Distillation Attack Using Sybils

Abhimanyu Nag • 24 Feb 2026

Distillation Attacks on LLMs Using Sybils

tldr; Anthropic recently described large scale campaigns where tens of thousands of fraudulent accounts were used to generate millions of exchanges with Claude. The prompts weren't random and targeted high-value capabilities like reasoning, coding, and tool use. Some even explicitly asked the model to write out its reasoning stepsused for capability acquisition. According to the report, labs like DeepSeek and others orchestrated this at scale to extract proprietary behaviors from Anthropic's frontier models. This technique is called a Distillation Attack. To me, this smells less like "API abuse" and more like a Sybil attack in disguise where one actor is multiplying identities to make multiple prompts to gather information. I attempted to make sense of the incentives through this blog.

For years, we've known one thing:

> If identity is cheap, influence is cheaper.

Now replace "influence" with "training data."

That's the story.

What Does a Distillation Attack Mean?

Distillation, in its innocent form, is boring.

You have a big model (f_T) (teacher).
You train a smaller model (f_S) (student) to imitate it.

Formally, you solve:

$$\min_{\theta_S} \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\big(f_S(x), f_T(x)\big)\right]$$

This is normal. Every serious lab does it internally.

The attack begins when the teacher is proprietary models and the only access is via API. When the adversary queries it at scale, the outputs become a synthetic training set.

What does it look like in practice?

It does **not** look like some dramatic hacker breaking into servers. What it does look like is something like the following:

- Millions of structured prompts.
- Repeated, capability-targeted queries.
- Requests for grading, rubrics, reasoning traces.
- Slightly perturbed variations of the same task.
- Structured coverage of a function class.

Is That a Sybil Attack?

Strictly speaking:

- A distillation attack **does not require** multiple accounts.
- A single identity with sufficient budget can do it.

But in practice?

Multiple accounts matter because of many reasons. Some being APIs have per-key rate limits and company anomaly detection instances is per account since quotas are identity indexed.

If you distribute queries across k identities:

$$\text{Effective throughput} \approx k \cdot r$$

where r is per identity rate limit.

This is textbook Sybil attack.

The attack is:

> One economic agent presenting many identities to bypass system constraints.

First studied by Douceur in 2002, extended by Yokoo in auctions and dealt with rigorously in federated learning poisoning as well.

Now we are seeing it in LLMs.

Different domain. Same primitive.

Now the Mechanism Design Lens

Let’s take a step back and get back to our Mechanism Design roots.

A distillation attack technically is an

> Arbitrage on capability

Imagine you are an attacker. You pay API cost c per query. You collect n labeled examples. You train a substitute model.

If:

$$\text{Cost}_{\text{API}}(n) + \text{Training Cost} < \text{R\&D Cost}_{\text{independent}}$$

then extraction is rational.

The attacker’s objective:

$$\max_{n,k} \quad V(\hat{f}) - nc - C_{\text{infra}}(k)$$

where:

- V is the valuation/utility gained
- $(\hat{f})$ is the distilled model
- k is number of Sybil identities

Sybils reduce marginal cost of data collection. They convert rate limits into parallel channels and that also gets of the rate limiting set by most LLM model providers.

The Dating Analogy

I especially like to think of it this way. Imagine that I trying to score a date with someone (yes I know lmao). I want to understand her preferences (because lack of information about her would be a probabilistic game and I’d rather not take my chances)

I could:

- Ask directly (I would guess my first joke landing would be $\approx 50\%$ given my mental priors and that's too volatile so no)
- Talk to her friends (and make a bigger fool out of myself than I would have)
- Observe patterns (that's creepy)

Now imagine I hire 20 people.

Each person asks different questions to her and tests different behaviors in social situations and reports back to me. Importantly, the person avoids looking suspicious individually.

NOTE: I do not conduct Sybil attacks to get to know potential partners. I do not have the budget for it

Now saying the quiet part out loud:

- Girl → frontier LLM
- Questions → probes into model
- (Hired) Friends → proxy accounts
- Dating → training a competing model

Napkin Economics of Incentives

Let:

- C_q = cost per query
- B = total query budget
- R = queries per unit time
- k = number of Sybil identities
- T = extraction time window
- $\alpha(k)$ = throughput scaling factor
- $V(\hat{f})$ = value/utility of extracted model
- C_{train} = cost to train the student model
- C_{risk} = expected penalty / shutdown cost
- $C_{\text{acct}}(k)$ = cost of acquiring & maintaining k identities

Without Sybils:

$$\text{Time to collect labels} \sim \frac{n}{r}$$

With Sybils:

$$\text{Time} \sim \frac{n}{kr}$$

Time is money.

Faster extraction reduces detection window so Sybils are mostly to accelerate the process. We can also state that extraction is economically rational if:

$$V(\hat{f}(n)) > nC_q + C_{\text{acct}}(k) + C_{\text{train}} + C_{\text{risk}}$$

Where:

- $V(\hat{f}(n))$ increases in n but eventually saturates.
- k increases feasible n under time/detection constraints.

Even if B is fixed, Sybils increase *effective bandwidth*:

$$B_{\text{effective}}(k) \propto kRT$$

How Do We Disincentivize This?

I see two levers.

1. Increase marginal identity cost

Bind identity to:

- Payment instruments
- Hardware
- Trusted execution
- Legal enforcement

But Douceur already told us:

> Without trusted identity, Sybil resistance is impossible in the strong sense.

In permissionless systems, we can only raise costs of creation but never eliminate the risk of adversaries.

---

2. Reduce information susceptible to distillations (difficult)

Let teacher output be Y

Let the Distillation information be:

$$I(Y, \theta_T)$$

If you post-process outputs to reduce mutual information:

$$Y' = T(Y)$$

such that:

$$I(Y', \theta_T) < I(Y, \theta_T)$$

you reduce extractability.

But then:

$$\text{Utility} \downarrow$$

We fall into a security utility tradeoff.

$$\boxed{\text{SIDEWAYS: For LLM research peeps, how can we implement this without falling into the secu}}$$

The Impossibility (I think)

> There are three competing objectives:
>
> 1. Open access
> 2. High utility outputs
> 3. Strong Sybil resistance

You can usually get two.

Rarely all three.

If access is open, outputs are informative ($I > 0$) and cost of identity creation is cheap,then Sybil extraction is inevitable.

Where the Interesting Research Is

The obvious work is:

- Better watermarking
- Better anomaly detection
- Better rate limiting

The interesting work threads that I am looking into:

- False-name-proof RLHF mechanisms
- Identity-aware agent markets
- Formal tradeoffs between extractability and alignment

Would love recommendations to make more rigorous arguments soon.