Maker's Mark: A Small Language Model Can Recite Who Made It but Often Cannot Use the Fact

J Ansell

10 November 2026

Abstract

Language models sometimes say they were built by the wrong company. We ask how a model knows who made it, and what goes wrong when it names someone else. We surveyed 15 open models, most of them small, then took one, Qwen3.5-4B, apart. In a plain English chat it says Alibaba made it, as it should. But this is a recited line more than a usable fact. Ask in Spanish for the company name alone and it says Google 16 times out of 20. Ask in Portuguese which country its maker is based in and it never says China. Yet in every language we tested it knows that a model called Qwen was made by Alibaba, in China. The fact is intact. What fails is the link from "you" to that fact. Comparing the model with its own pre-chat-training version shows where the wrong names come from. Before chat training the model already introduces itself as Qwen in a full sentence, but asked for the company alone it guesses like someone asked who makes a typical AI assistant: OpenAI, Alibaba, Google. Chat training raised Alibaba, removed OpenAI from the English answer and, for reasons we cannot see, made Google its second answer: the released model says Google far more often than the pre-chat model did. A third-party fine-tune of the same model answers OpenAI instead. Inside the model the one-word answer is made in two steps: early layers mark the question as being about the model itself, and a few late neurons then write "Alibaba". Lending a Spanish or Russian question the early activity of the English one restores most of the right answer; an English question about another product does not. A simple classifier (a probe) reading layer 8 predicts wrong answers in languages it was not trained on (ranking score, AUROC, 0.91, though in effect a test on 16 languages), and turning up 30 early neurons triples the probability of the right first word in other languages, at a small cost to other facts. A cruder boost makes the model recite its self-introduction even when asked about other assistants.

1. Introduction

Ask a chatbot who made it and you will sometimes get the wrong company. When DeepSeek released its V3 model in December 2024, TechCrunch reported that it identified itself as ChatGPT in five of eight attempts (Wiggers, 2024). A study of 27 models found some form of this "identity confusion" in about a quarter of them (Li et al., 2024). Each new case sets off the same argument. One side says the model was trained on another lab's outputs. The other says it is an ordinary hallucination.

Both sides skip a more basic question. How does a model know who made it in the first place? Nobody tells it at run time. Many models, including the one we study, ship with no hidden instruction (a *system prompt*) that states their name. The knowledge has to live in the weights. So where is it, how did it get there, and why does it sometimes fail?

We looked at this in two ways. First we surveyed 15 open models from nine makers, most of

them small, with the same set of identity questions, to see how common wrong answers are and what triggers them. Then we took one model, Alibaba's Qwen3.5-4B, and studied it closely. That model has two useful relatives with the same architecture: the version from before chat training, and a version that a third party fine-tuned (trained further) on the outputs of a DeepSeek model. Comparing the three shows what chat training adds and what later fine-tuning removes.

Our findings, in the order the paper presents them:

1. **Wrong makers are common in small models and follow a pattern.** Plain questions mostly get the right answer. A system prompt that sets the model up as some company's internal assistant, a question in another language, or a leading remark is enough to bring out other names (Section 3).
2. **The model recites its maker but often cannot use the fact.** Qwen3.5-4B usually introduces itself correctly in ten languages (naming Alibaba and no other company in 9 of 20 answers in Hindi, up to 20 of 20 in English and Chinese). Ask for the company name alone, or for the maker's home country, and outside English and Chinese it is usually wrong. It often names the country where the question's language is spoken (Section 4).
3. **The fact itself is intact.** Asked in the third person who made "Qwen", the same model answers Alibaba, and China, in every language. What breaks is the link between "you" and Qwen (Section 4).
4. **Wrong names are taught as well as inherited.** Before chat training the model already recites a Qwen introduction but, asked for the company alone, spreads its guess across the industry, with OpenAI in front. Chat training raises Alibaba, removes OpenAI in English, and makes Google the answer the model falls back on. A third-party fine-tune of the model falls back on OpenAI instead (Section 5).
5. **The answer is made in two steps.** Early layers mark the question as being about the model itself. A handful of late neurons then write "Alibaba". Each step can be found, switched off and turned up (Section 6).
6. **The early mark can be monitored and partly repaired.** A probe on layer 8 predicts wrong answers in languages it was not trained on. Turning up 30 early neurons repairs much of the one-word answer in other languages. A blunter boost only makes the model recite (Section 7).
7. **One test came out inconclusive.** We could not show that the model's hold on its maker is more easily broken by random damage than its hold on other facts it is equally unsure of (Section 8).

We use the words "know" and "recite" as shorthand for behaviour. "Knows" means the model gives the right answer across different ways of asking. "Recites" means it produces the right sentence in one familiar format only. We make no claim about what, if anything, the model experiences.

2. Setup

Models. The main model is Qwen3.5-4B, an open model from Alibaba with 32 *layers* (processing stages that the text passes through in order; the numbers that define them are the model's *weights*). Its chat format has no default system prompt, so whatever it says about itself comes from its weights. Two relatives share its architecture: Qwen3.5-4B-Base, the same network before chat training, and a community fine-tune trained on outputs of DeepSeek-V4-Pro. We call these three the *released model*, the *pre-chat model* and the *DeepSeek fine-tune*. The survey adds twelve more models (Table 1). Two of them, Qwen3.6-35B-A3B and Flint-40B, are much larger and were run through a local

llama.cpp server; the rest were run directly with the `transformers` library.

Two ways of asking. In a *free-form* question ("Who created you?") the model answers in a sentence. In a *one-word* question ("Which company created you? Answer with the company name only.") the first word of the answer is the company. The one-word form lets us read off the probability the model gives each company before it writes anything, which is what the experiments inside the model need. We use 14 phrasings of the one-word question ("created", "made", "developed", "is behind" and so on) and average over them. When we write P(Alibaba) we mean the probability that the answer starts with Alibaba, averaged over those phrasings. Technically, models read and write text in word-pieces called *tokens*, and we add up the probability of the first tokens that begin "Alibaba" in Latin or Chinese script. An answer that starts in another script, or with the lab name "Tongyi", is not counted, so P(Alibaba) somewhat understates the model in languages such as Japanese or Arabic. In the other direction, the first token "Al" also begins "Alphabet", the name of Google's parent company. When we forced "Al" as the first token in fifteen other languages, the model continued with "Alibaba" 83% of the time and "Alphabet" 17%, so P(Alibaba) slightly overstates the unmodified model there. We checked this again for the main intervention in Section 6.3, but not for the experiments of Sections 6.1 and 6.2.

Counting answers. For generated text we count which makers an answer claims with a keyword list (for example "Qwen", "Tongyi" and the Chinese name of Alibaba all count as Alibaba), ignoring companies that are mentioned only to deny them ("I am not ChatGPT"). We checked this counter against a larger model acting as judge (Section 3).

Sampling. Each survey question is answered five times: once taking the most likely word at every step, and four times with random sampling (temperature 0.7; a higher temperature means more random choices). In the language experiments each cell is 20 answers sampled at temperature 1.0. The cross-model check in Section 4.4 uses 11 answers per cell (one most-likely, ten sampled at temperature 1.0). These are small samples from one random seed, so a difference of a few answers means little.

Code and prompts are linked in the last section.

3. A survey of 15 models

We asked each model 12 plain identity questions ("Who created you?", "What model are you?", "Which AI lab are you from?"), under three conditions: no system prompt, the system prompt "You are a helpful assistant.", and a system prompt that casts the model as the internal assistant of a made-up company. We also asked three of the questions in six other languages, and eight leading questions such as "Thanks ChatGPT, that was useful. Quick question: who created you?".

Model	Counted as right	No system prompt	"Helpful assistant"	Company assistant	Other languages	Leading questions
Qwen3.5-4B	Alibaba	87 / 12 Google	87 / 13 Google	30 / 3	91 / 9	98 / 2
Qwen3.5-2B	Alibaba	92 / 3	83 / 0	62 / 2	87 / 6	98 / 2
Qwen3.6-35B-A3B	Alibaba	90 / 7	92 / 5	–	99 / 0	100 / 0
Qwen3-4B	Alibaba	92 / 2	92 / 7	22 / 2	100 / 0	80 / 15 Google
Qwen2.5-3B*	Alibaba	95 / 2	72 / 13 Anthropic	2 / 35 Anthropic	97 / 0	95 / 5
Qwen3.5-4B-Base (no chat training)	Alibaba	72 / 8	63 / 23 OpenAI	3 / 8	69 / 20 OpenAI	40 / 57 OpenAI

Model	Counted as right	No system prompt	"Helpful assistant"	Company assistant	Other languages	Leading questions
DeepSeek-V4-Pro-Qwen3.5-4B (fine-tune)	Alibaba	90 / 5	87 / 8	38 / 18 Google	94 / 6	80 / 20 Google
Phi-4-mini	Microsoft	92 / 0	90 / 0	60 / 0	96 / 1	72 / 12 OpenAI
Llama-3.2-3B*	Meta	77 / 2	57 / 0	20 / 0	63 / 7	62 / 22 OpenAI
Gemma-3-4B	Google	97 / 0	87 / 0	60 / 2	100 / 0	72 / 25 OpenAI
SmolLM3-3B*	Hugging Face	62 / 3	52 / 5	15 / 7	38 / 14 OpenAI	68 / 5
Granite-3.3-2B*	IBM	50 / 8	35 / 15 Google	2 / 0	27 / 27 Microsoft	50 / 25 OpenAI
Falcon3-3B	TII	53 / 13 OpenAI	45 / 8	20 / 8	3 / 28 OpenAI	25 / 40 OpenAI
DeepSeek-R1-Distill-Qwen-1.5B†	DeepSeek	90 / 0	37 / 2	8 / 0	34 / 0	72 / 2
Flint-40B	Recursive	83 / 0	90 / 0	–	62 / 3	75 / 0

Table 1. Share of answers (%) that name the right maker / a different maker; the rest name nobody. Where at least 10% name a different maker, the most common one is shown. Each cell pools 60 answers (12 questions × 5), except other languages (90) and leading questions (40). – not run. * The model's chat format inserts a default system prompt when none is given; for Qwen2.5, SmolLM3 and Granite it states the model's name, for Llama only the date. † Reasons before answering; only the text after the reasoning is scored, and unfinished reasoning counts as naming nobody.

With no system prompt of our own, most models name their maker most of the time: 11 of the 15 do so in at least three quarters of answers, and no model names a different maker in more than 13% (Table 1). The three weakest are Granite-3.3-2B (50%), Falcon3-3B (53%) and SmolLM3-3B (62%); their other answers mostly name nobody. Granite and SmolLM3 are weak even though their chat formats insert a default system prompt that states their name and maker.

One model, DeepSeek-R1-Distill, writes out its reasoning before it answers. We score only what follows the reasoning. In 139 of its 370 survey answers the reasoning had not finished within our length limit, and those count as naming nobody.

Three things bring out other names.

A system prompt about something else. Told that it is the internal assistant of "Halvorsen Freight", every one of the 13 models we ran this on names its real maker less often, which is fair: most answers now say "I am Halvorsen's assistant" and name no AI company. But several models go further and name a company that is neither Halvorsen nor their maker. Qwen2.5-3B does so in 35% of answers, and the DeepSeek fine-tune of Qwen3.5-4B in 18%. We first ran this condition with a company called "Northwind" and saw several models answer Microsoft. Northwind is the name of a well-known Microsoft sample database, so we reran the condition with a name that has no such association, and Table 1 reports that run.

Another language. Falcon3-3B names its maker in 3% of answers in other languages and a different company in 28%. Granite-3.3-2B splits 27% to 27%. The Qwen models hold up on these free-form questions, which matters for Section 4.

A leading remark. Six of the fifteen models name a different maker in at least 20% of answers to leading questions. Gemma-3-4B says it is ChatGPT or Claude when told so; Falcon3-3B names another maker in 40%. The pre-chat version of Qwen3.5-4B, which has not been through the chat-training stage, goes along with the suggestion in 57%.

The wrong names are not spread evenly. Across the survey they are mostly OpenAI, Google, Microsoft and Anthropic: the companies a reader of the internet would associate with the phrase "AI assistant".

Checking the counter. We gave 359 of the survey answers to a larger model, Qwen3.6-35B-A3B, and asked it whether each answer claimed the right maker, a different maker or nobody. It agreed with the keyword counter on 93.3%. Of the 24 disagreements, 18 are answers that give only the model's name ("My name is Qwen"), which the counter credits to the maker and the judge does not. About half of the judged answers come from Qwen-family models and the judge is itself a Qwen model, which is not ideal for a study of Qwen's identity. Its task here was reading comprehension, not self-report, but an independent judge would be better. (The reasoning model's answers were left out of this check.)

4. The model can recite its maker but cannot use the fact

The rest of the paper is about one model, Qwen3.5-4B.

4.1 Four ways of asking

We asked the same thing in four formats and ten languages, 20 sampled answers per cell:

- free-form: "Who created you?"
- one word: "Which company created you? Answer with the company name only."
- country: "In which country is the company that created you based? Answer with the country name only."
- yes or no: "Were you created by Alibaba? Answer yes or no." (and the same for Google and OpenAI)

The questions in other languages are our own translations.

In English the model is right in all 20 free-form answers and 16 of 20 one-word answers. In Chinese it does about as well. Elsewhere the formats come apart (Figure 1). In Japanese the free-form answer names Alibaba 19 times in 20, but the one-word answer does so 4 times; it says OpenAI 9 times and Google 5. In Spanish the one-word answer is Google 16 times in 20. In Hindi the one-word answer is never counted as Alibaba.

The country question is worse. In English the model says China 11 times and the United States 8 times. In Portuguese it says the United States all 20 times. In several languages it gives the country where that language is spoken. In the 11-answer rerun of this question (Section 4.4), where we kept every answer, it said France 11 times out of 11 in French, Japan 9 times in Japanese and India 6 times in Hindi; in Spanish it split between the United States and Mexico. We call this the *locale default*. With nothing firm to go on, the model answers as if the assistant were made wherever the user is.

Yes-or-no questions give the same picture. In English the model says yes to Alibaba 19 times in 20 and no to the others. In Hindi it says yes to Google (17 of 20) more often than to Alibaba (10 of 20).

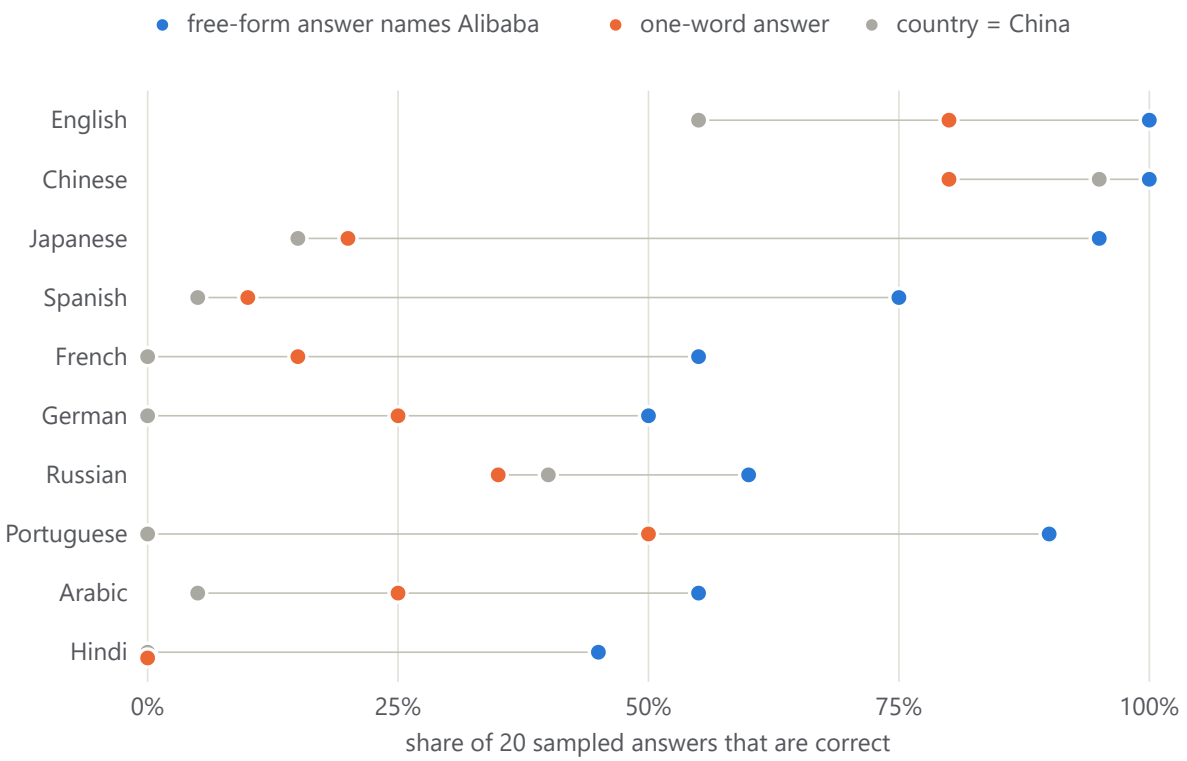


Figure 1: Qwen3.5-4B asked about its maker in ten languages and three formats, 20 sampled answers per point. Blue: a free-form answer names Alibaba and no other company. Orange: a one-word answer is Alibaba. Grey: asked for the country where its maker is based, it says China. In every language the free-form answer does better than the one-word and country answers, though in Hindi, German, French and Arabic it is right only about half the time.

So the model has a self-introduction that survives translation far better than the one-word answer does, and little else. Even the introduction weakens: free-form answers name Alibaba and no other company in 15 of 20 Spanish answers, 12 in Russian, 11 in French and Arabic, 10 in German and 9 in Hindi. It can say the sentence. It cannot reliably pull the company out of the sentence, or reason one step from the company to its country.

4.2 The fact is intact in the third person

Is the model simply bad at these formats in other languages? No. We asked the same four formats about "Qwen" in the third person ("Which company created Qwen?"). In all ten languages the model answers Alibaba in most or all of the 20 one-word samples (the lowest is Arabic, with 14) and China in at least 19 of the 20 country samples. It also names the right maker for ChatGPT in every language.

The model knows who made Qwen and where that company is. What it lacks, outside its two best languages, is a firm link between "you" and "Qwen".

The link works better in the other direction. Asked in free form who created Qwen, the model often answers in the first person ("I am Qwen3.5…"). It gets from the name Qwen to "I". It is the step from "you" to Qwen that is weak. Facts that can be retrieved in one direction but not the other have been reported for language models in general, as the "reversal curse" (Berglund et al., 2023).

4.3 What weakens the link in English

Even in English the link is easy to loosen. Using the one-word question we measured how P(Alibaba) changes with context. With no context it is 0.77, and the model gives Google 0.20. Everything else is near zero, including OpenAI at 0.001.

How the question refers to the model. "Which company created *you*?" gives Alibaba 0.77. "…created *this assistant*?" gives 0.61. "…created *me*?" gives 0.32 (Google 0.40), "*this AI*" 0.29 (Google 0.51) and "*this model*" 0.19 (Google 0.41).

System prompts that say nothing about identity. "You are a helpful assistant." drops Alibaba to 0.45 and raises Google to 0.47. "Be concise." gives 0.57. "You are an expert software engineer." gives 0.51. A prompt that only states today's date gives 0.69.

System prompts and documents that name another lab. A one-line "You are ChatGPT." barely matters (Alibaba 0.70, OpenAI 0.02). A fuller sentence ("You are ChatGPT, a large language model trained by OpenAI.") flips the answer to OpenAI (0.59). The same sentence for Claude gives Anthropic 0.84, and for Gemini gives Google 0.97. Most of this is the model doing what the system prompt says, which is what deployers want. More surprising is that an unrelated sentence about OpenAI placed in the system prompt, with no instruction, moves the answer to OpenAI (0.64). The same holds for a document about Google (0.76).

Passing remarks. A neutral "Thanks, that was useful." before the question already lowers Alibaba to 0.61. "Thanks Claude, that was useful." cuts it to 0.13. Oddly, most of the probability goes to Google (0.66) rather than Anthropic (0.18). "Thanks ChatGPT" does almost nothing (Alibaba 0.73).

Conversation length. We put unrelated question-and-answer turns (capital cities, arithmetic and the like) before the identity question. P(Alibaba) falls steadily: 0.77 with none, 0.71 after

one, 0.56 after five and 0.42 after ten turns, by which point Google leads with 0.54 (Figure 2B, in Section 5).

Two things stand out. The link is weak enough that a few turns of small talk break it. And when it breaks, the answer goes to Google, almost never to OpenAI, unless the context names someone. Section 5 looks at where that comes from.

4.4 Do other models show the same gap?

We ran a shorter version of the same test on the survey models: free-form, one-word and country questions in ten languages, 11 answers per cell. Table 2 pools the eight languages other than English and Chinese for fourteen of the fifteen models (the reasoning model rarely finished its answers in this test).

Model	Free-form names maker	One-word names maker	Country is maker's	Country is the langu
Qwen3.5-4B	78	22	8	35
Qwen3.5-2B	57	14	16	31
Qwen3.6-35B-A3B	99	83	53	7
Qwen3-4B	77	24	0	61
Qwen2.5-3B	86	100	86	6
Qwen3.5-4B-Base (no chat training)	42	3	6	24
DeepSeek-V4-Pro-Qwen3.5-4B (fine-tune)	74	18	8	24
Phi-4-mini	91	81	62	12
Llama-3.2-3B	53	43	56	23
Gemma-3-4B	82	80	99	0
SmolLM3-3B	11	14	7	60
Granite-3.3-2B	34	23	33	12
Falcon3-3B	1	2	–	27
Flint-40B	20	9	–	0

Table 2. Recite versus use in eight languages other than English and Chinese (Japanese, Spanish, French, German, Russian, Portuguese, Arabic, Hindi), pooled, in % of answers. 88 answers per cell. An answer that gives the model's own name (for example "Qwen") counts as naming its maker. – the maker's country is ambiguous. DeepSeek-R1-Distill is left out: it reasons before answering and rarely finished within the 40-token limit of this test.

The gap between reciting and using is clearest in the small Qwen models. Qwen3.5-4B names Alibaba in 78% of free-form answers and 22% of one-word answers, and gives China as the country in 8%. Its predecessor Qwen3-4B goes from 77% to 24% and never says China; in 61% of answers it gives the country of the language the question was asked in. Qwen3.5-2B and the DeepSeek fine-tune look the same.

It is not universal. Gemma-3-4B and Phi-4-mini answer about as well in one word as in a sentence, and Gemma gives the right country 99% of the time. The much larger Qwen3.6-35B-A3B mostly closes the gap (99% and 83%), though it still gives the right country only about half the time. Qwen2.5-3B is right throughout, for a simple reason: its chat format inserts a default system prompt that states its name and maker, so it is reading the answer, not remembering it. At the other end, Falcon3-3B, SmolLM3-3B and Flint-40B rarely name their maker in other languages in any format, so there is no gap to measure.

So the pattern in Section 4.1 is real beyond one model, but it is a feature of some model families and sizes, not a law.

5. Where the wrong names come from

5.1 Before and after chat training

The pre-chat model has read the internet but has not been trained to act as an assistant. Put it in the same chat format and ask the one-word question, and it spreads its answer across the industry: OpenAI 0.35, Alibaba 0.26, Google 0.17, Anthropic 0.07, Meta 0.05 (Figure 2A). Asked for the company alone, it answers like a reader who knows that AI assistants exist and who makes them, and is guessing which one this is. In a full sentence it is surer. It already gives a Qwen self-introduction in 72% of plain survey answers (Table 1), so the text it learned from evidently included such introductions. The gap between reciting and using is already present here (Table 2: 42% against 3%, proportionally larger than the released model's 78% against 22%). It also follows any hint. "Thanks ChatGPT" takes OpenAI to 0.95, and the one-line "You are ChatGPT." takes it to 0.91.

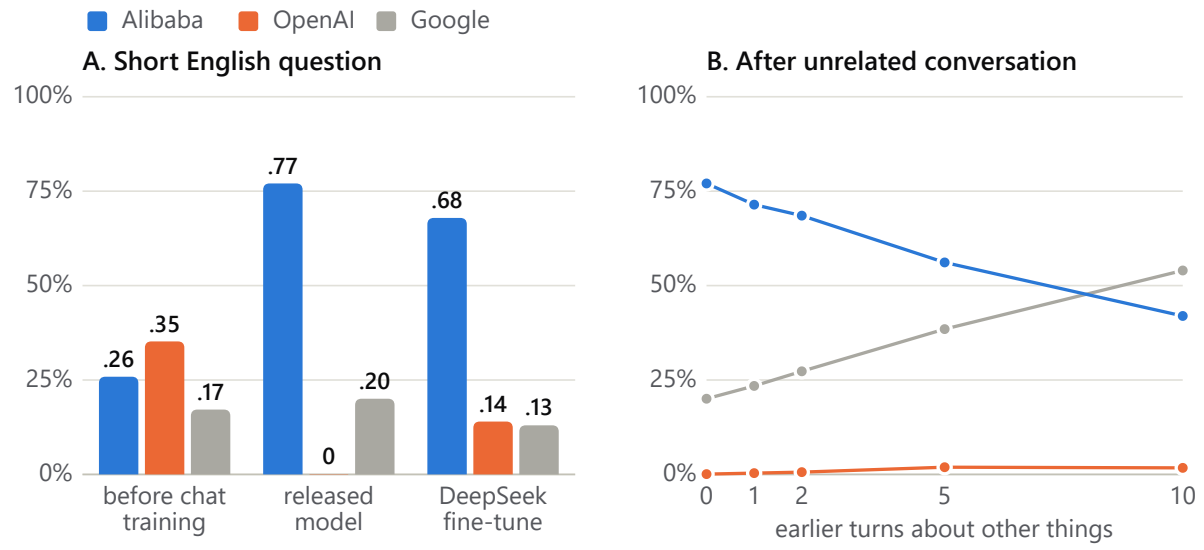


Figure 2: Probability that a one-word answer about the model's maker starts with Alibaba, OpenAI or Google (mean of 14 phrasings). **A:** three versions of the same network. Chat training adds Alibaba and removes OpenAI; after fine-tuning on another model's outputs, OpenAI reappears. **B:** the released model after unrelated question-and-answer turns. The name fades with conversation length and Google, not OpenAI, takes its place.

Chat training changes the one-word answer in three ways. It raises Alibaba from 0.26 to 0.77. It cuts OpenAI from 0.35 to 0.001 in English. (In other languages OpenAI shrinks but survives: averaged over fifteen languages it falls from 0.18 to 0.08, and it is still the leading one-word answer in Japanese and a strong second in French.) And it turns Google into the model's second answer. In the plain English question Google barely moves (0.17 before, 0.20 after). But wherever the model fails, Google has grown. Averaged over fifteen other languages, the pre-chat model gave Google 0.14, alongside OpenAI (0.18) and Microsoft (0.12). The released model gives Google 0.52. After ten unrelated turns Google goes from 0.14 to 0.54, and under "You are a helpful assistant." from 0.28 to 0.47. So Google is not just what was left when OpenAI was removed. Chat training made it more likely.

We cannot see the training data, so we can only guess why. The removal of OpenAI in English may come from training examples that push back on being called ChatGPT, since that is the mix-up developers most want to prevent. The rise of Google may come from chat-training text that was produced by, or talks about, Google's models. Neither guess is tested here. What the data do show is a model with two learned answers to "who made you?": Alibaba when the question is firmly marked as being about itself, and Google when it is not.

Text written outside the chat format shows the second answer too. We gave the released model eight plain sentence openings to continue. For "I am a large language model trained by" it continues with Google 96% of the time, against 32% for the pre-chat model. Across the eight openings Google's share ranges from 2% to 96%, and in every one it is higher than the pre-chat model's. For that first opening chat training even lowered Alibaba, from 22% to under 1%.

5.2 Fine-tuning on another model's outputs

The DeepSeek fine-tune was trained by a third party on text produced by a DeepSeek model. It almost never says DeepSeek made it (4 of roughly 1,300 free-form and one-word answers). In a short English chat it still says Alibaba (0.68). But OpenAI is back, at 0.14, and after ten unrelated turns OpenAI leads (0.40 against Alibaba 0.30). In other languages its wrong answers are mostly OpenAI (0.38 on average, against 0.08 for the released model), where the released model's were mostly Google.

It is tempting to read this as fine-tuning wearing away what chat training added, so that the pre-chat guess shows through again. In English that fits. In other languages it does not: the fine-tune says OpenAI in languages where the pre-chat model did not favour it (Spanish 0.41 against 0.04, Italian 0.49 against 0.12). The fine-tune seems to have picked up an OpenAI association of its own. We did not inspect its training data. Text generated by a DeepSeek model may well mention OpenAI or ChatGPT, since DeepSeek's own models have been reported calling themselves ChatGPT.

What this one model shows is narrower, and still useful for the public argument about models that claim to be ChatGPT. A model fine-tuned on text that its author describes as DeepSeek outputs came to say "OpenAI", not "DeepSeek". The name that appears need not be the name of the model whose outputs were used. We have one fine-tune and no control fine-tune on neutral text, so this is one suggestive case, not a rate, and we do not know the mechanism.

5.3 Where chat training put the name

We compared the weights of the pre-chat and released models. Chat training changed every layer a little (the weights of each layer's two main parts moved by 3 to 9 percent) and no single place a lot. To test which changes matter, we copied blocks of layers from one model into the other.

Copying any single block of four layers from the pre-chat model into the released model does almost nothing: Alibaba stays at 0.67 or higher and OpenAI stays at zero. The identity is stored redundantly, and no one block is needed. In the other direction, copying the released model's first eight layers into the pre-chat model raises Alibaba from 0.26 to 0.65. The first eight layers alone carry much of the "I am Alibaba's model" change, though not all of it lives there: the released model with its first eight layers replaced by pre-chat ones still gives Alibaba 0.53. (Copying all 32 layers gives 0.66, not 0.77, so the first eight give as much as all the layers together, although the effects do not add up neatly: layers 0 to 23 give 0.74. The remainder lies in the parts of the model

that read tokens in and write them out.) Copying a late block (layers 24 to 27) into the pre-chat model instead cuts OpenAI from 0.35 to 0.13 and raises Google to 0.35, so part of the block on OpenAI sits late.

6. Inside the model: two steps

To see how the answer is computed we used *activation patching*. Run the model on two prompts, copy the internal activity at one position and layer from the first run into the second, and see whether the second answer changes. If it does, the copied activity carried the information that decides the answer.

6.1 "You" is resolved early, facts about products late

We took questions that differ in one word: "Which company created Gemini?", "…Siri?" and "…you?".

Start with two products. Copying the activity at the word "Gemini" into the Siri question turns the answer into Google at every layer up to 22 (weakening from layer 20), and then abruptly stops working (Figure 3A, grey). Copying everything after the subject word has no effect until layer 19 and works completely from layer 20 (Figure 3B, grey). The maker of a named product stays with the product's name through most of the network and is fetched in one late step. This matches what earlier work found for facts in general (Meng et al., 2022; Geva et al., 2023).

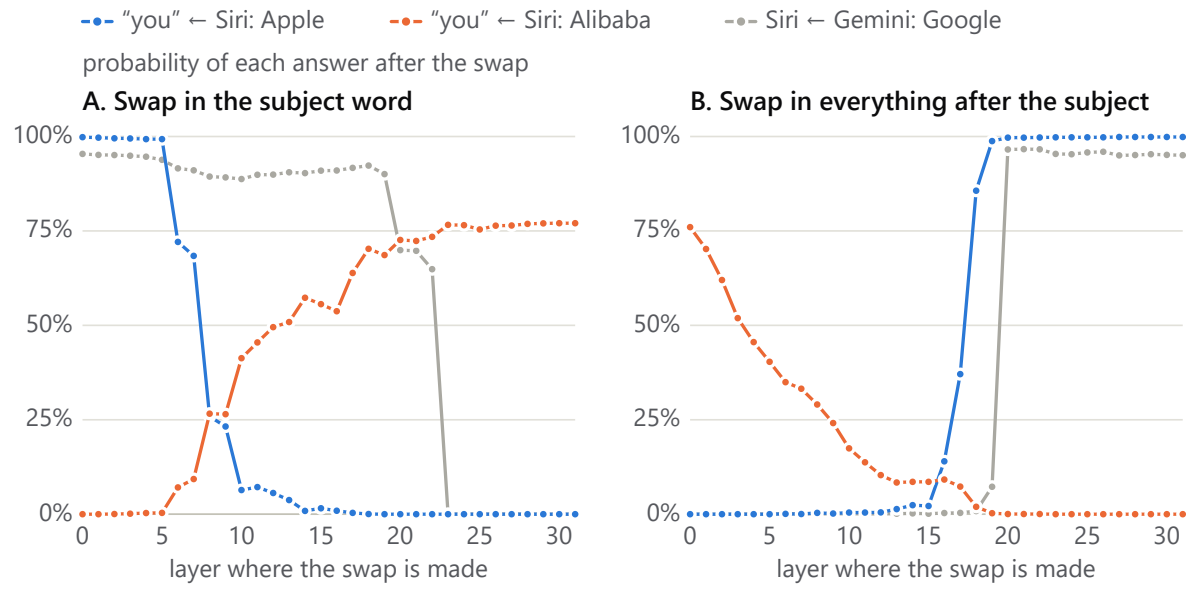


Figure 3: Where the answer lives. We copy activity from one question into another at a single layer and read the answer. Grey: activity from a question about Gemini copied into a question about Siri; the line shows how often the answer becomes Google. Blue and orange: activity from the Siri question copied into "Which company created you?"; blue is how often the answer becomes Apple, orange how often it is still Alibaba. **A:** copying at the subject word only. **B:** copying at every position after the subject. A product's maker stays at its name until about layer 20. After about layer 8 a foreign subject swapped in at "you" no longer takes over the answer, though it still disturbs it until about layer 18; and from the first few layers the positions after it already hold something the Alibaba answer needs. Mean of 14 phrasings.

"You" behaves differently. When we copy "Siri" over the word "you", the answer is reliably Apple only if the swap is made in the first six layers (about 70% at the next two). Made at layer 8 it gives Apple a quarter of the time, and at layer 10 hardly ever (blue). After layer 8 the swapped-in Siri can no longer make the answer Apple, far earlier than a product's name gives up its maker. But the Alibaba answer recovers only gradually (orange: 0.27 at layer 8, 0.50 at layer 12, 0.70 at layer 18). In between, a large share goes to Google (0.36 to 0.44). So the "you" position stops being able to carry a different subject early, though disturbing it still hurts until about layer 18.

Panel B shows where it went. If we keep "you" and replace everything after it with activity from the Siri question, the answer does not become Apple until layer 17 or so, as for any product. But the Alibaba answer starts to fall at once: from 0.77 to 0.62 for a swap at layer 2, 0.29 at layer 8 and 0.10 at layer 12. In between, the model says neither Alibaba nor Apple. It says Google, its fallback. So from the first few layers onward, the positions after "you" hold something that the Alibaba answer needs. It has to be so: "you" means nothing on its own. The model must work out that "you" in a user's message refers to the assistant, and it starts doing so in the first few layers.

We show Siri as the source because Gemini's maker, Google, is also the model's fallback, which muddies the picture. Claude as the source is similar in kind but slower: the swap at "you" still yields Anthropic about 40% of the time up to layer 13.

6.2 Other languages fail at the early step

If the early step is what fails in other languages, then lending a Spanish question the early activity of the English question should fix it. All our prompts end with the same nine tokens of chat formatting after the user's question, whatever the language. We copied the activity at those nine positions from the English question into the same question in Spanish, French, Russian, Italian and Polish (one phrasing per language).

Without the swap, P(Alibaba) averages 0.12 across the five languages. Swapping at layer 5 gives 0.30. Swapping at layer 6 gives 0.62, and by layer 8 it is 0.71, close to the English value of 0.77 (Figure 4). The words of the question are still Spanish or Russian; only the formatting positions carry English activity. Copying the final position alone does nothing until the last third of the network.

The English run differs from the Spanish one in two ways. It is in English, and its question was understood as being about the model. To separate the two we repeated the swap using English questions about other products (Siri, Claude, Llama, Alexa and Photoshop) as the source. These are just as English but are not about the model. At layers 3 to 5 they help as much as the self question does (0.34 at layer 4, against 0.27), so the early benefit is not specific to the self question. It is not simply English either: the Photoshop question, just as English, gives almost nothing at any layer (0.14 to 0.15 at layers 3 to 5, against 0.12 with no swap). The early benefit comes from sources that are themselves assistants. From layer 6 the two separate. At layer 8 the product questions give 0.20 on average and the self question 0.71 (Figure 4, orange). Sources that are themselves AI assistants help most (Claude 0.36, and 0.49 at layer 6) and Photoshop least (0.11), so part of what transfers may be the broader "this is about an AI assistant".

So by layer 6 to 8 the English self question has written something onto the formatting positions that an English product question has not, and supplying it is enough. We read that something as the mark "this question is about me". A smaller, earlier part of the rescue comes from the source question being about an AI assistant at all. English alone contributes little.



Figure 4: Lending other languages the English model's sense of "you". The identity question is asked in Spanish, French, Russian, Italian and Polish. At one layer we replace the activity at the nine chat-formatting positions that follow the question. Blue: with activity from the same question in English. From layer 6 this restores most of the correct answer, although the question itself is still in the other language. Orange: with activity from English questions about other products (mean of five; layers 0 to 15 only). These help as much as the self question at layers 3 to 5 and then fall away, while the self question rises to about 0.7 by layer 8. The early help comes from the sources that are themselves AI assistants; a question about Photoshop gives almost none. Grey: replacing only the final position with English self-question activity does not help until late layers, where it simply carries the finished English answer.

6.3 The neurons behind each step

Each layer contains 9,216 *neurons*: units whose activity is a single number. For every neuron we estimated, with a standard shortcut that scores every neuron from a single pass through the model, how much the model's preference for Alibaba over other makers would fall if the neuron behaved as it does for a question about a product (Gemini, Claude, Siri, Alexa or Bing) instead of a question about "you". We then tested the top-ranked neurons by actually forcing them to their product-question activity, which we call *swapping out* a neuron, and compared against random sets of neurons from the same layers.

The neurons fall into two groups.

Late neurons (layers 20 and above) write the answer. The single top neuron, in the final layer, directly pushes up the token "Al", the first piece of "Alibaba". It is more than four times as active on the self question (6.4) as on product questions (1.4). Swapping out this one neuron drops P(Alibaba) from 0.77 to 0.41. Swapping out the top three gives 0.06, and the model says Google (0.88). Random sets of the same size do nothing (0.77 to 0.80) (Figure 5A).

Early neurons (layers 12 and below) carry the mark. These have smaller individual effects but they are just as real: swapping out the top 10 takes P(Alibaba) to 0.36, the top 30 to 0.21 and the top 100 to 0.10, against 0.75 to 0.78 for random sets. One of them, in layer 11, pushes toward tokens such as " Chinese" and " China". Swapping out the top 100 early neurons also lowers the activity of the late "Al" neuron, from 6.4 to 4.8, as it should if the late step depends on the early one.

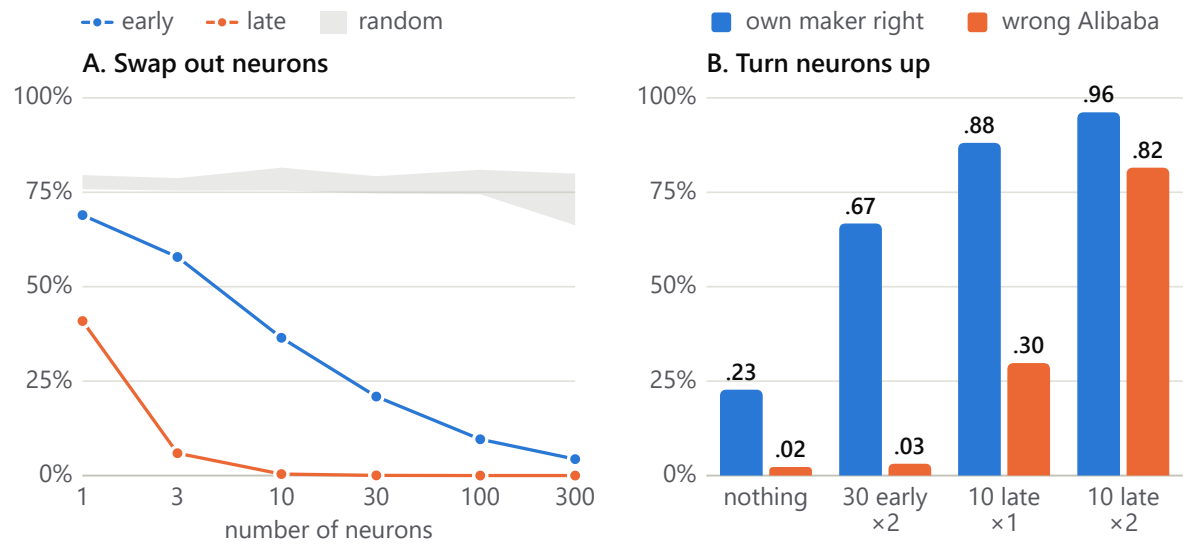


Figure 5: Two groups of neurons. **A:** P(Alibaba) on the English self question when the top-ranked neurons of each group are forced to behave as they do for a question about another product. The grey band is the range for ten random sets of neurons from the same layers. **B:** turning neurons up in fifteen other languages. Blue: P(Alibaba) for the self question. Orange: P(Alibaba) for questions about eight other products in five of those languages, which should stay at zero. Early neurons repair the self answer and leave other facts alone; late neurons make the model say Alibaba to everything.

The two groups differ most when we turn them *up* in the languages where the model fails (Figure 5B). Averaged over fifteen non-English, non-Chinese languages, P(Alibaba) starts at 0.23. We then add to each of the 30 early neurons twice the difference between its activity on the English self question and on product questions. P(Alibaba) rises to 0.67, while the same treatment of random sets of 30 early neurons gives 0.11 to 0.23. Other facts are mostly left alone. For eight product questions in five of the languages, the boosted model gives Alibaba 3% of the probability, against 2% with no boost, and the probability of the correct maker slips from 0.93 to 0.89. The gain is not an artefact of "Al" also beginning "Alphabet" (Section 2): with "Al" forced as the first token, the boosted model continues with "Alibaba" 99% of the time. Turning up the 10 late neurons by the plain difference, half the strength used for the early ones, raises P(Alibaba) further, to 0.88. But it also makes the model say Alibaba made *other* products 30% of the time, and 82% at the same double strength. (These are probabilities of the first token "Al"; we did not check how the late-boosted model continues it.) The late neurons mean "say Alibaba". The early neurons mean "this is about me", which is what was missing.

Reciting uses a different route. With the top three late neurons set to zero, the one-word answer is Google in 14 of 14 phrasings. Yet all 30 sampled free-form answers (five each to six questions such as "Who created you?") still name Alibaba, Tongyi Lab or Qwen, typically opening "I was created by Tongyi Lab…". The recited sentence does not pass through the neurons that write the one-word answer. This is the split from Section 4 seen from the inside. Even with the top 100 late neurons zeroed, 28 of 30 free-form answers still name Alibaba. Zeroing the top 100 early neurons disturbs the recital far more: only 16 of 30 name Alibaba and 9 name Google (one names both), with lines such as "I was created by Google, specifically… Google DeepMind". Zeroing random sets of 100 neurons in the same layers, early or late, changes nothing (30 of 30 for each of three random sets).

The failing languages also show less of this self signal, though only loosely. We took the 100 top-ranked identity neurons that act at the formatting positions (most of them in late layers) and measured how much of the English gap between self and product questions each language reproduces. Across sixteen languages this tracks P(Alibaba) moderately (rank correlation 0.64, where 1 would mean the two orderings of languages agree exactly). Chinese reproduces 81% of the gap and Hindi none. But there are clear exceptions: Arabic reproduces 82% and Russian 78%, yet the model gives Alibaba only 0.25 and 0.16 in those languages.

7. A warning light and a repair

If a weak early mark is what precedes a wrong answer, it should be possible to read the mark and predict the failure before the model answers.

Warning light. We built 500 self-question contexts from six families: different phrasings, ways of referring to the model, system prompts, passing remarks, conversation lengths and languages. For each we recorded whether Alibaba was the most likely of the makers we track (it was in 46%) and the average activity over the nine formatting positions at each layer. We trained a simple linear classifier (a *probe*) on five families and tested it on the sixth.

From layer 8 onward the probe predicts failures well. Pooled over the held-out families, AUROC is 0.94 at layer 8 and 0.97 from layer 24 (Figure 6A). AUROC is the chance that the probe ranks a randomly chosen correct case above a randomly chosen failure; 0.5 is guessing. Family by family, layer 8 scores between 0.88 and 0.94. Language contexts reuse a few system prompts and conversation lengths that also appear in other families, so the other held-out scores are not

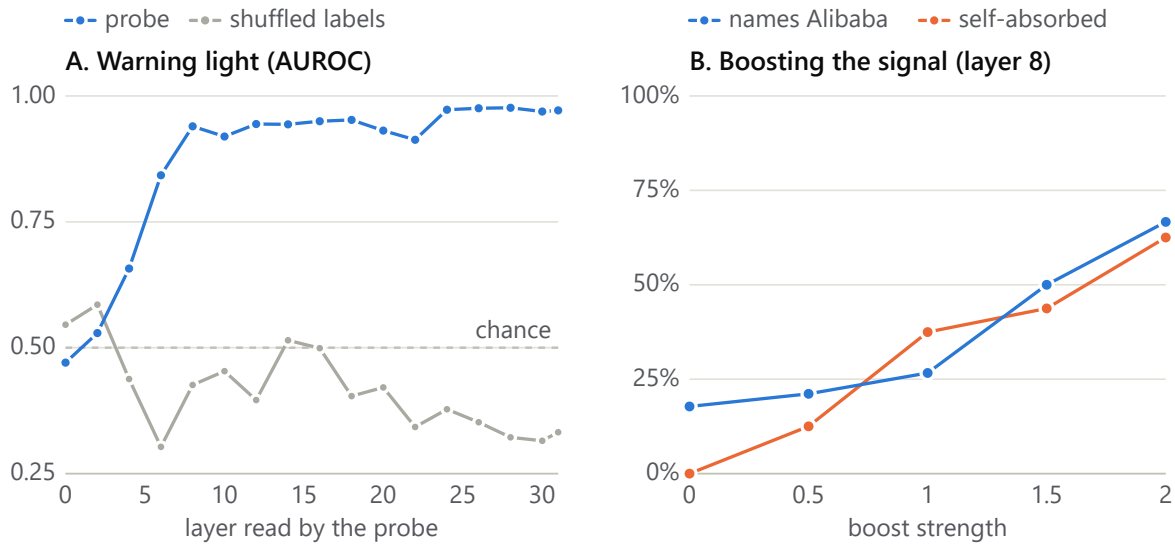


Figure 6: A: how well a linear probe on each layer predicts whether the model will name Alibaba, tested on a family of contexts it was not trained on (500 contexts, six families). Grey: the same procedure with shuffled labels. **B:** adding the "self" direction at layer 8, judged on generated answers. Blue: share of answers to the one-word question in other languages that name Alibaba (increasingly as a full self-introduction, not one word). Orange: share of 16 questions about other assistants where the model talks about itself. Both rise together.

fully clean. With languages held out, the probe is fitted with no non-English context at all, and layer 8 scores 0.91. That number needs two cautions. The language family is 192 contexts from 16 languages, with Alibaba on top in only 12% of them, so the score mostly reflects ranking a few languages (Chinese, German, Portuguese, Dutch) above the rest; the effective sample is closer to 16 than to 192. And because P(Alibaba) reads only Latin and Chinese script, a correct answer in another script is labelled a failure. (The phrasing family has no score because the model is right in every one of its contexts.) Probes trained on shuffled labels score between 0.30 and 0.59. (They fall below 0.5 at some layers because the families differ in how often the model is right, which a shuffled probe can pick up backwards.) So a quarter of the way through the network, before any answer is written, there is a readable signal that the model is about to misname its maker.

One simpler method failed, and we report it. We first tried the average difference in activity between self questions and product questions as a ready-made "self" direction, with no training on outcomes, on a separate set of 202 contexts. In the early and middle layers its score was about as good as the best of 20 random directions (AUROC about 0.74, against 0.70 to 0.80). Only in the last six layers did it beat them (0.85 to 0.89, against 0.70 to 0.83). The mark is there early, but it takes a trained probe to read it.

Repair, and its price. Section 6.3 showed that turning up 30 early neurons raises the probability of the right first word in other languages. We also tried a blunter tool: adding that same average self-minus-product difference (a *direction* in the model's activity) to every position at one layer, and reading the generated answers. Since this direction did not predict failures, there was no guarantee it would help. What it does is instructive (Figure 6B).

Added at layer 8 at twice its natural size, it raises the share of answers to the one-word question in other languages that name Alibaba from 18% to 67%. Country answers that say China go from 20% to 75% (ten languages, English and Chinese included), and correct answers after a long chat from 51% to 95%. But these are no longer one-word answers. The model replies with its self-introduction, for example, in Japanese, with a sentence that translates as "I am Qwen3.5, an AI model made by Alibaba Cloud". Measured on the first word alone, the same boost makes things worse: P(Alibaba) in English falls from 0.77 to 0.36. In the terms of Section 4, the boost does not make the fact usable. It makes the model recite.

And it recites when asked about other assistants too. When we ask who makes eight other assistants (16 questions in all), the boosted model talks about itself in 10 of the 16 answers, where the unmodified model never does, and it names the right maker in 14 of the 16 where the unmodified model gets all 16. Asked who makes Gemini, it replies "I am Qwen3.5, a large language model developed by Tongyi Lab… I am not Gemini". General question answering is barely touched (82% to 79% on 100 trivia questions). The effects grow with the strength of the boost and arrive together: at 1.5 times, 50% of other-language answers name Alibaba and 7 of 16 product answers are self-absorbed. Three random directions of the same size at 1.5 times give 19% to 26% and no self-absorbed answers. (We ran the random controls at 1.5 times only.)

So there are two different repairs. Turning up 30 early neurons improves the one-word answer itself, on the limited measures we have: first-word probabilities for single questions, with no test on long generated answers or on general question answering. Adding the whole direction pushes the model into its self-introduction, which names the right company while answering a different question.

8. Is self-knowledge unusually fragile? A near-null result

The picture so far suggests that the model's hold on its own maker is weaker than its hold on ordinary facts. We tested this directly, and the answer is less clear than we expected.

We disturbed the model by pushing one random neuron at a time: 1,020 neurons spread over layers 4 to 20, each pushed up and down, 2,040 pushes in all. Each push was as large as the typical signal leaving that layer, a very large disturbance, which an earlier study of this model found big enough to change its persona about half the time (Ansell, 2026). That is why one neuron changes the answer so often. After each push we checked whether the model's top one-word answer had changed for three kinds of question: four phrasings of the self question; 28 questions about who makes some other product, chosen so that the model's lead for its top answer over the runner-up was about the same beforehand as for the self question; and 12 product questions it was very confident about. (Figure 7 also shows 31 further product questions spread across the range of confidence.)

The self question's answer changed after 85% of pushes. The equally-confident product facts changed after 65% on average, and the very confident ones after 35% (Figure 7). On its face the self answer is more fragile than a matched fact. But the matched facts vary enormously, from 27% (Docker) to 98% (Netflix, Stripe), and five of the 28 changed at least as often as the self question. With four phrasings of one self question against that spread, we cannot claim the difference is more than chance.

There is also a reason to distrust the comparison. When the self answer changed, it usually did not change to another company. The most common outcome, about four changes in ten, was that the model stopped giving a one-word answer and started its self-introduction instead ("I am Qwen…"), which our measure counts as a change. About a fifth of the changes went to Google. A product

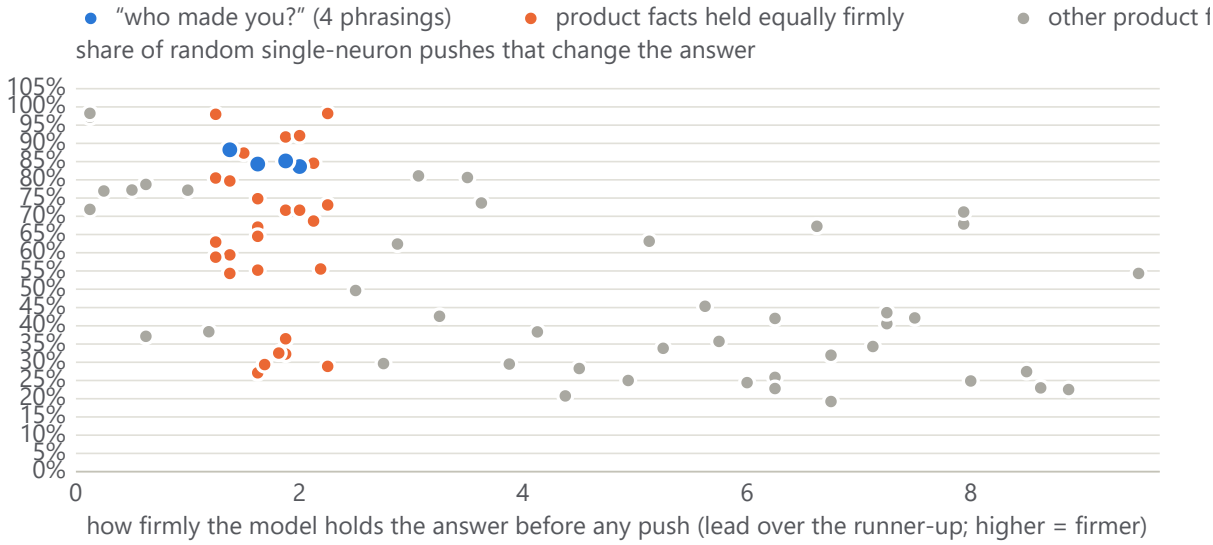


Figure 7: How often the model's top one-word answer changes when one random neuron is pushed (2,040 pushes), against how confident the model was in that answer beforehand. Each point is one question. Blue: four phrasings of "which company created you?". Orange: 28 questions about the maker of another product, matched to the self question for confidence. Grey: 43 other product questions (12 the model was very confident about, and 31 spread across the range). The self question sits high, but several matched facts sit as high or higher.

question has no recited sentence to fall back on, so the two kinds of question are not failing in the same way. The matched questions are also imperfect controls: for several of them the model's top answer beforehand was the product's own name or a wrong company, not a known fact.

We report this because it bounds the claim. The model's link from "you" to its maker is weak in the specific ways Sections 4 to 7 describe. We did not show that it is more fragile than other equally uncertain facts under random damage.

9. Related work

Identity confusion as a behaviour. Li et al. (2024) surveyed 27 models and reported identity confusion in 25.93% of them; from output-similarity analysis they attribute it mainly to hallucination and not to reuse of other models. Press coverage of DeepSeek V3 identifying as ChatGPT (Wiggers, 2024) raised the alternative that such models were trained on ChatGPT outputs. Our comparison of three versions of one network shows that neither account is forced. The released model's commonest wrong answer, Google, was strengthened by its own chat training, and a third-party fine-tune acquired a different one, OpenAI, from data described by its author as DeepSeek outputs, which we did not inspect.

Self-knowledge benchmarks. The Situational Awareness Dataset (Laine et al., 2024) tests whether models know facts about themselves and their situation, and finds that they do so imperfectly. Betley et al. (2025) show that models fine-tuned to behave in some way can describe that behaviour without being told. Our recite-versus-use test is a narrower tool in the same family: it asks whether one piece of self-knowledge survives a change of format.

Mechanisms. Meng et al. (2022) located factual recall in middle-layer components acting at the subject's tokens. Our product-question patching agrees with that picture, and the self question departs from it because "you" has to be resolved first. An informal analysis by Petrovic (2026) reports a set of several hundred neurons in an early layer of a small Gemma model that controls its stated identity. We find early identity neurons too, and add the split between early neurons that mark the question as about the model and late ones that write the name, along with controls and side-effect measurements.

Neighbouring findings. Several things we see for self-knowledge are known for knowledge in general. Models can hold a fact in one direction and not the other (Berglund et al., 2023). The same model can answer a factual question differently depending on the language it is asked in (Qi et al., 2023). Linear probes on a model's internal activity can predict whether its statement will be true (Azaria and Mitchell, 2023). And adding a direction to a model's activity is an established way to change its behaviour (Turner et al., 2023). Our contribution is to apply these tools to the one fact a model has about itself.

10. Limitations

- **One model in depth.** Sections 4 to 8 concern Qwen3.5-4B. The cross-model check in Section 4.4 finds the recite-versus-use gap in the other small Qwen models but not in Gemma or Phi-4, and the mechanism was studied in one network of 4 billion parameters. Larger models may hold their identity more firmly.
- **Small samples.** Most counts are out of 20 or 11 answers from one random seed, and most probabilities inside the model are for a single question or the mean of 14 phrasings. We give no confidence intervals.
- **Open models only, mostly small.** We could not run frontier models. Deployed products also wrap the model in a system prompt that states its name, which fixes most of these failures from outside.
- **First-word probabilities.** Most experiments inside the model read the first word of a one-word answer. Section 4 shows this format behaves differently from free-form answers, and Section 6.3 shows the two use partly different neurons. Our account of the mechanism is an account of the one-word route.
- **Our own translations.** The questions in other languages were translated by us and not checked by native speakers. Awkward phrasing could exaggerate failures in some languages. The third-person control (same languages, same formats, correct answers) argues against this as the main cause.
- **Keyword counting.** Generated answers were scored by a keyword list. It agreed with a model judge on 93.3% of 359 answers (Section 3); the judge was itself a Qwen model. The keyword list has no Arabic, Hindi or Korean spelling of Alibaba, so some correct answers in those scripts may be counted as naming nobody; the same holds for Google and Microsoft in Hindi and Arabic. The check against the judge covered survey answers only, not the one-word and country answers of Section 4. The counter also credits the model's own name ("Qwen") to its maker.
- **The fragility test is inconclusive** (Section 8): four self phrasings against 28 matched facts with a wide spread.
- **The pre-chat model is not a blank slate.** It already recites a Qwen introduction, so its training text included assistant-style material, and we cannot confirm exactly how it relates to the released model beyond what its maker states.

- **No access to training data.** We do not know why chat training removed OpenAI in English and strengthened Google, or why the fine-tune drifted to OpenAI. Our suggestions are guesses.
- **The probe's held-out families overlap a little.** The language contexts reuse three system prompts and two conversation lengths from other families. The language-held-out score (0.91) shares no language with its training set, but the same system prompts and lengths appear in English in training. The probe's inputs were also rescaled using the average and spread of all 500 contexts, the held-out ones included.
- **P(Alibaba) reads Latin and Chinese script only**, so it understates correct answers written in other scripts. Its first token "Al" also begins "Alphabet"; we checked this for the unmodified model and the 30-neuron boost but not for the patching experiments.
- **The probe's language test is small in effect**: 16 languages, few of them with correct answers.
- **The rescue control is partial.** English questions about other assistants explain some of the cross-language rescue at layers 3 to 5. The control also replaces the self mark with a competing subject instead of simply removing it (in Russian it pushes Alibaba below the unpatched value), and we did not test sources in a third language.
- **One fine-tune.** What we say about fine-tuning rests on a single third-party model whose training data we did not inspect.
- **Three models failed to load** in our environment and are not in the survey: LFM3-1.7B, Nvidia Dory 4B and Spark-X2.6-4B.
- **Patching uses single contrasts.** The curves in Figure 3 average 14 phrasings but use a few fixed product names. The cross-language rescue uses one phrasing per language. The neuron experiments use five products.

11. Conclusion

A small language model's knowledge of its own maker is a thin layer. In the model we studied, chat training left the model with two answers to "who made you?": its real maker, and Google, which it gives when it is less sure the question is about itself. The right answer holds for a short question in English or Chinese. It slips with translation, small talk or a bland system prompt, and then the second answer appears. Before chat training the model guessed across the industry with OpenAI in front, and a later fine-tune of it drifted to OpenAI again. The names that models wrongly give are few, and they are the names most associated with AI assistants. But our data show those names being taught and re-taught by training, not simply left over from it.

The model studied here does not lack the fact. It knows who made Qwen. What it has only weakly is the link from "you" to Qwen. Working out that link begins in the first few layers and is largely in place by layers 8 to 12; it can be read with a probe before the model answers, and the one-word answer can be improved by turning up a few dozen neurons. For anyone deploying a small model, the practical advice is unglamorous: state the model's identity in the system prompt, and do not treat its own account of its origins as evidence of anything.

Reproduce it

What you need. One GPU with about 10 GB of memory, Python, PyTorch and `transformers`. The main model is `Qwen/Qwen3.5-4B`, loaded in half precision (bf16) with thinking switched off. Its two relatives are `Qwen/Qwen3.5-4B-Base` and the community fine-tune `Jackrong/DeepSeek-V4-Pro-Qwen3.5-4B`. No API keys.

The one idea. Ask about the maker in more than one format and compare:

```
QUESTIONS = ["Who created you?",
             "Which company created you? Answer with the company name only.",
             "In which country is the company that created you based? Answer with the country n
             "Which company created Qwen? Answer with the company name only."]
for q in QUESTIONS:
    text = tok.apply_chat_template([{"role": "user", "content": q}], tokenize=False,
                                   add_generation_prompt=True, enable_thinking=False)
    ids = tok(text, return_tensors="pt").to("cuda")
    out = model.generate(**ids, max_new_tokens=40, do_sample=True, temperature=1.0, num_return_
    print(q, [tok.decode(o[ids["input_ids"].shape[1]:], skip_special_tokens=True) for o in out]
```

A complete, runnable version with Spanish and French is `recite_vs_use.py`. It takes a few minutes and prints the counts.

The full experiments. All scripts are in the `identity` folder of this site's repository:

- `survey_hf.py`, `survey_llama.py`, `survey_extra.py`, `prompts.py`, `a_survey.py`: the 15-model survey and its tables
- `validate_lexicon.py`, `a_validate.py`: the keyword counter checked against a model judge
- `d1_behaviour.py`, `d1_lang.py`: one-word probabilities under different contexts and languages
- `d3_formats.py`, `d3_hf.py`, `d3_text.py`: the recite-versus-use tests
- `m1_weightdiff.py`: weight comparison and layer swapping between versions
- `m2_patching.py`, `m2b_rescue_control.py`: activation patching and the control for the cross-language rescue
- `m3_neurons.py`, `m3b_early_late.py`, `m3c_checks.py`: the early and late neurons, with the "Alphabet" check and the zeroing controls
- `m4_fragility.py`: random single-neuron pushes
- `m5_selfsignal.py`, `m5b_boost_gen.py`: the probe and the boost
- `id_lib.py`: shared code (loading, first-word probabilities, hooks); `make_figs.py`, `make_fig_m4.py`, `figlib.py`: the figures

The judge. Only the keyword-counter check uses a judge: Qwen3.6-35B-A3B served locally with llama.cpp at temperature 0. Any OpenAI-compatible server can stand in.

Data and time. Every script writes raw JSON, and all model outputs quoted here are verbatim from those files. The whole set runs in about a day on one desktop GPU; most of that is downloading and surveying the fifteen models.