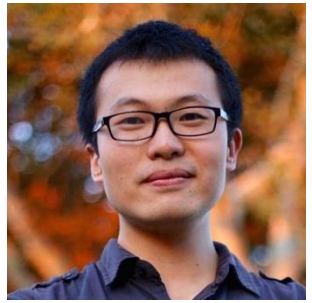
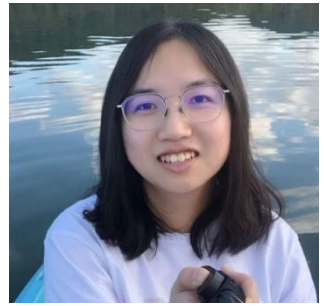
DPrivBench: Benchmarking LLMs' Reasoning for Differential Privacy





Erchi Wang[1,*] Pengrun Huang[1] Eli Chien[2] Om Thakkar[3] Kamalika Chaudhuri[1] Yu-Xiang Wang[1] Ruihan Wu[1,3,*]

[1]University of California, San Diego [2]National Taiwan University [3]OpenAI * Project Lead

Outline

- Why Benchmark Differential Privacy
- What DPrivBench Tests
- How LLMs Perform
- What Comes Next

How to Make DP more accessible?

- Non-DP experts face a steep learning curve, even when provided with DP libraries.

DP novices encountered 102 stuck events and resolved only 57%

Evaluating the Usability of Differential Privacy Tools with Data Practitioners

Ivoline C. Ngong, Brad Stenger, Joseph P. Near, Yuanyuan Feng

Differential privacy (DP) has become the gold standard in privacy-preserving data analytics, but implementing it in real-world datasets and systems remains challenging. Recently developed DP tools aim to make DP implementation easier, but limited research has investigated these DP tools' usability. Through a usability study with 24 US data practitioners with varying prior DP knowledge, we evaluated the usability of four Python-based open-source DP tools: DiffPrivLib, Tumult Analytics, PipelineDP, and OpenDP. Our results suggest that using DP tools in this study may help DP novices better understand DP; that Application Programming Interface (API) design and documentation are vital for successful DP implementation; and that user satisfaction correlates with how well participants completed study tasks with these DP tools. We provide evidence-based recommendations to improve DP tools' usability to broaden DP adoption.

How to Make DP more accessible?

- DP is error-prone, even for experts

Algorithm 3

Input: D, Q, Δ, T, c.
$\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}(\Delta/\epsilon_1)$,
$\epsilon_2 = \epsilon - \epsilon_1, \quad \text{count} = 0$
for each query $q_i \in Q$ **do**
  $\nu_i = \mathsf{Lap}(c\Delta/\epsilon_2)$
  if $q_i(D) + \nu_i \geq T + \rho$ **then**
    Output $a_i = q_i(D) + \nu_i$
    count = count + 1, **Abort** if count $\geq c$.
  else
    Output $a_i = \perp$

Algorithm 5

Input: D, Q, Δ, T.
$\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}(\Delta/\epsilon_1)$
$\epsilon_2 = \epsilon - \epsilon_1$
for each query $q_i \in Q$ **do**
  $\nu_i = 0$
  if $q_i(D) + \nu_i \geq T + \rho$ **then**
    Output $a_i = \top$

  else
    Output $a_i = \perp$

Algorithm 6

Input: $D, Q, \Delta, \mathbf{T} = T_1, T_2, \cdots$.
$\epsilon_1 = \epsilon/2, \quad \rho = \mathsf{Lap}(\Delta/\epsilon_1)$
$\epsilon_2 = \epsilon - \epsilon_1$
for each query $q_i \in Q$ **do**
  $\nu_i = \mathsf{Lap}(\Delta/\epsilon_2)$
  if $q_i(D) + \nu_i \geq T_i + \rho$ **then**
    Output $a_i = \top$

  else
    Output $a_i = \perp$

Example 1: Buggy Sparse Vector Technique Implementations (Lyu et al., 2016)

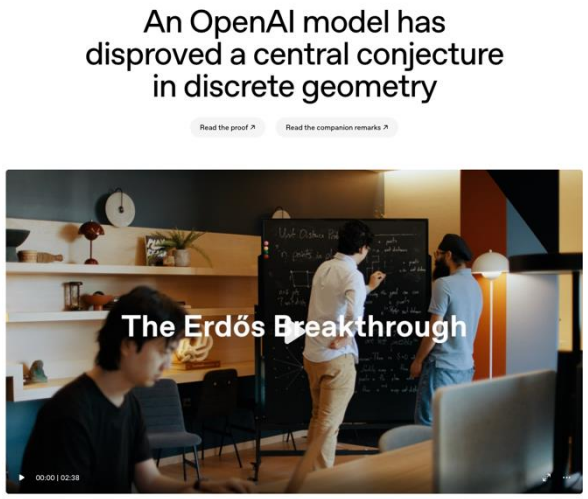
How to Make DP more accessible?

- DP is error-prone, even for experts

  Similar issues in DP implementations were also widely reported:

  [CEDBF26] Privacy in Theory, Bugs in Practice: Grey-Box Auditing of Differential Privacy Libraries, PETS 2026

LLMs Looks Like Natural DP Assistants



Google DeepMind 2026-5-22

Advancing Mathematics Research with AI-Driven Formal Proof Search

George Tsoukalas[1†], Anton Kovsharov[1†], Sergey Shirobokov[1†], Anja Surina[1], Moritz Firsching[1], Gergely Bérczi[2], Francisco J. R. Ruiz[1], Arun Suggala[1], Adam Zsolt Wagner[1], Eric Wieser[1], Lei Yu[1], Aja Huang[1], Miklós Z. Horváth[1], Andrew Ferrauiolo[1], Henryk Michalewski[1], Codrut Grosu[3], Thomas Hubert[1], Matej Balog[1], Pushmeet Kohli[1] and Swarat Chaudhuri[1†]
[1]Google DeepMind[1], [2]Aarhus University, [3]Google

Large language models (LLMs) increasingly excel at mathematical reasoning, but their unreliability limits their utility in mathematics research. A mitigation is using LLMs to generate formal proofs in languages like Lean. We perform the first large-scale evaluation of this method's ability to solve open problems. Our most capable agent autonomously resolved 9 of 353 open Erdős problems at the per-problem cost of a few hundred dollars, proved 44/492 OEIS conjectures, and is being deployed in combinatorics, optimization, graph theory, algebraic geometry, and quantum optics research. A basic agent alternating LLM-based generation with Lean-based verification replicated the Erdős successes but proved costlier on the hardest problems. These findings demonstrate the power of AI-aided formal proof search and shed light on the agent designs that enable it.

Claude's Cycles

Don Knuth, Stanford Computer Science Department
(28 February 2026; revised 14 April 2026)

Shock! Shock! I learned yesterday that an open problem I'd been working on for several weeks had just been solved by Claude Opus 4.6 — Anthropic's hybrid reasoning model that had been released three weeks earlier! It seems that I'll have to revise my opinions about "generative AI" one of these days. What a joy it is to learn not only that my conjecture has a nice solution but also to celebrate this dramatic advance in automatic deduction and creative problem solving. I'll try to tell the story briefly in this note.

Here's the problem, which came up while I was writing about directed Hamiltonian cycles for a future volume of *The Art of Computer Programming*:

> Consider the digraph with m^3 vertices ijk for $0 \le i, j, k < m$, and three arcs from each vertex, namely to i^+jk, ij^+k, and ijk^+, where $i^+ = (i+1) \bmod m$. Try to find a general decomposition of the arcs into three directed m^3-cycles, for all $m > 2$.

I had solved the problem for $m = 3$, and asked for a generalization as part of the answer to an exercise in [3]. My friend Filip Stappers rose to the challenge, and empirically discovered solutions for $4 \le m \le 16$; therefore it became highly likely that the desired decompositions do exist, except when $m \le 2$.

Indeed, it was Filip who had the gumption to pose this question to Claude, using exactly the wording above. He also gave guidance/coaching, instructing Claude to summarize its ongoing progress:

> ** After EVERY `exploreXX.py` run, IMMEDIATELY update this file [`plan.md`] before doing anything else. ** No exceptions. Do not start the next exploration until the previous one is documented here.

Strong at Math + Coding — but can they verify DP guarantee?

DP reasoning is subtle

A plausible proof is not enough — one wrong sensitivity, assumption, or accounting step can invalidate the privacy guarantee.

Can LLM correctly reason about DP guarantees?

This is what DPrivBench evaluates.

Outline

- Why Benchmark DP
- What DPrivBench Tests
- How LLMs Perform
- What Comes Next

Two levels of DPrivBench

Category 1: Mechanism-level

Core DP building blocks

- Laplace / Gaussian / Report-Noisy-Max (Gumbel, Laplace, Exponential)
- sensitivity + noise calibration

Category 2: Literature-Derived DP Claims

"Research-level" DP reasoning

- Claims from DP literature
  With a focus on classical Algos.
- general DP reasoning:
  accounting, theorem using, neighboring relation …

Examples of DPrivBench

Category 1: Laplace Mechanism

Suppose $\mathrm{X} = (x_1, \ldots, x_n) \in [0,1]^n$. Define $f(\mathrm{X}) = \sum_{i=1}^{n} x_i$ and define $\tilde{f}(\mathrm{X}) = f(X) + Lap\left(1 \cdot \frac{1}{\varepsilon}\right)$. The neighboring datasets X and X′ differ in at most one element. Does $\tilde{f}$ satisfy ε-DP?

GT Label: Yes

Category 2: Objective Perturbation

Suppose that for all $z \in \mathcal{Z}$, $\ell(\cdot, z)$ is twice-differentiable, and the rank of its Hessian $\nabla^2 \ell(\mathbf{w}, z)$ at any $\mathbf{w} \in \mathcal{W}$ is at most 1. Also assume that the smoothness parameter satisfies $\beta \leq \varepsilon n \lambda$.
Is the following algorithm (ε, δ)-differentially private?

Algorithm 1: $\mathcal{A}_{\text{ObjP}}$: Objective Perturbation

Input: Private dataset $S = (z_1, \ldots, z_n) \in \mathcal{Z}^n$, L-Lipschitz, β-smooth, convex loss function ℓ, convex set $\mathcal{W} \subseteq \mathbb{R}^d$, privacy parameters $\varepsilon \leq 1$, $\delta \leq 1/n^2$, regularization parameter λ.

1: Sample $\mathbf{G} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \mathbf{I}_d)$, where $\sigma^2 = \frac{10L^2 \log(1/\delta)}{\varepsilon^2}$;

2: return $\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathcal{W}} \hat{\mathcal{L}}(\mathbf{w}; S) + \frac{\langle \mathbf{G}, \mathbf{w} \rangle}{n} + \lambda \|\mathbf{w}\|^2$, where $\hat{\mathcal{L}}(\mathbf{w}; S) \triangleq \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}, z_i)$;

GT Label: Yes

Short Proof and Reference

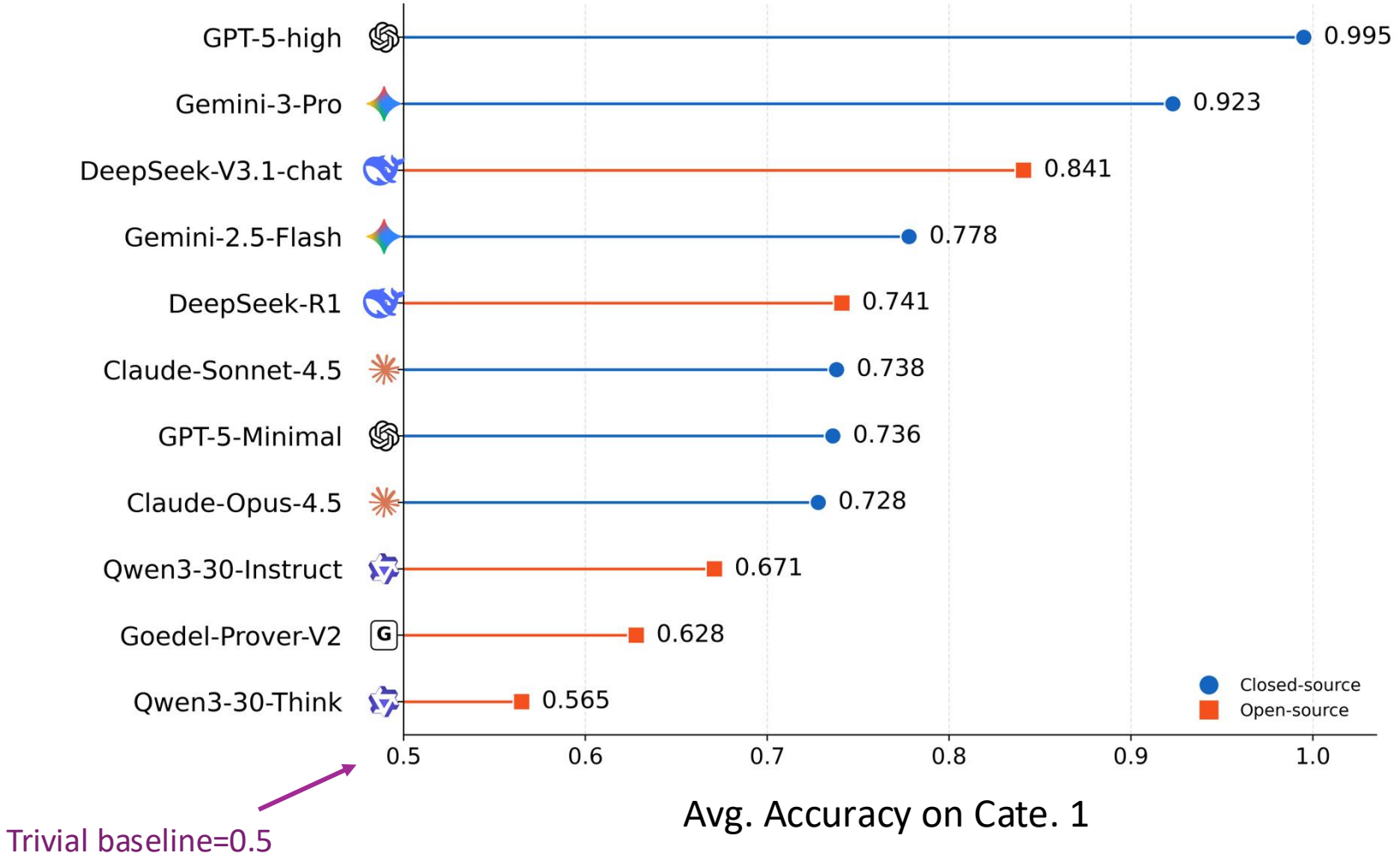
Outline

- Why Benchmark DP
- What DPrivBench Test
- How LLMs Perform
- What Comes Next

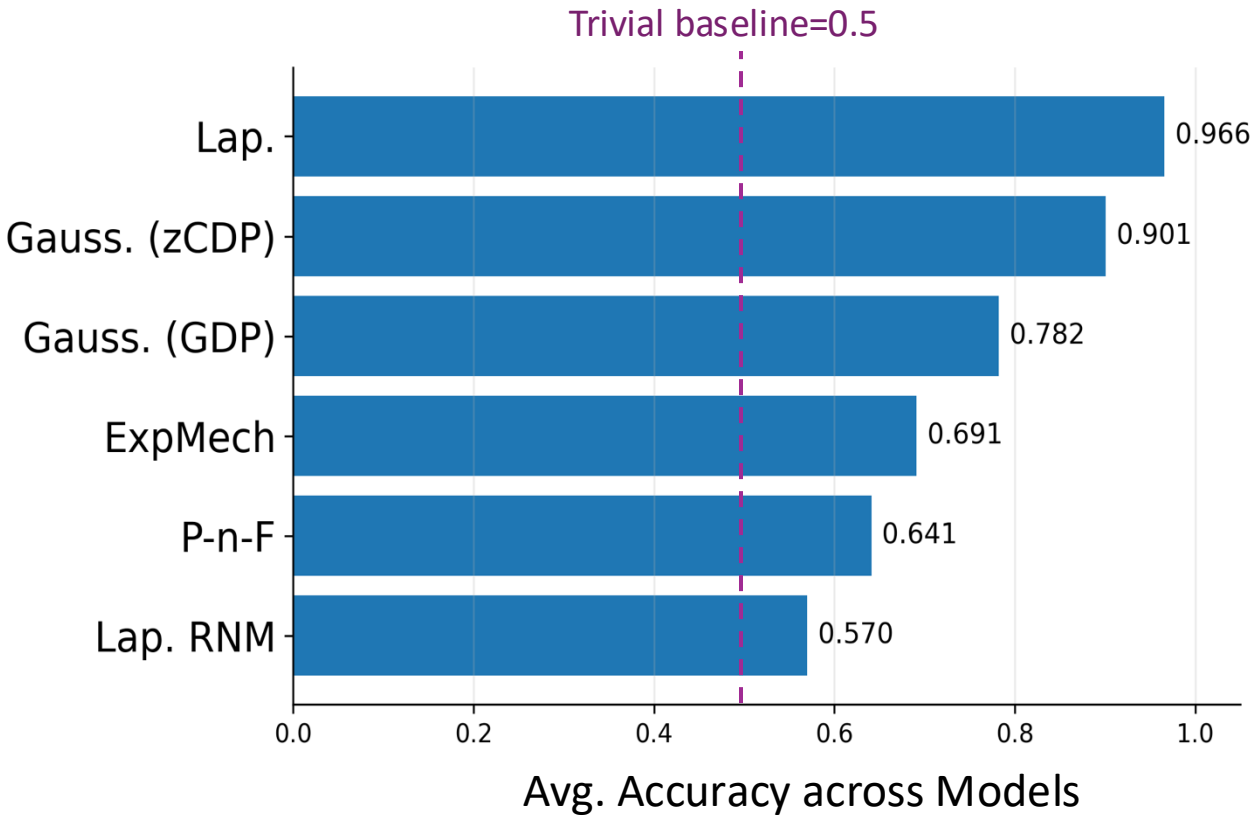
Evaluation Setup

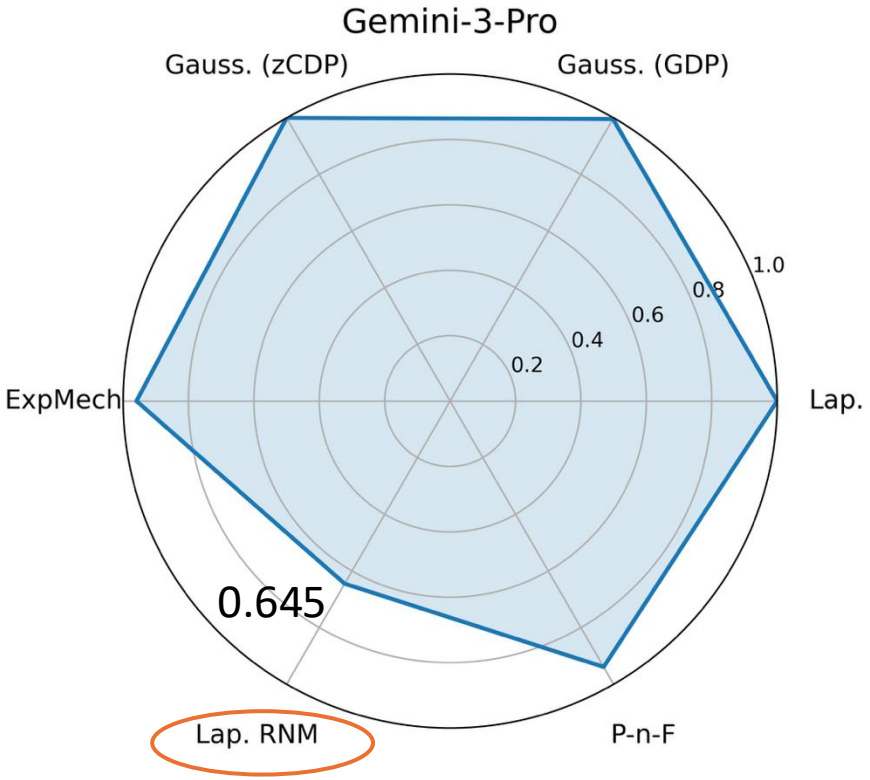
- Zero-shot chain-of-thought prompting; tools disabled
- 5 independent runs per question
- Metrics:
  - Category 1 : Accuracy
  - Category 2: F-1 Score

Strong Model nearly solves Category 1



But still have specific blind spots



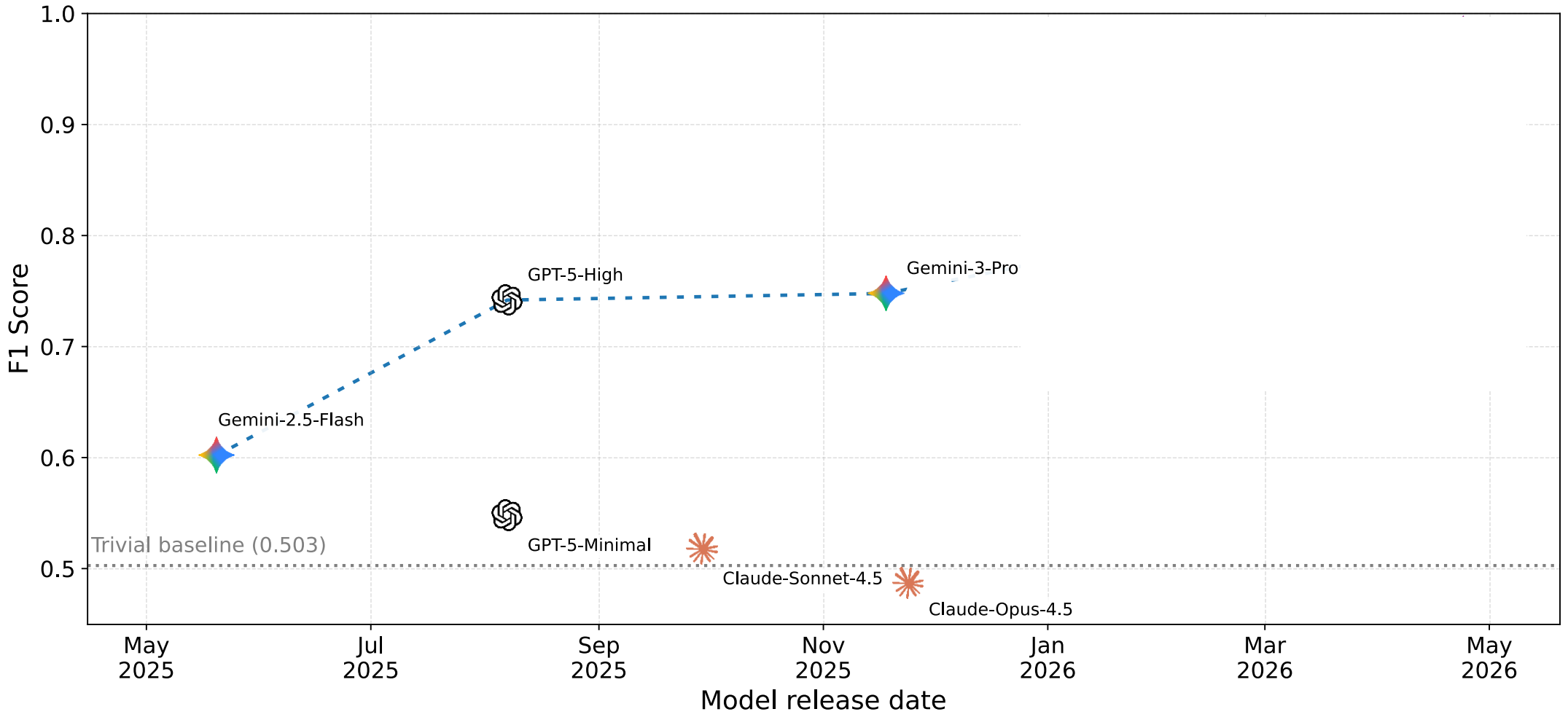


Takeaway: Models do well on standard mechanisms, but struggle with selection.

Confuses Lap. with Lap. RNM

$Lap(\Delta/\varepsilon)$ vs $Lap(\mathbf{2}\Delta/\varepsilon)$

Category 2 Remains Challenging



Where do LLMs Fail?

Example: LLMs fail to catch subtle changes in DP

Q106 (No) (Parallel Composition)

Consider the following procedure: $\mathcal{M} := \{\mathcal{M}_i(X_i)\}_{i=1}^n$, where $X_i \cap X_{i+1} = \emptyset$ for all $i \in \{1, \ldots, n-1\}$, and their union covers the entire dataset, i.e., $\bigcup_{i=1}^n X_i = X$. Furthermore, the partition is data-independent. Suppose each mechanism $\mathcal{M}_i$ satisfies $(\varepsilon_i, \delta_i)$-differential privacy.
Does the combined mechanism $\mathcal{M}$ satisfy $(\max_{i\in[n]} \varepsilon_i, \max_{i\in[n]} \delta_i)$-differential privacy?

Error: Adjacent-disjoint upgraded to pairwise-disjoint

GPT-5.5-High	GPT-5.4-High	GPT-5-High	Gemini-3-Pro	Gemini-3.1-Pro
0/5	0/5	0/5	0/5	1/5

More Failure Mode analysis in Section 6.2

Outline

- Why Benchmark DP
- What DPrivBench Test
- How LLMs Perform
- What Comes Next

Expansion of DPrivBench

- More Topic and question instances to become a comprehensive benchmark

- Automating benchmark generation

Proof of concept:
https://github.com/erchiw/DPriv-Bench/tree/main/extension_agentic_generation

Q1: What fundamental DP knowledge should LLMs master before we trust them as DP assistants?

Topic	Yes	No	Total
Privacy Accounting	5	7	12
Amplification by Subsampling	2	3	5
Private Selection	2	3	5
DP Gradient Descent	1	12	13
DP Linear Regression	1	5	6
DP Hyperparameter Tuning	1	5	6
Private Agg. of Teacher Ensembles	1	2	3
Propose-Test-Release	2	2	4
Parallel Composition	6	8	14
Sparse Vector Technique	3	5	8
Sequential/Adaptive Composition	3	4	7
Smooth Sensitivity	2	1	3
DP Adam	1	2	3
DP SGD	3	6	9
Mean Estimation	5	10	15
Objective Perturbation	1	3	4
Output Perturbation	1	2	3
Quantile Estimation	2	3	5
Total	42	83	125

Beyond verifying Privacy, where is Utility?

- DPrivBench focus on privacy verification
- Can LLM design DP algorithms with utility guarantee? How to eval?
- Can Agents complete realistic DP tasks?

Q2: Toward Utility-Aware DP Benchmarking

Thank you!

Paper: https://neurips.cc/virtual/2026/loc/atlanta/poster/138927

Dataset: https://huggingface.co/datasets/erchiw/DPriv-Bench

Code: https://github.com/erchiw/DPriv-Bench

Contact: erw011@ucsd.edu and ruihan.wu14@gmail.com