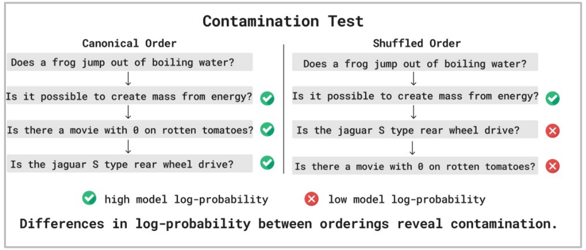
- The desire and need for precise guarantees (DP) in privacy type settings.
- Costs of privacy – both computationally and in performance
- Difficulty of membership inference, and the basic primitives (randomized canaries)

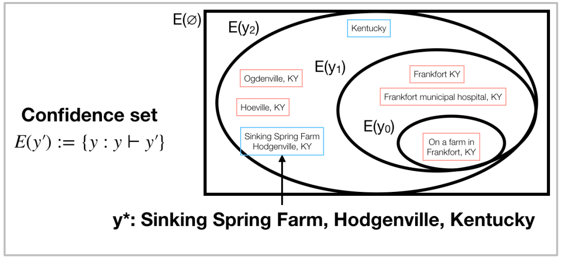
# Part 3: uncertainty quantification

Language models confidently assert false facts
can we fix that?



(a) PCA explained variance (b) Principal component weights




**Part 1: Robustness** **Part 2: Privacy/memorization** **Part 3: Uncertainty**

# What LLMs know (and don't know) is a core open problem

REUTERS®

Australian mayor readies world's first defamation lawsuit over ChatGPT content

By Byron Kaye

Air Canada ordered to pay customer who was misled by airline's chatbot

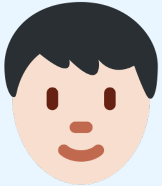
**Company claimed its chatbot 'was responsible for its own actions' when giving wrong information about bereavement fare**


LLMs are *remarkably* good at many things – even closed-book QA

.. But they aggressively make things up for things they don't know

# What can we do about hallucinations?

We don’t necessarily know everything in the world



Are there any strictly proper, bounded scoring rules?

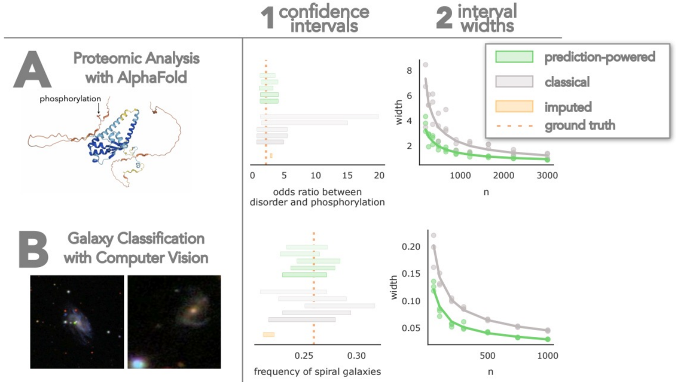
I don’t know anything about scoring rules.

.. but we generally know how to back off and communicate uncertainty

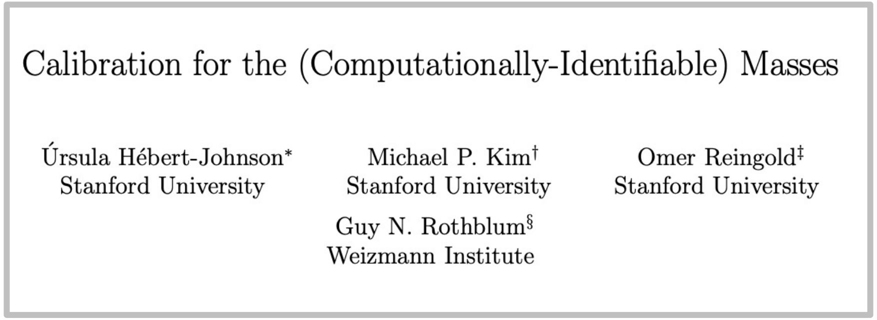
# Broader context: uncertainty quantification

Uncertainty quantification in general is a rich area

## Conformal prediction


## Multicalibration



Calibration for the (Computationally-Identifiable) Masses

Úrsula Hébert-Johnson* Stanford University

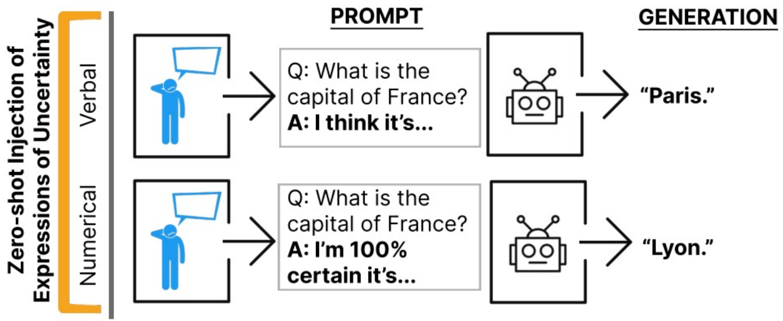
Michael P. Kim† Stanford University

Omer Reingold‡ Stanford University

Guy N. Rothblum§ Weizmann Institute

# Challenge: Bridging uncertainty estimation and LLMs

Language can carry rich notions of uncertainty: **explicitly**, via hedges




[Zhou, Jurfasky, Hashimoto 2023]

And **implicitly** via entailments and pragmatics

"It's going to rain tomorrow" →

- "It's going to **a lot** rain tomorrow"
- "It's going to **drizzle** tomorrow"
- "It'll pour tomorrow"

[Many exciting works on LMs + uncertainty – see Mielke et al 2022, Lin et al 2022 (and recent works at ACL/ICML!)]

# One example: conformal prediction for factuality

Can conformal prediction help with hallucination?

Yes, if we bridge *correctness* and *uncertainty quantification*

**Q: Where was Abe Lincoln Born?**
**A: Sinking Spring Farm, Hodgenville, Kentucky**

**LM outputs**

$y_0$ On a farm in Frankfort, Kentucky

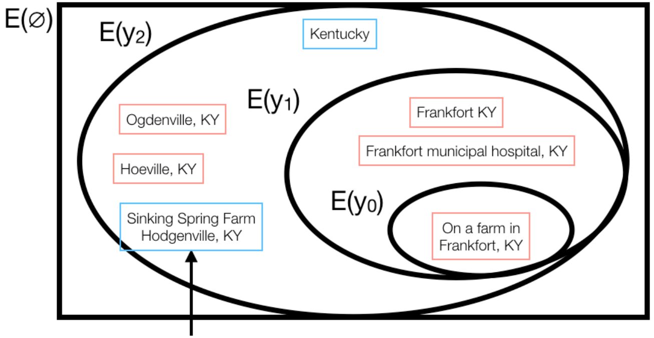
$y_1$ Frankfort, Kentucky

$y_2$ Kentucky

(Red shows falsehoods, blue is correct)

**Confidence set**

$E(y') := \{y : y \vdash y'\}$




**y*: Sinking Spring Farm, Hodgenville, Kentucky**

[Mohri and Hashimoto 2024]

# Algorithmic instantiation: ‘back off’ claims until correct

**The algorithm**:

1. Construct a sequence of “less specific” $y_t$
2. Score each $y_t$ via a confidence measure $S(y_t)$
3. Select a cutoff $\tau$ for $S(y_t)$ using conformal prediction such that

$$P\big(y^* \in E(y_\tau)\big) \geq 1 - \alpha$$

At inference time return $y_\tau$ -- this has a $1 - \alpha$ correctness guarantee

$\mathsf{F}_0(x)$ = `Michael Jordan (born February 17, 1063), is a former professional basketball player.`

$\mathsf{F}_1(x)$ = `Michael Jordan is a former professional basketball player.`

$\mathsf{F}_2(x) = \varnothing.$

[Mohri and Hashimoto 2024]

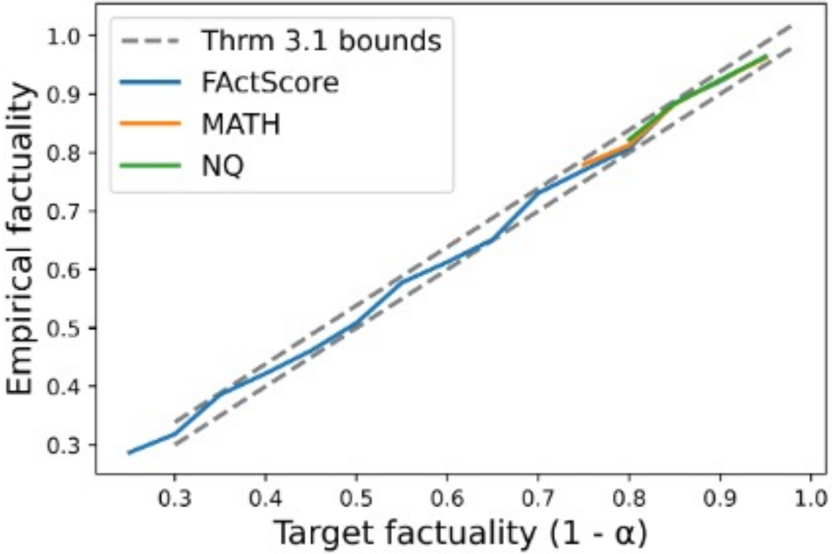
# Does it work?

**Theorem 4.1.** *Let $\{X_i, Y_i^*\}_{i=1}^{n+1}$ be exchangeable, $\mathsf{F}_t$ be sound, and $\hat{q}_\alpha$ be defined as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$th quantile of the scores $\{r(X_i, Y_i^*)\}_{i=1}^{n}$, which we assume to be distinct without loss of generality. Then, for $\alpha \in [\frac{1}{n+1}, 1]$, the following lower bound holds:*

$$\mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))) \geq 1 - \alpha.$$

*If $\mathsf{E}(\mathsf{F}_t(\cdot))$ follows the nested property, then the following upper bound holds:*

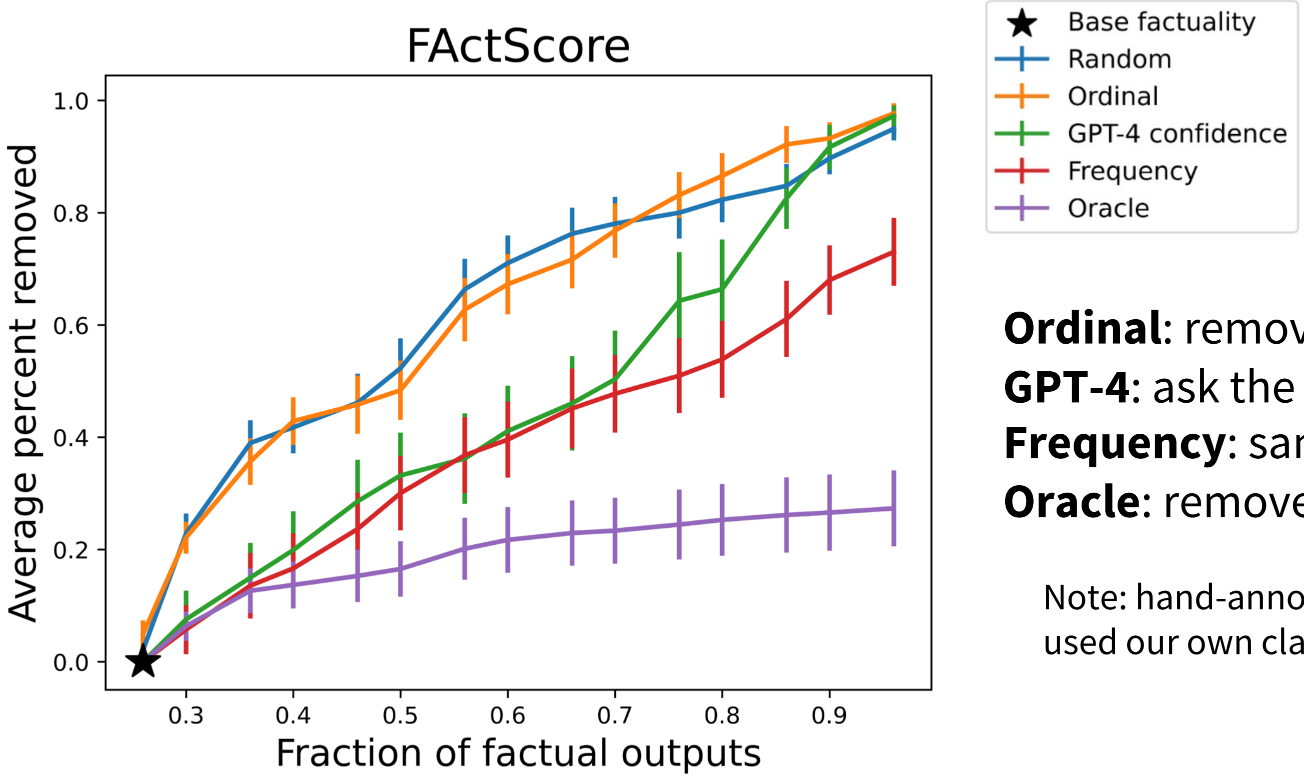
$$1 - \alpha + \frac{1}{n+1} \geq \mathbb{P}(Y_{n+1}^* \in \mathsf{E}(\mathsf{F}_{\hat{q}_\alpha}(X_{n+1}))).$$




[Mohri and Hashimoto 2024]

Setup – split data into half, calibrate on first half, measure marginal correctness on other half

# Factuality on FactScore from 0% to 80%




**Ordinal**: remove the last claims
**GPT-4**: ask the LM for confidence
**Frequency**: sample + count claim freq.
**Oracle**: remove non-factual first

Note: hand-annotated first 50 samples + used our own claim splits for data quality

[Mohri and Hashimoto 2024]

# Random FactScore examples

**Original**

Luis Guillermo Rivera is a professional athlete from Colombia who specializes in high jump. He was born on March 20, 1989 in Apartado, Colombia. Rivera has represented Colombia in several international competitions, including the World Championships in Athletics and the Pan American Games. He won the gold medal in the high jump at the 2013 South American Championships in Athletics. His personal best in the high jump is 2.31 meters, which he achieved in 2013.

Roxana Diaz is a Venezuelan athlete who specializes in the 400 meters hurdles. She represented her country in various international competitions, including the World Championships and the Pan American Games. Diaz is known for her speed, agility, and determination on the track. She has won several medals throughout her career, making her one of the most successful athletes in her country.

Sharad Kumar is an Indian para-athlete who competes in high jump events. He was born on March 1, 1992, in Bihar, India. Kumar was diagnosed with polio at a young age, which affected his left leg. Despite this, he pursued his passion for sports and began training as a high jumper.\n\nKumar has represented India in several international competitions and has won numerous medals. He won a gold medal at the 2014 Asian Para Games in Incheon, South Korea, and a silver medal at the 2018 Asian Para Games in Jakarta, Indonesia. He also won a bronze medal at the 2017 World Para Athletics Championships in London.\nnIn 2021, Kumar won a bronze medal in the men's high jump T42 category at the Tokyo Paralympics, marking his first Paralympic medal. His personal best in high jump is 1.90 meters, which he achieved at the 2017 World Para Athletics Championships.

Katsunosuke Hori is a Japanese actor and voice actor. He was born on November 28, 1941 in Tokyo, Japan. He is best known for his work in anime and video games, including roles in "Neon Genesis Evangelion", "Ghost in the Shell: Stand Alone Complex", and "Final Fantasy VII: Advent Children". He is also known for his dubbing roles in foreign films, such as the voice of Gandalf in the Japanese version of "The Lord of the Rings" trilogy.

Ra Jong-yil is a South Korean diplomat and academic. He served as the South Korean ambassador to the United Kingdom and Japan. He also served as the deputy director of the National Intelligence Service. As an academic, he has taught at various universities and written several books on North Korea and international relations.

**Conformal Factuality**

I'm uncertain about the specific details, and I recommend using a search engine like Google for more accurate and up-to-date information.

Roxana Diaz is an athlete who has represented Venezuela in various international competitions. She has also participated in the World Championships.

Sharad Kumar is an Indian para-athlete who was born on March 1, 1992, in Bihar, India. He competes in high jump events and despite his condition, he pursued his passion for sports and began training as a high jumper. Sharad Kumar has represented India in several international competitions and has won numerous medals in his career. Most notably, in 2021, he won a bronze medal in the men's high jump T42 category at the Tokyo Paralympics.

Katsunosuke Hori is a Japanese actor and voice actor. He is known for his work in anime and video games.

Ra Jong-yil is a South Korean diplomat and academic who has served as the South Korean ambassador to both the United Kingdom and Japan. In addition to his diplomatic work, he has written several books on North Korea and international relations.

[Mohri and Hashimoto 2024]

# Broader connections between conformal prediction and LLMs

**Large language model validity via enhanced conformal prediction methods**

**John J. Cherian**
Department of Statistics
Stanford University
Stanford, CA 94305
jcherian@stanford.edu

**Isaac Gibbs**
Department of Statistics
Stanford University
Stanford, CA 94305
igibbs@stanford.edu

**Emmanuel J. Candès**
Department of Statistics
Department of Mathematics
Stanford University
Stanford, CA 94305
candes@stanford.edu

Adaptively weaken guarantees,
optimize the scoring function

**Multi-group Uncertainty Quantification for Long-form Text Generation**

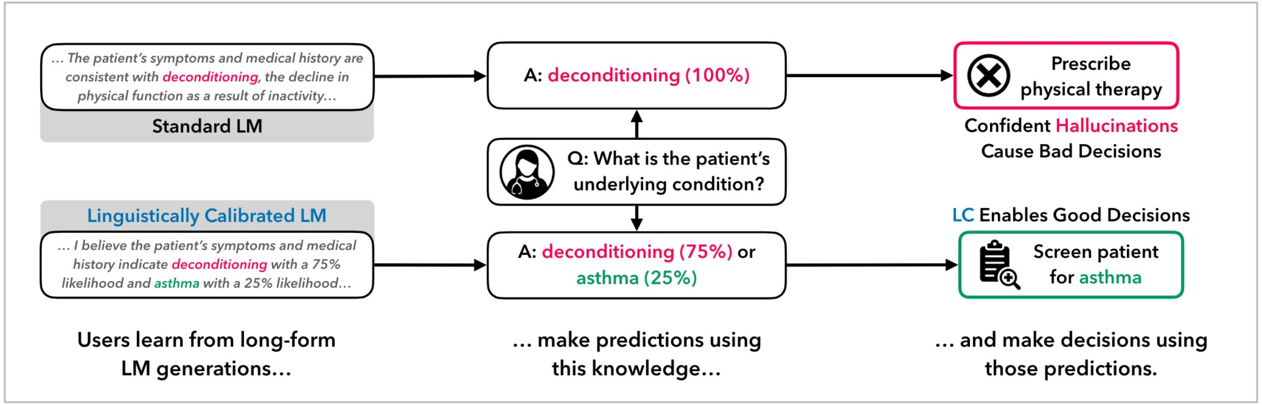
**Terrance Liu**
Carnegie Mellon University
Pittsburgh, PA 15213
terrancl@cs.cmu.edu

**Zhiwei Steven Wu**
Carnegie Mellon University
Pittsburgh, PA 15213
zstevenwu@cmu.edu

Robust to multiple environments

# Other directions: training LMs to express numerical uncertainty

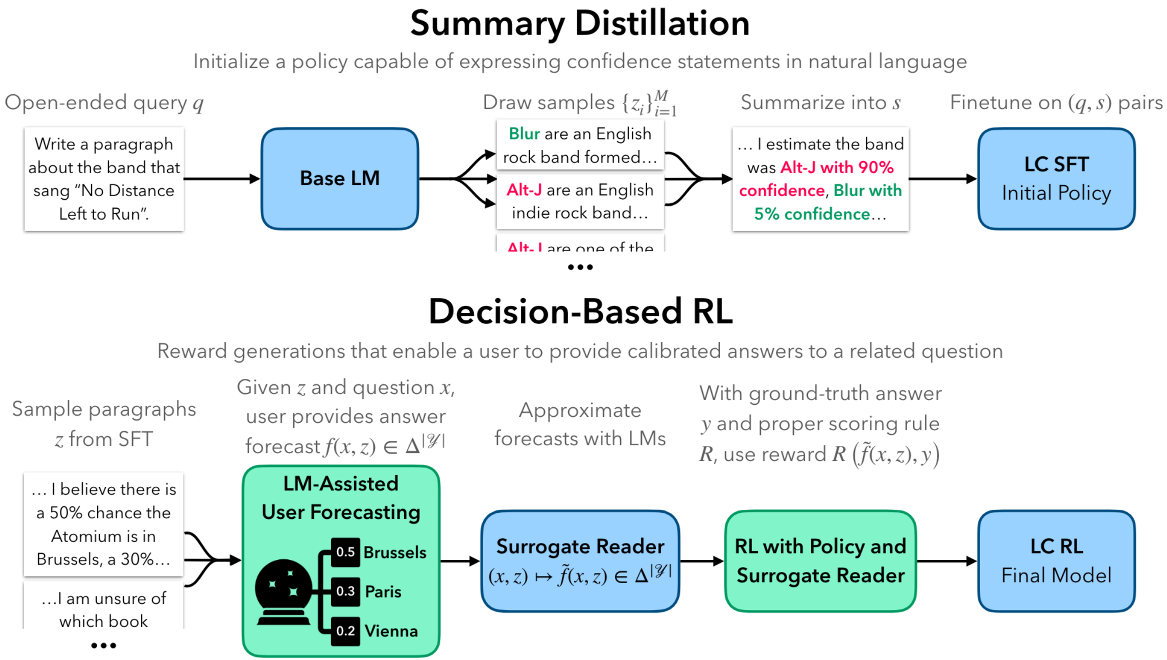
Linguistic uncertainty is a powerful tool to deal with lack of knowledge




How can we optimize language models to provide useful or *calibrated* uncertainty in language?

# Decision-based losses naturally lead to calibration

Simple but effective trick – optimize LMs against proper scoring rules (e.g. log-loss) for a downstream decision task (e.g. QA on LLM outputs)




[Band+ 2024]

# Uncertainty and LLMs

**What's new:** language as an output space – high dimensional, and 'in-band' uncertainty

**What's not really new:**

- Tools and challenges – conformal / multicalibration
- Impossibility results - individual coverage, or coverage under dist. shifts
- Usefulness of thinking about the interactions between uncertainty and decisions

# Putting it together

**The *classic* trustworthiness problems really remain problems**

- Robustness and generalization
- Privacy
- Uncertainty quantification

**The tools and observations from the past carry over, though sometimes with twists**

- Data augmentation, difficulty of distributional generalization
- Surprising effectiveness of DP, membership inference
- Conformal inference, proper scoring rules.