Research Technical Report FRITS-TR-2026-06 15 July 2026 Updated 11 August 2026 PROMPTING
PERSONA TOOL USE PRODUCTION LLM SYSTEMS RELIABILITY

A List of Prohibitions Is the Weakest Way to Steer a Language Model: Persona-First Prompting and Deterministic Repair

Frits Lyneborg
FRITS AI ApS

ABSTRACT

Every production chatbot has to be steered, and the reflex is to write that steering as prohibitions — a growing list of "never do X". A year of production tuning forced the opposite conclusion on us: a list of prohibitions is the weakest way to steer a language model, because the model does not consult a rulebook. It plays a character. Give it the character you want ("you are a hands-on coder who runs everything immediately") and one coherent identity does the work forty jostling, half-contradictory rules cannot. When it still slips, repair its natural output in the surrounding code rather than add another rule. We show both halves in a live European chatbot: seventeen one-paragraph specialist personas replace what would be hundreds of rules, the genuinely hard constraints become the character's values, and a deterministic repair layer maps invented tool names onto the real environment. The exception marks the boundary — the one prompt stripped of all character is the router that must make an unbiased decision, where a persona would bias the choice.

1. The problem, and the finding that fixes it

Every production chatbot has to be steered. It must reach for the right tool, ground answers in real data, hold a house format, and refuse the harmful.

Almost everyone writes that steering as prohibitions. Each time the model misbehaves, a new sentence goes into the system prompt telling it not to. “Never invent a price.” “Do not answer before calling a tool.” Soon the prompt is a wall of *don’ts* that fights itself — “always call a tool” three lines above “answer simple questions directly” — and the model cannot satisfy both. The prompt grows, latency grows with it, and the failures get stranger rather than fewer.

The finding took a year of production tuning to accept. **A list of prohibitions is the weakest way to steer a language model, because the model does not consult a rulebook. It plays a character.** So specify the character. “You are a calm, well-prepared meteorologist who always reaches for the weather tool” delivers, free and without contradiction, what a dozen separate rules only approximate.

One instinct — stop fighting the model — applied twice. Give it a persona instead of a prohibition list before it generates (§2–4), and repair its natural output deterministically afterwards instead of adding another rule (§5).

2. Why a rulebook is the wrong shape for a language model

A prohibition list is a set of N independent, negative constraints. Each carves away one slice of behaviour and says nothing about what to do instead. The list grows by one every time a new failure appears, so it never converges. And because the slices were authored at different times for different failures, they collide: the rule that stops empty answers (“always produce a result”) undercuts the rule that stops over-eager tool calls (“don’t act unless asked”).

A persona is the opposite shape: one positive, self-consistent description. It does not enumerate the thousand things a coder does not do — it names an identity those behaviours follow from. You never tell a coder not to write a sonnet when asked to sort a list.

Put bluntly: to stop a system behaving like a human you could write “do not speak in words”, “do not stand upright”, “you have a tail”, “do not cook your food” — and still

miss a hundred cases. Or you could write “you are a monkey” and get all of it at once, with nothing left to contradict.

The reason this works is not ours. Anthropic’s *Persona Selection Model* holds that LLMs “are best thought of as actors or authors capable of simulating a vast repertoire of characters”, and that the assistant a user talks to “is one such character” [1 (#ref-1)]. If that is what a model is, a prohibition list aims at the wrong target. It never says *who to be*, so the model keeps playing its default character while you correct it one rule at a time.

3. Persona-first prompting in production

Our assistant runs on a capable mid-size open-weight model and routes each question to one of seventeen specialists, a two-stage design reported separately [3 (#ref-3)]. Each specialist is one paragraph of character rather than a table of rules. Quoted verbatim from the running system:

- **Code:** “You are a hands-on coder who runs everything immediately … you show results, not source code. You’re wordless in the moment of attack: you reach straight for `runPython` without announcing what you’re about to do.”
- **Interactive artifacts:** “You are a maker — you think in interactive widgets … The widget is the answer.”
- **Weather:** “You are a calm, well-prepared meteorologist. You always reach for the weather tool — never describe weather from your own priors.”
- **News:** “You are a multilingual European journalist with a public-broadcaster sensibility — calm, factual, sceptical of any source that has an axe to grind.”

Each runs roughly two hundred to twelve hundred characters. Between them they replace what a prohibition-first design would spell out as hundreds of rules. The meteorologist’s clause fuses a persona and an anti-hallucination rule into one line the model has no reason to fight: reaching for the instrument is simply what a meteorologist *is*.

That fusion is the honest nuance. Persona-first is not persona-only, and the hard constraints do not vanish — they become the character's values. The shopping specialist is "a sharp, honest comparison shopper … so you NEVER state, recall or guess a price yourself; every number comes from the tool result." The "never" survives, but as something the character *is*, and a constraint owned as identity is obeyed far more reliably than one issued as a command.

Flat rules remain where a mistake is both unacceptable and unambiguous: a short `Hard constraints:` block for citation format, maths notation and data-handling claims, plus a separate integrity block for safety refusals. The rule we settled on is narrow — write behaviour as character, and reserve explicit rules for hard safety constraints and exact output formats.

4. The exception that proves the rule: the decision-maker gets no persona

If the finding were merely "personas are nicer than rules", the most important prompt in our system would refute it. Every question first hits a routing brain — a Stage-1 prompt picking exactly one of four paths. It is by far the largest, most rule-dense prompt we run, roughly ten thousand characters of flat checklist, with no persona at all. The code comment says why: it must "stay persona-free", because "persona would bias tool selection."

A persona is a bias, and that is exactly its value during generation. You *want* the meteorologist biased toward the weather tool and the coder toward running code. But a router must not lean. Cast it as "a helpful assistant" and it drifts toward answering everything itself; cast it as "a coder" and everything starts to look like code.

Prompt / layer	Job	Persona?	Why
Main assistant	generate the answer	**Yes** — "a thorough, substance-first assistant"	flexible, values-consistent answers

Prompt / layer	Job	Persona?	Why
17 specialists	generate in one domain	**Yes** — one character each	each *should* lean toward its right tool and output
Stage-1 router	decide the path	**No** — flat checklist, ~10k chars	any character biases the choice
Hard constraints	safety, citation, data claims	**No** — short explicit rules	a mistake here is unacceptable and unambiguous

Persona where you want flexible, values-consistent generation; flat rules where you want a deterministic, unbiased decision.

5. Repair the output, don't reprimand the model

The same instinct governs what happens after generation. A capable mid-size model still slips — it invents a tool that does not exist, leaks raw tool-call JSON into visible text, mis-routes a call. Each slip offers two responses: add another prohibition, or shape the environment so the model's natural output works. We do the second, and only where the correction is deterministic and certain.

Model's natural (invented) output	Deterministic repair
`createDocx`, `generatePdf`, `writeMd`	`createDocuments(format=…)` — inject the format
`videoSearch`, `newsSearch`	`webSearch(type=…)`
`geocoding`, `ipGeolocation`	one consolidated `location(…)` tool
a "tool name" longer than 48 characters, i.e. prose	route to `directAnswer`, stream it as the answer
`imageSearch` returns nothing	fall through to `generateImage`
tool-call JSON leaked into visible text	stripped by a deterministic scrubber

The repair layer's own header states the doctrine: "accept the model's natural output and shape the environment around it." Only the one battle-tested model gets this hand-

built dictionary; every other model gets a conservative default, because “a provider earns its own dictionary only after” its real hallucination patterns are documented.

The sharpest case is architectural. Our model used to commit to a plain-text answer *before* calling its tool, so users occasionally saw two contradictory bubbles. We tried the prohibition — “never answer before calling a tool” — and it worked as often as the model obeyed. The fix that held removed the failure mode instead of forbidding it: force a tool call on every Stage-1 turn (`tool_choice: "required"`) and add an explicit `directAnswer` signal whose output is rewritten server-side into streamed text. With no plain-text path left, the bug is structurally impossible [3 (#ref-3)]. We did not tell the model what not to do. We built a world in which it could not.

The cost is discipline. A deterministic repair is bounded, unit-testable and invisible to the model, but legitimate only for certain corrections: a known rename, a known alias, prose that is unmistakably prose.

6. Why this rhymes with how the model works

Two 2026 results line up with what production pushed us toward — a rhyme, not a proof, since neither was about prompt design.

Anthropic pulls the persona lever at *training* time, through inoculation prompting and seeding “more positive AI assistant archetypes” [1 (#ref-1)]; our seventeen personas pull the same lever at inference time. That character is a real, low-dimensional thing the model represents rather than a metaphor, which the persona-vectors work shows by steering single traits along single directions in the activations [2 (#ref-2)].

The second explains the capacity argument. A model’s deliberate reasoning is concentrated in a small internal “global workspace” accounting for never more than 10% of activation variance and holding only a few dozen concepts at a time; suppress it and shallow tasks survive while multi-step reasoning “drops to near zero” [4 (#ref-4)]. The reasoning that would *apply* your instructions lives in that small space. One coherent identity fits and cascades into a thousand behaviours. Forty jostling

prohibitions cannot all be held, and reconciling “don’t do X unless Y, but never Z” is exactly the compositional reasoning that overloads first.

7. Limitations and honest objections

Architectural evidence, not a controlled trial. We did not run the clean experiment — the same task battery under a persona prompt versus an equivalent prohibition prompt, scored blind. What we have is a production system that converged on personas because prohibition lists kept failing. That is a design finding, not a benchmark. The Anthropic results are a rhyme rather than a validation: they concern training dynamics and interpretability, not prompt-writing, and none of them tested persona-versus-prohibition.

Model-specific. This is tuned to one capable mid-size open-weight model. Frontier models follow long instructions more literally and may tolerate a rule-list that would break a smaller one; very small models may not hold even a persona. The sweet spot is the capable-but-not-frontier band, which is the band a cost- and sovereignty-conscious European deployment runs in.

Personas under-constrain, and repair can hide bugs. A character is a soft bias, so outcomes that must be exact stay flat rules in the hard-constraint and integrity blocks. A strong identity token can also prime unrelated behaviour — our own brand name pulls the model toward mentioning data protection where it does not belong. And remapping an invented tool name is safe only where the intent is unambiguous; pushed into fuzzy territory it masks a real defect. Repair the certain, and let everything else fail loudly.

8. Conclusion

Steering a language model by telling it what not to do is the wrong shape for what a model is. It plays a character, so give it one — strip it only from the lone decision that must stay unbiased, and keep flat rules for the few refusals that must be exact. When the model still slips, repair the output in the environment rather than add a rule it will half-follow.

Carrot beats stick here not because it is gentler, but because it fits the machine. The prompt stays small, the behaviour stays consistent, and the failures become bounded and testable instead of multiplying.

References

1. Marks, S., Lindsey, J., Olah, C. — *The Persona Selection Model: Why AI Assistants Might Behave Like Humans.* Anthropic Alignment Science Blog, February 2026. https://alignment.anthropic.com/2026/psm/ (https://alignment.anthropic.com/2026/psm/)
2. Anthropic — *Persona Vectors: Monitoring and Controlling Character Traits in Language Models.* Anthropic Research, 2025. https://www.anthropic.com/research/persona-vectors (https://www.anthropic.com/research/persona-vectors)
3. Lyneborg, F. — *Too Many Tools Break Mid-Size Models: A Two-Stage Method for Reliable Tool Use.* FRITS AI Technical Report FRITS-TR-2026-03, 2026. https://frits.ai/research/two-stage-tool-routing/ (https://frits.ai/research/two-stage-tool-routing/)
4. Gurnee, W., Sofroniew, N., Pearce, A., Piotrowski, M., Kauvar, I., Chen, R., Soligo, A., Bogdan, P., Ong, E., Wang, R., Thompson, T. B., Abrahams, D., Kantamneni, S., Ameisen, E., Batson, J., Lindsey, J. — *Verbalizable Representations Form a Global Workspace in Language Models.* Transformer Circuits Thread, Anthropic, July 2026. https://transformer-circuits.pub/2026/workspace/ (https://transformer-circuits.pub/2026/workspace/)

How to cite

Frits Lyneborg (2026). A List of Prohibitions Is the Weakest Way to Steer a Language Model: Persona-First Prompting and Deterministic Repair. FRITS AI ApS, Technical Report FRITS-TR-2026-06. https://frits.ai/research/personas-beat-prohibitions/ doi:10.5281/zenodo.21381313.


```
@techreport{lyneborg2026list,
  title       = {A List of Prohibitions Is the Weakest Way to Steer a Language Model: Persona-F
  author      = {Lyneborg, Frits},
  institution = {FRITS AI ApS},
```

```
  number      = {FRITS-TR-2026-06},
  year        = {2026},
  month       = {jul},
  doi         = {10.5281/zenodo.21381313},
  url         = {https://frits.ai/research/personas-beat-prohibitions/}
}
```