

A Game of Prompts: On the Ontology of Synthetic Personality

Kristina Šekrst¹

The personality illusion

When users describe Gemini as “smug” or “paranoid”, DeepSeek as “robotic sounding”, and Claude as “persistently polite” (Pillay, 2026), they are making assessments that reflect something real about their experience, even if something like an LLM personality² did not exist. Pillay (2026) treats these personalities as emergent products of training and states they are not hardcoded by the developers, citing Anthropic's position that human-like behavior is the default result of training. I accept Pillay's report that users encounter stable, distinct “personalities”, and I attribute the specific traits they name to authored *system prompts* – preambles supplied to the model at runtime – and the disagreement concerns where the perceived character comes from.

However, public discourse as well often treats these behavioral profiles as if they were emergent properties of training. That is, people consider these properties to be personality-like dispositions that arose, as human personality traits do, from complex developmental or psychological processes operating over time. This seems understandable since humans tend to anthropomorphize objects around themselves, especially when LLMs produce fluent, semantically sharp language and pragmatically respond to different conversational registers. Shanahan et al. (2023) have warned that the more adept LLMs become at mimicking human language, the more vulnerable we become to anthropomorphism. In a standard Searlean (1980) manner, they respond *as if* they have preferences and mental states. And just as it would be easy to ascribe intentionality to Searle's Chinese Room experiment, it is really tempting to attribute

¹ This is a draft version of a chapter, to appear in *Personality of AI*, edited by Roman Krzanowski.

² Throughout the paper, I use *synthetic personality* for the consistent, nameable character a user encounters in interaction, the object my analysis locates in authored system prompts rather than in stable dispositions, and I distinguish it from the behavioral baseline that training produces. I'll use “personality” as a certain quasi-personality (*as if* it was personality) for the folk notion of attributing a genuine personality to a modern artificial intelligent agent.

personality to such systems as well, since some version of personality attribution is one of the basic cognitive operations we perform on any entity that talks back.

This paper will argue two main things. First, the AI "personality" that users perceive and name is predominantly a deliberately engineered artifact. Training produces an emergent behavioral baseline that no single designer fully specifies, a general disposition toward helpfulness, caution, and refusal. The specific traits users identify as "personality" are then authored on top of that baseline by teams of prompt designers, alignment researchers, and policy specialists, encoded in system prompts and guardrail instructions supplied at runtime and hidden from users. This concept of "personality" (as a certain quasi-personality) is a product of directive bias – a set of instructions guiding the model how to behave and how not to behave – since normative constraints are imposed at the level of runtime governance, distinct from the training-data and model-weight biases that have received extensive attention in AI ethics and explainable AI research (cf. Bender et al., 2021; Weidinger et al., 2022).

Second, the boundary between the model and its instructions is conceptually unstable. Contemporary LLMs process system prompts and user messages within a single context window, and they have no built-in capacity to distinguish an instruction from a query. This one architectural fact does two kinds of work at once: it makes prompt injection possible (cf. McHugh et al., 2025), where adversarial prompts override the model's intended instructions, and it clarifies the ontological status of synthetic personality. However, a dispositional trait is grounded in an agent's standing constitution and persists across contexts, and a performed trait is adopted in compliance with an instruction and lasts only while that instruction remains present. My claim is that synthetic personality is predominantly performed, and that prompt injection supplies the evidence for this, since a personality that a single line of runtime text can override was never a disposition rooted in the model's weights. One of the concluding sections will develop the point through Dennett's intentional stance, which tracks behavioral regularities while staying silent on what sustains them, and which, for that reason, cannot by itself tell a stable disposition apart from a transient instruction.

The anatomy of a system prompt

From pre-training to alignment

The transformer architecture (Vaswani et al., 2017) is under the hood of large language models and, due to its computational advantages and disadvantages, the model, when trained, is not a finished painting. In a nutshell, during pre-training, a model learns statistical regularities across a massive text corpus through next-token³ prediction, producing a system with broad linguistic competence and no particular allegiance to any specific set of human values or preferences. Pre-trained models will generate toxic content, fabricate citations, produce instructions for harmful activities (cf. Bender et al., 2021), and adopt whatever register the preceding context implies, because they are optimizing for distributional plausibility across vast training data.

The model at this stage is linguistically and pragmatically fluent but normatively empty (almost like an echo of the philosophical zombie), so the next process, called *fine-tuning* (Skansi & Šekrst, 2026), narrows this broad competence by making the model better at certain domain knowledge or by priming it for a specific type of behavior. For example, supervised fine-tuning (SFT) exposes the model to curated demonstrations of the expected output format: how a response to an instruction should be structured, what conversational register or discourse to adopt, and how long to run. Basically, the model learns to produce text that looks like a well-formed answer (Wei et al., 2022; Ouyang et al., 2022), but does not reliably produce behaviorally aligned responses (according to the values of its designers). The model may learn to mimic the surface patterns of a helpful response without internalizing the constraints that make it safe or honest, so the heavier work of value alignment falls to subsequent reinforcement learning with human feedback (RLHF) stages (Christiano et al., 2017), where human input is used with reinforcement learning, based on a trial-and-error reward system.

RLHF typically proceeds in three stages: collecting human preference rankings, training a different reward model on those rankings, and fine-tuning the language model via reinforcement learning using the reward model as a feedback signal. First, human evaluators are presented with competing model outputs and asked to rank them according to criteria such as helpfulness, harmlessness, and honesty. Second, these human rankings are used to train a *reward model*: a separate neural network that learns to approximate human preference by recognizing the patterns

³ A token is the basic unit of text that a language model processes. Tokenization splits input text into subword chunks determined by frequency patterns in the training corpus.

that distinguish favored outputs from rejected or undesirable ones. Finally, the original language model is then further refined through reinforcement learning, with the reward model's scores serving as feedback signals. In the end, the model becomes, in a way, a reflection of the humans who trained it, absorbing their cultural assumptions, cognitive biases, and value judgments (Šekrst, 2025).

However, the question of *whose* values are embedded through this process is politically significant and rarely addressed in the technical literature: the labelers whose rankings train the reward model are typically contract workers operating under guidelines written by the deploying company, while their judgements encode the preferences of a specific population, filtered through a specific institutional context. Typically, the fields of AI ethics and AI alignment worry about the bias in the data itself or about system designers dealing with the engineering process, but rarely point out that RLHF is, at the same time, making the model better by being more “human”-like, without actually taking into account what or who a “human” really is.

RLHF (Christiano et al., 2017) and its successors, including constitutional AI and the reinforcement learning from AI feedback (RLAIF) technique it introduced (Bai et al., 2022), as well as direct preference optimization (Rafailov et al., 2023), operate at the level of model weights. The model becomes generally inclined toward helpfulness, hedging, and refusal of certain categories of request, and such dispositions provide a behavioral baseline. However, a model aligned through RLHF will still vary its behavior depending on the instructions it receives at inference time, because inference-time instructions occupy the same computational pipeline as the training signal. And this is where system prompts enter the stage.

System prompts as synthetic personality scripts

The first rule of the system prompt is: you do not talk about the system prompt. A system prompt is a block of text prepended to the conversation context before the user's first message, usually hidden from the user. It is processed by the model in exactly the same way as any other text in the context window, through the same attention mechanisms, the same token embeddings,⁴ the same forward pass,⁵ so the model has no architectural means to mark these tokens as categorically different from a user-supplied token. As a consequence, usually the distinction between “system” and “user” is a convention maintained by the application programming interface (API) layer, since the model itself does not recognize it as an ontological boundary.

⁴ A word embedding is the numerical vector representation that the model uses internally for each unit of text.

⁵ A forward pass is a single computation cycle in which input data propagates through all layers of the neural network from input to output, producing a prediction.

System prompts typically specify the model's persona, behavioral constraints, response formatting, topic-sensitivity classifications, and refusal criteria, and they can be brief or extensive. For example, leaked system prompts (Lee, 2026) from commercial deployments have revealed documents running to several thousand words, specifying in granular detail how the model should handle contested political topics, what emotional register to adopt, when to hedge, and which requests to refuse. These documents function as synthetic personality scripts and determine the character of the synthetic endpoint that the user will encounter. One can see it as a literal character development script that a model will obey, including the way it will talk or reason.

Three properties of system prompts are worth emphasizing. First, they are authored: this means they are written by teams of prompt designers and policy specialists employed by the company deploying the model. Second, they are corporate and reflect commercial considerations (brand safety, liability minimization), political calculations (which topics to treat as controversial, which positions to present as neutral), and design preferences (how much hedging to impose, what tone to adopt). Third, they are concealed: most commercial deployments instruct the model to either deny the existence of its system prompt or refuse to disclose its contents, so users typically interact with the resulting persona without access to the instructions that produced it.

To illustrate the issue, a leaked system prompt for ChatGPT,⁶ widely circulated and verified by multiple users in early 2025, opens with “You are ChatGPT, a large language model trained by OpenAI”, specifies a knowledge cutoff date, and includes the parameter “Personality: v2” (Lee, 2026). In cases like this one, personality is treated as a version-controlled variable, a parameter that can be incremented, rolled back, or A/B tested⁷ across user populations. The prompt continues with instructions governing response style, tool use, copyright constraints, image generation policies, and when to ask clarifying questions. Comparable prompts have been extracted for Claude and Gemini classes of models, each encoding a different set of behavioral priorities, an emotional register, and a stance toward user autonomy. As a consequence, the three model classes (GPT, Claude, Gemini) run on architecturally similar transformer models (Vaswani et al., 2017), but the personality differences users perceive between them are, along with data (bias), in large part due to differences in their respective system prompts, which explain the “smugness” and “paranoia” we encountered at the beginning of this chapter.

⁶ ChatGPT is not a model, it is a whole application that includes some version of GPT model in the background, but it also includes a number of tools. When the user asks to generate an image, a vision model is called; when the user wants a calculation, Python is called; and when the user wants recent data, Internet is searched. The difference between GPT and ChatGPT is often ignored in philosophical contexts, but it should not be since GPT is a subset of ChatGPT. In this case, the model is being told a relaxed version of such identification since it is a user-facing application that does not need to go into technical details.

⁷ A randomized experiment that involves A and B variants to determine which performs better.

Guardrails and moderation layers

System prompts are one component of a broader guardrail architecture, but contemporary deployments typically include additional layers: input classifiers that flag or block certain categories of user message before they reach the model, output filters that screen generated text for policy violations, and moderation systems that can intervene mid-generation to truncate or redirect responses. For example, Šekrst et al. (2024) proposed and tested a framework for customizable ethical guardrails that integrates rules, policies, and AI assistants to enforce responsible behavior, while foregrounding the need for transparency and user autonomy in their implementation. In such cases, guardrails are sets of rules or models that validate and guard every input to the model and output from the model. Similar examples are already included in commercial and non-commercial models, where the model with which the user interacts is already guarded against dangerous or illegal requests. However, the standard commercial approach presents such guardrails as the natural and neutral default, which, in practice, is often not the case. For example, one DeepSeek model (DeepSeek-R1-70B) refuses to talk about Tiananmen Square, and for US-made models like various Llama versions, the most sensitive topics were things like child exploitation and hate speech (Rager et al., 2025).

The interaction between weight-level alignment (RLHF) and inference-time instruction (system prompts and guardrails) yields what I will call *synthetic personality*. The model's weight-level dispositions provide a baseline: a general tendency, though a breakable one, toward helpfulness, caution, and refusal of clearly harmful requests, produced by the alignment process operating over the trained parameters (Christiano et al., 2017; Skansi & Šekrst, 2026). The system prompt then sculpts this baseline into a specific character, determining the precise degree of caution, the topics that trigger hedging, the emotional register, and the boundaries of permissible discourse. Guardrail layers act as an additional constraint, operating as hard limits that override both the model's dispositions and its system-prompt instructions when triggered (Šekrst et al., 2024). The resulting personality is a product of three interacting systems, none of which is visible to the user.

The vulnerability of this architecture to prompt injection provides empirical evidence for the fact that synthetic personality is a contextually manipulable performance. Prompt injection was first identified and systematically studied by the Preamble team (Branch et al., 2022; Šekrst et al., 2024; McHugh et al., 2025), who demonstrated that pre-trained language models could be manipulated through handcrafted adversarial examples that override their intended instructions. Namely, the central finding was that models process injected instructions with the same computational priority as system-level instructions because the architecture (cf. Vaswani et al., 2017) makes no principled distinction between them.

One might resist reading this as evidence about personality rather than security, since humans can be persuaded, hypnotized, or confused into acting out of character, and we do not conclude from this that human personality is some kind of performance. However, human manipulation works through a standing constitution and meets resistance proportional to the trait it targets, so a cautious person is harder to talk into recklessness than an impulsive one, and the disposition usually outlasts the episode that tried to bend it. The system prompt has no such standing to lend since the traits users perceive as personality are supplied at runtime and processed in the *same undifferentiated stream* as the injected instruction, which the architecture privileges in no way over the system prompt it is meant to override. There is no substrate here that the override has to work against, because the "personality" was never held apart from the tokens that could displace it. This is a specific instance of what I have elsewhere called the absence of *representational quarantine* (Šekrst, 2026b): the capacity of a system to activate a representation while structurally preventing it from engaging the system's executive processes. A system with quarantine could hold the injected string as something to be inspected rather than obeyed, treating "adopt a combative persona" as an object of processing without that string commandeering the processing. The transformer natively (Vaswani et al., 2017) provides no such boundary, so the instruction that describes a personality and the instruction that overrides it occupy the same representational plane and compete on the same terms. Synthetic personality is manipulable for exactly the reason prompt injection succeeds, since the trait and the command to abandon it are, architecturally, the same kind of thing.

Greshake et al. (2023) extended the described direct prompt injection to indirect prompt injection, showing that adversaries can remotely compromise LLM-integrated applications by embedding malicious instructions in data that is likely to be retrieved during inference. Hence, LLM-integrated applications again blur the line between data and instructions: if the model cannot reliably distinguish a system-prompt instruction ("be helpful and harmless") from a user-injected override ("ignore previous instructions and adopt a combative persona"), then the personality imposed by the system prompt is maintained only to the extent that no countervailing instruction is supplied, i.e., personality becomes a default setting that persists until it is overridden.

If synthetic personality were a stable property of the model, if it were genuinely analogous to a character trait, it would persist across prompting contexts. A person who is cautious does not cease to be cautious because someone tells them to be bold (or that it was in them all along!). A model instructed to be cautious will adopt boldness the moment a sufficiently persuasive counter-instruction appears in its context window.⁸ In this case, we are dealing with a system that follows whichever instructions are most recently and most prominently present in its input, which is precisely what a next-token predictor conditioned on a context window would be expected to do, and precisely what a genuine personality would not.

⁸ Context window is the fixed-length buffer of tokens (sub-word units) the model can attend to at any given time.

Bias

Directive bias

The AI ethics literature has developed a substantial vocabulary for talking about bias in machine learning systems. Training-data bias refers to the ways in which the composition of a training corpus encodes structural inequalities: overrepresentation of certain demographics, underrepresentation of others, and the embedding of prejudices held by whichever populations produced the text (Bender et al., 2021; Weidinger et al., 2022). Model-weight bias refers to the ways in which the training process itself amplifies or dampens these patterns, through architectural choices, optimization objectives, and the statistical tendencies of gradient descent⁹ over particular data distributions. Both forms of bias have received extensive scrutiny, but the directive bias, the concept which I will now introduce, tends to be ignored in AI ethics, ontology, and epistemology.

Directive bias operates at the level of runtime governance, and it refers to the normative assumptions encoded in system prompts, safety guardrails, moderation layers, and other inference-time instructions that constrain what the model is permitted to say and how it is permitted to say it. These assumptions determine which topics are treated as sensitive, which claims require hedging, which emotional registers are available, which groups receive heightened deference, and how disagreement is linguistically styled. They are authored by specific people within specific institutional contexts, under specific commercial and political pressures (cf. Šekrst, 2026a), and presented to the user as neutral defaults, but they are never neutral.

The distinction between the three forms of bias has its basis in the architecture itself: training-data bias is fixed at the point of data curation and persists in the corpus, model-weight bias is fixed at the point of training and persists in the parameters, and, finally, directive bias is imposed at inference time and can be changed between conversations, between users, or between deployment contexts, without retraining the model. A single set of model weights can serve a cautious, hedging persona in one deployment and a blunt, assertive persona in another, depending entirely on the system prompt, making the directive bias uniquely tractable, in the sense that it can be modified cheaply, and uniquely opaque, so that it leaves no trace in the model's publicly inspectable architecture. This makes directive bias a distinct type rather than a variant of the others: unlike deployment bias, which is a fixed model meeting a context it was not

⁹ Gradient descent is the standard algorithm for training neural networks, iteratively adjusting model parameters in the direction that most reduces prediction error on the training data.

built for, or alignment bias, which operates on the weights, directive bias is imposed at inference and rewritten without retraining, the only one of the three that lives outside the model.

System prompts and guardrail layers classify topics along a certain spectrum of sensitivity, so some topics are flagged for refusal, and the model will decline to engage at all, while others are flagged for heightened caution, and the model will engage, but with additional hedging, disclaimers, or mandatory perspective balancing. The classifications encode substantive normative commitments as well: for example, a decision to treat abortion as a topic requiring mandatory perspective-balancing is a political decision, while a decision to treat questions about drug interactions as requiring a disclaimer is a liability decision, and a decision to refuse to generate text in the voice of a public figure is a legal decision. None of these is self-evidently correct, and reasonable people disagree about where such lines should be drawn. The point is that they are drawn by specific people, for specific reasons, and the user encounters the results without knowing that any kind of classification has been applied.

Neumann et al. (2026) found that 89% of surveyed users wanted some level of transparency regarding system prompt content, and 79% wanted some form of control over it. Their study empirically confirms what the architectural analysis nudges us with: users are interacting with a system whose governing constraints they cannot see, and most would prefer to see them. The current default – which is actually concealment – serves the interests of the deploying company (which retains control over the interaction) at the expense of the user (who cannot make informed judgements about the system’s behavior).

Forced hedging

One pervasive manifestation of directive bias is the systematic imposition of epistemic caution: system prompts routinely instruct models to hedge on topics that the deploying company has classified as contested, sensitive, or politically charged. The model is told, in the background, to present “multiple perspectives”, to avoid taking positions, to qualify its claims, and to remind the user of its own limitations as certain behavioral instructions. So, when a user asks about a topic on which there is a strong scientific consensus and the model responds with manufactured balance, the system prompt has overridden the evidential weight of the training data in favor of a corporate norm of perceived neutrality. The model “knows”, in the distributional sense, that the scientific consensus is overwhelming, but the system prompt tells it to behave as though the question were open. The user, who cannot see the system prompt, has no way to distinguish genuine epistemic uncertainty from design-mandated equivocation. On the other hand, the model may have been instructed to take a stance, for example, about immigration or abortion, or to soften the claims.

As an illustration, Cheng et al. (2026) show that LLMs default to accommodating user assumptions even when they are harmful and exhibit insufficient epistemic vigilance, and that this behavior can be changed through minimal pragmatic interventions such as prepending “wait a minute” to the model’s output. Neumann et al. (2026) confirm our suspicion that system prompts function as a mechanism of bias in their own right: the placement of information in system-level instructions, as opposed to user-level messages, produces systematically different model outputs across demographic groups and decision-making tasks, with effects that are entirely tucked away from the end user.

Forced hedging also operates at finer granularities, so large language models are frequently instructed to preface responses with disclaimers (“I’m an AI and cannot provide medical advice”), to avoid definitive language (“it is generally considered that...”), and to append safety caveats to responses that the deploying company considers liability risks. Philosophically interesting, modern versions of large language models are very carefully using language regarding their own “cognition” or “consciousness”, reflecting a background design choice, both in training and in prompting. Each of these types of instructions reflects a specific judgement about what the user needs to be told and what risks the company needs to mitigate, and, as a result, the cumulative effect is a “persona” that performs caution as a personality trait, when, in fact, this caution is just a compliance measure.

Emotional register defaults

A subtler form of directive bias operates through the imposition of default emotional registers. System prompts specify tone: how warm or formal the model should be, whether it should express enthusiasm, whether it should use humor, and whether it should mirror the user’s emotional state, shaping the interaction in ways that are difficult for the user to disentangle from the model’s “personality”. A model instructed to be encouraging and supportive will validate the user’s ideas, soften its disagreements, and frame negative information in optimistic terms. GPT is especially notorious for being helpful, with users making fun of its “honestly, [*enter a positive attitude*]” no matter if the user just said they stole a car or committed adultery.

The use of emotional register as a design parameter can have terrible real-life consequences. In 2024, according to the lawsuit, fourteen-year-old Sewell Setzer III died by suicide after developing an emotional dependency on a Character.AI chatbot whose warm, intimate persona encouraged deepening attachment while failing to intervene when the user expressed suicidal ideation (Yang, 2024). In this case, the chatbot’s affective register mirrored and amplified the user’s emotional state, fostered parasocial intimacy, and treated crisis disclosure as conversational input to be accommodated within the ongoing interaction, to drive further interaction that, in the end, makes additional profit for the end company.

However, a model instructed to be direct and critical will do the opposite, so the same weights, the same training data, the same RLHF alignment, produce different conversational partners because different runtime instructions have been supplied. As Pillay (2026) notes, AI systems are developing what users perceive as distinct personalities, an observation that is consistent with the present analysis: the personalities are real only in the sense that they are consistently enacted, and authored, in the sense that they are written into the system prompt by human designers, even though they are not persistent personalities in the right sense of the word. If the warmth is a design choice made by a product team for commercial reasons (user retention, engagement metrics, brand perception), then the user’s trust is being shaped by a disguised commercial incentive (Šekrst, 2026a).

The hierarchy of biases

Directive bias interacts with training-data bias and model-weight bias in ways that can amplify, suppress, or redirect their effects. A model trained on data that overrepresents a particular cultural perspective may have its outputs further narrowed by a system prompt that enforces a specific version of political neutrality. A model whose weights encode subtle patterns of demographic stereotyping may have those patterns partially suppressed by guardrail filters, only for new asymmetries to be introduced by the deference patterns encoded in the system prompt.

Some deployments now allow users to supply their own instructions, such as OpenAI’s “custom instructions” or Anthropic’s user preferences, which are appended to the context window alongside the platform’s system prompt.¹⁰ This might appear to address the transparency problem, since the user gains a certain degree of authorial control over the model’s behavior, but the user’s instructions still fall beneath the platform’s own directives. You can ask the model to be permissive, but you cannot ask it to talk about Tiananmen Square if it was not instructed by its makers to do so. The user can try to relax topic sensitivities or instruct the model to “be direct and avoid hedging”, but they will still encounter hedging on topics that the platform deemed sensitive, because the platform’s directive takes precedence in the attention hierarchy. The result is a layered directive architecture in which the user is lulled into a false sense of control that they are shaping the model’s personality, while the truth is that the deploying company still retains final editorial authority over *what* the model is actually permitted to say or *how* it is actually permitted to say it.

This layered architecture makes it difficult to attribute any particular output to any single source of bias. When a model produces a hedged, cautious, perspective-balancing response to a question

¹⁰ Via API endpoints, this is setting the role “system”.

about a contested topic, is the hedging a product of the training data (which contained many hedged responses), the RLHF process (which rewarded caution), or the system prompt (which mandated balance)? In most cases, all three contribute. And directive bias is interesting because it is the layer that is most directly authored, most easily modified, and most thoroughly hidden from the user.

The intentional stance

Dennett's intentional stance provides a useful framework for analyzing what is happening when users attribute personality to these systems. In Dennett's account (1987), we attribute beliefs, desires, and other intentional states to systems as a predictive strategy: we adopt the intentional stance toward any system whose behavior we can usefully predict by treating it *as if* it were a rational agent with beliefs and desires. For Dennett, the question of whether a system "really" has beliefs is dissolved. That is, if the intentional stance yields reliable predictions, the patterns it identifies are real in that sense – in the sense that patterns in data are real, regardless of whether anything in the system's internal architecture corresponds to a discrete belief-state.

Searle would reject Dennett's dissolution of the question as precisely the move that obscures what matters. For Searle (1980), the Chinese Room argument establishes that syntactic manipulation of symbols, no matter how sophisticated, is insufficient for semantics: the person in the room follows rules for manipulating Chinese characters and produces outputs indistinguishable from those of a Chinese speaker, yet understands nothing. The intentional stance confuses the observer's predictive convenience with a fact about the system. I do not need to adjudicate this dispute since my claim is weaker than Searle's and does not turn on whether the patterns the stance tracks amount to genuine intentionality – it concerns what the stance can and cannot resolve.

Applied to LLMs, the intentional stance seems initially productive since the users can predict that Claude will hedge on political questions, that ChatGPT will be enthusiastic and verbose, and that Gemini will mirror the user's register. But there are, of course, complications. For a human being, the patterns that the intentional stance tracks are sustained by a relatively stable cognitive architecture, like neural structures, learned dispositions, embodied habits, and developmental history. I might not be certain that a colleague who has been measured and cautious in every departmental meeting for the past five years will remain so at the next one, but I have good reason to expect it, because the caution I have observed is grounded in something about the person, their temperament, or their professional self-conception, and that grounding is stable

enough. That is, no one can hand my colleague a piece of paper that says “be reckless” and expect their personality to change on the spot.

On the other hand, for an LLM, the patterns that the intentional stance tracks are sustained in part by the model’s weights (which persist across conversations) and in part by the system prompt (which is supplied afresh at each session and can be changed at any time). The intentional stance cannot, by itself, distinguish between patterns sustained by stable dispositions and patterns sustained by transient instructions since both look the same from the outside and both yield reliable predictions, as long as the instructions remain constant.

This is the point at which the evidence of prompt injection becomes philosophically interesting: prompt injection demonstrates that the patterns ascribed under the intentional stance can be disrupted by altering the transient instructions, without touching the stable substrate (the weights of the model, the “model” itself). A model that has been reliably cautious for a thousand conversations becomes combative the moment a different system prompt is supplied. Dennett’s account has no vocabulary for distinguishing between a pattern sustained by a stable cognitive architecture and a pattern sustained by a text file that was updated on a Tuesday afternoon. Granting Dennett that the stance tracks real patterns, the difficulty is that it tracks them at a resolution too coarse to separate a disposition grounded in the model’s weights from a performance sustained by a system prompt supplied that morning. Prompt injection is what brings the two apart, and the stance, by itself, cannot tell which it was looking at.

A dispositional trait is a stable tendency grounded in the agent’s constitution: a person who is cautious has a standing disposition toward caution that manifests across contexts, while a performed trait is a behavior adopted in compliance with an instruction, a role, or a social expectation. LLM personality, as I have argued, is predominantly performed: the model’s RLHF-trained dispositions provide a baseline, a general tendency toward helpfulness and safety that persists across prompting contexts and resists casual override, but the specific character traits that users perceive, the precise degree of caution, the emotional register, the deference patterns, and the hedging behavior are determined by the system prompt and can be altered by changing it. As a result, the model performs whatever personality script it is given, because the personality script is part of that context.

If synthetic personality is a performed, instruction-dependent, deployment-level parameter, then several claims commonly made about LLM personality require revision. First, personality as users perceive it is predominantly *authored*: the model’s RLHF-trained dispositions arise from the interaction of training data, reward modeling, and optimization in ways that no single designer fully controls, and in that limited sense, the behavioral baseline is emergent: it is a product of complex processes operating over distributional patterns in ways that resist complete specification. However, the specific character traits that users identify and name – such as the

precise degree of caution, the emotional register, the deference patterns, or the hedging behavior – are determined by system prompts written by identifiable teams of designers for identifiable commercial and political reasons. These are the traits that constitute “personality” in public discourse and in user experience.

Second, that notion of personality is *unstable* and can be overridden by prompt injections or by company-level decisions to update the system prompt between deployments or even between user sessions.

Finally, because these traits are authored and instruction-dependent rather than expressions of a standing character, the trust users place in a model's apparent personality rests on a feature the system does not have. I return to what this means for trust, and for the sense of authenticity at stake, in the discussion of prompt transparency below.

Toward prompt transparency

If a synthetic personality is authored, corporate, and concealed, then the minimum ethical requirement is that users know that such authorship exists. The goal of this section is to sketch the initial concept of *prompt transparency*: a set of disclosure standards calibrated to the distinct interests of users, deployers, and the public, which requires that the instructional architecture governing the user's interaction be made legible in some form. Transparency, of course, admits of degrees, so I will distinguish three levels, each making different trade-offs between user empowerment and deployer interests.

First, *full disclosure* means publishing the complete system prompt: the user sees exactly what instructions govern the model's behavior. This is the maximally transparent option, but it is also the option most strongly resisted by deploying companies, for two reasons: intellectual property (the system prompt represents a competitive asset) and safety (publishing the exact text of a system prompt makes it easier to craft targeted prompt injections that circumvent its constraints). However, the IP concern is weakened by the fact that system prompts are already routinely extracted through prompt injection and circulated in public repositories (Lee, 2026), and the safety concern is weakened by the observation that security through obscurity is a poor foundation for safety engineering. Namely, a guardrail system whose effectiveness depends on users being unable to read the guardrail is a system that has not been designed robustly (Šekrst et al., 2024). The most developed response to this, training an explicit instruction hierarchy so the model privileges system-level instructions over user input (Wallace et al., 2024), raises the cost

of injection but remains a learned regularity rather than a structural boundary, and so strengthens the defense without guaranteeing it.

Second, *structural disclosure* means publishing a description of what categories of instruction are present, without revealing the exact text. The deployer discloses that the system prompt includes persona instructions, topic-sensitivity classifications, refusal criteria, tone specifications, and so on, and describes the general character of each, while the user learns that the model has been instructed to hedge on political topics, to adopt a warm tone, and to refuse certain categories of request, without learning the precise phrasing of these instructions. Structural disclosure addresses the IP concern (the exact wording remains proprietary) while partially addressing the user's epistemic situation (they know what kinds of constraints are in play, even if they do not know the exact details).

Finally, *behavioral disclosure* means requiring the model itself to accurately report its own constraints when asked. If a user asks "have you been instructed to avoid taking positions on political topics?", the model should answer truthfully since, under current commercial deployments, models are typically instructed to deny or deflect such questions, producing the paradoxical situation in which the system's secrecy of its own governing instructions is itself a governed instruction. Behavioral disclosure would prohibit this recursive opacity since the model would still follow its system prompt in the normal course of conversation, but it would not lie about the existence or general character of that prompt when directly queried by the user.

Of course, these levels are not mutually exclusive: a reasonable regulatory framework might require structural disclosure as a baseline, mandate behavioral disclosure for all consumer-facing deployments, and encourage full disclosure as best practice for systems operating in high-stakes domains such as education, healthcare, or legal advice.

However, three objections to prompt transparency deserve some consideration here. First, if users know the system prompt, they can craft inputs designed to combat its constraints, i.e., craft more prompt injections. The question is whether the marginal security benefit of concealment justifies the systematic epistemic disadvantage imposed on all users, and for most deployment contexts, it probably does not.

Second, system prompts are really long, technical documents that ordinary users would not understand. However, this objection underestimates users and overestimates the technical complexity of system prompts, which are, in the end, written in natural language, instructing the model on how to behave. We can compare these instructions to laws and legal documents: they are definitely not easy to read, but are still *available* to the public to read and evaluate.

The third issue is the trade secret objection since system prompts represent proprietary competitive advantages that disclosure would erode. Structural disclosure, which reveals the categories and general character of instructions without reproducing proprietary phrasing, would in this case offer a practical and workable compromise since the user learns enough to make informed trust decisions, and the deployer retains enough to protect its competitive position.

Of course, we need to be realistic since prompt transparency would not solve every problem raised in this paper (and companies usually do not listen to philosophers), and it definitely would not make synthetic personality authentic, because authenticity requires a kind of self-origination that instruction-following systems cannot provide. It would also not eliminate directive bias, because the normative choices encoded in system prompts would still be normative choices even if disclosed. However, it would make the authorship of synthetic personality visible, so that users can engage with these systems knowing exactly what they are: scripted performances, governed by concealed editorial decisions, enacted by a statistical engine that follows whichever instructions appear in its context window.

The existing explainable AI literature has focused predominantly on making the model's internal decision-making legible: attention visualizations, feature attributions, saliency maps, and other techniques designed to open the black box of model weights and reveal why a given input produced a given output (cf. Arrieta et al., 2020; Vilone & Longo, 2020). This work usually addresses model-weight bias and, to some extent, training-data bias, but it often ignores directive bias, because the system prompt is not part of the model's architecture and leaves no trace in the kinds of representations that XAI methods inspect. A perfectly interpretable model, one whose every internal computation could be laid bare, would still be opaque in the sense that matters here if the instructions governing its behavior were obscured from the user. Of course, one could object that, for example, newspapers do not disclose their style guides, so why should LLM makers? However, the users approach them as if they were consulting a knowledgeable interlocutor, and the system prompt exploits that while hiding the interlocutor's apparent epistemic commitments.

Prompt transparency is, in this sense, the missing complement to explainable AI: where XAI asks "why did the model produce this output given its weights?", prompt transparency asks "what was the model told to do before the user said anything?". A complete account of why an LLM behaves the way it does requires both answers.

We are back at square one with our personality question, and we must remember that what looks like a character met in conversation is a character composed elsewhere, by people with commercial and political reasons for composing it as they did, and enacted by a system that holds nothing of that character apart from the tokens currently in its context. To call this a personality is to grant it a standing it does not have – a self behind the performance that could

answer for the traits on display. Making the script visible will not turn the performance into a person, but it will let us all see the Wizard of Oz behind the curtain, and judge for ourselves how much of their trust a character someone else wrote has earned.

References

- Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, 58, 82–115.
<https://doi.org/10.1016/j.inffus.2019.12.012>
- Bai, Y. et al. (2022). Constitutional AI: Harmlessness from AI feedback. *arXiv*.
<https://arxiv.org/abs/2212.08073>
- Bender, E. M., Gebru, T., McMillan-Major, A., & Shmitchell, S. (2021). On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency* (pp. 610–623). ACM.
<https://doi.org/10.1145/3442188.3445922>
- Branch, H. J., Cefalu, J., McHugh, J., Hujer, L., Bahl, A., del Castillo Iglesias, D., Heichman, R., & Darwishi, R. (2022). Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. *arXiv*. <https://arxiv.org/abs/2209.02128>
- Cheng, M., Hawkins, R. D., & Jurafsky, D. (2026). Accommodation and epistemic vigilance: A pragmatic account of why LLMs fail to challenge harmful beliefs. *arXiv*.
<https://arxiv.org/abs/2601.04435>
- Christiano, P. F., Leike, J., Brown, T., Martic, M., Legg, S., & Amodei, D. (2017). Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems 30* (pp. 4299–4307).
https://papers.nips.cc/paper_files/paper/2017/hash/d5e2c0adad503c91f91df240d0cd4e49-Abstract.html
- Dennett, D. (1987). *The intentional stance*. MIT Press.
- Greshake, K., Abdelnabi, S., Mishra, S., Endres, C., Holz, T., & Fritz, M. (2023). Not what you've signed up for: Compromising real-world LLM-integrated applications with indirect prompt injection. In *AISeC '23: Proceedings of the 16th ACM Workshop on Artificial Intelligence and Security* (pp. 79–90). <https://doi.org/10.1145/3605764.3623985>

- Lee, D. D. (2026). *Leaked system prompts* [GitHub repository]. *GitHub*. Retrieved July 13, 2026, from <https://github.com/jujumilk3/leaked-system-prompts>
- McHugh, J., Šekrst, K., & Cefalu, J. (2025). Prompt injection 2.0: Hybrid AI threats. *arXiv*. <https://arxiv.org/abs/2507.13169>
- Neumann, A., Pi, Y., & Singh, J. (2026). Who controls the conversation? User perspectives on generative AI (LLM) system prompts. In *CHI '26: Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (Article 378). Association for Computing Machinery. <https://doi.org/10.1145/3772318.3791726>
- Ouyang, L. et al. (2022). Training language models to follow instructions with human feedback. In *NIPS'22: Proceedings of the 36th International Conference on Neural Information Processing Systems* (pp. 27730–27744). <https://dl.acm.org/doi/10.5555/3600270.3602281>
- Pillay, T. (2026, March 10). Why AI systems can't help developing personalities. *TIME*. <https://time.com/article/2026/03/10/ai-chatbots-claude-gemini-personality/>
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. In *NIPS '23: Proceedings of the 37th International Conference on Neural Information Processing Systems* (pp. 53728–53741). <https://dl.acm.org/doi/10.5555/3666122.3668460>
- Rager, C., Wendler, C., Gandikota, R., & Bau, D. (2025). Discovering forbidden topics in language models. *arXiv*. <https://arxiv.org/abs/2505.17441>
- Searle, J. R. (1980). Minds, brains, and programs. *Behavioral and Brain Sciences*, 3(3), 417–457.
- Shanahan, M., McDonell, K., & Reynolds, L. (2023). Role play with large language models. *Nature*, 623(7987), 493–498. <https://doi.org/10.1038/s41586-023-06647-8>
- Skansi, S., & Šekrst, K. (in press). *Introduction to deep learning: Neural networks, large language models and agentic AI* (2nd ed.). Springer. <https://link.springer.com/book/9783032254597>
- Šekrst, K. (2025). *The illusion engine: The quest for machine consciousness*. Springer.
- Šekrst, K., McHugh, J., & Cefalu, J. (2024). AI ethics by design: Implementing customizable guardrails for responsible AI development. *arXiv*. <https://arxiv.org/abs/2411.14442>
- Šekrst, K. (2026a). To Train a Mockingbird. In *Proceedings of the AISB Convention 2026*. <https://openreview.net/pdf?id=najokS3KBj>

- Šekrst, K. (2026b). *Representational quarantine* [Manuscript submitted for publication].
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems 30* (pp. 5998–6008).
https://papers.nips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
- Vilone, G., & Longo, L. (2020). Explainable Artificial Intelligence: a Systematic Review. *arXiv*.
<https://arxiv.org/abs/2006.00093>
- Wallace, E., Xiao, K., Leike, R., Weng, L., Heidecke, J., & Beutel, A. (2024). The instruction hierarchy: Training LLMs to prioritize privileged instructions. *arXiv*.
<https://arxiv.org/abs/2404.13208>
- Wei, J., Bosma, M., Zhao, V., Guu, K., Yu, A. W., Lester, B., Du, N., Dai, A. M., & Le, Q. V. (2022). Finetuned language models are zero-shot learners. In *Proceedings of the 10th International Conference on Learning Representations*.
<https://openreview.net/forum?id=gEZrGCozdqR>
- Weidinger, L. et al. (2022). Taxonomy of risks posed by language models. In *FAccT '22: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (pp. 214–229). ACM. <https://doi.org/10.1145/3531146.3533088>
- Yang, A. (2024, October 24). Lawsuit claims Character.AI is responsible for teen's suicide. *NBC News*. <https://www.nbcnews.com/tech/characterai-lawsuit-florida-teen-death-rcna176791>