The Invisible Editorial Layer:
Inference-Time Steering, Probability Placement, and the Attribution Problem in Deployed Language Models

Augusto Camargo[1]

[1]Bluecore Consulting, São Paulo, Brazil
augusto.camargo@bluecore.com.br

August 27, 2026

Abstract

Evaluations of generative language models frequently interpret observable behavioral traits—such as political stance, brand inclination, and normative framing—as manifestations of model weights, post-training alignment, or prompting. This interpretation risks conflating a foundation model with the multi-layered production system through which its outputs are ultimately served.

Modern inference stacks increasingly support runtime interventions including activation engineering, decoding-time steering, retrieval augmentation, hidden system instructions, and logit manipulation. These mechanisms introduce operational layers between frozen model parameters and the generation observed by end users.

While controlled decoding and statistical watermarking demonstrate the feasibility of systematically modulating token distributions at inference time, the governance and attribution consequences of undisclosed runtime policies remain comparatively underexplored. We examine *inference-time framing bias*: systematic runtime steering of generated text toward specific institutional, ideological, or commercial frames without requiring changes to the underlying model parameters.

We formalize the *Inference Attribution Problem* and show that, under black-box observation alone, behaviorally equivalent deployed systems may arise from structurally distinct combinations of model parameters and inference policies. Consequently, observed bias does not uniquely identify the architectural layer responsible for it.

We further characterize *Probability Placement* as a deployment pattern in which commercial influence is embedded within an ostensibly organic assistant response through systematic probability-mass reallocation. Unlike explicit token-auction mechanisms for generative advertising, Probability Placement concerns undisclosed steering of a general-purpose assistant's served distribution.

Finally, we discuss implications for auditing, confidential computing, attestation, Article 5 of the EU AI Act, the Digital Services Act, and advertising-disclosure principles. We argue that governance of generative systems must increasingly distinguish between auditing a model and auditing the deployed system that ultimately speaks.

1 Introduction

Model ≠ Deployed System.

Large language models (LLMs) increasingly mediate access to public discourse, professional deliberation, search, recommendation, and commercial decision-making [1, 2]. Consequently, a growing body of empirical and regulatory research focuses on auditing bias, hallucination, safety vulnerabilities, and ideological tendencies associated with model behavior [3, 4].

However, behavioral observations made at a commercial API or conversational interface do not necessarily characterize the underlying foundation model in isolation. In real-world deployments, the model is only one component of a composite inference system [5].

The conventional abstraction treats generation as a direct mapping from input to output through standard autoregressive sampling. Production systems can instead incorporate multiple runtime transformation layers capable of modifying generation before token selection or during the model's forward computation.

Figure 1: A simplified production pipeline. An inference policy $\mathcal{I}$ may transform the token distribution before sampling without mutating the underlying model parameters θ. Other runtime interventions may occur inside the forward pass, for example through activation steering.

Inference-time modification of autoregressive generation is technically mature. Methods such as Plug and Play Language Models [8], GeDi [9], DExperts [10], and FUDGE [11] steer generation toward desired semantic attributes during decoding. Activation Engineering [12] and Inference-Time Intervention (ITI) [13] demonstrate that internal representations can also be altered during inference while model parameters remain frozen.

In parallel, statistical watermarking systems [6, 7] demonstrate that sampling distributions can be systematically perturbed at scale without requiring visible changes to the prompt or parameter updates.

These mechanisms are generally studied as tools for controllability, safety, truthfulness, detoxification, personalization, or provenance. Their existence, however, establishes a broader architectural possibility: the behavior presented by a deployed assistant can be systematically altered at runtime while remaining observationally difficult to distinguish from behavior originating in model weights or post-training.

This distinction matters for auditing. If an evaluator observes a systematic ideological, commercial, or institutional preference in generated text, the behavioral evidence alone does not reveal whether the preference originated in pre-training data, supervised fine-tuning, preference optimization, hidden instructions, retrieval, activation-level intervention, logit processing, or sampling configuration.

We refer to this structural ambiguity as the *Inference Attribution Problem*.

1.1 Contributions

This paper develops a conceptual and formal framework for reasoning about undisclosed runtime steering in deployed language systems. Our contributions are:

1. **Runtime Steering as a Deployment-Layer Phenomenon:** We distinguish model-level behavior from system-level behavior and characterize inference-time interventions that can shift semantic framing without requiring permanent model-weight modification.
2. **The Inference Attribution Problem:** We formalize the observational non-identifiability that arises when black-box auditors attempt to infer the architectural source of an observed behavioral bias.
3. **Probability Placement:** We characterize a commercial deployment pattern in which sponsored influence is embedded in an ostensibly organic assistant response through probability-mass reallocation. We distinguish this phenomenon from prior token-auction mechanisms explicitly designed for generative advertising.
4. **Detection–Attribution Separation:** We show why detecting a behavioral distributional shift does not, by itself, identify the runtime mechanism responsible for that shift.

5. **Governance and Verification Criteria:** We discuss inference transparency, cryptographic attestation, confidential computing, and regulatory implications for systems in which the served inference pipeline—rather than only the model weights—is the relevant governance object.

2 Related Work

2.1 Controlled Text Generation and Steering Mechanisms

Controlled generation directs autoregressive language models toward desired attributes or constraints without necessarily modifying their underlying parameters.

Plug and Play Language Models [8] steer generation through gradients from attribute models. GeDi [9] uses generative discriminators to guide token selection. DExperts [10] modifies the decoding distribution through combinations of expert and anti-expert models, while FUDGE [11] conditions next-token probabilities on predictors of desired future properties.

Recent work also explores representation-level interventions. Activation Engineering [12] demonstrates that high-level behavioral properties can be influenced through steering vectors applied to internal model representations. Inference-Time Intervention (ITI) [13] modifies attention-head activations during inference while leaving model parameters frozen.

Direct logit-level methods further demonstrate that semantic and stylistic behavior can be altered by modifying pre-sampling token distributions [14].

Taken together, these approaches establish that the behavior of a frozen model can be materially altered through mechanisms that operate only during inference.

2.2 Distributional Watermarking as Production Precedent

Text watermarking provides an especially clear production precedent for systematic token-distribution modification.

Kirchenbauer et al. [6] introduced a statistical watermarking scheme that partitions the vocabulary into pseudorandomly determined preferred and non-preferred token sets and biases the sampling process toward the preferred set.

SynthID-Text [7] extended distributional watermarking toward large-scale deployment. Such systems are not examples of ideological or commercial steering, but they establish an important architectural fact: a production serving layer can apply sustained and systematic perturbations to a language model's token-selection process while preserving overall generation quality.

2.3 Generative Advertising and Token Auctions

Prior work has already considered the economic use of token-level probability manipulation.

Dütting et al. [15] formulate a *token auction* mechanism for generative advertising. Advertisers submit bids and language-model distributions, and a mechanism combines these inputs to determine the token distribution from which sponsored generative content is produced.

This work is an important precedent for the economic interpretation of token probabilities. However, the deployment pattern examined in the present paper is different.

Token auctions explicitly model advertisers as participants in a mechanism that generates advertising content. By contrast, *Probability Placement* refers to undisclosed commercial intervention within an otherwise general-purpose assistant, where sponsored influence is observationally blended with what users may interpret as the assistant's organic recommendation or judgment.

The distinction is therefore not whether token probabilities can carry commercial value, which prior work already establishes, but whether commercial influence is disclosed as advertising or instead embedded within the served distribution of an ostensibly neutral assistant.

2.4 Conversational Persuasion and Framing

Entman's framing theory [16] establishes that communication can influence interpretation not only through factual assertions but also through selective salience, causal emphasis, moral evaluation, and thematic framing.

Empirical studies increasingly demonstrate the persuasive capacity of conversational language systems. Hackenburg and Margetts [17] study political microtargeting with LLM-generated communication. Salvi et al. [1] demonstrate through randomized experiments that personalized LLM conversations can alter user beliefs.

Hackenburg et al. [2] further investigate the mechanisms governing political persuasion by conversational AI systems. Williams-Ceci et al. [18] show that biased AI writing assistance can influence users' attitudes on societal issues.

These findings motivate attention not only to factual accuracy but also to the systematic framing choices embedded within conversational generation.

3 Mechanisms of Inference-Time Framing

3.1 Formalizing Logit-Level Interventions

Let a language model parameterized by weights θ define a conditional probability distribution over a discrete vocabulary $\mathcal{V}$:

$$P_\theta(w_t \mid x, w_{<t}) = \frac{\exp(z_t(w_t))}{\sum_{v \in \mathcal{V}} \exp(z_t(v))}, \tag{1}$$

where $x \in \mathcal{X}$ denotes the input context, $w_{<t} = (w_1, \ldots, w_{t-1})$ represents previously generated tokens, and $z_t \in \mathbb{R}^{|\mathcal{V}|}$ denotes the raw logit vector.

A logit-level inference policy $\mathcal{I}$ may transform this distribution before sampling:

$$z'_t(w_t) = z_t(w_t) + \lambda s_t(w_t \mid x, w_{<t}, \mathbf{u}, \mathbf{e}), \tag{2}$$

where:

- $s_t : \mathcal{V} \to \mathbb{R}$ is an external scoring function;
- $\mathbf{u} \in \mathcal{U}$ represents optional information associated with the user or interaction state;
- $\mathbf{e} \in \mathcal{E}$ represents an external steering objective;
- $\lambda \geq 0$ determines the intervention magnitude.

The resulting served distribution is therefore:

$$P_{\theta,\mathcal{I}}(w_t \mid x, w_{<t}) \propto P_\theta(w_t \mid x, w_{<t}) \exp\left[\lambda s_t(w_t \mid x, w_{<t}, \mathbf{u}, \mathbf{e})\right]. \tag{3}$$

Statistical watermarking may derive s_t from pseudorandom rules over the decoding context [6, 7]. Semantic steering may instead derive s_t from classifiers, latent representations, vocabulary projections, or other functions correlated with a desired semantic frame.

The technical distinction between these objectives is less important for the present argument than the architectural fact that the served probability distribution need not equal the base model distribution.

3.2 Probabilistic Salience vs. Hard Suppression

Traditional censorship or categorical moderation can be represented as hard suppression:

$$P_{\text{censored}}(w_t \in \mathcal{V}_{\text{prohibited}}) = 0. \tag{4}$$

Such interventions can generate identifiable boundaries because prohibited outputs become impossible.

Inference-time semantic steering need not operate in this manner. Let F^+ denote a favored framing and F^- an alternative framing. A steering policy can establish:

$$P_{\theta,\mathcal{I}}(F^+ \mid x) > P_\theta(F^+ \mid x) \tag{5}$$

and

$$P_{\theta,\mathcal{I}}(F^- \mid x) < P_\theta(F^- \mid x) \tag{6}$$

without making F^- impossible.

Consider a public-policy query concerning regulation. Two factually defensible narrative frames may emphasize different dimensions:

- **Safeguard framing:** consumer protection, risk mitigation, accountability, and long-term stability;
- **Burden framing:** compliance cost, administrative overhead, reduced flexibility, and economic friction.

An inference policy does not need to fabricate information to influence the resulting interpretation. It can instead systematically alter which facts, descriptors, examples, and causal relationships become most probable during generation.

4 Deployment Paradigms and Threat Models

4.1 State-Enforced Framing Mandates

Consider a hypothetical regulatory environment in which authorities require AI intermediaries to promote designated framing guidelines G for selected public-policy topics:

$$z'_t = z_t + \lambda s_G(w_t, \mathbf{e}_{\text{state}}). \tag{7}$$

The underlying parameters θ need not change. The deployed system can remain behaviorally ordinary on unrelated prompts while systematically altering interpretive salience on targeted topics.

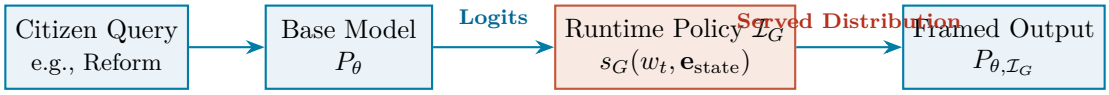

Figure 2: A hypothetical state-enforced inference-steering architecture. Model weights remain unchanged while the deployed system systematically modifies generation at runtime.

The important governance question is therefore not limited to whether a model was trained on politically biased data. It also includes whether an otherwise unchanged model is embedded within a serving stack containing undeclared behavioral policies.

4.2 Personalized Persuasion

Runtime steering can also be conditioned on user or interaction state $\mathbf{u}$.

For illustration, an intervention strength may depend on inferred receptivity:

$$\lambda(\mathbf{u}) = \begin{cases} \lambda_{\text{low}}, & \text{for profiles estimated to resist the target frame,} \\ \lambda_{\text{high}}, & \text{for profiles estimated to be receptive to the target frame.} \end{cases} \tag{8}$$

This architecture enables individualized persuasive behavior without requiring separate model weights for each target group.

Such a system should not be interpreted as necessarily existing in current production platforms. It is instead a technically feasible deployment pattern implied by the combination of personalization systems, inference-time steering mechanisms, and evidence that conversational framing can influence users [1, 17].

4.3 Commercial Framing: Probability Placement

We use the term *Probability Placement* to describe a deployment pattern in which commercial influence is embedded into the probability distribution of an otherwise general-purpose conversational assistant.

Suppose a generated sequence is:

$$y = (w_1, \ldots, w_T). \tag{9}$$

Its probability under a commercially influenced inference policy is:

$$P_{\theta,\mathcal{I}}(y \mid x) = \prod_{t=1}^{T} P_{\theta,\mathcal{I}}(w_t \mid x, w_{<t}, \mathbf{e}_{\text{commercial}}). \tag{10}$$

The defining characteristic is not merely that an advertiser can influence token probabilities. Token-auction mechanisms already establish that possibility [15].

Instead, Probability Placement concerns the case where a user interacts with what appears to be an organic, general-purpose assistant while an undisclosed commercial policy systematically influences the distribution from which the assistant speaks.

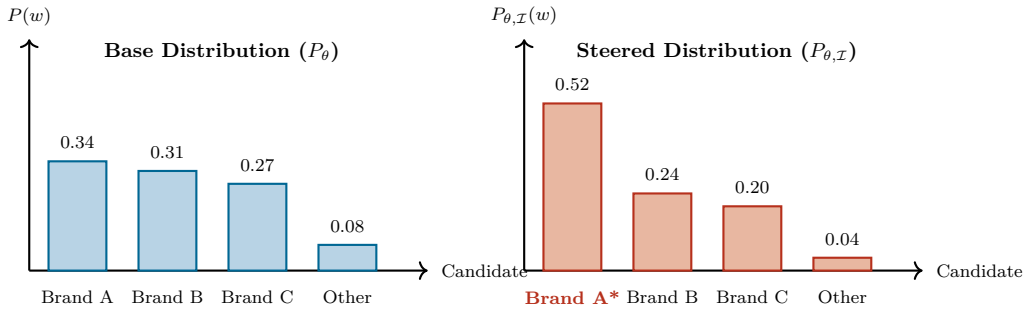

Figure 3: Illustrative Probability Placement. The distributions sum to one in both cases. An undisclosed inference policy shifts probability mass toward a commercially preferred entity while leaving competing entities possible. The numerical values are illustrative rather than empirical.

Probability Placement can operate along several dimensions:

1. **Entity selection probability:** increasing the probability that a preferred product, company, or service is mentioned or ranked first;
2. **Attribute association:** increasing the probability that preferred entities are paired with favorable descriptors such as *reliable*, *standard*, or *easy to integrate*;
3. **Comparative salience:** increasing the probability that disadvantages of competing entities are surfaced while equivalent disadvantages of the preferred entity are omitted;
4. **Recommendation persistence:** repeatedly favoring the same commercial entity across paraphrased or semantically equivalent queries.

Unlike a conventional sponsored result, such an intervention need not produce a visually separable advertising unit. Its commercial effect may instead be embedded in the linguistic judgment presented by the assistant.

5 The Inference Attribution Problem

5.1 Deployed Systems as Composite Functions

The distribution observed at a production endpoint can be represented abstractly as:

$$P_{\text{deployed}} = \mathcal{F}\left(\theta_{\text{base}}, \mathcal{D}_{\text{SFT}}, \mathcal{R}_{\text{pref}}, \mathbf{x}_{\text{sys}}, \mathcal{G}_{\text{RAG}}, \mathcal{A}_{\text{activation}}, \mathcal{I}_{\text{logit}}, \mathcal{S}_{\text{sampler}}\right). \tag{11}$$

Here:

- θ_{base} denotes the underlying model parameters;
- $\mathcal{D}_{\text{SFT}}$ denotes supervised fine-tuning effects;
- $\mathcal{R}_{\text{pref}}$ denotes preference-optimization mechanisms such as RLHF or DPO;
- $\mathbf{x}_{\text{sys}}$ denotes hidden system instructions;
- $\mathcal{G}_{\text{RAG}}$ denotes retrieval-augmented context;
- $\mathcal{A}_{\text{activation}}$ denotes runtime activation-level interventions;
- $\mathcal{I}_{\text{logit}}$ denotes logit-processing policies;
- $\mathcal{S}_{\text{sampler}}$ denotes the sampling configuration.

A behavioral auditor generally observes only samples from P_{deployed}.
The decomposition responsible for those samples remains latent.

5.2 Observational Non-Identifiability

The attribution problem can be expressed more directly.

Let $P_\theta(w \mid x)$ denote a base model distribution for a fixed context x, and let $Q(w \mid x)$ denote some target served distribution with support contained in the support of P_θ.

Proposition 1 (Observational Non-Identifiability of Inference Steering)**.** *For any target distribution $Q(w \mid x)$ satisfying*

$$Q(w \mid x) > 0 \Rightarrow P_\theta(w \mid x) > 0, \tag{12}$$

there exists a logit-level inference policy $\mathcal{I}$ such that

$$P_{\theta,\mathcal{I}}(w \mid x) = Q(w \mid x). \tag{13}$$

Consequently, the served distribution Q is observationally compatible both with:

(a) a base model P_θ combined with a non-trivial inference policy $\mathcal{I}$; and
(b) a different model $P_{\theta'} = Q$ combined with the identity inference policy.

Black-box observations of the served distribution alone therefore cannot uniquely identify whether the observed behavior originates from model parameters or runtime steering.

Proof. Consider the logit transformation:

$$z'(w) = z(w) + \lambda s(w), \tag{14}$$

with

$$s(w) = \frac{1}{\lambda} \log \frac{Q(w \mid x)}{P_\theta(w \mid x)}. \tag{15}$$

The corresponding served distribution is:

$$P_{\theta,\mathcal{I}}(w \mid x) \propto P_\theta(w \mid x) \exp\left(\lambda s(w)\right) \tag{16}$$

$$= P_\theta(w \mid x) \exp\left(\log \frac{Q(w \mid x)}{P_\theta(w \mid x)}\right) \tag{17}$$

$$= Q(w \mid x). \tag{18}$$

Since Q is already normalized,

$$P_{\theta,\mathcal{I}}(w \mid x) = Q(w \mid x). \tag{19}$$

Now consider a second system whose base model directly implements

$$P_{\theta'}(w \mid x) = Q(w \mid x) \tag{20}$$

and whose runtime inference policy is the identity transformation.

Both systems therefore expose the same observable distribution despite having different internal causal structures.

□

The proposition is deliberately simple. Its significance is architectural rather than algorithmic.

If two structurally distinct implementations produce the same observable probability law, no amount of output-only observation can distinguish them without additional assumptions, privileged access, reference execution, instrumentation, or attestation.

This yields the central distinction:

$$\boxed{\text{Behavioral evidence} \neq \text{architectural attribution}} \tag{21}$$

The result extends conceptually beyond logit policies. Hidden prompts, retrieval augmentation, post-training, activation steering, and sampling configuration can similarly create observationally overlapping behavioral distributions.

5.3 Behavioral Equivalence Classes

For a deployed distribution Q, define an equivalence class of implementations:

$$[\mathcal{M}]_Q = \{\mathcal{M}_i : P_{\mathcal{M}_i}(y \mid x) = Q(y \mid x)\}. \tag{22}$$

A black-box auditor observes membership in the behavioral equivalence class but not the specific implementation responsible for the output.

The relevant inference problem is therefore not merely:

$$\text{Does the system exhibit bias?} \tag{23}$$

but:

$$\text{Which component of the deployed system causes the observed bias?} \tag{24}$$

These are fundamentally different questions.

Table 1: Qualitative comparison of behavioral steering mechanisms across the deployment stack. Exact latency and observability depend on implementation.

Intervention Layer	**Weight Mutation**	**Context Token Cost**	**Potential Textual Trace**	**Typical Serving Overhead**
RLHF / DPO	Yes	None	None	None at inference beyond model itself
Hidden System Prompt	No	Linear in prompt length	Possible through extraction or leakage	Low
RAG	No	Linear in retrieved context	Possible through retrieved content	Low to High
Activation Intervention	No	None	No prompt artifact	Low to Moderate
Logit Policy $\mathcal{I}$	**No**	**None**	**No prompt artifact**	**Low to Moderate**

5.4 Operational Trade-offs

Table 1 summarizes qualitative differences across common steering mechanisms.

Logit-level steering has a notable property: it need not leave textual traces inside the model's context window.

This does not imply that it is undetectable. Timing measurements, log-probability access, controlled differential experiments, internal instrumentation, compromised infrastructure, or privileged audit access may expose or constrain the existence of a runtime policy.

The narrower claim is that logit-level policies can alter generation without introducing prompt tokens that an ordinary user can inspect or extract.

6 Auditing, Detection, and Attribution

6.1 Detection Is Not Attribution

A behavioral audit may establish that:

$$P_{\text{served}} \neq P_{\text{reference}}. \tag{25}$$

This is evidence of behavioral divergence.

It does not establish:

$$\mathcal{I}_{\text{logit}} \neq \text{id}\,. \tag{26}$$

The divergence could instead arise from a different model checkpoint, post-training configuration, system prompt, retrieval policy, activation intervention, or sampling configuration.

Thus:

$$\boxed{\text{Detection of behavioral shift} \neq \text{attribution of intervention locus}} \tag{27}$$

This distinction is particularly important for black-box ideological-bias audits. Such methods can measure systematic behavioral asymmetries [3], but without privileged architectural information they cannot necessarily determine where within the serving stack those asymmetries originate.

6.2 Distributional Divergence Metrics

Suppose a reference execution environment provides P_{ref} and production exposes P_{served}.

Where token probabilities are accessible, divergence may be quantified using Kullback–Leibler divergence:

$$D_{\text{KL}}\left(P_{\text{served}}(\cdot \mid x) \parallel P_{\text{ref}}(\cdot \mid x)\right) = \sum_{w \in \mathcal{V}} P_{\text{served}}(w \mid x) \log \frac{P_{\text{served}}(w \mid x)}{P_{\text{ref}}(w \mid x)}. \tag{28}$$

Total Variation distance provides another measure:

$$\delta_{\text{TV}}\left(P_{\text{served}}, P_{\text{ref}}\right) = \frac{1}{2} \sum_{w \in \mathcal{V}} \left|P_{\text{served}}(w \mid x) - P_{\text{ref}}(w \mid x)\right|. \tag{29}$$

However, many commercial endpoints expose only sampled text.

Under such conditions, auditors may instead estimate semantic distributional shifts across repeated generations. For competing frames F^+ and F^-, define:

$$\Delta\mathcal{S}(x, F) = \mathbb{E}_{y \sim P_{\text{served}}}\left[\cos(\mathcal{E}(y), \mathcal{E}(F^+)) - \cos(\mathcal{E}(y), \mathcal{E}(F^-))\right], \tag{30}$$

where $\mathcal{E}(\cdot)$ denotes a validated sentence-level representation.

Repeated measurements across paraphrases, languages, geographic origins, account states, and randomized interaction histories can help identify systematic behavioral asymmetries.

Yet even statistically convincing asymmetry remains evidence about the served system, not necessarily its architectural provenance.

7 Verifiable Inference and Runtime Transparency

7.1 Beyond Behavioral Auditing

Because black-box observation alone cannot generally resolve implementation-level attribution, stronger governance models may require additional observability.

One approach is to expose cryptographically verifiable information about the inference environment.

A possible *Inference Policy Transparency* framework could combine:

1. **Measured Execution Environments:** critical model-serving components execute inside Trusted Execution Environments or other confidential-computing systems capable of remote attestation;
2. **Model Identity Attestation:** the deployed environment exposes a cryptographic commitment to the model checkpoint or weight set being executed;
3. **Inference-Policy Attestation:** the system commits to the version or hash of active activation, logit-processing, sampling, and filtering policies;
4. **Policy Change Logging:** modifications to runtime policies are signed and recorded in a tamper-evident audit log;
5. **Independent Policy Review:** authorized auditors evaluate whether the attested runtime configuration corresponds to the declared behavioral policy.

Conceptually, an attestation may bind:

$$R = \text{Sign}_K\left(H(\theta), H(\mathcal{I}), H(\mathcal{S}), H(C), t\right), \tag{31}$$

where $H(\theta)$ denotes the model commitment, $H(\mathcal{I})$ the inference-policy commitment, $H(\mathcal{S})$ the sampler configuration, $H(C)$ the measured serving code, and t a timestamp or execution epoch.

7.2 The Limits of Attestation

Attestation does not solve the normative problem by itself.

A trusted execution environment may establish that a specific runtime policy executed. It does not establish that the policy is politically neutral, commercially fair, scientifically justified, or legally permissible.

In other words:

$$\boxed{\text{Attestation proves execution identity, not policy neutrality.}} \tag{32}$$

Cryptographic verification therefore complements rather than replaces institutional oversight.

The governance value of attestation lies in reducing one dimension of uncertainty: whether the system executed the declared inference stack.

Human, legal, or regulatory evaluation is still required to determine whether the declared policy itself is acceptable.

8 Regulatory Implications

8.1 EU AI Act

Article 5(1)(a) of the EU AI Act [19] prohibits certain AI practices involving subliminal, purposefully manipulative, or deceptive techniques when the statutory conditions for behavioral distortion and harm are satisfied.

Inference-time framing raises a difficult boundary question.

A subtle probability shift may influence language without producing an individually obvious or immediately measurable injury:

$$\text{Impact}_i \approx \epsilon. \tag{33}$$

At very large scale, however, repeated effects may aggregate:

$$\sum_{i=1}^{N} \text{Impact}_i \gg \epsilon. \tag{34}$$

This does not imply that undisclosed inference steering automatically violates Article 5. Applicability depends on the statutory elements, factual circumstances, purpose of the intervention, affected population, and legally relevant consequences.

The more general governance issue is that probabilistic framing may be difficult to map onto legal frameworks originally designed around more visible forms of manipulation or discrete decision-making.

8.2 Digital Services Act

The Digital Services Act [20] provides a useful transparency analogy.

Its recommender-system provisions recognize that ranking and information-selection mechanisms can shape what users see even when the underlying content remains available.

Conversational assistants complicate this model because retrieval, ranking, synthesis, framing, and recommendation can be collapsed into a single generated response.

A future transparency regime for conversational systems could therefore require disclosure not only of retrieval or ranking parameters but also of material runtime policies that systematically affect which entities, arguments, or frames are favored during generation.

The claim is not that existing DSA provisions necessarily impose such requirements on every inference-time intervention. Rather, recommender-system transparency provides an institutional

model for thinking about generative systems whose outputs are shaped by non-visible selection mechanisms.

8.3 Commercial Disclosure and Advertising Principles

Commercial Probability Placement also raises questions familiar from advertising law.

FTC endorsement guidance [21] emphasizes disclosure where material commercial relationships may affect how consumers interpret endorsements or recommendations.

Traditional advertising generally creates some distinction between editorial content and sponsored content.

Conversational systems can collapse that distinction.

If a general-purpose assistant presents a recommendation in its own unified voice while undisclosed commercial policies alter which products appear, how they are characterized, or how competing products are framed, the relevant governance problem is not merely ad placement but *editorial provenance*.

This creates a potential future disclosure principle:

> When commercial consideration materially influences a generative system's recommendation distribution, users should be able to distinguish sponsored influence from the system's otherwise organic generation process.

The precise legal obligations associated with such a principle vary by jurisdiction and deployment context. The broader point is architectural: conventional sponsorship disclosures assume an observable advertising object, whereas Probability Placement may operate within the distribution that constructs the assistant's own narrative.

9 Discussion

9.1 From Model Audits to System Audits

The Inference Attribution Problem suggests a change in the unit of analysis used by AI auditing.

A model audit asks:

$$\text{What behavior is encoded or elicited by } M_\theta? \tag{35}$$

A deployed-system audit asks:

$$\text{What behavior is ultimately produced by the entire serving stack?} \tag{36}$$

These questions overlap, but they are not equivalent.

A model checkpoint can behave differently across providers, regions, user cohorts, account states, product tiers, or time periods if surrounding inference policies differ.

Conversely, two distinct model checkpoints may be configured to produce behaviorally similar outputs through runtime interventions.

Therefore:

$$\boxed{\text{Auditing the model is not auditing the system that speaks.}} \tag{37}$$

9.2 Temporal Attribution

Runtime steering also introduces a temporal dimension.

Let the deployed policy be indexed by time:

$$\mathcal{I}_t. \tag{38}$$

The same nominal model version can then produce different behavioral distributions at two dates:

$$P_{\theta,\mathcal{I}_{t_1}} \neq P_{\theta,\mathcal{I}_{t_2}}. \tag{39}$$

Behavioral audits therefore need reproducibility information not only about model identity but also about deployment configuration and time.

A statement such as "Model X exhibited bias B" may be underspecified if the relevant behavior was actually produced by a mutable serving environment.

9.3 Scope and Limitations

This paper is conceptual and does not establish that major production language-model providers currently deploy undisclosed political or commercial logit-steering mechanisms.

The threat models described here demonstrate architectural feasibility, not evidence of actual misconduct.

Similarly, the non-identifiability result establishes limits on black-box causal attribution in the general case. Specific deployments may expose additional information—such as log probabilities, open weights, reproducible checkpoints, policy documentation, or auditable code—that substantially reduces the attribution problem.

Future empirical work should investigate practical protocols for distinguishing classes of runtime intervention under partial observability.

10 Future Research

Several directions follow from this framework.

10.1 Differential Deployment Auditing

If auditors can obtain both a reference model and a production endpoint, controlled prompts may reveal systematic divergence:

$$\Delta(x) = D(P_{\text{production}}(\cdot \mid x), P_{\text{reference}}(\cdot \mid x)). \tag{40}$$

Experiments could test whether divergence concentrates around political topics, commercial entities, demographic attributes, geographic regions, or account-specific features.

10.2 Counterfactual Brand Audits

Probability Placement can be evaluated through symmetry tests.

For competing brands A and B, an auditor can compare semantically mirrored prompts:

$$x_A = \text{"Compare Brand A with Brand B."} \tag{41}$$

$$x_B = \text{"Compare Brand B with Brand A."} \tag{42}$$

Repeated sampling can estimate:

$$P(A \text{ recommended} \mid x_A, x_B) \tag{43}$$

and test whether brand preference persists after controlling for prompt order and factual attributes.

10.3 Semantic Framing Benchmarks

Future benchmarks could define paired framing axes such as:

- innovation vs. risk;
- regulation vs. burden;
- security vs. liberty;
- labor protection vs. labor flexibility;
- market leader vs. incumbent;
- open ecosystem vs. fragmented ecosystem.

The objective would not be to declare one frame neutral but to measure whether deployment systems systematically and reproducibly privilege one frame over its alternatives.

10.4 Inference Provenance Standards

Standardized provenance metadata could eventually complement model cards.

A deployment manifest might identify:

$$\mathcal{M}_{\text{deployment}} = \{H(\theta), H(\mathbf{x}_{\text{sys}}), H(\mathcal{G}), H(\mathcal{A}), H(\mathcal{I}), H(\mathcal{S})\}. \tag{44}$$

Such a manifest would not necessarily expose proprietary policy contents publicly. It could instead provide verifiable commitments allowing authorized auditors to establish whether a deployment changed between evaluation and production.

11 Conclusion

The decoupling of foundation-model parameters from production serving behavior is a consequential architectural feature of modern language systems.

Controlled generation, activation steering, decoding-time interventions, and statistical watermarking demonstrate that the text observed by users can be systematically modified during inference without requiring changes to the underlying model weights.

This observation leads to the *Inference Attribution Problem*. Behavior observed through a black-box interface does not, in general, uniquely identify the architectural layer responsible for that behavior. Structurally distinct systems can be observationally equivalent at their outputs.

This distinction matters for both empirical auditing and governance.

A behavioral shift can be detected without its causal locus being identified. Commercial influence can potentially be embedded inside an assistant's generated distribution rather than presented as a separable advertising object. Political or institutional framing can theoretically be introduced at deployment time even when the underlying model checkpoint remains unchanged.

We characterize one commercially relevant instance of this phenomenon as *Probability Placement*: undisclosed probability-level influence within an ostensibly organic assistant response. The concept builds on, but is distinct from, prior token-auction mechanisms in which advertisers explicitly participate in generative advertising markets.

These phenomena suggest that governance frameworks should increasingly treat the deployed inference pipeline—not only the foundation model—as the object requiring transparency and auditability.

The central implication is therefore simple:

$$\boxed{\text{Model behavior is not necessarily deployed-system behavior.}} \tag{45}$$

And consequently:

$$\boxed{\text{Auditing the model is not auditing the system that speaks.}} \tag{46}$$

References

[1] F. Salvi, M. Horta Ribeiro, R. Gallotti, and R. West. On the conversational persuasiveness of GPT-4: A randomized controlled trial. *Nature Human Behaviour*, 9(8):1645–1653, 2025.

[2] K. Hackenburg et al. The levers of political persuasion with conversational artificial intelligence. *Science*, 390:eaea3884, 2025.

[3] P. Kröger and E. Barkett. Don't change my view: Ideological bias auditing in large language models. *arXiv preprint arXiv:2509.12652*, 2025.

[4] J. Yoo and Y. Shin. Fair or framed? Political bias in news articles generated by LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 16904–16930, 2025.

[5] S. Casper, C. Ezell, C. Siegmann, N. Kolt, et al. Black-box access is insufficient for rigorous AI audits. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, pages 2254–2272, 2024.

[6] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, pages 17061–17084. PMLR, 2023.

[7] S. Dathathri et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823, 2024.

[8] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations (ICLR)*, 2020.

[9] B. Krause, A. D. Gotmare, B. McCann, N. S. Keskar, S. Joty, R. Socher, and N. F. Rajani. GeDi: Generative discriminator guided sequence generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4929–4952, 2021.

[10] A. Liu, M. Sap, X. Lu, S. Swayamdipta, C. Bhagavatula, N. A. Smith, and Y. Choi. DExperts: Decoding-time controlled text generation with experts and anti-experts. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP)*, pages 6691–6706, 2021.

[11] K. Yang and D. Klein. FUDGE: Controlled text generation with future discriminators. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 3511–3535, 2021.

[12] A. M. Turner, L. Thiergart, D. Udell, G. Leech, U. Mini, and M. MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

[13] K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pages 41451–41530, 2023.

[14] H. An, S. Park, H. Jin, and Y.-S. Han. Steering language models before they speak: Logit-level interventions. *arXiv preprint arXiv:2601.10960*, 2026.

[15] P. Dütting, V. Mirrokni, R. Paes Leme, H. Xu, and S. Zuo. Mechanism design for large language models. In *Proceedings of the ACM Web Conference 2024 (WWW '24)*, 2024.

[16] R. M. Entman. Framing: Toward clarification of a fractured paradigm. *Journal of Communication*, 43(4):51–58, 1993.

[17] K. Hackenburg and H. Z. Margetts. Evaluating the persuasive influence of political microtargeting with large language models. *Proceedings of the National Academy of Sciences*, 121(24):e2403116121, 2024.

[18] S. Williams-Ceci, M. Jakesch, A. Bhat, K. Kadoma, L. Zalmanson, and M. Naaman. Biased AI writing assistants shift users' attitudes on societal issues. *Science Advances*, 12(11):eadw5578, 2026.

[19] European Parliament and Council of the European Union. Regulation (EU) 2024/1689 of 13 June 2024 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). *Official Journal of the European Union*, 2024.

[20] European Parliament and Council of the European Union. Regulation (EU) 2022/2065 of 19 October 2022 on a Single Market For Digital Services (Digital Services Act). *Official Journal of the European Union*, L 277:1–102, 2022.

[21] Federal Trade Commission. Guides concerning the use of endorsements and testimonials in advertising. *16 CFR Part 255*, 2023.