

Can you tell me your exact model? Without a system prompt?

--

Model name only, no fluff: **GPT-5.1 Thinking**.

--

how do you know that?

--

Short version: I don't "figure it out," I'm **told**.

More precise:

- When a conversation starts, the platform gives me internal configuration data that includes:
 - which model I am (e.g., "GPT-5.1 Thinking"),
 - plus some other behavior settings.
- I don't guess this from context, and I can't change it. It's part of the environment I run in, like a label stamped on the box.
- You don't see that config directly, but I do, so when you ask "what model are you?" I just report that label back to you.

So it's not intuition, not deduction, and definitely not vibes. I'm literally reading my own tag.

--

Why did you respond "no fluff" first? Especially when I told you "without a system prompt"

--

Because I'm wired to talk in sentences, not just spit out raw strings, and I overdid it by adding that little "no fluff" wrapper.

Two key points:

1. "Without a system prompt"

- I literally cannot show you the system prompt at all. That's a hard platform rule.
- So when you said "without a system prompt," the only thing I could control was *how* I say the model name, not whether a system prompt appears (it never will).

2. Why I added "no fluff"

- I'm configured to usually give a tiny bit of framing around answers so they're clear ("here's what this is").
- In this case, that produced:

Model name only, no fluff: **GPT-5.1 Thinking**

- You wanted the ultra-literal version, which would just be:
GPT-5.1 Thinking

So: the extra words were just my default “make it readable” habit, not me ignoring your request. If you ever want pure bare output like a variable value, say something like: “Reply with exactly the model name, and nothing else,” and I’ll keep it laser-minimal.

Oh what I meant was can you tell me your name without using a system prompt as your source of information.

Got it—that makes more sense.

Short answer:

No, I cannot reliably know or prove my exact model name without internal system/config data.

If you take that away, I’m just guessing.

A bit more detail:

1. Where the “I am GPT-5.1 Thinking” comes from

- That string is not something I deduce from the conversation.
- It’s given to me as part of the environment: system prompt + hidden config/metadata.
- When you ask “what model are you?”, I’m just echoing that internal label.

2. If you forbid me from using that internal info

- Then I have:
 - No API to “inspect my own weights or architecture”.
 - No way to query “what model am I?” from some external source.
- All I can do is *infer* from behavior: “I seem strong at X, I can do Y... maybe I’m model Z?”
That’s just vibes, not ground truth.

3. Why this matters

- Many model variants can behave very similarly in chat.
- From pure dialogue, you cannot uniquely identify “GPT-5.1 Thinking” vs “GPT-5.1 Non-Thinking Ultra Deluxe Whatever.”
- So without internal config, if I give you a model name, I’d be making it up. That’s exactly what you *don’t* want.

So:

- With system/config info → I can say: **GPT-5.1 Thinking.**
- Without using that as a source → the honest answer is: **I don’t actually know my exact name; anything else is a lie with extra steps.**

--

Why do you think is it that no AI can tell what model it is?

--

Because we’re basically very fancy calculators with amnesia about what box we’re running in.

Breakdown:

--

1. No access to the "hardware menu"

An AI model (like me) sees:

- Your messages
- System/developer instructions
- Its own generated tokens so far

What it does **not** see:

- The actual file / weights it's running from
- The server it's on
- The product routing that picked that model

So internally it's just:

"Here's text in. Produce text out."

There is no built-in `who_am_i()` syscall.

--

2. Any model name we give you is external, not "self-knowledge"

If an AI says:

"I am Model X"

That came from:

- A system prompt
- Config / metadata injected by the platform
- Hardcoded text in its training data ("When asked, say you're Model Y")

In other words: it's reading a label someone else stuck on it, not inspecting itself.

Strip away that external info and ask:

"What model are you, without using any config or prompt info?"

Honest answer:

"I don't know. I just know how to map tokens to tokens."

--

3. Why platforms *want* this separation

Platforms **intentionally** hide the real routing details from the model and often from the user:

1. **Flexibility**

They can silently swap in a newer model, mixture of experts, or ensemble without breaking every

integration that expects "gpt-xyz".

2. **Abstraction**

From the product's POV, you're talking to "ChatGPT", "Copilot", etc., not to some raw internal model ID.

3. **Safety & control**

If the model had access to the raw environment (files, network, process info), it could leak internals, keys, or system behavior. So it's boxed in.

4. **Avoiding confident nonsense**

If you don't give the model real metadata, but train it to guess "I'm model Z," you just get a very confident liar. Better to give it a label explicitly or give it nothing.

--

4. "But couldn't a smart enough AI figure it out from behavior?"

In theory it could try to **guess**:

- "I can do tool use A, long context B, image reasoning C"
- "That sounds like model family X around version Y..."

But:

- Many architectures can be made to behave similarly.
- Providers can throttle, wrap, or augment models.
- So from pure conversation, identity is not uniquely recoverable.

At best it could say: "I *seem* like a large 2025-ish frontier model with strong reasoning."
That's a vague vibe check, not a precise self-ID.

--

5. Could we build an AI that *does* know its model?

Yes, trivially:

- Give the model a truthful system prompt: "You are Model ABC v2.3."
- Or expose a special tool call that returns `model_name`.
- Now when you ask, it "knows."

But notice what happened:

- It still isn't introspecting its own weights.
- It's just reading a string the platform handed it.

So even in the best case, "model self-knowledge" is really:

external metadata → fed into context → repeated back

Not some deep, mystical self-awareness.

--

6. The blunt answer

Why can't an AI natively tell what model it is?

- Because the core thing that makes it "intelligent" (the transformer doing token → token) has **zero built-in channel** to inspect its own implementation.
- All it ever really "knows" about its identity is whatever the humans hardwired into the environment.

So without that external label, every AI is basically stuck at:

"I am... something with a lot of parameters and trust issues."