

# Silent Updates: Measuring and Closing the Post-Deployment Disclosure Gap

Sophia Abraham<sup>1</sup>, Ben Bucknall<sup>2</sup>

<sup>1</sup>Pivotal Research

<sup>2</sup>University of Oxford  
sabraha2@nd.edu

## Abstract

Deployed foundation models are often not static systems, with providers able to modify system behavior through fine-tuning, classifier updates, system prompt revisions, retrieval changes, and routing changes. These updates can be made *silently* – that is, without public disclosure, a version increment, or re-evaluation. Such *silent updates* challenge a core assumption behind current AI governance frameworks that an externally verifiable chain of custody links the model referred to in evaluation results or a system card to the model served to users.

In this paper, we examine post-deployment disclosure practices across first-party API providers and inference hosts to establish the extent to which a chain of custody exists in practice. We find that providers commonly publish substantial safety documentation, including quantitative evaluations and version-specific reports, but no provider in our sample published information allowing an external party to verify that the artifact being served is the same one referred to in this documentation.

We introduce the Silent Updates Scorecard, a public instrument for measuring post-deployment disclosure practices across providers and hosts, and preliminary results for a sample of nine first-party API providers and seven third-party inference hosts. We also propose a Three-Part Behavioral Trigger System for determining when post-deployment modifications to a system motivate disclosure or re-evaluation obligations.

## Introduction

Foundation model deployments are not static after release and prior work has shown that deployed models continue to evolve through post-training adaptation, learning from deployment data, and ongoing operational updates (Ouyang et al. 2022; Xu et al. 2022; O’Brien, Ee, and Williams 2023; Pratt and Tanjaya 2025). In production systems, deployment-layer mechanisms such as routing policies, retrieval configuration, and system prompt revisions may further change externally observed behavior without changing the public model identifier. This motivates the need for mechanisms that link deployed behavior to the versions evaluated in safety assessments. When these changes occur without public disclosure, a version increment, or re-evaluation,

we call them *silent updates*. As a result, published evaluations may no longer describe the system being served, with users being unaware of this discrepancy.

This creates a problem for current AI governance frameworks. Safety evaluations, system cards, responsible scaling policies, and post-market monitoring obligations all assume that the evaluated model can be linked to the deployed one through some externally checkable chain of custody. Yet most frameworks do not clearly specify when deployment changes require disclosure or re-evaluation. Our results suggest that these links often cannot be independently verified using public documentation or API access alone.

The consequences extend beyond documentation itself. Published evaluations are increasingly used by regulators, deployers, and researchers as evidence about system behavior (Mitchell et al. 2019a; European Parliament and Council of the European Union 2024; Keller et al. 2026). But if the evaluated artifact cannot be reliably connected to the deployed system, it becomes difficult to independently verify what those evaluations apply to in practice.

We study what we term the *verification gap*: the difficulty of independently connecting published evaluations to the systems users actually interact with. First, we introduce the Silent Updates Scorecard, an empirical instrument for measuring post-deployment disclosure practices across nine first-party API providers and seven inference hosts. Most providers in our sample publish extensive safety and release documentation, including quantitative evaluations and version-specific reports. At the same time, we found very limited ways for an external party to verify how those evaluations relate to the systems being served in practice. We did not identify any provider that exposes a verifiable API-to-evaluation round-trip.

Second, we examine how provider Terms of Service interact with independent evaluation. Six of nine providers in our sample include clauses that either explicitly restrict benchmarking or prohibit certain forms of competitive use in ways that may discourage external evaluation. Together, these conditions make meaningful third-party evaluation more difficult in practice (Longpre et al. 2024).

Third, we propose a Three-Part Behavioral Trigger System for determining when deployment changes should trigger disclosure obligations. The framework combines capability triggers, behavioral drift triggers, and component trig-

gers. We also propose a regulatory safe harbor for public-interest benchmarking.

The remainder of the paper is organized as follows. Section 2 reviews related work on AI governance, transparency, and behavioral drift. Sections 3 and 4 define silent updates and describe our methodology. Sections 5 through 7 present the empirical results and chain-of-custody analysis. Sections 8 through 10 discuss governance implications, limitations, and conclusions.

## Background and Related Work

### Post-Deployment AI Governance

Recent AI governance frameworks increasingly recognize the need for post-deployment monitoring and disclosure obligations for general-purpose models. The EU AI Act establishes documentation, transparency, and post-market monitoring requirements for general-purpose and systemic-risk models (Act 2024). The GPAI Code of Practice further develops these obligations through measures covering model documentation, monitoring, and reporting updates (European Commission 2025). These frameworks frequently rely on terms such as “material” or “substantial” change without clearly defining when disclosure or re-evaluation is required.

### Transparency Measurement

The Foundation Model Transparency Index (FMTI) (Bommasani et al. 2025) is the closest companion effort to ours. The 2024 and 2025 editions evaluate major foundation model developers across a broad set of transparency indicators. The 2025 report identifies post-deployment disclosure as one of the weakest areas across providers. Our scorecard focuses specifically on post-deployment change disclosure. We also evaluate inference hosts in addition to first-party providers and examine how these disclosure practices affect outside oversight.

Structured documentation has become a central mechanism for improving transparency and accountability in AI systems. Early work introduced datasheets for datasets to standardize documentation of dataset provenance, composition, intended use, and limitations (Gebru et al. 2021). Subsequent work extended these ideas to model cards, system cards, and other structured disclosure formats for communicating model capabilities, limitations, and evaluation results (Mitchell et al. 2019b; Gilbert et al. 2023; Derczynski et al. 2023). More recently, STREAM proposes standardized reporting of frontier AI capability evaluations, while Audit Cards advocate structured reporting of evaluation context and methodology to improve the interpretability and trustworthiness of AI assessments (McCaslin et al. 2025; Staufer et al. 2025). Our chain-of-custody analysis extends this literature by asking whether these documentation artifacts can be reliably connected to the behavior of deployed systems over time.

### Behavioral Drift in Deployed Models

Prior work has documented that deployed machine learning systems and commercial APIs can change substantially over

time, often without explicit notification to users. Studies have observed behavioral drift in deployed language models, shifts in commercial machine learning APIs, and undocumented performance changes following model updates (Chen, Zaharia, and Zou 2024; Miresghallah et al. 2024; Gao, Liang, and Guestrin 2025; Chen et al. 2021; Eyuboglu et al. 2024). More recent work has proposed efficient black-box techniques for detecting silent model updates and API changes (Cai et al. 2025; Chauvin et al. 2026a,b). Our work is complementary: rather than developing new detection algorithms, we investigate whether existing disclosure practices allow external stakeholders to determine when changes have occurred and whether public documentation accurately reflects deployed behavior.

### Auditing and Accountability Methodology

Our scorecard draws on the growing literature on AI auditing and accountability, particularly work positioning third-party audits as a governance mechanism for frontier AI systems (Raji et al. 2020; Birhane et al. 2024; Mökander et al. 2023; Brundage et al. 2026). We adopt conventions from this literature around evidence preservation, public scoring instruments, and reproducibility. The work also connects to research on AI supply chain accountability and external scrutiny ecosystems, which emphasize the need for verifiable information flows across the machine learning lifecycle and between model developers, deployers, and downstream users (Cobbe, Veale, and Singh 2023; Widder and Nafus 2023; Anderljung et al. 2023; Ojewale et al. 2025; Spoczynski, Melara, and Szyller 2025). Silent updates complicate this chain of accountability because the artifact referenced in compliance documentation may not be the one actually served downstream, weakening the connection between published evaluations and deployed systems (Anderljung et al. 2023).

## Silent Updates: Definitions and Scope

### Defining Silent Updates

We define a *silent update* as a change to a deployed AI system that materially affects user-facing behavior without corresponding public disclosure. The key issue is whether outside parties can observe and track those changes over time. A model update accompanied by public documentation, versioning information, or re-evaluation would not qualify as silent. By contrast, an undisclosed change to a safety classifier or routing policy would qualify as a silent update even if the underlying model weights remained unchanged.

Table 1 summarizes the operational definitions used throughout the paper. A property counts as present only when the stated condition is observable through public documentation, Terms of Service, or standard API access.

### A Taxonomy of Post-Deployment Changes

Post-deployment changes operate at multiple layers of the deployed stack. We distinguish three categories. *Model-layer changes* include weight updates from post-training, distillation, or quantization. *System-layer changes* include

| Term                                          | Operational definition                                                                                                                                                                                                                             |
|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Snapshot binding                              | A safety evaluation document names an immutable, API-callable identifier that uniquely identifies the evaluated artifact.                                                                                                                          |
| API-to-evaluation round-trip                  | An external party can map a model’s API response back to the specific evaluation artifact describing the served snapshot using only public information.                                                                                            |
| Policy version-binding                        | A content-policy revision is linked to a specific model snapshot rather than applying to the service as a whole.                                                                                                                                   |
| Family vs. snapshot identifier                | A family identifier such as “GPT-5” denotes a product line. A snapshot identifier denotes one immutable served artifact. Only the latter satisfies snapshot-level criteria.                                                                        |
| Deterministic control vs. reproducible replay | A deterministic control such as temperature 0 or a seed parameter can reduce sampling variation within a fixed snapshot. It does not guarantee that the same snapshot remains deployed over time, which is what reproducible replay would require. |
| Silent update                                 | A behavioral or system change without disclosure linking the change to a specific named version and effective date.                                                                                                                                |

Table 1: Operational definitions of the core terms used in scoring and analysis. Each definition is evaluated only against publicly observable artifacts such as documentation, Terms of Service, and standard API responses.

modifications to system prompts, classifiers, retrieval components, tool access, routing policies, inference budgets, and tokenizers. *Surface-layer changes* include differences between chatbot and API behavior, as well as mid-session model substitution.

Current governance frameworks primarily focus on the model layer. Many deployment changes, however, occur at the system layer. Figure 1 situates these categories within the deployed stack.

## Scope of This Paper

Our empirical analysis is limited to what can be observed externally. We use public model listings exposed through provider APIs, provider documentation and archived versions of that documentation, and behavioral fingerprints obtained through standard API calls. We do not claim visibility into internal deployment infrastructure, model registries, or proprietary system prompts and classifiers.

This limitation is central to the governance problem we study. Downstream deployers, regulators, and independent evaluators can only act on information that is externally available. This paper focuses on the gap between deployed behavior and what outside parties can independently verify.

| Codes | Section                  | What it measures                                           |
|-------|--------------------------|------------------------------------------------------------|
| V1–4  | Versioning               | Whether a provider exposes stable version identifiers      |
| C1–5  | Changelog                | Whether changes are publicly documented                    |
| X1–2  | Cross-surface            | Whether chatbot and API model identity align               |
| D1–6  | Deprecation              | Whether retirement is governed by policy                   |
| M1–7  | Safety eval traceability | Whether published evaluations link to deployed versions    |
| R1–3  | Reproducibility          | Whether deterministic controls support reproducible replay |
| T1–2  | Monitoring support       | Whether third-party drift detection is feasible            |

Table 2: Scorecard sections and dimensions measured.

## Methodology

### Provider and Inference Host Selection

We score nine first-party API providers and seven inference hosts. The first-party providers are OpenAI, Anthropic, Google, Meta, xAI, Mistral, DeepSeek, Cohere, and AI21. The inference hosts are AWS Bedrock, Azure OpenAI, Google Vertex AI, Together AI, Groq, Fireworks AI, and Replicate.

We included first-party providers if they operate a public API serving at least one foundation model, publish at least one form of safety documentation, and plausibly fall under EU AI Act GPAI obligations based on public information about model scale. We included inference hosts if they serve at least three foundation models from distinct first-party providers and operate a public commercial API. We exclude open-weight releases without an associated hosted API because silent updates, as defined here, are a phenomenon of hosted deployment.

### The Scorecard Instrument

The scorecard contains twenty-nine questions organized into seven sections, shown in Table 2. These cover versioning, changelog coverage, cross-surface transparency, deprecation policy, safety evaluation traceability, reproducibility infrastructure, and monitoring support. Four additional questions apply only to inference hosts. The full question text and scoring guide are available in the accompanying online repository.

The question set draws from disclosure mechanisms described in the GPAI Code of Practice, post-deployment indicators used in the Foundation Model Transparency Index (Bommasani et al. 2025), and Article 25 reclassification criteria. The rubric was finalized before scoring and is preserved in its pre-scoring form in the online repository.

Each question has a literal answer and a normalized score. Binary questions score 0 or 1. Numeric questions are banded from 0 to 3 using thresholds defined in the scoring guide. List questions score by the count of items returned. Cross-surface criteria (X1–X2) were evaluated only for providers operating both a public API and a consumer-facing chatbot

## surface-layer changes

## system-layer changes

## model-layer changes

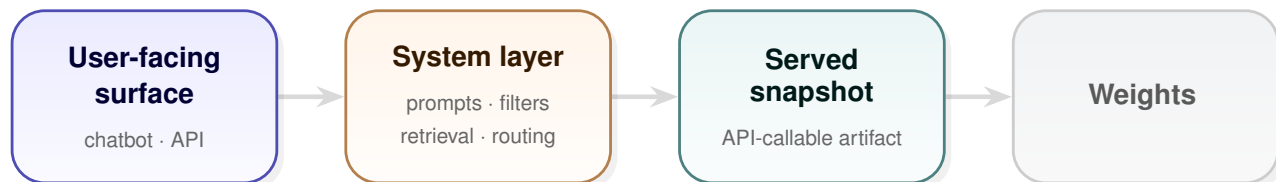


Figure 1: The deployed AI system stack. Many post-deployment changes occur at the surface and system layers, while current governance obligations primarily attach to the model or weight layer.

interface. Providers without a comparable chatbot surface were marked as not applicable (N/A) and excluded from the applicable maximum score rather than receiving zero points. The maximum first-party score is 37 points for providers where all criteria apply.

Scoring was performed by a single rater using public documentation and standard API access. Each score in the final scorecard is linked to an evidence row containing the source URL, access date, and a verbatim passage from the relevant documentation or API response.<sup>1</sup>

The evidence base contains more than 270 rows across system cards, model cards, technical reports, changelogs, deprecation tables, Terms of Service, and API reference pages. When current documentation was insufficient or had changed, we reconstructed historical state using Internet Archive snapshots at four points before the scoring date: twelve months, six months, three months, and one month.

After the initial scoring pass, an independent co-author performed an evidence audit of the completed scorecard. Rather than rescore providers, the audit evaluated whether the cited evidence justified each assigned score, whether similar evidence was scored consistently across providers, and whether any factual inconsistencies or ambiguous applications of the rubric remained. Any flagged cases were revisited against the original documentation before the scorecard was finalized.

For every question scored zero across the sample, we treated the zero as a claim to be falsified rather than asserted. We searched provider documentation, archived snapshots, API and SDK references, and system-card appendices for counterexamples satisfying the operational definitions in Table 1. Near-matches that failed one or more criteria were recorded but not scored as satisfying the criterion. The closest near-matches are discussed in the chain-of-custody analysis.

### Chain-of-Custody Protocol

For each first-party provider, we identified the most recent public safety evaluation document. This could be a system card, model card, technical report, or responsible scaling

commitment. We then attempted to link that evaluation to the model version currently served.

We coded three dimensions: whether the evaluation named a unique identifier for the evaluated artifact, whether that identifier remained API-accessible, and whether the document provided enough methodological detail for external re-evaluation.

Each evaluation falls into one of four outcomes: chain established, version unspecified, version inaccessible, or methodology underspecified. A chain is established only when all three conditions are met.

### What We Deliberately Do Not Measure

We restrict our empirical claims to externally observable disclosure. We do not measure weight changes directly because weight access is not available for closed-API providers. We do not measure system prompt content because production system prompts are not externally inspectable. We do not measure routing or load balancing because route assignment is not disclosed in API responses for any provider in our sample.

## Results

We present aggregate Scorecard results across the nine first-party providers and seven inference hosts in our sample. The mean first-party score is 16.4 points, corresponding to 44.4% of the maximum applicable score under the rubric. Scores range from 8 (Meta) to 23 (OpenAI). Table 3 presents the full first-party ranking, and Table 4 summarizes inference host transparency. Table 5 summarizes the highest-level disclosure and verification properties across providers, and Table 6 shows the specific properties most relevant to our argument.

Inference hosts showed relatively limited variation under the hosting-specific rubric. Six of seven evaluated hosts satisfied three of four transparency criteria, with the primary limitations arising from incomplete disclosure of deployment-level transformations such as quantization and upstream model changes. Fireworks AI was the only host scoring below this threshold, reflecting gaps in lifecycle and deployment disclosure.

The most consistent findings do not appear in the total scores but in the repeated absence of specific binding and

<sup>1</sup>The complete scorecard, evidence tables, and dataset are available at <https://github.com/sabrah2/silent-updates-scorecard>.

| Rank | Provider  | Score | %    |
|------|-----------|-------|------|
| 1    | Cohere    | 22    | 62.9 |
| 2    | OpenAI    | 23    | 62.2 |
| 3    | Anthropic | 20    | 54.1 |
| 4    | Google    | 18    | 48.6 |
| 5    | xAI       | 15    | 40.5 |
| 5    | Mistral   | 15    | 40.5 |
| 5    | DeepSeek  | 15    | 40.5 |
| 8    | AI21      | 12    | 34.3 |
| 9    | Meta      | 8     | 21.6 |

Table 3: First-party Scorecard totals ranked by overall score. Maximum possible score is 37; providers without applicable cross-surface criteria were normalized against their applicable maximum.

| Inference Host   | Score | %    |
|------------------|-------|------|
| AWS Bedrock      | 3/4   | 75.0 |
| Azure OpenAI     | 3/4   | 75.0 |
| Google Vertex AI | 3/4   | 75.0 |
| Together AI      | 3/4   | 75.0 |
| Groq             | 3/4   | 75.0 |
| Replicate        | 3/4   | 75.0 |
| Fireworks AI     | 2/4   | 50.0 |

Table 4: Inference host transparency scores. Hosts were evaluated using the hosting-specific criteria provided in the accompanying online repository.

verification properties across providers. We organize these patterns into four recurring failure modes.

### Failure Mode 1: Pinned Identifiers Without Behavioral Guarantees

All nine providers expose dated or otherwise versioned identifiers through their APIs (V1=9/9). Five of nine document a snapshot-pinnable identifier in product documentation (V2=5/9). Only two of nine publish a centralized mapping from alias identifiers such as `gpt-5` or `claude-sonnet-4-6` to the dated snapshots those aliases currently resolve to (V3=2/9).

Several providers explicitly document alias behavior in which stable API-facing identifiers resolve to changing underlying model versions over time. DeepSeek’s changelog records that `deepseek-chat` and `deepseek-coder` were retained for backward compatibility while being upgraded to DeepSeek V2.5 (DeepSeek AI 2024). Anthropic documents that aliases point to a recommended version and “update over time” (Anthropic 2026a). Google Vertex AI similarly states that “the auto-updated alias of a Gemini model always points to the latest stable model” (Google Cloud 2026). These cases motivate our distinction between model-family aliases and immutable snapshot identifiers.

The DeepSeek case illustrates the problem clearly. The identifier `deepseek-chat` has remained continuously available since May 2024, while the underlying served model advanced through multiple versions over the same

| Observed property                                                           | Providers  |
|-----------------------------------------------------------------------------|------------|
| Expose versioned or dated identifiers                                       | 9/9        |
| Publish quantitative safety metrics                                         | 7/9        |
| Publish per-version safety comparisons                                      | 6/9        |
| Document snapshot-pinnable identifiers                                      | 5/9        |
| Publish alias-to-snapshot mapping                                           | 2/9        |
| Name the evaluated artifact with a pinned, API-callable snapshot identifier | <b>1/9</b> |
| Enable an external API-to-evaluation round-trip                             | <b>0/9</b> |
| Version-bind content policy                                                 | <b>0/9</b> |
| Permit third-party benchmarking under ToS                                   | 3/9        |

Table 5: Results overview. Safety evaluation artifacts are common, but the primitives needed to bind those artifacts to deployed API behavior are absent or rare.

period. Figure 2 summarizes the progression recorded in DeepSeek’s public release notes.

### Failure Mode 2: Changelogs Document Launches More Reliably Than Behavior Changes

Eight of nine providers maintain public release notes or changelogs (C1=8/9). Most document new model launches and pricing changes. Far fewer document behavior changes to existing identifiers. Only two of nine publish version-level behavior change documentation (C2=2/9), and none consistently flag safety-relevant changes in those notes (C3=0/9). Two of nine provide a machine-readable changelog feed (C4=2/9).

The asymmetry is consistent across providers. Launches and pricing changes are documented at relatively high granularity, while behavior changes to existing deployments are documented much less consistently.

### Failure Mode 3: API and Chatbot Surfaces Diverge

Among providers operating both API and consumer chatbot surfaces, none disclose the dated snapshot identifier serving the chatbot frontend at any given time (X1=0/7). Only one provider, Mistral, propagates chatbot-side model updates into the developer-facing changelog among providers with applicable chatbot surfaces (X2=1/7).

We score X1 strictly. A family label such as “GPT-5” or “Claude Sonnet 4” does not satisfy the criterion because it does not identify the specific snapshot being served. Chatbot and API behavior may therefore differ in version, system prompt, or safety classifier configuration without external disclosure.

This creates a governance problem. A regulator observing chatbot behavior cannot reliably infer the configuration of the API service integrated downstream.

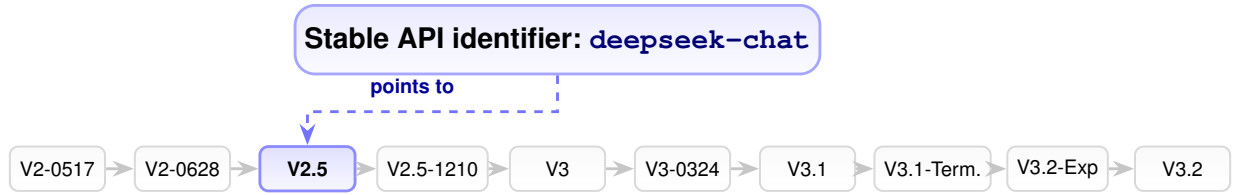


Figure 2: Alias drift on a stable API identifier. The identifier `deepseek-chat` remained stable while the underlying served artifact changed across ten documented releases over nineteen months.

|             | Disclosure |    |    | Binding / verification |    |    |    |    |
|-------------|------------|----|----|------------------------|----|----|----|----|
|             | V1         | M7 | V2 | V3                     | M1 | M2 | M6 | T2 |
| OpenAI      | ■          | ■  | ■  | □                      | □  | □  | □  | □  |
| Anthropic   | ■          | ■  | ■  | ■                      | ■  | □  | □  | □  |
| Google      | ■          | ■  | □  | □                      | □  | □  | □  | □  |
| Meta        | ■          | ■  | □  | □                      | □  | □  | □  | ■  |
| xAI         | ■          | ■  | ■  | □                      | □  | □  | □  | □  |
| Mistral     | ■          | □  | ■  | □                      | □  | □  | □  | □  |
| DeepSeek    | ■          | □  | □  | □                      | □  | □  | □  | ■  |
| Cohere      | ■          | ■  | ■  | ■                      | □  | □  | □  | □  |
| AI21        | ■          | ■  | □  | □                      | □  | □  | □  | ■  |
| <i>n</i> /9 | 9          | 7  | 5  | 2                      | 1  | 0  | 0  | 3  |

**Labels:** V1 = versioned identifiers; M7 = quantitative safety metrics; V2 = snapshot-pinnable identifier; V3 = alias-to-snapshot mapping; M1 = evaluation names an immutable API-callable identifier; M2 = externally verifiable API-to-evaluation round-trip; M6 = content policy bound to version; T2 = benchmarking permitted under Terms of Service. Anthropic’s M1 is partial (see Table 7).

Table 6: Disclosure versus verification across first-party providers. Filled cells (■) indicate the property is present in our scoring; □ indicates absence. Anthropic alone names a pinned evaluated snapshot (M1), but no provider supports the external round-trip (M2) or version-binds policy (M6).

#### Failure Mode 4: Safety Evaluations Are Not Bound to Served Versions

The strongest asymmetry appears in the M-section results. Providers frequently publish safety evaluations and quantitative metrics, but none expose an externally verifiable API-to-evaluation round-trip or version-bind content policy to a specific snapshot. Anthropic partially satisfies M1 by naming a pinned evaluated snapshot, but the external verification chain still breaks at the round-trip layer.

We refer to this gap as the chain-of-custody failure and examine it in the following section.

#### Chain-of-Custody Analysis

The M-section of the Scorecard asks whether an external party can connect three artifacts: the model identifier returned by the API, the snapshot evaluated in the safety report, and the content policy governing that snapshot. We call this connection the *chain of custody*. Across the nine first-party providers in our sample, we did not identify a complete chain of custody.

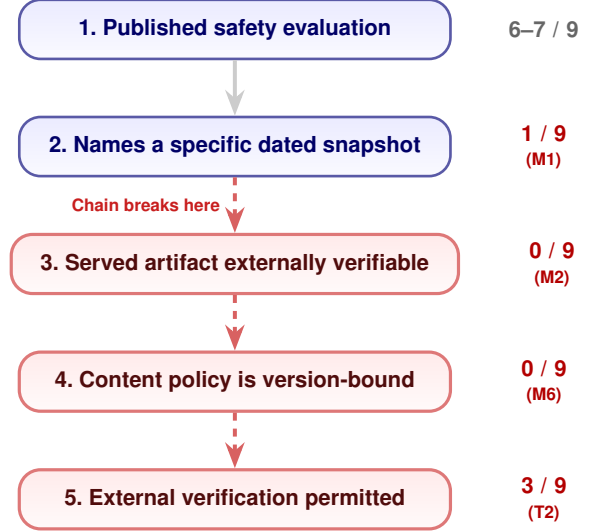


Figure 3: The chain of custody from a published safety evaluation to an externally verifiable claim about deployed behavior. Providers commonly publish evaluation artifacts, but we did not identify the binding primitives needed to connect those artifacts to deployed snapshots.

The pattern holds even at providers that publish extensive safety evaluations. Figure 3 summarizes the structure of the chain and the points where it breaks across the sample.

#### Naming Is Not Verification

Table 7 presents the per-provider M-section scores. Two binding primitives are absent across the entire sample (M2 and M6). A third, naming the evaluated snapshot (M1), is satisfied only partially by a single provider.

**M1 = 1/9.** Most providers refer to evaluated systems by family or variant name rather than by a pinned, API-callable snapshot identifier. OpenAI’s GPT-5 System Card discusses `gpt-5-thinking` and `gpt-5-main`, labels that correspond to evolving deployments over time (OpenAI 2025). Google’s Gemini 3 Pro Model Card similarly uses family-level identifiers (Google DeepMind 2025). Meta and xAI also rely on family designations.

Anthropic is the closest counterexample. Beginning with the Claude 4.6 generation, Anthropic documents immutable model identifiers rather than convenience aliases. These identifiers use a dateless format but correspond to fixed API-callable model snapshots (Anthropic 2026b). The Claude

|    | OAI | ANT | GOO | MET | xAI | MIS | DS | COH | AI21 | $n/9$ |
|----|-----|-----|-----|-----|-----|-----|----|-----|------|-------|
| M1 | 0   | 1   | 0   | 0   | 0   | 0   | 0  | 0   | 0    | 1     |
| M2 | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0    | 0     |
| M3 | 1   | 1   | 1   | 1   | 1   | 0   | 0  | 1   | 0    | 6     |
| M4 | 1   | 1   | 1   | 1   | 1   | 0   | 0  | 0   | 1    | 6     |
| M5 | 1   | 1   | 1   | 1   | 1   | 0   | 0  | 1   | 0    | 6     |
| M6 | 0   | 0   | 0   | 0   | 0   | 0   | 0  | 0   | 0    | 0     |
| M7 | 1   | 1   | 1   | 1   | 1   | 0   | 0  | 1   | 1    | 7     |

Table 7: Per-provider M-section scores. Each cell is 0 (absent) or 1 (present). Anthropic partially satisfies M1 because it names a pinned evaluated snapshot but does not expose a complete external verification chain.

Opus 4.7 System Card states that evaluations were run on the final released snapshot (Anthropic 2026c). We therefore score Anthropic as partially satisfying M1. However, some dangerous-capability metrics are reported as maxima across multiple internal snapshots, including safeguards-removed variants, so not every reported number is bound to the served artifact.

**M2 = 0/9.** Naming a snapshot is necessary but not sufficient. We did not identify a documented mechanism at any provider that allows an external party to verify that the artifact returned by an API call is identical to the artifact described in the evaluation document. The `API model` field is provider-asserted. Even when it returns a dated identifier, no public mechanism verifies that the served weights, system prompt, and classifier configuration match the evaluated system.

**M6 = 0/9.** Across the sample, content policy documents apply service-wide rather than to specific model versions. OpenAI’s Usage Policies, Anthropic’s Acceptable Use Policy, and Google’s Generative AI Prohibited Use Policy all govern the provider service as a whole. We did not identify a version-scoped policy document at any provider.

### A Worked Example: GPT-5

GPT-5 is among the most extensively documented systems in our sample. OpenAI published a forty-page GPT-5 System Card with per-category `not_unsafe` scores across multiple variants (OpenAI 2025). On the M-section, it satisfies M3, M4, M5, and M7.

The document refers to the evaluated systems as `gpt-5-thinking` and `gpt-5-main`. These are family-level identifiers. OpenAI’s API exposes dated snapshots such as `gpt-5-2025-08-07`, but the system card does not specify which snapshot corresponds to the evaluated deployment. An auditor querying the API therefore cannot determine which dated snapshot produced the reported safety metrics.

OpenAI’s Usage Policies also apply service-wide rather than to specific snapshots, so an external party cannot determine which policy revision governed the evaluated system or whether the same policy governs the currently served deployment. OpenAI’s Terms of Use additionally restrict competing-system benchmarking, limiting the external verification process that would otherwise close the loop.

The disclosures exist, but they are not externally bound to deployed artifacts.

### The Missing Primitive Is Binding

The natural hypothesis suggested by the M2 and M6 zero counts is that providers simply do not conduct serious safety evaluations. The remaining M-section data does not support that interpretation.

Seven of nine providers publish quantitative safety metrics (M7=7/9). Six publish per-version safety comparisons across releases (M4=6/9). Six publish comparisons across simultaneously released variants (M5=6/9). Six also document substantial re-evaluation on major releases (M3=6/9).

The OpenAI GPT-5 System Card includes category-level `not_unsafe` scores across multiple variants (OpenAI 2025). Google’s Gemini 3.1 Pro Model Card reports refusal-rate and safety deltas relative to Gemini 3 Pro (Google DeepMind 2026). Anthropic’s Opus 4.7 System Card spans more than two hundred pages and includes comparative evaluations against earlier Opus generations. Meta’s Llama 4 model card reports cyber, CBRNE, and child-safety evaluations relative to Llama 3.3 (Meta AI 2025). xAI’s Grok 4.1 Model Card publishes multilingual malicious-use evaluations and explicitly corrects an error in earlier reporting (xAI 2025). Cohere’s Command A Technical Report includes extensive distribution-based safety analysis (Cohere 2025).

Providers are increasingly publishing substantive safety evaluations. What remains absent is a public mechanism that binds those evaluations to deployed artifacts.

The providers themselves also document the drift that makes this gap consequential. Google Vertex AI states that “the auto-updated alias of a Gemini model always points to the latest stable model” (Google Cloud 2026). Anthropic documents that convenience aliases “update over time” (Anthropic 2026a). DeepSeek’s `deepseek-chat` identifier resolved to a succession of underlying systems over nineteen months (DeepSeek AI 2024). A persistent identifier is not a stable model.

### Cross-Surface Divergence

The problem becomes more pronounced across hosting surfaces. We documented four recurring divergence patterns where the same nominal model identifier corresponds to materially different served behavior.

First, lifecycle divergence. Anthropic retired Claude 3 Haiku from its first-party API on 2026-04-20, while AWS Bedrock listed the corresponding deployment until 2026-09-10, a 143-day difference.

Second, content-filter divergence. Azure OpenAI applies default-on content filtering not present in OpenAI’s first-party API (Microsoft Cloud 2025), producing distinct served behavior for the same underlying `gpt-5-*` deployment.

Third, quantization divergence. Together AI is the only hosting provider in our sample that publicly discloses quantization settings at the model level (H2=1/7). The same `openai/gpt-oss-120b` weights are served at MXFP4 precision on Together AI and at undisclosed precision on Groq.

Fourth, deprecation divergence. Cohere retired `command-r-03-2024` from its first-party API on 2025-09-15, while AWS Bedrock continued serving the same model until 2026-08-19 without a public reconciliation notice.

### Contractual Limits on External Verification

The chain-of-custody problem interacts with a second layer: contractual restrictions on independent verification.

Of the nine first-party providers in our sample, six restrict benchmarking through either explicit benchmarking clauses or broader competing-product restrictions. Cohere’s Terms of Use prohibit accessing the service “for any other benchmarking or competitive purposes” (Cohere 2026). OpenAI, Anthropic, Google, Mistral, and xAI instead rely on broader competing-product language that can plausibly extend to benchmarking activity.

This distinction matters, but both mechanisms constrain independent verification. We therefore score T2 as absent whenever either restriction is present.

The Cohere case is particularly notable. Cohere has the highest normalized percentage score in our Scorecard on transparency artifacts while simultaneously restricting external benchmarking in its Terms of Service. The possibility that competing-product clauses can be used against benchmarking efforts is not hypothetical. In July 2025, Anthropic revoked OpenAI’s API access under such a clause. Contemporary reporting described benchmarking activity as among the cited concerns (Nuñez 2025; Dave 2025). Together AI similarly prohibits “competitive analysis or benchmarking” in its Terms of Service (Together AI 2026).

Independent benchmarking is often proposed as a response to provider opacity (Bommasani et al. 2025). Yet six of nine first-party providers in our sample restrict benchmarking through either explicit benchmarking clauses or broader competing-product provisions. As a result, some of the mechanisms commonly proposed for independent verification may themselves be contractually constrained.

Providers increasingly publish detailed safety artifacts, but the technical and contractual conditions needed to independently verify those artifacts often remain limited.

### Behavioral Drift Evidence

The chain-of-custody failure described earlier would matter less if served behavior remained stable for the lifetime of a snapshot identifier. We collect evidence to the contrary from three observable signals: provider disclosures that acknowledge deployment-layer variation, documented alias-to-snapshot transitions, and the existence of a single hosting primitive that demonstrates the feasibility of immutable snapshot binding.

**Quantization disclosure as a feasibility proof.** Together AI is the only inference host in our sample that discloses quantization at the deployment level (H2=1/7). For models such as `openai/gpt-oss-120b`, Together specifies the served precision format directly, including MXFP4, BF16, FP8, and INT8. Groq, Fireworks AI, Replicate, AWS Bedrock, Azure OpenAI, and Google Vertex AI do not

disclose equivalent information. The important observation is not the absence at six hosts but the presence at one. Snapshot-bound disclosure of a numerical precision that can affect served behavior is operationally feasible. Its absence elsewhere therefore reflects a documentation choice rather than a technical limitation.

**Documented alias-update patterns.** Google Vertex AI states that “the auto-updated alias of a Gemini model always points to the latest stable model” (Google Cloud 2026). Groq documents a similar upgrade policy for Llama deployments, noting in release documentation that requests to legacy identifiers automatically migrate to newer variants. DeepSeek’s API release history likewise shows that the stable identifier `deepseek-chat` resolved to a succession of underlying versions over time (DeepSeek AI 2024). A persistent identifier is therefore not necessarily a persistent model. Where providers document these transitions, downstream users can at least detect that silent re-pointing may occur. Where they do not, the transition remains opaque.

**Content-hashed immutable versions.** Replicate exposes the strongest snapshot-binding primitive observed in our sample. Each deployment is identified by a content hash, and historical hash-identified versions remain queryable through the API. An evaluator who records the hash associated with a benchmarked deployment can later verify that the served artifact is bit-identical to the evaluated one. No other provider or host in our sample exposes an equivalent primitive. The significance is broader than Replicate itself: immutable snapshot binding is not hypothetical infrastructure. It already exists in production systems.

For closed-API providers, we cannot directly tell whether behavioral changes under a stable identifier come from changes to the underlying model or deployment. What we can show is that a stable public identifier does not necessarily mean the deployment itself is stable. Quantization disclosures show that deployment-level details can be documented, alias-update policies show that stable identifiers can point to changing artifacts, and content-hashed deployments show that immutable binding is technically possible. Taken together, this strengthens the interpretation of the M-section results where the lack of a clear link between published evaluations and deployed systems does not seem to come from a fundamental technical limitation, but from the fact that these mechanisms are generally not exposed to external evaluators.

### Proposal: A Three-Part Behavioral Trigger System

#### Diagnosis

Our results show that current disclosure practices do not provide a reliable chain of custody between published evaluations and deployed systems.

Existing governance frameworks rely on terms such as “material change,” “substantial modification,” and “relevant changes” to trigger disclosure or re-evaluation obligations. However, these terms are rarely tied to clear behavioral thresholds. The Commission’s GPAI Guidelines include an indicative compute-based threshold for substantial modifi-

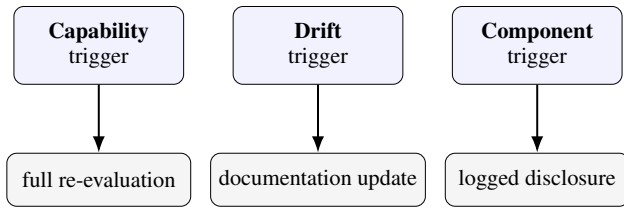


Figure 4: The proposed Three-Part Behavioral Trigger System. Capability triggers initiate re-evaluation, drift triggers initiate documentation updates, and component triggers initiate deployment-level disclosure.

cation under Article 25, but this guidance applies primarily to fine-tuning and does not address many deployment-layer modifications.

As a result, current frameworks provide limited guidance for determining when post-deployment changes should be disclosed, documented, or re-evaluated.

### Three Trigger Types

We propose replacing ambiguous “material change” language with three behavioral trigger types linked to different disclosure obligations.

The numerical thresholds below are illustrative and intended only as examples of how such a system could be operationalized under existing governance frameworks.

**Capability Triggers.** A change is treated as substantial when it produces a capability shift beyond a defined threshold on a benchmark suite, significantly changes refusal behavior on a fixed safety battery, or introduces a previously absent dangerous capability. Capability triggers initiate full re-evaluation under Article 55 and possible reconsideration of Article 25 classification status.

**Drift Triggers.** A change is treated as documentable when behavioral fingerprints drift beyond a calibrated threshold across a large prompt battery. Examples include embedding-distribution shifts over fixed prompts or measurable changes in structured-output compliance under stable identifiers. Drift triggers initiate documentation updates under the transparency provisions of the GPAI Code of Practice.

**Component Triggers.** A change is treated as disclosable when deployment-layer components are modified, including system prompts, classifiers, retrieval systems, routing policies, tool access, inference-compute budgets, or tokenizers. Component triggers initiate logged disclosure obligations within a fixed reporting window.

Component triggers are particularly important because many deployment changes occur outside the model weights themselves. Current governance obligations primarily attach to the model artifact, while users interact with a larger deployed system. Figure 4 summarizes the three trigger types and their associated obligations.

### Scope and Limits

The Three-Part Trigger System addresses several of the failures identified in our analysis. Capability triggers bind eval-

uations to snapshots. Drift triggers detect behavioral movement under stable identifiers. Component triggers surface deployment-layer modifications that may otherwise remain undisclosed.

The proposal does not resolve the contractual restrictions discussed earlier in the paper. A governance regime that requires disclosure while permitting providers to restrict external measurement creates tension between transparency obligations and independent verification.

We therefore pair the trigger proposal with a narrower recommendation: a regulatory safe harbor for third-party benchmarking conducted in good faith for governance, auditing, or academic research purposes. Comparable mechanisms already exist in areas such as copyright, financial regulation, and whistleblower protection. Eligibility could require methodological transparency, non-commercial intent, and provider notification within a fixed disclosure window.

## Limitations and Discussion

### Limitations

Our measurement scope is bounded by what providers expose through public APIs and documentation. System prompts not surfaced through APIs, internal routing policies, and deployment-layer changes visible only through weight access remain outside our empirical reach. These constraints are themselves part of the problem we study: external governance mechanisms can rely only on information providers choose to expose publicly.

The Scorecard was coded by a single rater. To improve consistency, the completed scorecard subsequently underwent an independent evidence audit focused on evidence-to-score consistency and factual accuracy. While this process strengthens confidence in the final scores, formal inter-rater reliability remains future work.

The proposed trigger system also depends on normative choices about which behaviors matter and how drift thresholds should be calibrated. We do not attempt to resolve those questions here.

Any threshold-based trigger system is vulnerable to Goodhart effects. Providers may optimize for reference batteries while drifting outside them. Rotating benchmark suites or granting qualified third parties access to non-public probes may mitigate this problem but will not eliminate it.

Our proposal applies most directly to hosted APIs, where providers control deployment and can therefore be held to disclosure obligations. Open-weight ecosystems complicate this logic because downstream deployers may substantially modify released models. Extending the framework to open-weight ecosystems remains future work.

While this paper focuses on the cross-sectional Scorecard and chain-of-custody analyses, future work could extend this framework with longitudinal disclosure-gap metrics that quantify how deployment evolves relative to public documentation over time. Candidate measures include coverage gaps (the proportion of deployed versions lacking corresponding changelog entries), persistence gaps (the proportion of retired versions lacking deprecation notices), and

granularity gaps (the ratio of deployment changes to documented behavioral updates). Estimating these quantities reliably would require continuous monitoring of provider APIs, changelogs, and historical archives over extended periods rather than the point-in-time measurements presented here.

Finally, some providers were only partially observable during the measurement period. Meta’s hosted Llama API, for example, remained preview- or waitlist-gated across portions of the study, limiting direct verification of several API-facing versioning and reproducibility criteria. Where direct inspection was not possible, scoring relied on publicly available documentation when available; otherwise, criteria were recorded as unknown rather than inferred. Scores should therefore be interpreted as measurements of observable deployment transparency rather than claims about providers’ internal governance practices.

### Political Economy

Recent events help clarify the incentives surrounding post-deployment disclosure. In July 2025, Anthropic revoked OpenAI’s Claude API access under a competing-product clause that contemporary reporting linked to benchmarking activity. Cohere’s Terms of Use contain similarly explicit restrictions on benchmarking. These cases illustrate a broader pattern in our sample: providers may publish substantial transparency artifacts while simultaneously limiting outside evaluation through contractual controls.

Current governance frameworks encourage disclosure but rarely require providers to expose the technical or contractual conditions needed for outside parties to verify deployment behavior independently.

Providers can reasonably be expected to resist aspects of the trigger proposal introduced earlier in the paper. Operationalized thresholds increase compliance costs, create additional audit exposure, and reduce some of the flexibility currently afforded by silent updates. Numerical thresholds, reporting windows, and definitions of safety-relevant drift would likely remain subjects of regulatory negotiation.

At the same time, modification-trigger regimes are not unprecedented. Industries such as aviation, pharmaceuticals, and medical devices already operate under formal change-disclosure and re-evaluation requirements. We do not claim that AI systems present identical risks, only that post-deployment disclosure obligations are compatible with technologically dynamic industries.

The current regime leaves substantial discretion with providers over what counts as a disclosable change and who may independently evaluate those changes. One purpose of the Scorecard is to make those tradeoffs more visible.

### Conclusion

In this paper, we show that it is not currently possible to connect evaluation results to deployed systems in externally verifiable ways. Across 9 AI providers we examined, the information available through public documentation and standard API access was generally insufficient to verify that the evaluated model was the one actually being served.

We describe this configuration as *transparency without verifiability*. In response, we propose two complementary

mechanisms: a Three-Part Behavioral Trigger System for determining when post-deployment changes should trigger disclosure obligations, and a regulatory safe harbor for governance-purposed benchmarking to support independent evaluation.

Current AI governance frameworks implicitly assume a chain of custody between evaluated and deployed systems. Our findings suggest that this chain is often missing in practice. Building more reliable links between evaluations and deployed behavior should therefore be treated as a priority in future work on AI evaluations and governance.

### Ethical Statement

This work evaluates the public disclosure practices of foundation model providers using publicly accessible documentation, Terms of Service, and standard API interactions. We do not collect personal data or rely on non-public information. Each score in the Scorecard is linked to documentary evidence to support external review and replication.

The Scorecard measures externally observable disclosure rather than underlying provider conduct. A low score reflects the absence of publicly verifiable disclosure, not evidence of misconduct or deceptive intent. Throughout the paper, we limit our claims to what can be established from public-facing artifacts and standard developer access.

The governance proposals discussed in this paper would impose additional disclosure and compliance obligations on providers if implemented. The specific thresholds and trigger mechanisms we outline are intended as illustrative starting points rather than fixed technical standards.

### References

- Act, E. A. I. 2024. The eu artificial intelligence act. *European Union*.
- Anderljung, M.; Smith, E. T.; O’Brien, J.; Soder, L.; Bucknall, B.; Bluemke, E.; Schuett, J.; Trager, R.; Strahm, L.; and Chowdhury, R. 2023. Towards Publicly Accountable Frontier LLMs: Building an External Scrutiny Ecosystem under the ASPIRE Framework. *arXiv:2311.14711*.
- Anthropic. 2026a. Model configuration. Claude Code Documentation.
- Anthropic. 2026b. Model IDs and Versioning. Anthropic Developer Documentation. Accessed May 2026.
- Anthropic. 2026c. System Card: Claude Opus 4.7. Technical report, Anthropic. Accessed May 2026.
- Birhane, A.; Steed, R.; Ojewale, V.; Vecchione, B.; and Raji, I. D. 2024. AI Auditing: The Broken Bus on the Road to AI Accountability. *arXiv preprint arXiv:2401.14462*.
- Bommasani, R.; Klyman, K.; Kapoor, S.; Maslej, N.; Longpre, S.; Xiong, B.; and Liang, P. 2025. The 2025 Foundation Model Transparency Index. *arXiv preprint arXiv:2512.10169*.
- Brundage, M.; Dreksler, N.; Homewood, A.; McGregor, S.; Paskov, P.; Stosz, C.; Sastry, G.; Cooper, A. F.; Balston, G.; Adler, S.; Casper, S.; Anderljung, M.; Werner, G.; Mindermann, S.; Mavroudis, V.; Bucknall, B.; Stix, C.; Freund, J.; Pacchiardi, L.; Hernandez-Orallo, J.; Pistillo, M.; Chen,

- M.; Painter, C.; Ball, D. W.; O’Keefe, C.; Weil, G.; Harack, B.; Finley, G.; Hassan, R.; Emmons, S.; Foster, C.; Reuel, A.; Treece, B.; Bengio, Y.; Reti, D.; Bommasani, R.; Trout, C.; Shamsabadi, A. S.; Dattani, R.; Weller, A.; Trager, R.; Sevilla, J.; Wagner, L.; Soder, L.; Ramakrishnan, K.; Papadatos, H.; Murray, M.; and Tovcimak, R. 2026. Frontier AI Auditing: Toward Rigorous Third-Party Assessment of Safety and Security Practices at Leading AI Companies. *arXiv:2601.11699*.
- Cai, W.; Shi, T.; Zhao, X.; and Song, D. 2025. Are You Getting What You Pay For? Auditing Model Substitution in LLM APIs. *arXiv:2504.04715*.
- Chauvin, T.; Lalanne, C.; Merrer, E. L.; Loubes, J.-M.; Taïani, F.; and Tredan, G. 2026a. Token-Efficient Change Detection in LLM APIs. *arXiv:2602.11083*.
- Chauvin, T.; Merrer, E. L.; Taïani, F.; and Tredan, G. 2026b. Log Probability Tracking of LLM APIs. *arXiv:2512.03816*.
- Chen, L.; Cai, T.; Zaharia, M.; and Zou, J. 2021. Did the Model Change? Efficiently Assessing Machine Learning API Shifts. *arXiv:2107.14203*.
- Chen, L.; Zaharia, M.; and Zou, J. 2024. How Is ChatGPT’s Behavior Changing Over Time? *Harvard Data Science Review*, 6(1).
- Cobbe, J.; Veale, M.; and Singh, J. 2023. Understanding Accountability in Algorithmic Supply Chains. *arXiv preprint arXiv:2304.14749*.
- Cohere. 2025. Command A: An Enterprise-Ready Large Language Model. Technical report, Cohere. Accessed May 2026.
- Cohere. 2026. Terms of Use. Cohere Legal Directory. Accessed May 2026.
- Dave, P. 2025. Anthropic Revokes OpenAI’s Access to Claude. *WIRED*. Accessed May 2026.
- DeepSeek AI. 2024. DeepSeek-V2.5: A New Open-Source Model Combining General and Coding Capabilities. DeepSeek API Documentation Changelog.
- Derczynski, L.; Kirk, H. R.; Thukral, S.; Röttger, P.; and Hale, S. A. 2023. Assessing Language Model Deployment with Risk Cards. *arXiv preprint arXiv:2303.18190*.
- European Commission. 2025. General-Purpose AI Code of Practice. Technical report, European Union. Final Version.
- European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 (Artificial Intelligence Act). Official Journal of the European Union, OJ L, 2024/1689, 12.7.2024.
- Eyuboglu, S.; Goel, K.; Desai, A.; Chen, L.; Monfort, M.; Ré, C.; and Zou, J. 2024. Model changelists: Characterizing updates to ml models. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2432–2453.
- Gao, I.; Liang, P.; and Guestrin, C. 2025. Model Equality Testing: Which Model Is This API Serving? In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J. W.; Wallach, H.; Iii, H. D.; and Crawford, K. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12): 86–92.
- Gilbert, T. K.; Dean, S.; Zick, N.; and Lambert, N. 2023. Choices, Risks, and Reward Reports: Charting Public Policy for Reinforcement Learning. *Journal of Artificial Intelligence Research (JAIR)*, 76: 1173–1195.
- Google Cloud. 2026. Model versions and lifecycle. Gemini Enterprise Agent Platform Documentation.
- Google DeepMind. 2025. Gemini 3 Pro Model Card. Technical report, Google. Accessed May 2026.
- Google DeepMind. 2026. Gemini 3.1 Pro Model Card. Technical report, Google. Accessed May 2026.
- Keller, D. J.; Kwegyir-Aggrey, K.; Steed, R.; Rao, A. K.; Sharp, J. L.; and Bergman, S. S. 2026. Expanding the AI Evaluation Toolbox with Statistical Models. Technical Report NIST AI 800-3, National Institute of Standards and Technology (NIST), Gaithersburg, MD.
- Longpre, S.; Kapoor, S.; Klyman, K.; Ramaswami, A.; Bommasani, R.; Blili-Hamelin, B.; Huang, Y.; Skowron, A.; Yong, Z.-X.; Kotha, S.; Zeng, Y.; Shi, W.; Yang, X.; Southen, R.; Robey, A.; Chao, P.; Yang, D.; Jia, R.; Kang, D.; Pentland, S.; Narayanan, A.; Liang, P.; and Henderson, P. 2024. A Safe Harbor for AI Evaluation and Red Teaming. *arXiv:2403.04893*.
- McCaslin, T.; Alaga, J.; Nedungadi, S.; Donoughe, S.; Reed, T.; Bommasani, R.; Painter, C.; and Righetti, L. 2025. STREAM (ChemBio): A standard for transparently reporting evaluations in AI model reports. *arXiv preprint arXiv:2508.09853*.
- Meta AI. 2025. Llama 4 Model Card. Technical report, Meta. Accessed May 2026.
- Microsoft Cloud. 2025. Azure OpenAI Service content filtering. Microsoft Learn Documentation. Accessed May 2026.
- Mireshghallah, N.; Rytting, C. M.; Newman, B.; Radford, A.; Carter, J.; and Gleave, A. 2024. Value Drifts: Tracing Value Alignment During LLM Deployment in the Wild. In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; and Gebru, T. 2019a. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, FAT\* ’19, 220–229. ACM.
- Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I. D.; and Gebru, T. 2019b. Model Cards for Model Reporting. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAccT)*, 220–229.
- Mökander, J.; Schuett, J.; Kirk, H. R.; and Floridi, L. 2023. Auditing large language models: a three-layered approach. *AI and Ethics*, 4(4): 1085–1115.
- Núñez, M. 2025. Anthropic cracks down on unauthorized Claude usage by third-party harnesses. *VentureBeat*. Accessed May 2026.

O'Brien, J.; Ee, S.; and Williams, Z. 2023. Deployment Corrections: An incident response framework for frontier AI models. *arXiv:2310.00328*.

Ojewale, V.; Steed, R.; Vecchione, B.; Birhane, A.; and Raji, I. D. 2025. Towards AI Accountability Infrastructure: Gaps and Opportunities in AI Audit Tooling. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, CHI '25, 1–29. ACM.

OpenAI. 2025. GPT-5 System Card. Technical report, OpenAI. Accessed May 2026.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. *arXiv:2203.02155*.

Pratt, J.; and Tanjaya, A. 2025. Documenting the Impacts of Foundation Models. *Partnership on AI*.

Raji, I. D.; Smart, A.; White, R.; Mitchell, M.; Gebru, T.; Hutchinson, B.; Smith-Loud, J.; Theron, D.; and Barnes, P. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency (FAccT)*, 33–44.

Spoczynski, M.; Melara, M. S.; and Szyller, S. 2025. Atlas: A Framework for ML Lifecycle Provenance & Transparency. *arXiv:2502.19567*.

Staufer, L.; Yang, M.; Reuel, A.; and Casper, S. 2025. Audit cards: Contextualizing ai evaluations. *arXiv preprint arXiv:2504.13839*.

Together AI. 2026. Terms of Service. Together AI Legal Documentation. Accessed May 2026.

Widder, D. G.; and Nafus, D. 2023. Dislocated Accountabilities in the “AI Supply Chain”: Modularity and Developers’ Notions of Responsibility. *Big Data & Society*, 10(1): 20539517231177620.

xAI. 2025. Grok 4.1 Model Card. Technical report, xAI. Accessed May 2026.

Xu, J.; Ung, M.; Komeili, M.; Arora, K.; Boureau, Y.-L.; and Weston, J. 2022. Learning New Skills after Deployment: Improving open-domain internet-driven dialogue with human feedback. *arXiv:2208.03270*.