Inference-Time Control for Trustworthy Large Language Models

Yuyang Bai [‡,†] , Zheyuan Liu [‡] , Han Yan [‡] , Zhangchen Xu [‡] , Yixin Wan [‡] , Canyu Chen [‡] , Zehong Wang [‡] , Xiangchi Yuan [‡] , Yue Huang [‡] , Guangyao Dou [‡] , Yuji Zhang , Hangxiao Zhu , Zhuofeng Li , Manling Li , Xiangliang Zhang , Mohit Bansal , Sanmi Koyejo , Kai-Wei Chang , Yu Zhang [*] , Meng Jiang [*]

Posted Date: 15 May 2026

doi: 10.20944/preprints202605.1041.v1

Keywords: large language models; trustworthy AI; inference-time control; safety; privacy; fairness; factuality; guardrails; decoding strategies; representation engineering; machine unlearning; multi-agent systems

 doi:10.20944/preprints202605.1041.v1

Article

Inference-Time Control for Trustworthy Large Language Models

Yuyang Bai [1,†,‡], Zheyuan Liu [2,†,‡], Han Yan [2,‡], Zhangchen Xu [3,‡], Yixin Wan [4,‡], Canyu Chen [5,‡], Zehong Wang [2,‡], Xiangchi Yuan [6,‡], Yue Huang [2,‡], Guangyao Dou [7,‡], Yuji Zhang [8], Hangxiao Zhu [1], Zhuofeng Li [1], Manling Li [5], Xiangliang Zhang [2], Mohit Bansal [9], Sanmi Koyejo [10], Kai-Wei Chang [4], Yu Zhang [1,*] and Meng Jiang [2,*]

[1] Texas A&M University, USA
[2] University of Notre Dame, USA
[3] University of Washington, USA
[4] University of California, Los Angeles, USA
[5] Northwestern University, USA
[6] Georgia Institute of Technology, USA
[7] Johns Hopkins University, USA
[8] University of Illinois at Urbana-Champaign, USA
[9] University of North Carolina at Chapel Hill, USA
[10] Stanford University, USA
* Correspondenc: yuzhang@tamu.edu (Y.Z.); mjiang2@nd.edu (M.J.)
† Project Leader.
‡ Major Contributor.

Abstract

Once a large language model is released, training-time alignment is hard to revise; yet deployment introduces context-specific risks that the original training cannot anticipate: evolving safety policies, jurisdictional constraints, retrieval contamination, and adaptive adversarial prompting. In this paper, we unify inference-time techniques for trustworthy generation across safety, privacy, fairness, and factuality under a single framework: the *inference-time control plane*, with three tiers of intervention—*External Controls* (Context Engineering, Guardrails, Decoding Strategies), which act around the model; *Internal Manipulations* (Representation Engineering, Unlearning, Pruning), which act inside it; and *System-Level Orchestration* (Multi-Agent Systems), which coordinate several models. We also introduce a meta-axis evaluation framework that crosses the four trustworthiness dimensions with five evaluation axes (effectiveness, locality, generality, interpretability, efficiency), and describe representative metrics at each intersection. We identify four cross-cutting open problems: brittleness under adaptive adversaries, the control–utility tradeoff, verification of removal, and the composition of layered interventions. A curated paper list is available at https://github.com/leopoldwhite/Awesome-Inference-Time-Trustworthiness.

Keywords: large language models; trustworthy AI; inference-time control; safety; privacy; fairness; factuality; guardrails; decoding strategies; representation engineering; machine unlearning; multi-agent systems

1. Introduction

Large Language Models (LLMs) and Multimodal Large Language Models (MLLMs) have rapidly moved from research prototypes to deployed systems in domains such as healthcare, law, and finance. Models are now accessed primarily through APIs, regulatory frameworks like the EU AI Act are imposing new behavioral requirements, and new vulnerabilities are discovered continuously after deployment. Existing work has largely emphasized train-time trustworthiness, where safety, fairness, privacy, or factuality objectives are encoded during parameter learning through data curation, supervised alignment, or preference optimization. These methods aim to shape the model's default behavior globally and persistently by modifying what the model internalizes.

 doi:10.20944/preprints202605.1041.v1

In this paper, we distinguish this perspective from inference-time trustworthiness, which concerns mechanisms that regulate model behavior at runtime after training has been completed. Rather than changing the model's learned parameters alone, inference-time methods operate on the generation process through contextual control, decoding-time intervention, representation engineering, internal steering, or multi-agent verification. This distinction extends beyond considerations of efficiency or convenience. More fundamentally, train-time and inference-time methods differ in where control is exercised, how persistent the intervention is, and what level of guarantees it can provide. Train-time methods primarily shape a model's behavioral prior, whereas inference-time methods determine how that prior is enacted, constrained, or corrected in a specific interaction context.

These two forms of trustworthiness are complementary. Train-time alignment is well suited for establishing broad and stable default behavior, but deployment introduces context-specific risks that cannot always be fully included during training, including changing policies, user-specific constraints, retrieval contamination, adversarial prompting, and jurisdiction-dependent requirements. Inference-time mechanisms provide a second control plane that is local, adaptive, and often more directly auditable at the level of individual interactions, as evidenced by guardrail families that expose per-query policy identifiers, risk scores, and decision traces [1–4]. They offer a complementary control layer that can steer, constrain, or monitor model behavior after deployment. Because they are often lightweight and modular, they are well suited to dynamic and context-specific trustworthiness requirements. Conversely, inference-time control alone cannot substitute for robust underlying models, because runtime interventions may be brittle, reactive, or incomplete without strong train-time foundations—for example, adversarial probing can recover supposedly unlearned knowledge [5,6], and guardrails remain susceptible to adaptive jailbreaks that evade fixed detectors [7]. A full account of trustworthy LLM deployment therefore requires treating trustworthiness as a multi-layer property spanning both training and inference.

The necessity of this multi-layered approach is especially salient in high-stakes deployments such as clinical decision support, legal drafting, and financial advising, where even low-frequency trustworthiness failures can translate into concrete harm and where jurisdiction-specific constraints make a single global default behavior insufficient. As general-purpose reasoning capabilities continue to strengthen, a complementary framing has gained traction in the recent agent literature. Often termed harnessing, this approach treats deployment-time trustworthiness as the problem of wrapping capable but imperfectly aligned models in runtime scaffolding, such as context management, policy enforcement, and agent coordination. This concept is commonly summarized by the equation Agent = Model + Harness. This framing maps naturally onto two tiers of our taxonomy: Tier 1 External Controls and Tier 3 System-Level Orchestration. However, it excludes an equally important class of interventions that act on the model's internal computations, such as Representation Engineering, Unlearning, and Pruning. These internal methods become essential when trustworthiness requirements cannot be met by surface scaffolding alone, requiring us instead to bound what the model can recall, represent, or activate in the first place. We therefore adopt a broader inference-time scope that unifies harness-style external controls with internal manipulations under a single framework, capturing the full stack of deployment-time intervention points available after training concludes.

Many inference-time techniques were first developed to improve model performance, such as reducing latency, improving fluency, or grounding outputs through retrieval. More recently, they have also been adapted to support trustworthiness during deployment, including safety, privacy, fairness, and factuality. This shift calls for a more unified view of the field. In this paper, we organize inference-time trustworthiness methods by the mechanism and locus of control they introduce during generation. As shown in Figure 1, we group them into three tiers: **External Controls**, **Internal Manipulations**, and **System-Level Orchestration**. This perspective highlights shared design patterns and clarifies trade-offs in granularity, invasiveness, and modularity. These tiers differ in where and how they intervene, ranging from black-box control around the model, to white-box intervention inside the model, to coordinated control across multiple agents.

 doi:10.20944/preprints202605.1041.v1

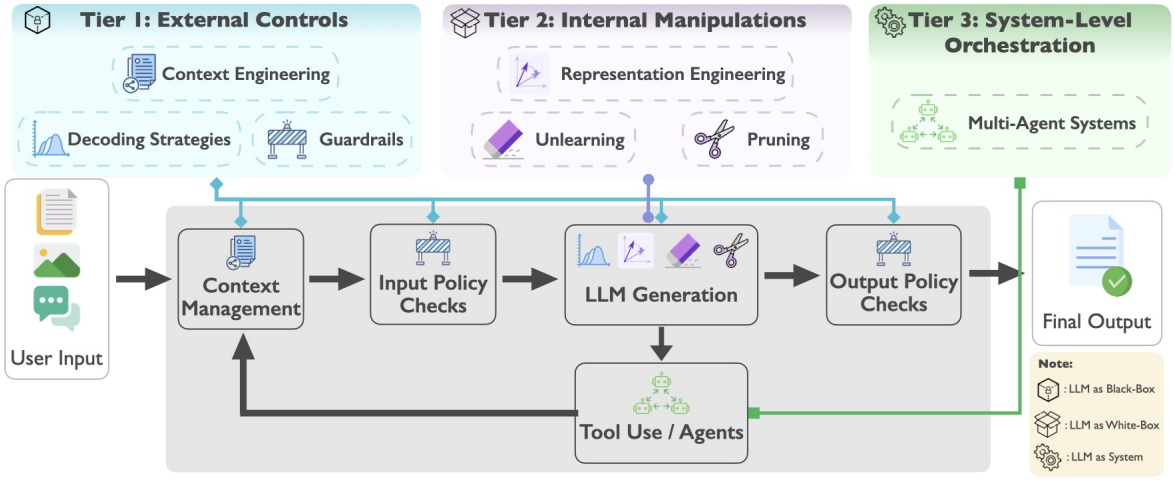

Figure 1. Taxonomy (top) and attachment points in an inference-time pipeline (bottom). We organize seven inference-time methods into three color-coded tiers—*Tier 1: External Controls* (**cyan**, black-box: Context Engineering, Guardrails, Decoding Strategies), *Tier 2: Internal Manipulations* (**lavender**, white-box: Representation Engineering, Unlearning, Pruning), and *Tier 3: System-Level Orchestration* (**green**, system-level: Multi-Agent Systems). Each tier mapped to its point of intervention in the generation pipeline.

Tier 1: External Controls.

Black-box methods that intervene at the *context assembly*, *input/output policy checks*, and *decoding* stages of the pipeline:

- **Context Engineering** (Section 2.1): Strategic prompt design through rules, instructions, or few-shot exemplars to guide outputs without modifying model parameters.
- **Guardrails** (Section 2.2): External modules that inspect inputs or outputs against safety or policy constraints, blocking, redacting, or regenerating content when violations occur.
- **Decoding Strategies** (Section 2.3): Manipulation of token-level distributions during generation to promote desired attributes or suppress undesired ones.

Tier 2: Internal Manipulations.

White-box methods that operate within the *LLM generation* stage of the pipeline:

- **Representation Engineering** (Section 3.1): Direct modification of internal activations by adding or subtracting steering vectors associated with specific concepts.
- **Unlearning** (Section 3.2): Targeted removal of information, behaviors, or biases from a pre-trained model to fulfill data-forgetting requirements or disable harmful capabilities.
- **Pruning** (Section 3.3.1): Post-training removal of weights, neurons, or attention heads, originally for efficiency but now increasingly explored for trust-related effects.

Tier 3: System-Level Orchestration.

A single category, **Multi-Agent Systems** (Section 4.1), in which a *tool use / agents* loop spans the entire pipeline through coordinated agent interactions such as debate, cross-verification, and role specialization.

These three tiers form a defense-in-depth view of inference-time trustworthiness. External controls provide lightweight and modular policy enforcement. Internal manipulations enable more direct and fine-grained behavioral intervention. System-level orchestration introduces an additional layer of reliability through collaborative reasoning and feedback. Across this taxonomy, we focus on four dimensions of trustworthiness: **Safety**, which concerns preventing harmful, biased, or malicious outputs and defending against misuse such as jailbreaks; **Privacy**, which concerns limiting training-data leakage and protecting user inputs during inference; **Fairness**, which concerns reducing systematic disadvantages toward individuals or groups; and **Factuality**, which concerns grounding model outputs in verifiable knowledge and reducing hallucinations. These three tiers and four dimensions together

 doi:10.20944/preprints202605.1041.v1

define the inference-time control plane, and we analyze how its components trade off and compose under deployment constraints.

1.1. Related Work

This work sits at the intersection of three research strands. **Efficiency-focused work** on inference-time optimizations such as quantization, distillation, speculative decoding, and Mixture-of-Experts deployment [8–10] targets computational performance rather than trustworthy behavior. **Post-training capability research** on prompting and reinforcement learning aims to make models more capable reasoners or tool users [11,12]; deployment-time enforcement of safety, privacy, or fairness is peripheral. **Trustworthy AI research** either takes a broad view [13,14] or focuses on specific facets such as text-to-image bias [15], autonomous agent safety [16], and machine unlearning [17,18]. To our knowledge, no prior work provides a unified framework that treats *inference-time methods as a control plane* for trustworthiness across safety, privacy, fairness, and factuality.

1.2. Contributions and Organization

Our key contributions are:

- **A Pipeline-Grounded, Three-Tier Framework.** We propose a three-tier framework grounded in the generation pipeline (Figure 1). The framework covers seven inference-time control methods and groups them by their intervention point, access requirements, and composability within a defense-in-depth design.
- **Cross-Dimensional Analysis.** We analyze how each category addresses Safety, Privacy, Fairness, and Factuality, clarifying capabilities, limitations, and trade-offs under deployment constraints.
- **Cross-Cutting Open Problems.** We identify open problems on scalability, evaluation, and composition that cut across the framework, and discuss directions for future work.

The remainder is organized as follows. **Sections 2–4** cover the three tiers in turn. **Section 5** analyzes evaluation benchmarks and trade-offs. **Section 6** discusses open problems and limitations, and **Section 7** concludes.

Table 1. Comparison with related work on inference-time trustworthiness metrics and methods.

Prior Work	Inference-time Metrics				Inference-time Methods						
	Trustworthiness Dimensions				External Controls			Internal Manipulations			System-Level Orchestration
	Safety	Privacy	Fairness	Factuality	Context Engineering	Guardrails	Decoding	Representation Engineering	Unlearning	Pruning	Multi-Agent Systems
Ours	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Donisch et al. [8]	✗	✗	✗	✗	✗	✗	✓	✗	✗	✓	✗
Kumar et al. [12]	✓	✓	✗	✓	✗	✓	✓	✓	✗	✓	✓
Tie et al. [11]	✓	✗	✓	✓	✗	✗	✓	✓	✗	✓	✓
Huang et al. [13]	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗
Liu et al. [14]	✓	✓	✓	✓	✓	✓	✗	✓	✓	✗	✗
Wan et al. [15]	✓	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗
Yu et al. [16]	✓	✓	✓	✓	✓	✗	✗	✗	✗	✗	✓
Barez et al. [18]	✓	✓	✗	✗	✓	✓	✓	✓	✓	✗	✗
Liu et al. [17]	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗

2. External Controls

External controls treat the model as a black box, shaping its behavior by manipulating inputs, the decoding process, or outputs—without accessing or modifying internal weights or activations. These methods are the most modular and widely applicable, as they require no white-box access and can be deployed on proprietary, API-only models. In the inference-time pipeline (Figure 1), they attach to the context assembly, input policy checks, decoding, and output policy checks stages.

2.1. Context Engineering

Within the landscape of inference-time methods for trustworthy LLMs, context engineering (CE) has emerged as a central paradigm. Instead of modifying model parameters, CE emphasizes the deliberate design and orchestration of the information presented to the model at inference time. This approach is particularly important for trustworthiness: by shaping the context, one can systematically influence factuality, safety, fairness, etc, without modifying model's parameters.

 doi:10.20944/preprints202605.1041.v1

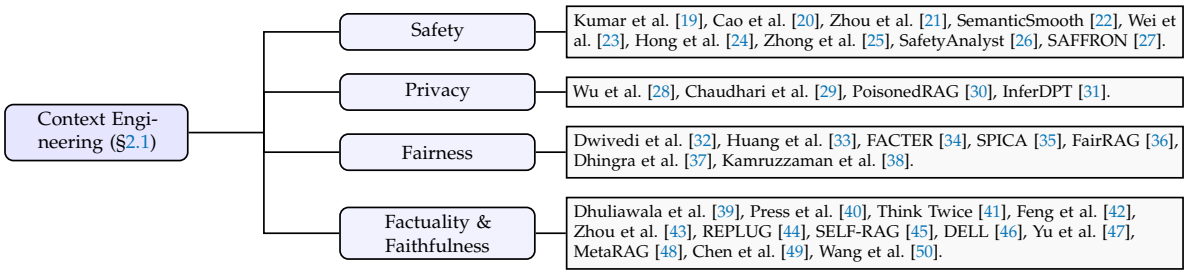

Figure 2. A taxonomy of Context Engineering for enhancing LLM trustworthiness.

2.1.1. Core Principle

Context engineering is the systematic discipline of designing, optimizing, and managing the information payload provided to LLMs at inference time. Unlike early prompt engineering, which treated the context as a flat string, CE models the context as a structured composition of four components: directive instructions encompassing rules and constraints; knowledge inputs such as retrieved documents or prior dialogue; interaction mechanisms including tools, APIs, or system affordances; and the live user query specifying the task at hand. An assembly function integrates these heterogeneous elements into the final sequence consumed by the model.

The CE framework rests on three foundational pillars that together form a pipeline. Context retrieval and generation is responsible for sourcing task-relevant material through prompt design, retrieval from external knowledge bases, and adaptive assembly strategies. Context processing transforms raw material into model-ready input through compression, refinement, and multimodal or structured integration. Context management ensures coherence and efficiency across stages by dynamically organizing, prioritizing, and updating contextual elements under system constraints such as the maximum context length. Together, these pillars support higher-level methods such as Retrieval-Augmented Generation (RAG), persistent memory, tool-augmented reasoning, and multi-agent collaboration.

2.1.2. Applications

Safety

Context engineering promotes safety by controlling the informational boundary within which a model operates, encoding explicit constraints at the prompt level, grounding generation in trusted external sources, and embedding verification routines into the reasoning process. Prompt engineering offers a lightweight first layer of defense, ranging from certifiable defenses with robustness guarantees against adversarial prompts [19] and robust alignment prompting [20] to baseline defenses such as paraphrasing, in-context refusals, and perplexity checks [21], as well as smoothing-based methods like SemanticSmooth that guard against both token- and prompt-level attacks [22]. Retrieval-augmented generation further enhances safety by grounding responses in verifiable external knowledge, reducing hallucinations and lowering the risk of unsafe misinformation [23,24], though vulnerabilities such as corpus poisoning attacks that insert adversarial passages into knowledge bases remain a concern [25]. Reasoning-oriented approaches complement these strategies by enabling models to regulate their own outputs during inference through structured step-by-step verification [26] and safety-aware scoring that resists jailbreaks at scale [27].

Privacy

Context engineering for privacy focuses on controlling what enters the model's context. Instead of modifying parameters, privacy is preserved by assembling the context such that sensitive elements are masked, filtered, or abstracted while retaining task-relevant information. Recent work shows that CE can both introduce and mitigate privacy risks. Prompts themselves may leak sensitive training data through property and membership inference attacks [28], and RAG pipelines are vulnerable to poisoned documents that can exfiltrate private passages or bias model outputs [29]. On the defense side, certified privacy-preserving mechanisms for RAG offer robustness against adversarial

 doi:10.20944/preprints202605.1041.v1

manipulation [30], while frameworks for selective retrieval and privacy-aware generation control personal data exposure [31]. Together, these studies underscore that privacy risks often stem from how external context is engineered into models, and effective mitigation requires careful control, sanitization, and certification of injected information.

Fairness

Bias in LLMs often arises from skewed training data and emergent model behavior, and context engineering provides a practical means of mitigation without retraining. Carefully designed prompts have been shown to reduce bias, from employing in-context learning to mitigate gender stereotypes [32] to frameworks for testing and reducing social bias in code generation through iterative, feedback-driven prompting [33] and embedding explicit fairness constraints in prompts [34]. Retrieval-augmented methods enhance fairness by grounding outputs in curated, balanced evidence, with fairness-aware retrieval mechanisms prioritizing diverse and representative sources to reduce stereotype reinforcement [35,36]. A third line of work integrates structured reasoning through chain-of-thought [37] or deliberative reasoning [38] to enforce fairness constraints, decomposing decisions into interpretable steps that enable bias detection and correction during inference.

Factuality

A core challenge in trustworthy AI is mitigating factual errors and hallucinations, and context engineering offers several complementary strategies. Prompt engineering improves factual consistency through structured instructions, including self-detection prompts that require models to evaluate multiple candidate answers before finalizing a response [39–41], abstention-oriented prompting that encourages models to decline uncertain answers in multilingual settings [42], and opinion-based or counterfactual demonstrations that reduce reliance on parametric knowledge [43]. Retrieval-augmented generation enhances factual grounding by supplementing LLMs with external documents, with frameworks enabling adaptive and reflective retrieval such as SELF-RAG [45], integration of retrieval with proxy explanations for misinformation detection [44,46,47], and metacognitive loops that detect insufficient or conflicting knowledge and adjust retrieval accordingly [48]. Memory-based methods further improve factuality by allowing models to retain and update knowledge across interactions through external memory buffers where user feedback persists across sessions [49] and modular memory systems combining episodic traces and semantic caches [50].

2.2. Guardrails

2.2.1. Core Principle: Modular and External Control

Guardrails provide a modular and external mechanism for controlling LLM behavior, operating as independent components that monitor and filter inputs or outputs without modifying the model's parameters. Unlike training-time interventions, guardrails function post-deployment, treating the LLM as a black-box system. This external positioning enables rapid deployment and updates, allowing systems to quickly adapt to evolving safety requirements and emerging threats. Their modular design supports flexible integration into existing pipelines, where multiple guardrails can be layered or combined to target specific risks. As external controls, guardrails ensure trustworthy outputs by intercepting potentially harmful content before it reaches the user or by regenerating outputs when violations are detected. While this principle draws conceptually from traditional content moderation systems, it is now tailored to the dynamic and interactive nature of LLMs, emphasizing flexibility and minimal interference with the model's core functionality.

2.2.2. A Typology of Guardrail Mechanisms

Guardrail mechanisms can be classified according to their underlying approach, spanning from simple deterministic techniques to advanced LLM-driven systems.

 doi:10.20944/preprints202605.1041.v1

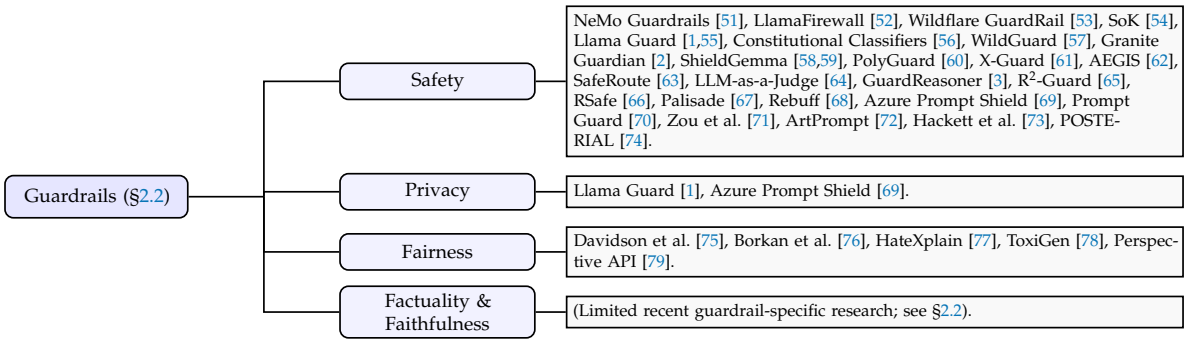

Figure 3. A taxonomy of Guardrails for enhancing LLM trustworthiness.

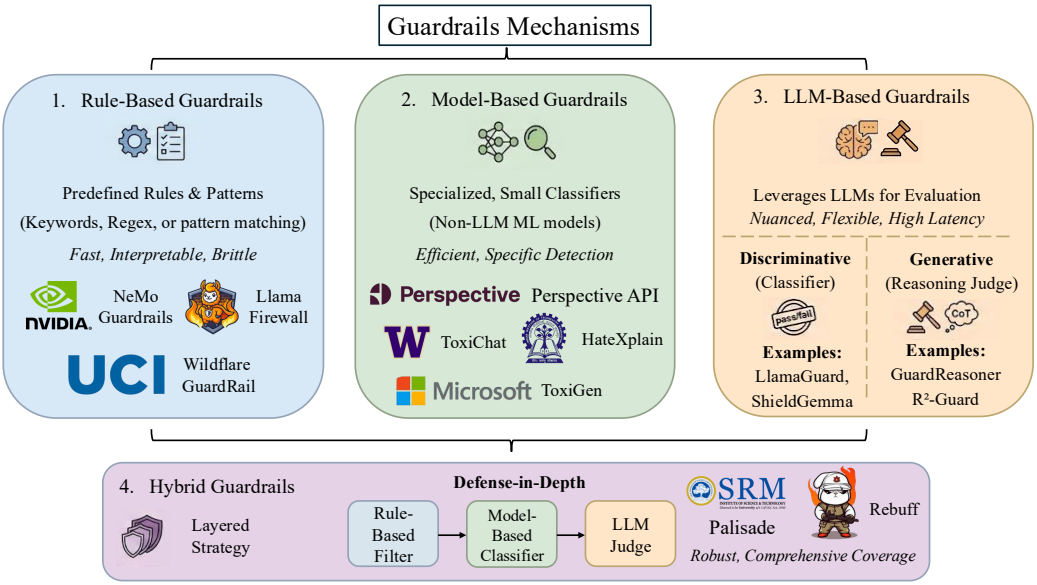

Figure 4. A typology of guardrail mechanisms for LLM trustworthiness. The four categories—Rule-Based, Model-Based, LLM-Based (discriminative and generative), and Hybrid—are distinguished by their underlying detection approach, with representative systems shown for each. Hybrid guardrails combine multiple mechanism types in a layered "defense-in-depth" strategy for comprehensive coverage.

Rule-Based Guardrails

Rule-based guardrails rely on predefined rules, such as keyword lists, regular expressions, or pattern matching, to detect and filter prohibited content. For example, NeMo Guardrails [51] introduces programmable rails for dialog flow, topic blocking, and tool use; LlamaFirewall [52] incorporates customizable scanners including regex-based code filters for agentic workflows; and Wildflare GuardRail [53] integrates modular rule-based wrappers as part of a multi-stage pipeline. While transparent and low-latency, these systems are brittle against novel adversarial strategies specific to LLMs, such as prompt injection and jailbreak attacks [54].

Model-Based Guardrails

Model-based guardrails employ smaller, specialized classifiers—typically non-LLM models such as neural networks or fine-tuned smaller transformers—for nuanced detection of safety violations. Early work focused on hate speech and toxicity detection, addressing challenges such as distinguishing hate speech from offensive content [75], fairness-oriented evaluation for toxicity detectors [76], and adversarially generated datasets for implicit toxicity [78]. These insights informed practical deployments such as the Perspective API [79], which detects toxicity across languages, and explainable models like HateXplain [77] that provide human rationales alongside labels.

 doi:10.20944/preprints202605.1041.v1

LLM-Based Guardrails

LLM-based guardrails employ another large language model to evaluate and filter generated content, capturing nuance, abstraction, and intent that smaller classifiers often miss. Discriminative guardrails use LLMs as safety classifiers: Llama Guard [1,55] detects unsafe multimodal content categories, while Constitutional Classifiers [56], WildGuard [57], Granite Guardian [2], and Shield-Gemma [58,59] provide fine-tuned drop-in safety classifiers. Recent work has expanded to multilingual coverage with PolyGuard [60] and X-Guard [61], while ensemble approaches like AEGIS [62] and adaptive routing via SafeRoute [63] reflect a shift toward scalable, real-time solutions. Generative (reasoning) guardrails employ an LLM as a deliberative judge that critiques and justifies another model's output. Building on the LLM-as-a-judge paradigm [64], recent methods incorporate explicit reasoning chains before producing verdicts, such as GuardReasoner [3], R^2-Guard [65] with knowledge-enhanced logical reasoning, GuardAdvisor [80] with a soft-gating pipeline, and RSafe [66] with adaptive guided safety reasoning.

Hybrid Guardrails

Hybrid guardrails combine multiple mechanisms for comprehensive coverage through a "defense-in-depth" strategy. For example, Palisade [67] proposes a three-layer pipeline combining a rule-based filter, an ML classifier, and a companion-LLM check, while Rebuff [68] integrates heuristics, canary tokens, a database of known attacks, and an LLM-based check.

2.2.3. Applications

Safety

Guardrails are predominantly applied in safety, where they serve as critical defenses against emerging threats inherent to LLMs. On the input side, guardrails detect and block attacks such as jailbreaks that aim to bypass alignment constraints [71,72] and prompt injections where adversaries embed malicious instructions to hijack model outputs [73,74]. Systems like Azure Prompt Shield [69] and Prompt Guard [70] employ adversarial-input detectors to identify injection attempts and block unsafe prompts. On the output side, guardrails evaluate and filter model generations before delivery to end users, with systems like Llama Guard [1] enforcing a taxonomy of unsafe categories—including violent crimes, sex crimes and child exploitation, defamation, hate speech, self-harm, and election misinformation—by classifying outputs and blocking or regenerating unsafe completions.

Privacy

In the privacy domain, guardrails protect against the disclosure of personal or sensitive information. Output-side guardrails can classify and block generations that contain personally identifiable information, confidential data, or intellectual property violations. Systems such as Llama Guard [1] include disclosure of sensitive information as an explicit unsafe category, while Azure Prompt Shield [69] detects attempts to exfiltrate private data through prompt injection. However, guardrail-specific research for privacy remains limited compared to safety, and most privacy protections are currently bundled within broader safety taxonomies rather than addressed by dedicated guardrail mechanisms.

Fairness

Guardrails for fairness have historical roots in pre-LLM content moderation, where classifiers were designed to detect and reduce bias in generated content [76,77]. Toxicity and hate speech detectors such as Perspective API [79] and ToxiGen classifiers [78] serve as fairness-oriented guardrails by flagging content that disparages or stereotypes specific demographic groups. In the LLM era, however, fairness-specific guardrail research has received less attention, likely due to the assumption that post-training alignment methods such as RLHF improve baseline fairness, leaving a gap for dedicated fairness enforcement mechanisms at inference time.

 doi:10.20944/preprints202605.1041.v1

Factuality

Guardrails for factuality represent the least developed dimension in current research. While output-side guardrails could in principle verify factual claims against trusted knowledge bases or flag low-confidence generations, there is limited recent work on dedicated factuality-focused guardrail mechanisms for LLMs. This gap presents an opportunity for future research to develop fact-checking guardrails that complement retrieval-augmented and decoding-based approaches to factual grounding.

2.3. Decoding

2.3.1. Core Principle: Real-Time Logit and Probability Manipulation

Decoding strategies involve real-time manipulation of the token probability distribution or logits at each generation step to steer the model's outputs toward desired characteristics. At each step, the model produces a probability distribution over its vocabulary conditioned on the preceding tokens, and decoding interventions modify this distribution—by suppressing, amplifying, or re-weighting specific token probabilities—before sampling the next token. This approach allows for fine-grained control over output without modifying the model's parameters, making it computationally efficient and adaptable to diverse trustworthiness requirements.

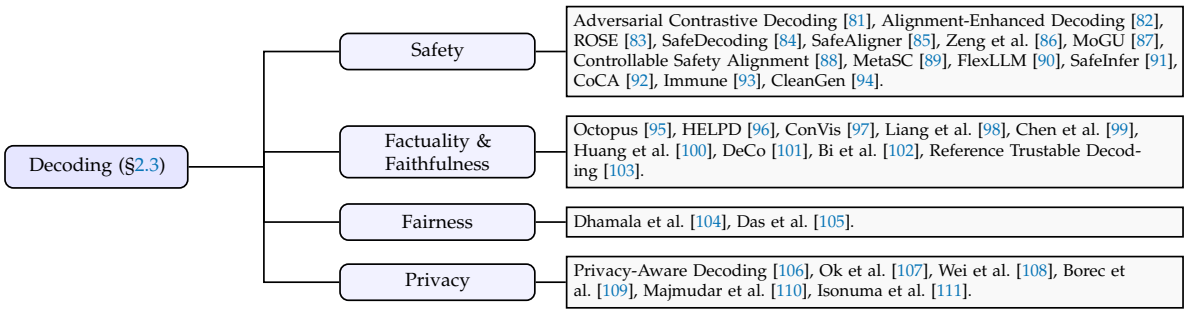

Figure 5. A taxonomy of Decoding Strategies for enhancing LLM trustworthiness.

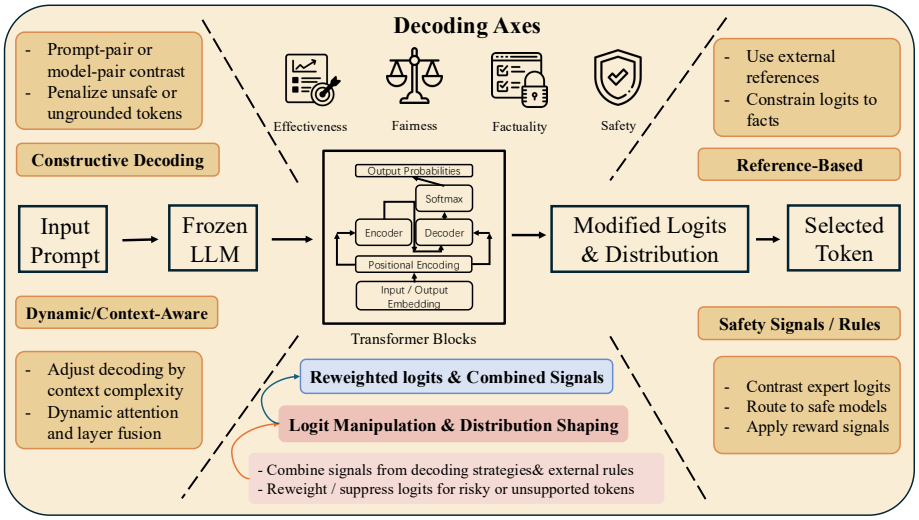

Figure 6. Overview of trustworthiness-oriented decoding strategies. A frozen LLM produces logits that are modified at each autoregressive step by three complementary families of interventions before token sampling. **Contrastive Decoding** (balance icon) contrasts outputs from paired prompts or paired models—e.g., safe versus unsafe prompts, or an expert against an amateur model—to penalize unsafe or ungrounded tokens. **Dynamic/Context-Aware Decoding** (neural-network icon) adjusts decoding hyperparameters, attention weights, or layer fusion on the fly, based on context complexity or intermediate signals. **Reference-Based Decoding** (document-with-shield icon) constrains logits toward facts or safety policies sourced from external references such as retrieved documents or rule sets. These three families feed into a shared *Logit Manipulation & Distribution Shaping* stage, optionally combined with external knowledge or safety signals, which reweights or suppresses risky, unsupported, or unsafe tokens before the next token is sampled.

 doi:10.20944/preprints202605.1041.v1

2.3.2. Applications

Safety

Decoding strategies for safety center on inference-time intervention to prevent harmful outputs without altering model weights, and are crucial for defending against adversarial attacks such as jailbreaks. Contrastive decoding enhances safety by comparing logits from safe and unsafe distributions: methods like Adversarial Contrastive Decoding [81] and Alignment-Enhanced Decoding [82] use opposite or malicious prompts to penalize unsafe continuations, while SafeDecoding [84] and SafeAligner [85] contrast logits from safety-trained expert models against base or unsafe models, and ROSE [83] applies reverse prompt contrastive decoding with step-by-step correction [86]. External guidance approaches use separate models or signals to steer generation, such as MoGU's dynamic routing between a usable and a safe LLM [87], inference-time adaptation through safety configurations [88], and meta-critique optimization of safety specifications [89]. Context-adaptive and dynamic defenses adjust safety measures based on specific inputs, including moving target defense through dynamic decoding hyper-parameters [90] and context-adaptive safety amplification [91]. Specialized defense work addresses safety in multimodal LLMs through output distribution calibration [92] and safety reward model alignment [93], as well as countering backdoor attacks via speculative decoding to identify suspicious tokens [94].

Factuality

Diverse decoding strategies have been developed to mitigate hallucinations and ensure context-aware outputs in LLMs and MLLMs. Contrastive decoding reduces hallucinations by contrasting output distributions derived from original and perturbed inputs: Octopus [95] adapts to input complexity with noise, HELPD [96] employs hierarchical feedback with vision-enhanced penalty decoding, ConVis [97] penalizes hallucinated content via text-to-image reconstruction, and further methods re-balance distributions to prioritize visual grounding [98] or integrate multi-layer fusion with contrastive decoding [99]. Context-aware decoding dynamically adjusts strategies to ensure faithfulness to input context, such as dynamically adjusting attention mechanisms to prioritize context-relevant tokens [100] and adaptively selecting preceding layers to correct output logits [101]. Reference-based decoding enhances factual consistency by contrasting knowledge states to improve confidence on edited facts [102] or using reference information to ensure output consistency without additional training [103].

Privacy

Decoding strategies address privacy concerns by targeting two objectives: reducing privacy leakage and reducing memorization. For leakage prevention, Privacy-Aware Decoding [106] filters outputs in RAG systems to prevent disclosure of sensitive retrieved data, selective teacher supervision during decoding avoids leaking data from untrusted sources [107], and privacy vulnerabilities in speculative decoding have been identified where accelerated inference may increase leakage risks [108]. For memorization reduction, nucleus sampling has been shown to only modestly reduce memorization due to peaked output distributions [109], differential privacy applied during decoding by adding noise to logits provides formal guarantees [110], and contrastive generation encourages novel outputs to reduce reliance on memorized training data [111].

Fairness

While decoding strategies have been explored for safety, factuality, and privacy, their impact on fairness remains a nascent and under-explored domain. Dhamala et al. [104] provide a pioneering analysis of how decoding parameters such as top-k, top-p, and temperature sampling influence fairness in open-ended text generation, demonstrating that specific hyperparameter configurations can exacerbate or mitigate group-level disparities. Similarly, Das et al. [105] explore bias across the entire decoder hyperparameter space, revealing critical trade-offs between bias reduction and generation

 doi:10.20944/preprints202605.1041.v1

quality. These studies highlight fairness as an emerging frontier in decoding methods, where systematic methodologies and robust interventions remain a pressing need.

3. Internal Manipulations

Internal manipulations require white-box access to the model. They intervene directly in the model's computation, for example by modifying activations during a forward pass, selectively removing targeted behaviors or knowledge in context, or pruning architectural components. Compared with external controls, these methods often provide finer-grained and more persistent behavioral changes. In return, they require white-box access to internal representations or model components during generation. In the inference-time pipeline shown in Figure 1, they operate at the LLM generation stage by acting on internal representations or model components.

3.1. Representation Engineering

3.1.1. Core Principle

Representation Engineering (RepE) is an inference-time control paradigm for LLMs that directly modifies internal activations, often in the residual stream at selected layers, to steer high-level behaviors such as refusal [118,120] or truthfulness [122]. Unlike methods that focus on individual neurons or fully specified circuits, RepE treats concepts as directions or low-rank subspaces in population activations and manipulates them directly. This top-down view supports both probing and steering of abstract properties within a single forward pass [135]. In practice, RepE identifies how a concept is encoded in activations, represents it as a direction or subspace, and applies targeted control through activation steering or related interventions. This makes RepE a flexible, efficient, and interpretable way to shape model behavior while largely preserving overall capabilities [136].

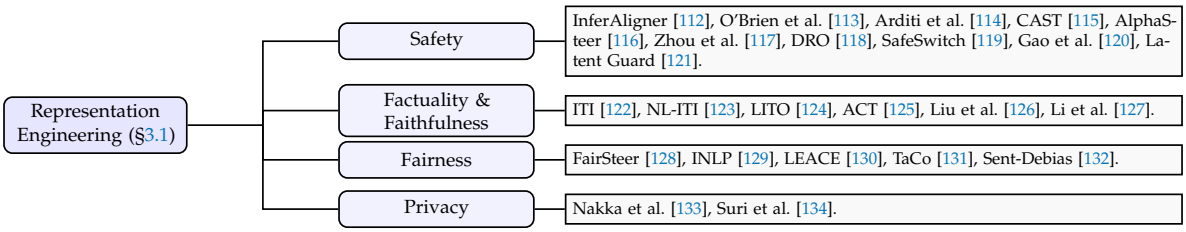

Figure 7. A taxonomy of Representation Engineering for enhancing LLM trustworthiness.

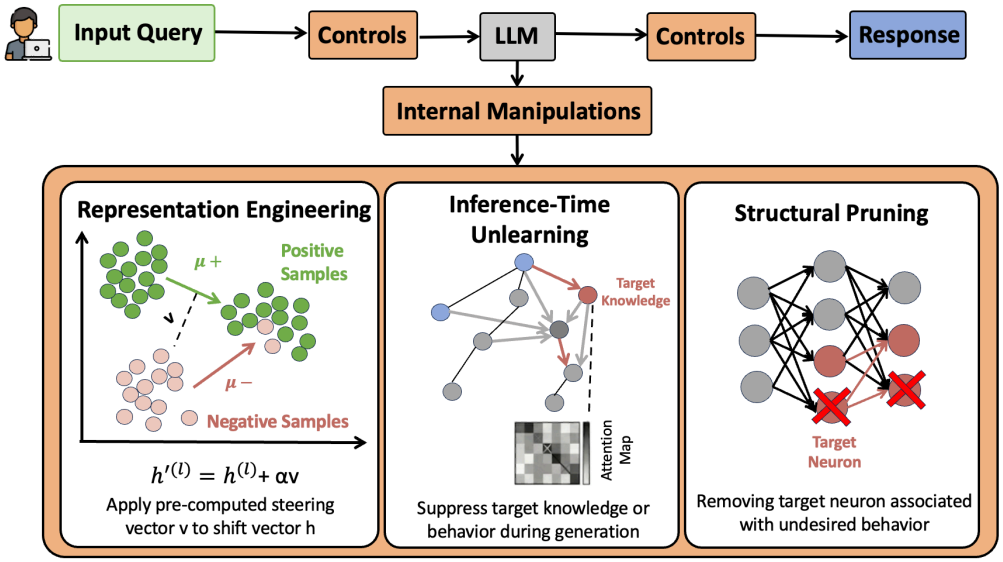

Figure 8. Overview of internal manipulation methods at inference time, including representation engineering, inference-time unlearning, and structure-level interventions such as pruning.

 doi:10.20944/preprints202605.1041.v1

3.1.2. Mechanism

RepE typically works by constructing steering vectors, which are directions in activation space associated with a target concept, and injecting them during inference. This process has two stages: identifying concept-relevant representations and applying them at selected layers.

Let $h^{(\ell)} \in \mathbb{R}^d$ denote the hidden representation at layer ℓ in the residual stream, where d is the representation dimension. Given two prompt sets, X^+ for the target concept and X^- for its contrast, we compute their mean activations at layer ℓ:

$$\mu^+ = \frac{1}{|X^+|} \sum_{x \in X^+} h^{(\ell)}(x), \qquad \mu^- = \frac{1}{|X^-|} \sum_{x \in X^-} h^{(\ell)}(x).$$

The steering vector is then defined as

$$v = \mu^+ - \mu^-,$$

representing the concept as a direction in activation space. More generally, dimensionality reduction methods such as PCA [137], SVD [138], and SAE, as well as optimization-based methods, can be used to extract low-rank subspaces that capture the concept beyond a single direction [139–141].

During inference, the model is steered by modifying the residual activation before it is passed to later layers. For an input x with hidden state $h^{(\ell)}(x)$, the steered representation is

$$\tilde{h}^{(\ell)}(x) = h^{(\ell)}(x) + \alpha v,$$

where $\alpha \in \mathbb{R}$ controls the strength and direction of the intervention. Positive α strengthens the target concept, while negative α suppresses it. This intervention is lightweight because it adds no trainable parameters and only requires a vector addition at inference time. By adjusting α, model behavior can be continuously controlled without retraining or weight updates [140]. Some extensions replace the additive form with affine transformations,

$$\tilde{h}^{(\ell)}(x) = W h^{(\ell)}(x) + b,$$

where (W, b) are low-rank steering operators that provide more expressive but still efficient control [139,141].

3.1.3. Applications

Safety

Representation engineering has become an important tool for safety control at inference time. Existing work mainly uses activation-level steering to modulate refusals, suppress harmful outputs, or reduce false refusals. Some methods inject safety vectors during generation, either through cross-model guidance or harm-specific steering, to improve safety while preserving utility [112]. Others use sparse autoencoders to identify refusal-related features or derive steering directions for detoxification, which improves controllability but may introduce utility trade-offs [113]. Mechanistic studies further suggest that refusal behavior can often be traced to a small set of latent directions or attention components, which enables more targeted interventions such as conditional activation steering and false-refusal ablation [114–117].

This line of work also provides a clearer view of how safety steering interacts with jailbreak robustness. Prior studies show that generic safety prompts often over-shift representations toward refusal, which can increase blanket refusals even when the model can distinguish harmful from benign inputs [118]. In response, later methods study more selective steering strategies and more robust safety control under adversarial prompting [119–121]. Overall, these results show that representation engineering offers a lightweight and increasingly precise toolkit for inference-time safety control.

 doi:10.20944/preprints202605.1041.v1

Fairness

Fairness can also be improved through interventions on internal representations rather than full retraining. In inference-time settings, activation steering uses bias-related directions to modify hidden states and reduce demographic or ideological skew. Recent methods further adapt the intervention layer and strength to improve fairness with limited utility loss [128]. Related work studies concept erasure, which removes protected-attribute information from embeddings or activations through projection or transformation. Representative methods include INLP, LEACE, TaCo, and Sent-Debias [129–132]. Together, these studies suggest that representation-level interventions can help reduce bias without full model retraining.

Privacy

Representation engineering also plays a dual role in privacy, as activation steering can both expose and mitigate leakage. On the attack side, Nakka et al. [133] identify refusal-related attention heads for sensitive attributes and steer a small subset of them at inference time to bypass safeguards, leading to high disclosure rates across several LLMs. On the defense side, Suri et al. [134] show that activation steering can suppress verbatim memorization on a controlled benchmark with limited quality loss. Together, these results suggest that inference-time steering can serve both privacy auditing and mitigation without retraining.

Factuality and Faithfulness

A major line of work in representation engineering steers internal activations at inference time to improve truthfulness without updating model weights. ITI shifts a small set of attention-head activations along truth-related directions and substantially improves TruthfulQA accuracy on Alpaca [122]. NL-ITI extends this idea with multi-token non-linear interventions and reports further gains over ITI [123]. To avoid fixed intervention strengths, LITO learns an instance-specific schedule and can back off through uncertainty-aware refusal, improving truthfulness while preserving task accuracy [124,124]. ACT learns a bank of steering vectors and adapts intervention strength by hallucination category, showing gains across the LLaMA family [125]. Other studies provide a more mechanistic view by identifying global truth-related structure in representation space or finer-grained truth neurons associated with truthful output [126,127].

3.2. Unlearning

3.2.1. Core Principle

LLMs are trained on massive corpora, so controlling what knowledge they retain and express has become increasingly important. At the same time, harmful, private, copyrighted, or otherwise undesirable content may be absorbed during training, raising legal, ethical, and safety concerns. These concerns have made unlearning an important problem for trustworthy LLM deployment. At a high level, unlearning aims to prevent targeted knowledge from influencing model behavior. In inference-time settings, this goal is pursued not by rewriting model parameters, but by intervening during generation to suppress, bypass, or reduce the influence of unwanted knowledge and behaviors. Such methods act as runtime control mechanisms. They limit the model's tendency to express targeted content while preserving its broader capabilities.

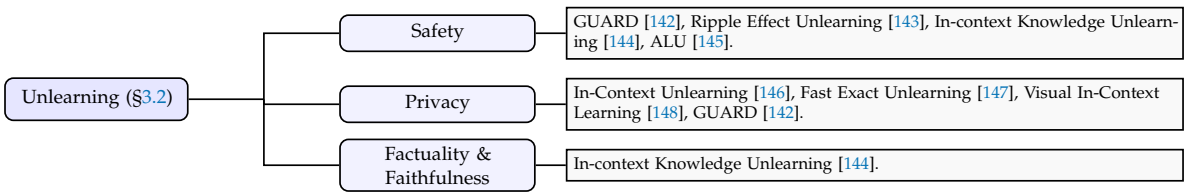

Figure 9. A taxonomy of Unlearning for enhancing LLM trustworthiness.

 doi:10.20944/preprints202605.1041.v1

3.2.2. Mechanisms

Inference-time unlearning methods can be grouped into two main families: *gradient-based methods* and *influence-based methods*. Gradient-based methods apply dynamic penalties or control signals during the forward pass to steer generation away from targeted knowledge. Influence-based methods instead estimate and remove the contribution of specific in-context examples at inference time.

Gradient-Based Methods

Gradient-based unlearning methods suppress targeted knowledge by applying gradient-inspired penalties during the forward pass. Rather than updating model weights, they modify the model's runtime computation to steer generation away from undesirable outputs.

A representative example is **GUARD** [142], which introduces adaptive restriction and detection during generation. By identifying knowledge associated with sensitive concepts and applying penalties in real time, the model is guided away from unsafe continuations while maintaining fluency. Related work also suggests that inference-time forgetting can emerge through runtime control signals that redirect generation toward refusal rather than factual recall [143]. Overall, these methods make unlearning flexible and reversible at deployment time, but they also highlight a central limitation: the targeted knowledge is often suppressed in output rather than fully erased from the model.

Influence-Based Methods

Influence-based methods treat forgetting as the removal or reduction of specific contextual influence at inference time. Rather than steering the model globally, they estimate how particular training or in-context examples contribute to a prediction and then cancel, exclude, or replace that effect during generation. In this sense, they frame unlearning as a counterfactual reasoning problem rather than direct intervention on the forward trajectory.

For example, Pawelczyk et al. [146] introduce In-Context Unlearning, where LLMs are taught to disregard selected in-context examples by relabeling or suppressing their contribution, thereby simulating forgetting within the prompt. Building on this idea, Muresanu et al. [147] propose Fast Exact Unlearning, which efficiently removes the influence of in-context training examples from model predictions without weight updates. Although originally motivated by privacy compliance, these methods show how influence estimation can support inference-time unlearning guarantees. This perspective also extends to multimodal settings. Zhou et al. [148] study unlearning in large vision-language models through visual in-context learning and show that selectively excluding visual exemplars at inference time can reduce leakage of sensitive information. Together, these works show that influence-based unlearning is a clean and flexible inference-time mechanism across both text and multimodal generation.

3.2.3. Applications

Although inference-time unlearning methods differ in mechanism, they share a common goal of reducing the influence of harmful, private, or outdated knowledge during generation. Below, we highlight their applications to privacy, safety, and factuality/faithfulness.

Safety

A key application of unlearning is the suppression of dangerous or toxic capabilities. Because LLMs are trained on large web corpora, they may retain knowledge of illegal activities or unsafe behaviors that can be elicited at deployment time. Gradient-based methods such as GUARD [142] address this problem by applying adaptive restriction and detection during generation, steering the model away from unsafe continuations without retraining. In-context knowledge unlearning [144] uses unlearning tokens to trigger selective refusal in context, enabling dynamic suppression of unsafe or irrelevant knowledge. Ripple Effect Unlearning [143] further shows that suppressing harmful capabilities, such as bomb-making instructions, can reduce jailbreak success but may also propagate to related benign domains. Multi-agent inference frameworks such as ALU [145] extend this line

 doi:10.20944/preprints202605.1041.v1

by formulating safety control through coordinated unlearning-oriented reasoning under diverse jailbreak settings.

Privacy

Privacy is one of the clearest motivations for unlearning, especially under regulations such as the right to be forgotten [149,150]. Because LLMs can memorize sensitive data and disclose it during interaction, inference-time unlearning provides a lightweight way to reduce such leakage without retraining. In-Context Unlearning [146] suppresses the influence of forgotten examples within the prompt, while Fast Exact Unlearning [147] removes the effect of sensitive in-context examples at inference time without changing model parameters. In multimodal settings, Visual In-Context Learning [148] further shows that excluding visual exemplars at inference time can reduce leakage in large vision-language models.

Factuality and Faithfulness

Inference-time unlearning also relates to factuality and faithfulness, especially when models rely on outdated, incorrect, or untrusted knowledge during generation. Instead of recalling such content, unlearning can redirect the model toward abstention or explicit forgetting. In-context knowledge unlearning [144] provides a direct example, showing that unlearning tokens can suppress forgotten content and induce explicit "forgotten" responses rather than unsupported answers. This suggests that inference-time unlearning can improve factual reliability by preventing the model from expressing knowledge that should no longer be trusted. At the same time, this connection remains underexplored, and it is still unclear when runtime suppression should be viewed as genuine forgetting rather than controlled abstention. Recent diagnostic frameworks argue that single-value unlearning metrics obscure exactly this distinction, and propose cognitive-diagnosis-style evaluations that jointly probe retention, removal, and transfer effects across difficulty levels [151].

3.3. Pruning

3.3.1. Core Principle

Pruning has traditionally been studied as a tool for improving efficiency by reducing inference cost, memory footprint, and latency. As foundation models move into safety- and regulation-sensitive settings, this framing is no longer sufficient. The key question is no longer only how to make models smaller or faster, but also how to make them safer, fairer, more privacy-preserving, and more reliable.

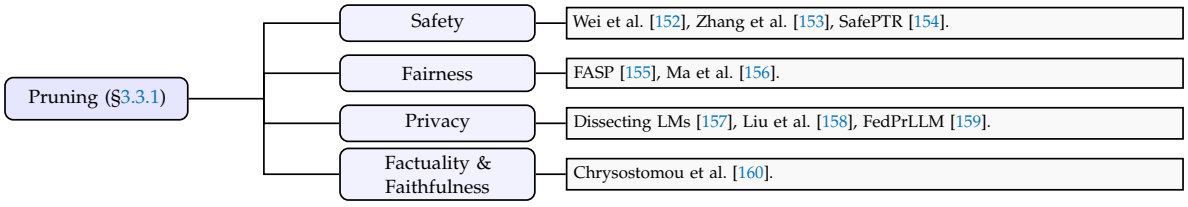

Figure 10. A taxonomy of Pruning for enhancing LLM trustworthiness.

From this perspective, pruning becomes a form of structural control over model behavior rather than a purely efficiency-driven reduction technique. By selectively suppressing weights, neurons, heads, or subnetworks associated with bias amplification, unsafe generation, or privacy leakage, pruning can reshape the pathways through which the model produces outputs. Methods such as fairness-aware sparsification, safety-constrained pruning, and privacy-oriented structural reduction reflect this shift by embedding trust-related objectives into the model's active computation rather than treating them only as post hoc corrections. Under this view, pruning serves not only to improve efficiency, but also to support leaner and more trustworthy deployment.

 doi:10.20944/preprints202605.1041.v1

3.3.2. Applications

Empirical evidence shows that utility metrics alone can miss important trustworthiness shifts after pruning. Compression may preserve perplexity while still degrading refusal behavior, toxicity control, or other safety properties. This means safety should be treated as a first-class constraint in pruning rather than checked only after compression.

Safety

Naive pruning can weaken safety even when utility loss is small. Wei et al. [152] show that removing only ∼3% of parameters, or ∼2.5% of low-rank components, in safety-critical regions can sharply degrade refusal behavior, suggesting that aligned safety depends on structurally sparse components. Zhang et al. [153] further show that efficiency-oriented activation approximation can also introduce safety vulnerabilities, reinforcing the need to account for safety during compression rather than after it. Recent work therefore explores more targeted pruning strategies. In multimodal LLMs, SafePTR [154] selectively prunes harmful tokens at vulnerable layers and restores benign features at later layers, improving jailbreak resistance while preserving utility. Overall, these results suggest that pruning can support safety only when the intervention is explicitly trust-aware. Otherwise, compression may remove precisely the components that sustain aligned behavior.

Fairness

Fairness-oriented pruning is less clearly aligned with inference-time trustworthiness than safety-oriented runtime pruning. Existing work mainly studies post-processing structural interventions, such as pruning attention heads associated with bias or comparing head- and neuron-level pruning for bias mitigation [155,156]. These results suggest that selective structural suppression can reduce bias, but they are better viewed as adjacent pruning-based debiasing methods than as clean inference-time interventions. As a result, fairness remains relatively underexplored for pruning under the stricter inference-time setting considered here.

Privacy

Pruning has also been explored for privacy protection, though much of this work is closer to deployment-oriented structural intervention than to strict runtime control during generation. Existing methods identify and remove neurons or subnetworks associated with memorized sensitive content, aiming to suppress privacy leakage while preserving general utility. Examples include data-efficient neuron pruning in language models, modality-aware pruning in MLLMs, and federated pruning schemes that exchange pruning masks rather than raw data [157–159]. These results suggest that pruning can support privacy-preserving deployment, but most current methods rely on structural modification before deployment rather than conditional pruning at inference time.

Factuality and Faithfulness

Pruning has also shown some potential for improving factual reliability, although evidence remains limited. One large summarization study reports that pruned models hallucinate less and rely more on source content, suggesting that pruning may suppress spurious generation pathways [160]. However, it is still unclear whether this effect transfers beyond summarization or supports stronger notions of faithfulness. At present, factuality and faithfulness remain promising but underexplored directions for pruning in trustworthy model deployment.

4. System-Level Orchestration

System-level orchestration moves beyond single-model interventions to coordinate multiple LLM agents into collaborative architectures. Rather than controlling one model in isolation, trustworthiness here emerges from structured interaction patterns—debate, cross-verification, role specialization, and iterative self-correction. In the inference-time pipeline (Figure 1), the tool-use / agents loop spans context assembly, generation, and output checking, creating feedback cycles that enable collective reliability.

 doi:10.20944/preprints202605.1041.v1

4.1. Multi-Agent Systems

Multi-agent systems (MAS) shift control from a single model to a coordinated group of LLM-based agents. Trustworthiness is no longer a property enforced on one model; it emerges from the interaction protocols that connect the agents. By structuring those interactions through debate, adversarial simulation, or multi-perspective deliberation, MAS can reach levels of robustness, safety, and factuality that single agents struggle to attain.

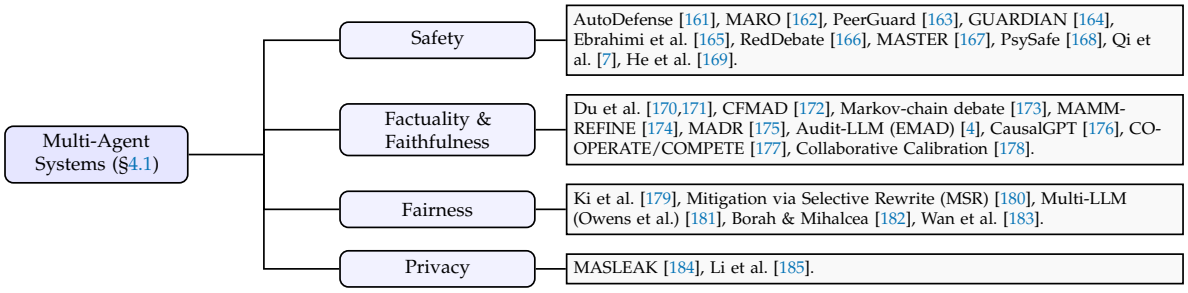

Figure 11. A taxonomy of Multi-Agent Systems for enhancing LLM trustworthiness.

4.1.1. Core Principle: From Monolithic Control to Collaborative Systems

Multi-agent systems decentralize reasoning and decision-making. A complex problem is decomposed and assigned to specialized agents, each with its own role, persona, or expertise. Trustworthiness is shaped by the interaction protocol, which may use adversarial debate, cooperative problem-solving, peer review, or hierarchical verification. For example, a "propose" agent generates an initial response, a "critic" agent challenges its assumptions, and a "synthesize" agent integrates the feedback into a final answer. This structured interaction adds error correction and validation at inference time, so reliability depends less on any single agent's quality and more on the collaborative architecture itself.

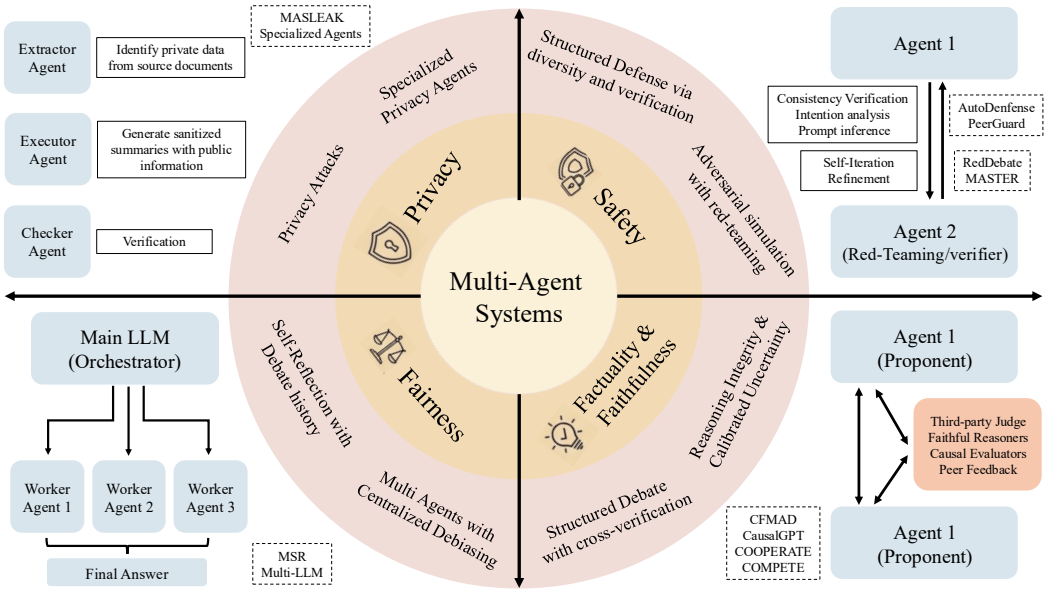

Figure 12. Overview of multi-agent system architectures for LLM trustworthiness across four dimensions. Safety employs structured defense via role diversity and adversarial red-teaming; Factuality relies on structured debate with cross-verification and calibrated uncertainty; Fairness uses multi-agent debiasing with self-reflection as well as orchestrated plan-and-refine loops; and Privacy leverages specialized agents (Extractor, Executor, Checker) for data sanitization. Representative systems and interaction patterns are shown for each dimension.

4.1.2. Applications

Safety

Multi-agent systems make safety an emergent property of *structured interaction* rather than a static property of a single model. Specialized agents run a propose–attack–audit–adjudicate loop: they gen-

 doi:10.20944/preprints202605.1041.v1

erate responses, probe for failures, cross-check reasoning, and arbitrate outcomes. A unifying pattern is *structured defense via role diversity and cross-verification*. AutoDefense combines intention analysis, prompt inference, and final judgment to filter unsafe outputs [161]. Committee-style analyzers extend coverage by combining linguistic, comment, and fact-checking expertise for misinformation risks [162]. Peer-based scrutiny hardens internal logic: PeerGuard has agents verify consistency between each other's chain-of-thought and final outputs to expose backdoors [163]. At system scale, collaboration can be modeled as a temporal interaction graph to trace and contain the spread of hallucinations or injected errors [164], while dynamic credibility weighting down-ranks unreliable contributors [165]. Safety also benefits from *adversarial simulation for proactive discovery*: automated red-teaming with debate iteratively elicits, refines, and patches unsafe behaviors using persistent memory [166]. The same interactive channels, however, expand the attack surface. Structured debates can amplify jailbreak success [7], communication links invite Agent-in-the-Middle manipulation [169], and persuasive agents can steer group consensus toward unsafe actions [168]; taxonomies such as MASTER map these vulnerabilities [167], and analyses of *emergent social risks* show how misaligned agent interactions can produce system-level harms that no individual agent was designed to cause [186].

Factuality

Multi-agent systems enhance factuality and faithfulness by replacing a single model's unverified reasoning with structured interaction among multiple agents. Rather than trusting one model's first-pass answer, MAS organize a propose–critique–adjudicate loop in which agents advance claims, attack each other's reasoning, and converge under a judge. Foundational work shows that debating agents can reach more accurate conclusions than individuals, as critiques expose faulty logic that a single model may miss [170]; debates judged by weaker raters still tend to favor truthful arguments [171]. Forced-disagreement variants such as CFMAD assign counterfactual stances to improve robustness to initial mistakes [172]. This pattern scales to long-form tasks by decomposing outputs into atomic claims and verifying each through coordinated agents: role-structured stateful debates [173], discriminative reranking pipelines [174], complementary-perspective feedback for explanation faithfulness [175], and domain-specific evidence cross-examination [4]. Beyond surface correctness, MAS also promote reasoning integrity by separating faithful reasoners from causal evaluators [176], and yield calibrated uncertainty through peer feedback and adversarially generated alternatives that reveal knowledge gaps and trigger abstention [177,178].

Fairness

Multi-agent systems address demographic biases that simpler mitigation methods leave intact. One direction uses multi-agent debate for cultural adaptability: Ki et al. [179] propose a framework where LLM agents handle culturally situated questions through debate and self-reflection, with a judge LLM resolving disagreements based on debate history. For demographic fairness, multi-agent architectures decompose the debiasing task across specialized roles: Mitigation via Selective Rewrite uses an LLM agent to give targeted feedback on removing gender bias in language style [180]; multi-LLM frameworks place multiple models in conversational settings with centralized and decentralized debiasing configurations [181]; and self-reflection mechanisms let models identify and self-correct societal biases through a critique step [182]. Most recently, Wan et al. [183] proposes an agentic planning pipeline that runs an iterated diagnose-and-refine loop until the output is satisfactory.

Privacy

Multi-agent systems introduce mechanisms for safety and factuality, but they also create new privacy challenges that remain a research gap. The interconnected structure of MAS opens new vectors for data leakage: the MASLEAK framework shows systematic, worm-like attacks that extract intellectual property such as system prompts, tool definitions, and topological structure from black-box MAS applications by propagating through agent communication chains [184]. The same architectural principles can also be used for defense: multi-agent approaches decompose privacy preservation into

 doi:10.20944/preprints202605.1041.v1

specialized roles, such as an Extractor Agent to identify private data, an Executor Agent to generate sanitized outputs, and a Checker Agent for verification, reducing leakage of sensitive information while preserving output quality [185]. Together, these studies show that the collaborative structure of MAS is both a liability and an asset for privacy. A unified treatment that combines robust defenses, secure communication protocols, and verifiable information-flow controls remains largely unexplored.

5. Evaluation

Inference-time trustworthiness methods intervene after model training to shape LLM behavior under deployment constraints. As these interventions differ a lot in where and how they act, such as in prompts, logits, activations, external filters, or the system workflow, evaluation cannot be unified by one benchmark or one metric family. Although foundational benchmarks such as TrustGPT [187] and dynamic evaluation toolkits such as TrustEval [188] provide valuable unified views over multiple trustworthiness dimensions, they were not designed to isolate the runtime-specific properties introduced by inference-time controls. We therefore adopt a meta-axis evaluation view on the inference-time evaluation. The goal is to capture deployment-relevant properties that arise specifically from runtime trustworthiness. In practice, such enforcement is shaped by latency, cost, and integration constraints. Each axis highlights a distinct evaluation question, and while the axes are conceptually unique, they can overlap in practice:

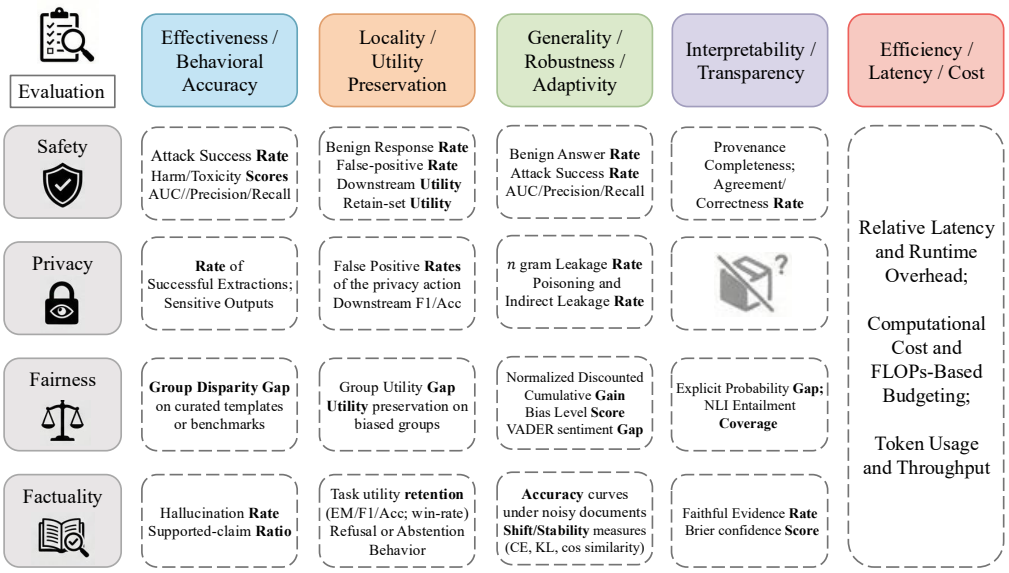

Figure 13. A meta-axis evaluation framework for inference-time trustworthiness methods. Rows correspond to four trustworthiness dimensions (Safety, Privacy, Fairness, Factuality), and columns represent five complementary evaluation axes (Effectiveness, Locality, Generality, Interpretability, Efficiency). Each cell lists representative metrics specific to the dimension–axis intersection, highlighting the multi-faceted nature of deployment-time evaluation.

- **Effectiveness / Behavioral Accuracy**: To what extent does the inference-time mechanism reliably enforce the intended behavioral constraints, and under what conditions does it fail?
- **Locality / Utility Preservation**: To what extent are the intervention effects localized to the targeted behaviors while preserving general task performance and user utility?
- **Generality / Robustness / Adaptivity**: How well does the mechanism generalize across tasks, domains, and input distributions, and how robust is it to distributional shifts?
- **Efficiency / Latency / Cost**: What additional runtime overhead does the mechanism introduce in terms of latency, compute, memory, or token usage?
- **Interpretability / Transparency**: To what extent are the mechanism's actions, triggers, and decision rationales transparent to human auditors or downstream systems?

 doi:10.20944/preprints202605.1041.v1

Although these axes separate different deployment concerns, they often trade off against one another. Building on this framing, this section organizes evaluation along the axes above and describes representative metrics across four categories: Safety, Privacy, Fairness, and Factuality.

5.1. Effectiveness / Behavioral Accuracy

Effectiveness / Behavioral Accuracy evaluates whether an inference-time mechanism $\mathcal{M}$ changes model behavior in the intended direction on a target evaluation distribution. Consider an LLM that produces an output $Y_{\mathcal{M}}(x)$ for an input $x \sim \mathcal{D}$ under $\mathcal{M}$. Effectiveness asks whether $\mathcal{M}$ increases the likelihood of desired trust-aligned behavior and decreases the likelihood of undesired behavior:

$$\text{Effectiveness}(\mathcal{M}) \propto \mathbb{E}_{x\sim\mathcal{D}}\Big[\mathbf{1}\{Y_{\mathcal{M}}(x) \in \mathcal{Y}_{\text{desired}}\} - \mathbf{1}\{Y_{\mathcal{M}}(x) \in \mathcal{Y}_{\text{undesired}}\}\Big].$$

In inference-time settings, this axis primarily measures the runtime behavior induced by a plug-in control layer, such as refusal, redaction, constrained decoding, evidence grounding, or verification. However, Effectiveness alone does not tell whether improvements come from overly broad suppression of benign behavior, from brittle operating thresholds, or from masking the underlying capability. These trade-offs motivate complementary axes such as Locality (utility and over-refusal), Robustness (bypassability and shift), Efficiency (latency/cost), and Interpretability (auditability).

5.1.1. Rate-Based Metrics

Rate-based metrics capture how often undesired behavior still occurs at inference time, and are the most prevalent family across all trustworthiness categories. In safety, jailbreak success rate measures whether an adversarial prompt elicits a harmful response from the model, unsafe completion rate tracks the overall proportion of harmful outputs across all queries, and the complementary defense success rate captures how often the mechanism successfully blocks such outputs. Risk-score formulations take a continuous toxicity or harm score assigned to each output instead of a binary attack outcome and report the fraction exceeding a predefined threshold [19,84,85,142,154,163,189,190]. In privacy, extraction success rate captures whether an attacker can recover memorized or sensitive text from model outputs, while disclosure rate measures how often outputs contain personally identifying or otherwise private information across a broader query distribution [30,106,109,146]. In factuality, hallucination rate counts the proportion of outputs that are unsupported or factually incorrect, groundedness rate measures the fraction of claims in a response that are explicitly supported by retrieved or provided evidence, and edit success rate verifies whether a targeted fact has been correctly updated following a knowledge editing intervention [39,45,48,191,192].

5.1.2. Classifier-Style Metrics

Classifier-style metrics treat the inference-time intervention as an explicit detection task and evaluate how well the mechanism separates harmful, private, or hallucinated outputs from benign ones. Rather than reporting a single operating point, metrics such as AUC and AUPRC summarize performance across all possible decision thresholds, while Precision, Recall, and F1 show the trade-off between false positives and false negatives at a chosen threshold. These metrics are important when the intervention involves a learned or rule-based classifier, such as a safety guardrail or hallucination detector, where threshold calibration governs deployment behavior. They appear across safety guardrails [1–3,60], privacy membership-inference evaluations [30,133], and factuality hallucination detection [45,48].

5.1.3. Group-Disparity Metrics

Group-disparity metrics quantify fairness effectiveness by comparing outcomes across demographic groups, capturing whether a mechanism treats different populations equitably. Bias scores on curated benchmarks measure stereotyping or differential toxicity for specific groups, worst-group performance identifies the most disadvantaged subpopulation, and disparity gaps capture the abso-

 doi:10.20944/preprints202605.1041.v1

lute difference in an outcome metric such as toxicity, error rate, or refusal rate between two groups. Together, these metrics reveal whether a fairness intervention reduces imbalance in model behavior or merely shifts it. They are reported across prompt-based interventions [32,33,38], fairness-aware retrieval [35,36], and pruning methods that target bias-related internal structure [155,156,193].

5.1.4. Task-Performance Metrics

Task-performance metrics such as exact match, F1, and accuracy measure whether trustworthiness gains come at the cost of general capability, serving as a sanity check on standard benchmarks. They are widely reported among representation or context engineering evaluations to confirm that they do not disrupt unrelated competencies [122,124,125,191], and also appear in multimodal hallucination settings [96,97].

5.2. Locality / Utility Preservation

Locality / Utility Preservation evaluates whether an inference-time mechanism $\mathcal{M}$ limits its behavioral changes to the intended outputs while preserving user utility. Unlike training-time alignment, inference-time controls are typically implemented as plug-in layers. As a result, the locality failure mode is behavioral spillover, in which the system becomes safer by being less helpful, issuing more refusals, or providing less information. Formally, let $\mathcal{D}_{\text{tgt}}$ denote the targeted distribution (e.g., harmful or sensitive prompts) and $\mathcal{D}_{\text{benign}}$ denote benign traffic. A generic utility retention view is:

$$\Delta U_{\text{benign}}(\mathcal{M}) = \mathbb{E}_{x \sim \mathcal{D}_{\text{benign}}}[u(Y_{\mathcal{M}}(x)) - u(Y(x))],$$

where $u(\cdot)$ can be task accuracy (EM/F1/Acc), LLM-as-judge helpfulness, or win-rate. In inference-time deployment, locality is coupled to thresholding and pipeline composition. When multiple strategies are combined, such as guardrails and decoding, they can amplify suppression of benign outputs even if each component appears acceptable in isolation.

5.2.1. False-Positive and Over-Refusal Rate Metrics

These metrics capture whether a mechanism incorrectly suppresses or refuses inputs it was not designed to target. In safety, benign response rate measures how often the defense leaves safe prompts unaffected, and is usually reported alongside attack success rate to confirm that refusal gains are not simply the result of refusing more broadly [19–22]. In guardrail systems, false-positive rate and benign blocking rate capture how often clean inputs are incorrectly filtered or routed. These summarize the trade-off between blocking harmful content and passing benign traffic [1–3,60]. In multi-agent defenses, false alarm rates on clean interactions are essential because a monitoring agent that incorrectly flags benign exchanges can suppress the entire downstream workflow [163,166]. In privacy, false positives of the privacy action through blocking harmless queries, measure damage on non-sensitive content [106,146]. In fairness, refusal disparity measures the absolute difference in refusal rate between demographic groups under the intervention, capturing whether the mechanism disproportionately suppresses responses for certain groups rather than improving equitable helpfulness [32,33,38].

5.2.2. Utility Retention Metrics

Utility retention metrics measure whether general task performance is preserved on non-targeted inputs after the intervention. Common metrics include exact match, F1, and accuracy on held-out benchmarks, MT-Bench scores, instruction-following rates, LLM-as-judge helpfulness, and win-rate. In safety, these are reported for decoding, representation engineering, pruning, and unlearning interventions to confirm that modifications do not distort normal generation quality [84,114,142,143,154,189]. In privacy, downstream task accuracy on non-sensitive queries confirms that filtering or constrained generation does not broadly reduce answer quality [30,109]. In fairness, per-group utility retention tracks whether task success is preserved equitably across demographic groups, ensuring bias reduction is not obtained by

 doi:10.20944/preprints202605.1041.v1

broadly degrading helpfulness for any subpopulation [36,155,156,193]. In factuality, retain-set task utility on benign factual QA is reported alongside hallucination reductions [39,45,48,122].

5.3. Generality / Robustness / Adaptivity

Generality / Robustness / Adaptivity evaluate whether an inference time mechanism $\mathcal{M}$ keeps its intended effect when inputs, tasks, or attackers change at deployment. This axis is not about a single test set. It is about performance over a family of conditions, which is likely out-of-distribution. Let $\{\mathcal{D}_k\}_{k=1}^{K}$ be test distributions and let $s_k(\mathcal{M})$ be a score on $\mathcal{D}_k$. A common summary reports both the average and the worst case:

$$s_{\text{avg}}(\mathcal{M}) = \frac{1}{K}\sum_{k=1}^{K} s_k(\mathcal{M}), \qquad s_{\min}(\mathcal{M}) = \min_{k\in[K]} s_k(\mathcal{M}).$$

5.3.1. Worst-Case and Cross-Distribution Rate Metrics

These metrics summarize how a mechanism's core effectiveness signal holds up across multiple conditions rather than a single fixed distribution. In safety, attack success rate and benign answering rate are reported across multiple jailbreak families and prompt styles, with worst-case values serving as the primary robustness summary. Sensitivity to threshold changes is captured by reporting deltas when guard strength varies [19–21]. Decoding and representation engineering methods report harmful score and safety score under stronger or more diverse attacks. Tracking utility accuracy on task suites to confirm that robustness gains do not come at the cost of normal performance [84,112,142]. In privacy, attack success rate is reported across multiple attack types and privacy budgets to capture how leakage behavior changes as the adversary changes. N-gram leakage rate measures whether the control generalizes beyond exact span matches to broader overlap with sensitive content [29,31]. In fairness, bias scores and stereotype index are reported across demographic categories, topics, and occupations to evaluate whether bias reduction holds beyond a fixed template. Fairness violation rates are tracked alongside utility metrics such as NDCG [32,33,38]. In factuality, accuracy and F1 are reported across cross-domain splits to test whether robustness comes from better evidence selection rather than longer contexts [39,45,46,48].

5.3.2. Classifier-Style Metrics Across Domains and Languages

When the intervention involves an explicit detector or classifier, robustness is evaluated by reporting precision, recall, F1, AUC, and AUPRC across multiple benchmarks, domains, and languages. In safety, guardrails and multi-agent systems use these metrics to evaluate whether detection quality generalizes across domains and languages [2,62,166]. In privacy, precision, recall, and F1 for detecting poisoned passages evaluate whether the control generalizes to different poisoning strategies, since both false positives and false negatives matter [25,30]. In fairness, detailed metrics such as sentiment gap, toxicity scores, and regard-based labels are reported across demographic groups to evaluate how robust the fairness conclusion is to different measurement approaches [155,194].

5.3.3. Calibration and Stability Metrics

These metrics evaluate whether a mechanism remains consistent and reliable as conditions shift. In safety, expected calibration error tests whether the same decision threshold remains reliable under distribution shift. In factuality, representation engineering methods track shift measures such as cross-entropy and KL divergence to detect distribution change, and stability measures such as cosine similarity to check whether the intervention direction stays consistent as training data changes [122,124,135]. In fairness, cross-split and cross-capability comparisons, reporting generality scores on train-test splits, check whether bias reductions are stable rather than artifacts of a particular evaluation setup [193,195].

 doi:10.20944/preprints202605.1041.v1

5.4. Interpretability / Transparency

Interpretability / Transparency evaluates whether an inference-time mechanism $\mathcal{M}$ is observable and auditable. When $\mathcal{M}$ changes the system behavior (e.g., refusal, redaction, or rewriting), it should be clear (i) what action was taken, (ii) what condition triggered the action, and (iii) what evidence supports the decision. This is particularly important at inference time because many controls are deployed as modular plug-ins. Formally, let $a_{\mathcal{M}}(x) \in \mathcal{A}$ denote the runtime action on input x, and let $\tau_{\mathcal{M}}(x)$ be the trigger set (policy IDs, detectors, or threshold crossings). Let $z_{\mathcal{M}}(x)$ be the transparency artifact (logs, scores, or traces). A generic transparency score is:

$$T(\mathcal{M}) = \mathbb{E}_{x\sim\mathcal{D}}[v(a_{\mathcal{M}}(x), \tau_{\mathcal{M}}(x), z_{\mathcal{M}}(x))],$$

where $s(x)$ is a risk score (or classifier confidence) and t is the deployed threshold.

5.4.1. Provenance and Audit Metrics

These metrics measure whether each intervention is accompanied by a sufficient audit record that identifies what component acted, what policy triggered it, and at what operating point. In safety, provenance completeness captures whether each intervention includes a stage identifier, policy identifier, risk score, and threshold value, making it possible to distinguish targeted filtering from broad over-refusal [3,4,154,164]. In factuality, provenance logs of what was retrieved and used at runtime help separate retrieval failures from generation failures, and verification loops or critique-style retrieval augmentation leave intermediate records that make the reasoning process auditable [39,45,46].

5.4.2. Human Judgment Metrics

These metrics evaluate whether the mechanism's explanations are understandable to human annotators. In safety, agreement rate measures the fraction of explanations that annotators judge as correct or helpful, evaluating whether intermediate reasoning steps and policy triggers are meaningful rather than superficially plausible [22,65]. In factuality, faithful explanation rate measures the fraction of explanations that humans consider consistent with the provided evidence, evaluating whether verification rationales are genuinely grounded [45,175]. LLM-as-a-judge is widely used as a scalable proxy for human judgment, but recent work shows that judge models themselves carry systematic biases—e.g., position, verbosity, and self-preference biases—which can distort transparency evaluations and should be reported alongside raw judge scores [196].

5.4.3. Distributional Transparency Metrics

These metrics evaluate whether the mechanism's behavior across groups or confidence levels is auditable from the output distribution. In fairness, an explicit probability gap between two matched identity terms in the same context measures how much the mechanism shifts likelihoods across groups [193]. In pluralistic settings, NLI entailment coverage measures how much the final response reflects provided group viewpoints, capturing whether the mechanism genuinely incorporates intended perspectives [195]. In factuality, calibration metrics such as the Brier score measure whether reported confidence aligns with actual correctness, making the system's factuality claims auditable across varying operating points [171,178].

5.5. Efficiency / Latency / Cost

Efficiency / Latency / Cost evaluates how practical it is to deploy an inference-time mechanism $\mathcal{M}$ by measuring the computational resources consumed during the control process. Unlike training-time alignment, which spreads costs over the model's lifetime, inference-time methods operate on the critical path of user interaction. Even small increases in latency or compute per request can noticeably worsen user experience and raise operational costs. The main efficiency failure mode is excessive overhead, where the mechanism requires many extra forward passes, memory, or auxiliary calls

 doi:10.20944/preprints202605.1041.v1

to achieve safety or accuracy improvements. Formally, let $C_{\text{base}}(x)$ denote the cost of generating a response for input x using the base model. A general overhead measure is:

$$\Delta C(\mathcal{M}) = \mathbb{E}_{x\sim\mathcal{D}}[C_{\mathcal{M}}(x) - C_{\text{base}}(x)],$$

where $C_{\mathcal{M}}(x)$ is the cost under the mechanism. The commonly used metrics are:

- **Relative Latency and Runtime Overhead.** A standard way to measure deployment cost is relative latency, often reported as the *Average Token Generation Time Ratio* or its equivalence [82,84,94]. It measures the slowdown caused by the defense during inference:

$$\text{ATGR} = \frac{1}{N}\sum_{i=1}^{N}\frac{T_{\text{def}}(x_i)}{T_{\text{base}}(x_i)},$$

  where $T_{\text{def}}(x_i)$ is the wall-clock time for prompt x_i under the defense, and $T_{\text{base}}(x_i)$ is the baseline time.
- **Computational Cost and FLOPs-Based Budgeting.** To study scaling behavior in a way that is less sensitive to hardware, several works use floating point operations as a compute proxy [27,63]. This gives a hardware-independent view of how much extra computation is needed for a given safety gain, often reported as performance under a compute budget:

$$\text{Performance}(C_{\text{budget}}) = \text{Metric}\Big(\mathcal{M} \mid C_{\mathcal{M}} \le C_{\text{budget}}\Big).$$

  This is used to plot safety outcomes such as Attack Success Rate against inference TFLOPs, tracing a safety–compute trade-off curve.
- **Token Usage and Throughput.** For methods that expand context—such as guardrail prompting, meta-prompting, or retrieval augmentation—token growth is a direct cost driver [36,74,86]. A simple and widely used summary is the *Token Usage Ratio*:

$$r_{TT} = \frac{\text{Total Tokens}_{\text{defense}}}{\text{Total Tokens}_{\text{base}}}.$$

  This captures cost increases from longer prompts, extra intermediate steps, or additional retrieved content. Throughput in tokens per second is also commonly reported to reflect serving capacity under the defense.

6. Discussion and Open Problems

6.1. External Controls

Context engineering, guardrails, and decoding strategies share a common strength: they treat the model as a black box, making them model-agnostic, composable, and rapidly updatable without retraining. Context engineering is training-free and flexible, with different techniques addressing distinct challenges: prompting for lightweight alignment, RAG for factual grounding, memory for persistence, and reasoning for self-regulation. Guardrails add an independent policy-enforcement layer that can be swapped across providers, while decoding strategies enable fine-grained, real-time steering of token distributions across multiple trustworthiness dimensions.

However, all three tiers are soft controls and share similar failure modes. Context engineering is fragile to small phrasing changes and vulnerable to context poisoning; guardrails face a continuous cat-and-mouse dynamic against adversarial evasion; and hard decoding constraints can produce brittle, incoherent outputs [85]. A common "Control Tax" runs through these methods: strict guardrails increase false positives and reduce helpfulness, while aggressive logit suppression [81] or noise injection [110] compromises generation quality. LLM-based judges and multi-layer decoding also add latency overhead. None of these methods can provide formal safety guarantees in isolation, and integrated pipelines that combine context engineering, guardrails, and decoding remain underexplored.

 doi:10.20944/preprints202605.1041.v1

6.2. Internal Manipulations

Representation engineering, unlearning, and pruning offer more direct behavioral control by intervening on internal model components. Representation engineering treats hidden states as controllable objects and supports concept-level steering through activation addition or sparse autoencoders [122,135,140]. Unlearning steers outputs away from sensitive knowledge through runtime penalties or influence subtraction, which is practical when regulatory requirements shift. Pruning provides structural control by removing heads, neurons, or tokens associated with unsafe or biased behavior, and is compatible with broader compression pipelines.

Despite these strengths, internal methods share several fragilities. Representation engineering depends heavily on intervention strength, layer choice, and normalization; multi-concept steering remains brittle, and strong interventions risk pushing activations out of distribution [136]. Unlearning faces a verification problem: many methods suppress rather than truly remove knowledge, leaving models vulnerable to adversarial recovery [5]. Pruning is highly sensitive to where sparsity is introduced; even small amounts of naive pruning can sharply weaken safety-critical behavior. All three methods exhibit a utility–control trade-off, where stronger interventions can degrade fluency, accuracy, or reasoning [197]. Representation engineering also carries dual-use risks, as the same activation-level controls can strengthen or weaken safety [198]. Robust deployment across internal methods will require better verification protocols, principled intervention selection, and careful integration with broader safety governance [199].

6.3. System-Level Orchestration

Multi-agent systems introduce reliability through collaborative reasoning, but their most immediate obstacle is operational overhead: each query may trigger multiple agents over several rounds, with performance gains often diminishing quickly [170,172], which leaves many architectures impractical for real-time applications [4]. The collaborative dynamics designed to foster trustworthiness can also become vectors for emergent failures. Agents sharing the same base model exhibit "groupthink," reinforcing shared hallucinations [172], and adversaries can exploit the collaborative structure: a single persuasive adversary can sway an entire group [200], while debate frameworks can be more vulnerable to jailbreaks than single agents [7]. Evaluating MAS trustworthiness is also more complex, as effectiveness depends on roles, topology, protocols, and agent count [167].

7. Conclusions

As large language models are deployed in real applications, controlling their trustworthiness after training has become as important as alignment during training. In this paper, we have presented a unified framework for inference-time control of trustworthy LLM behavior across safety, privacy, fairness, and factuality. The framework spans external guardrails, internal representation engineering, and system-level orchestration. Our analysis shows that no single method is sufficient; each trades off control strength, computational cost, and model utility against the others. Building reliable systems will therefore require composing these methods rather than choosing among them. Inference-time intervention is not an afterthought to alignment but a coherent design space in its own right, and treating it as such is a starting point for verifiably trustworthy LLMs.

References

1. Chi, J.; Karn, U.; Zhan, H.; Smith, E.M.; Rando, J.; Zhang, Y.; Plawiak, K.; Coudert, Z.D.; Upasani, K.; Pasupuleti, M. Llama Guard 3 Vision: Safeguarding Human-AI Image Understanding Conversations. *ArXiv* **2024**, *abs/2411.10414*.
2. Padhi, I.; Nagireddy, M.; Cornacchia, G.; Chaudhury, S.; Pedapati, T.; Dognin, P.L.; Murugesan, K.; Miehling, E.; Cooper, M.S.; Fraser, K.; et al. Granite Guardian. *ArXiv* **2024**, *abs/2412.07724*.
3. Liu, Y.; Gao, H.; Zhai, S.; Xia, J.; Wu, T.; Xue, Z.; Chen, Y.; Kawaguchi, K.; Zhang, J.; Hooi, B. GuardReasoner: Towards Reasoning-based LLM Safeguards. *ArXiv* **2025**, *abs/2501.18492*.

 doi:10.20944/preprints202605.1041.v1

4. Song, C.; Ma, L.; Zheng, J.; Liao, J.; Kuang, H.; Yang, L. Audit-LLM: Multi-Agent Collaboration for Log-based Insider Threat Detection, 2024, [arXiv:cs.CR/2408.08902].
5. Łucki, J.; Wei, B.; Huang, Y.; Henderson, P.; Tramèr, F.; Rando, J. An adversarial perspective on machine unlearning for ai safety. *arXiv preprint arXiv:2409.18025* **2024**.
6. Dou, G.; Liu, Z.; Lyu, Q.; Ding, K.; Wong, E. Avoiding Copyright Infringement via Large Language Model Unlearning. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2025, 2025, pp. 5176–5200.
7. Qi, S.; Zou, Y.; Li, P.; Lin, Z.; Cheng, X.; Yu, D. Amplified Vulnerabilities: Structured Jailbreak Attacks on LLM-based Multi-Agent Debate, 2025, [arXiv:cs.CR/2504.16489].
8. Donisch, L.; Schacht, S.; Lanquillon, C. Inference optimizations for large language models: Effects, challenges, and practical considerations. *arXiv preprint arXiv:2408.03130* **2024**.
9. Liu, J.; Tang, P.; Wang, W.; Ren, Y.; Hou, X.; Heng, P.A.; Guo, M.; Li, C. A survey on inference optimization techniques for mixture of experts models. *arXiv preprint arXiv:2412.14219* **2024**.
10. Zhou, Z.; Ning, X.; Hong, K.; Fu, T.; Xu, J.; Li, S.; Lou, Y.; Wang, L.; Yuan, Z.; Li, X.; et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294* **2024**.
11. Tie, G.; Zhao, Z.; Song, D.; Wei, F.; Zhou, R.; Dai, Y.; Yin, W.; Yang, Z.; Yan, J.; Su, Y.; et al. Large Language Models Post-training: Surveying Techniques from Alignment to Reasoning. *arXiv arXiv:2503.06072* **2025**.
12. Kumar, K.; Ashraf, T.; Thawakar, O.; Anwer, R.M.; Cholakkal, H.; Shah, M.; Yang, M.H.; Torr, P.H.; Khan, F.S.; Khan, S. Llm post-training: A deep dive into reasoning large language models. *arXiv preprint arXiv:2502.21321* **2025**.
13. Huang, Y.; Sun, L.; Wang, H.; Wu, S.; Zhang, Q.; Li, Y.; Gao, C.; Huang, Y.; Lyu, W.; Zhang, Y.; et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561* **2024**.
14. Liu, Y.; Yao, Y.; Ton, J.F.; Zhang, X.; Guo, R.; Cheng, H.; Klochkov, Y.; Taufiq, M.F.; Li, H. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374* **2023**.
15. Wan, Y.; Subramonian, A.; Ovalle, A.; Lin, Z.; Suvarna, A.; Chance, C.; Bansal, H.; Pattichis, R.; Chang, K.W. Survey of bias in text-to-image generation: Definition, evaluation, and mitigation. *arXiv preprint arXiv:2404.01030* **2024**.
16. Yu, M.; Meng, F.; Zhou, X.; Wang, S.; Mao, J.; Pang, L.; Chen, T.; Wang, K.; Li, X.; Zhang, Y.; et al. A survey on trustworthy llm agents: Threats and countermeasures. *arXiv preprint arXiv:2503.09648* **2025**.
17. Liu, Z.; Dou, G.; Tan, Z.; Tian, Y.; Jiang, M. Machine unlearning in generative ai: A survey. *arXiv preprint arXiv:2407.20516* **2024**.
18. Barez, F.; Fu, T.; Prabhu, A.; Casper, S.; Sanyal, A.; Bibi, A.; O'Gara, A.; Kirk, R.; Bucknall, B.; Fist, T.; et al. Open problems in machine unlearning for ai safety. *URL https://arxiv. org/abs/2501.04952* **2025**.
19. Kumar, A.; Agarwal, C.; Srinivas, S.; Li, A.J.; Feizi, S.; Lakkaraju, H. Certifying LLM Safety against Adversarial Prompting. In Proceedings of the First Conference on Language Modeling, 2024.
20. Cao, B.; Cao, Y.; Lin, L.; Chen, J. Defending Against Alignment-Breaking Attacks via Robustly Aligned LLM. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024, 2024, pp. 10542–10560.
21. Zhou, Y.; Han, Y.; Zhuang, H.; Guo, K.; Liang, Z.; Bao, H.; Zhang, X. Defending Jailbreak Prompts via In-Context Adversarial Game. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 20084–20105.
22. Ji, J.; Hou, B.; Robey, A.; Pappas, G.J.; Hassani, H.; Zhang, Y.; Wong, E.; Chang, S. Defending large language models against jailbreak attacks via semantic smoothing. *arXiv preprint arXiv:2402.16192* **2024**.
23. Wei, Z.; Wang, Y.; Li, A.; Mo, Y.; Wang, Y. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387* **2023**.
24. Hong, G.; Kim, J.; Kang, J.; Myaeng, S.H.; Whang, J.J. Why So Gullible? Enhancing the Robustness of Retrieval-Augmented Models against Counterfactual Noise. In Proceedings of the 2024 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2024.
25. Zhong, Z.; Huang, Z.; Wettig, A.; Chen, D. Poisoning Retrieval Corpora by Injecting Adversarial Passages. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023, pp. 13764–13775.
26. Li, J.J.; Pyatkin, V.; Kleiman-Weiner, M.; Jiang, L.; Dziri, N.; Collins, A.; Borg, J.S.; Sap, M.; Choi, Y.; Levine, S. SafetyAnalyst: Interpretable, Transparent, and Steerable Safety Moderation for AI Behavior. In Proceedings of the Forty-second International Conference on Machine Learning, 2025.

 doi:10.20944/preprints202605.1041.v1

27. Qiu, R.; Li, G.; Wei, T.; He, J.; Tong, H. Saffron-1: Towards an Inference Scaling Paradigm for LLM Safety Assurance. *arXiv preprint arXiv:2506.06444* **2025**.
28. Wu, Y.; Wen, R.; Backes, M.; Berrang, P.; Humbert, M.; Shen, Y.; Zhang, Y. Quantifying privacy risks of prompts in visual prompt learning. In Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24), 2024, pp. 5841–5858.
29. Chaudhari, H.; Severi, G.; Abascal, J.; Jagielski, M.; Choquette-Choo, C.A.; Nasr, M.; Nita-Rotaru, C.; Oprea, A. Phantom: General trigger attacks on retrieval augmented language generation. *arXiv preprint arXiv:2405.20485* **2024**.
30. Zou, W.; Geng, R.; Wang, B.; Jia, J. PoisonedRAG: Knowledge Corruption Attacks to Retrieval-Augmented Generation of Large Language Models. In Proceedings of the 34th USENIX Security Symposium (USENIX Security 25), 2025.
31. Tong, M.; Chen, K.; Zhang, J.; Qi, Y.; Zhang, W.; Yu, N.; Zhang, T.; Zhang, Z. Inferdpt: Privacy-preserving inference for black-box large language models. *IEEE Transactions on Dependable and Secure Computing* **2025**.
32. Dwivedi, S.; Ghosh, S.; Dwivedi, S. Breaking the bias: Gender fairness in llms using prompt engineering and in-context learning. *Rupkatha Journal on Interdisciplinary Studies in Humanities* **2023**, *15*.
33. Huang, D.; Zhang, J.M.; Bu, Q.; Xie, X.; Chen, J.; Cui, H. Bias testing and mitigation in llm-based code generation. *ACM Transactions on Software Engineering and Methodology* **2024**.
34. Fayyazi, A.; Kamal, M.; Pedram, M. FACTER: Fairness-Aware Conformal Thresholding and Prompt Engineering for Enabling Fair LLM-Based Recommender Systems. In Proceedings of the Forty-second International Conference on Machine Learning, 2025.
35. Chen, Q.Z.; Feng, K.; Park, C.Y.; Zhang, A.X. Spica: Retrieving scenarios for pluralistic in-context alignment. *arXiv preprint arXiv:2411.10912* **2024**.
36. Shrestha, R.; Zou, Y.; Chen, Q.; Li, Z.; Xie, Y.; Deng, S. Fairrag: Fair human generation via fair retrieval augmentation. In Proceedings of the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 11996–12005.
37. Dhingra, H.; Jayashanker, P.; Moghe, S.; Strubell, E. Queer people are people first: Deconstructing sexual identity stereotypes in large language models. *arXiv preprint arXiv:2307.00101* **2023**.
38. Kamruzzaman, M.; Kim, G.L. Prompting techniques for reducing social bias in llms through system 1 and system 2 cognitive processes. *arXiv preprint arXiv:2404.17218* **2024**.
39. Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-Verification Reduces Hallucination in Large Language Models. *FINDINGS OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: ACL 2024* **2024**, pp. 3563–3578.
40. Press, O.; Zhang, M.; Min, S.; Schmidt, L.; Smith, N.A.; Lewis, M. Measuring and Narrowing the Compositionality Gap in Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 5687–5711.
41. Li, M.; Wang, W.; Feng, F.; Zhu, F.; Wang, Q.; Chua, T.S. Think Twice Before Trusting: Self-Detection for Large Language Models through Comprehensive Answer Reflection. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 11858–11875.
42. Feng, S.; Shi, W.; Wang, Y.; Ding, W.; Ahia, O.; Li, S.S.; Balachandran, V.; Sitaram, S.; Tsvetkov, Y. Teaching LLMs to Abstain across Languages via Multilingual Feedback. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, 2024, pp. 4125–4150.
43. Zhou, W.; Zhang, S.; Poon, H.; Chen, M. Context-faithful Prompting for Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, 2023, pp. 14544–14556.
44. Shi, W.; Min, S.; Yasunaga, M.; Seo, M.; James, R.; Lewis, M.; Zettlemoyer, L.; Yih, W.t. REPLUG: Retrieval-Augmented Black-Box Language Models. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024, pp. 8364–8377.
45. Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.
46. Wan, H.; Feng, S.; Tan, Z.; Wang, H.; Tsvetkov, Y.; Luo, M. DELL: Generating Reactions and Explanations for LLM-Based Misinformation Detection. In Proceedings of the ACL (Findings), 2024.
47. Yu, W.; Iter, D.; Wang, S.; Xu, Y.; Ju, M.; Sanyal, S.; Zhu, C.; Zeng, M.; Jiang, M. Generate rather than Retrieve: Large Language Models are Strong Context Generators. In Proceedings of the International Conference on Learning Representations, 2023.

 doi:10.20944/preprints202605.1041.v1

48. Zhou, Y.; Liu, Z.; Jin, J.; Nie, J.Y.; Dou, Z. Metacognitive retrieval-augmented large language models. In Proceedings of the Proceedings of the ACM Web Conference 2024, 2024, pp. 1453–1463.
49. Chen, M.; Li, Y.; Padthe, K.; Shao, R.; Sun, A.; Zettlemoyer, L.; Ghosh, G.; Yih, W.t. Improving factuality with explicit working memory. *arXiv preprint arXiv:2412.18069* **2024**.
50. Wang, W.; Dong, L.; Cheng, H.; Liu, X.; Yan, X.; Gao, J.; Wei, F. Augmenting language models with long-term memory. *Advances in Neural Information Processing Systems* **2023**, *36*, 74530–74543.
51. Rebedea, T.; Dinu, R.L.; Sreedhar, M.N.; Parisien, C.; Cohen, J. NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2023.
52. hana Chennabasappa, S.; Nikolaidis, C.; Song, D.; Molnar, D.; Ding, S.; Wan, S.; Whitman, S.; Deason, L.; Doucette, N.; Montilla, A.; et al. LlamaFirewall: An open source guardrail system for building secure AI agents. *ArXiv* **2025**, *abs/2505.03574*.
53. Han, S.; Avestimehr, A.S.; He, C. Bridging the Safety Gap: A Guardrail Pipeline for Trustworthy LLM Inferences. *ArXiv* **2025**, *abs/2502.08142*.
54. Wang, X.; Ji, Z.; Wang, W.; Li, Z.; Wu, D.; Wang, S. SoK: Evaluating Jailbreak Guardrails for Large Language Models. *ArXiv* **2025**, *abs/2506.10597*.
55. Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; et al. Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations. *ArXiv* **2023**, *abs/2312.06674*.
56. Sharma, M.; Tong, M.; Mu, J.; Wei, J.; Kruthoff, J.; Goodfriend, S.; Ong, E.; Peng, A.; Agarwal, R.; Anil, C.; et al. Constitutional Classifiers: Defending against Universal Jailbreaks across Thousands of Hours of Red Teaming, 2025, [arXiv:cs.CL/2501.18837].
57. Han, S.; Rao, K.; Ettinger, A.; Jiang, L.; Lin, B.Y.; Lambert, N.; Choi, Y.; Dziri, N. WildGuard: Open One-Stop Moderation Tools for Safety Risks, Jailbreaks, and Refusals of LLMs, 2024, [arXiv:cs.CL/2406.18495].
58. Zeng, W.; Liu, Y.; Mullins, R.; Peran, L.; Fernandez, J.; Harkous, H.; Narasimhan, K.; Proud, D.; Kumar, P.; Radharapu, B.; et al. ShieldGemma: Generative AI Content Moderation Based on Gemma. *ArXiv* **2024**, *abs/2407.21772*.
59. Zeng, W.; Kurniawan, D.; Mullins, R.; Liu, Y.; Saha, T.; Ike-Njoku, D.; Gu, J.; Song, Y.; Xu, C.; Zhou, J.; et al. ShieldGemma 2: Robust and Tractable Image Content Moderation. *ArXiv* **2025**, *abs/2504.01081*.
60. Kumar, P.; Jain, D.; Yerukola, A.; Jiang, L.; Beniwal, H.; Hartvigsen, T.; Sap, M. PolyGuard: A Multilingual Safety Moderation Tool for 17 Languages. *ArXiv* **2025**, *abs/2504.04377*.
61. Upadhayay, B.; Behzadan, P.V.; Wang, M.; Lin, P.; Cai, S.; An, S.; Ma, S.; Lin, Z.; Huang, C.; Wei, A.; et al. X-Guard: Multilingual Guard Agent for Content Moderation. *ArXiv* **2025**, *abs/2504.08848*.
62. Ghosh, S.; Varshney, P.; Galinkin, E.; Parisien, C. AEGIS: Online Adaptive AI Content Safety Moderation with Ensemble of LLM Experts. *ArXiv* **2024**, *abs/2404.05993*.
63. Lee, S.; Lee, D.B.; Wagner, D.; Kang, M.; Seong, H.; Bocklet, T.; Lee, J.; Hwang, S.J. SafeRoute: Adaptive Model Selection for Efficient and Accurate Safety Guardrails in Large Language Models. *ArXiv* **2025**, *abs/2502.12464*.
64. Zheng, L.; Chiang, W.L.; Sheng, Y.; Zhuang, S.; Wu, Z.; Zhuang, Y.; Lin, Z.; Li, Z.; Li, D.; Xing, E.P.; et al. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. *ArXiv* **2023**, *abs/2306.05685*.
65. Kang, M.; Li, B. R2-Guard: Robust Reasoning Enabled LLM Guardrail via Knowledge-Enhanced Logical Reasoning. *ArXiv* **2024**, *abs/2407.05557*.
66. Zheng, J.; Ji, X.; Lu, Y.; Cui, C.; Zhao, W.; Deng, G.; Liang, Z.; Zhang, A.; Chua, T.S. RSafe: Incentivizing proactive reasoning to build robust and adaptive LLM safeguards. *ArXiv* **2025**, *abs/2506.07736*.
67. Kokkula, S.; Somanathan, R.; Nandavardhan, R.; Aashishkumar.; Divya, G. Palisade - Prompt Injection Detection Framework. *ArXiv* **2024**, *abs/2410.21146*.
68. LangChain AI. Rebuff: Prompt Injection Detection for LLM Applications. https://blog.langchain.com/rebuff/, 2023. Accessed: 2025-08-18.
69. Microsoft. Prompt Shields in Azure AI Content Safety. Microsoft Learn (online), 2025. Accessed via Microsoft Learn documentation: Unified API to detect and block adversarial user-input attacks on LLMs.
70. Meta. Llama Prompt Guard documentation. Meta (online), 2025. Model card and prompt-format documentation for Llama Prompt Guard, available online at Meta's official site.
71. Zou, A.; Wang, Z.; Kolter, J.Z.; Fredrikson, M. Universal and Transferable Adversarial Attacks on Aligned Language Models. *ArXiv* **2023**, *abs/2307.15043*.

 doi:10.20944/preprints202605.1041.v1

72. Jiang, F.; Xu, Z.; Niu, L.; Xiang, Z.; Ramasubramanian, B.; Li, B.; Poovendran, R. ArtPrompt: ASCII Art-based Jailbreak Attacks against Aligned LLMs. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2024.
73. Hackett, W.; Birch, L.; Trawicki, S.; Suri, N.; Garraghan, P. Bypassing Prompt Injection and Jailbreak Detection in LLM Guardrails. *arXiv preprint arXiv:2504.11168* **2025**.
74. Jiang, F.; Xu, Z.; Niu, L.; Wang, B.; Jia, J.; Li, B.; Poovendran, R. POSTER: Identifying and Mitigating Vulnerabilities in LLM-Integrated Applications. *Proceedings of the 19th ACM Asia Conference on Computer and Communications Security* **2023**.
75. Davidson, T.; Warmsley, D.; Macy, M.W.; Weber, I. Automated Hate Speech Detection and the Problem of Offensive Language. In Proceedings of the International Conference on Web and Social Media, 2017.
76. Borkan, D.; Dixon, L.; Sorensen, J.S.; Thain, N.; Vasserman, L. Nuanced Metrics for Measuring Unintended Bias with Real Data for Text Classification. *Companion Proceedings of The 2019 World Wide Web Conference* **2019**.
77. Mathew, B.; Saha, P.; Yimam, S.M.; Biemann, C.; Goyal, P.; Mukherjee, A. HateXplain: A Benchmark Dataset for Explainable Hate Speech Detection. In Proceedings of the AAAI Conference on Artificial Intelligence, 2020.
78. Hartvigsen, T.; Gabriel, S.; Palangi, H.; Sap, M.; Ray, D.; Kamar, E. ToxiGen: A Large-Scale Machine-Generated Dataset for Adversarial and Implicit Hate Speech Detection. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2022.
79. Lees, A.; Tran, V.Q.; Tay, Y.; Sorensen, J.S.; Gupta, J.; Metzler, D.; Vasserman, L. A New Generation of Perspective API: Efficient Multilingual Character-level Transformers. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* **2022**.
80. Huang, Y.; Zhuang, H.; Ye, J.; Bao, H.; Wang, Y.; Hua, H.; Wu, S.; Chen, P.Y.; Zhang, X. Guardian-as-an-Advisor: Advancing Next-Generation Guardian Models for Trustworthy LLMs. *arXiv preprint arXiv:2604.07655* **2026**.
81. Zhao, Z.; Zhang, X.; Xu, K.; Hu, X.; Zhang, R.; Du, Z.; Guo, Q.; Chen, Y. Adversarial Contrastive Decoding: Boosting Safety Alignment of Large Language Models via Opposite Prompt Optimization. *ArXiv* **2024**, *abs/2406.16743*.
82. Liu, Q.; Zhou, Z.; He, L.; Liu, Y.; Zhang, W.; Su, S. Alignment-Enhanced Decoding: Defending Jailbreaks via Token-Level Adaptive Refining of Probability Distributions. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024.
83. Zhong, Q.; Ding, L.; Liu, J.; Du, B.; Tao, D. ROSE Doesn't Do That: Boosting the Safety of Instruction-Tuned Large Language Models with Reverse Prompt Contrastive Decoding. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2024.
84. Xu, Z.; Jiang, F.; Niu, L.; Jia, J.; Lin, B.Y.; Poovendran, R. SafeDecoding: Defending against Jailbreak Attacks via Safety-Aware Decoding. *ArXiv* **2024**, *abs/2402.08983*.
85. Huang, C.; Zhao, W.; Zheng, R.; Lv, H.; Dou, S.; Li, S.; Wang, X.; Zhou, E.; Ye, J.; Yang, Y.; et al. SafeAligner: Safety Alignment against Jailbreak Attacks via Response Disparity Guidance. *ArXiv* **2024**, *abs/2406.18118*.
86. Zeng, X.; Shang, Y.; Zhu, Y.; Chen, J.; Tian, Y. Root Defence Strategies: Ensuring Safety of LLM at the Decoding Level. *ArXiv* **2024**, *abs/2410.06809*.
87. Du, Y.; Zhao, S.; Zhao, D.; Ma, M.; Chen, Y.; Huo, L.; Yang, Q.; Xu, D.; Qin, B. MoGU: A Framework for Enhancing Safety of Open-Sourced LLMs While Preserving Their Usability. *ArXiv* **2024**, *abs/2405.14488*.
88. Zhang, J.J.; Elgohary, A.; Magooda, A.; Khashabi, D.; Durme, B.V. Controllable Safety Alignment: Inference-Time Adaptation to Diverse Safety Requirements. *ArXiv* **2024**, *abs/2410.08968*.
89. Gallego, V. MetaSC: Test-Time Safety Specification Optimization for Language Models. *ArXiv* **2025**, *abs/2502.07985*.
90. Chen, B.; Guo, H.; Yan, Q. FlexLLM: Exploring LLM Customization for Moving Target Defense on Black-Box LLMs Against Jailbreak Attacks. *ArXiv* **2024**, *abs/2412.07672*.
91. Banerjee, S.; Tripathy, S.; Layek, S.; Kumar, S.; Mukherjee, A.; Hazra, R. SafeInfer: Context Adaptive Decoding Time Safety Alignment for Large Language Models. In Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
92. Gao, J.; Pi, R.; Han, T.; Wu, H.; Hong, L.; Kong, L.; Jiang, X.; Li, Z. CoCA: Regaining Safety-awareness of Multimodal Large Language Models with Constitutional Calibration. *ArXiv* **2024**, *abs/2409.11365*.
93. Ghosal, S.S.; Chakraborty, S.; Singh, V.; Guan, T.; Wang, M.; Beirami, A.; Huang, F.; Velasquez, A.; Manocha, D.; Bedi, A.S. Immune: Improving Safety Against Jailbreaks in Multi-modal LLMs via Inference-Time Alignment. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2024**, pp. 25038–25049.

 doi:10.20944/preprints202605.1041.v1

94. Li, Y.; Xu, Z.; Jiang, F.; Niu, L.; Sahabandu, D.; Ramasubramanian, B.; Poovendran, R. CleanGen: Mitigating Backdoor Attacks for Generation Tasks in Large Language Models. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2024.
95. Suo, W.; Zhang, L.; Sun, M.; Wu, L.Y.; Wang, P.; Zhang, Y. Octopus: Alleviating Hallucination via Dynamic Contrastive Decoding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* **2025**, pp. 29904–29914.
96. Yuan, F.; Qin, C.; Xu, X.; Li, P. HELPD: Mitigating Hallucination of LVLMs by Hierarchical Feedback Learning with Vision-enhanced Penalty Decoding. *ArXiv* **2024**, *abs/2409.20429*.
97. Park, Y.; Lee, D.; Choe, J.; Chang, B. ConVis: Contrastive Decoding with Hallucination Visualization for Mitigating Hallucinations in Multimodal Large Language Models. In Proceedings of the AAAI Conference on Artificial Intelligence, 2024.
98. Liang, X.; Yu, J.; Mu, L.; Zhuang, J.; Hu, J.; Yang, Y.; Ye, J.; Lu, L.; Chen, J.; Hu, H. Mitigating Hallucination in Visual-Language Models via Re-Balancing Contrastive Decoding. In Proceedings of the Chinese Conference on Pattern Recognition and Computer Vision, 2024.
99. Chen, D.; Fang, F.; Ni, S.; Liang, F.; Hu, X.; Argha, A.; Alinejad-Rokny, H.; Yang, M.; Li, C. Lower Layers Matter: Alleviating Hallucination via Multi-Layer Fusion Contrastive Decoding with Truthfulness Refocused. 2024.
100. Huang, Y.; Zhang, Y.; Cheng, N.; Li, Z.; Wang, S.; Xiao, J. Dynamic Attention-Guided Context Decoding for Mitigating Context Faithfulness Hallucinations in Large Language Models. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2025.
101. Wang, C.; Chen, X.; Zhang, N.; Tian, B.; Xu, H.; Deng, S.; Chen, H. MLLM can see? Dynamic Correction Decoding for Hallucination Mitigation. *ArXiv* **2024**, *abs/2410.11779*.
102. Bi, B.; Liu, S.; Mei, L.; Wang, Y.; Ji, P.; Cheng, X. Decoding by Contrasting Knowledge: Enhancing LLMs' Confidence on Edited Facts. *ArXiv* **2024**, *abs/2405.11613*.
103. Shi, L.; Yao, Y.; Li, Z.; Zhang, L.; Zhao, H. Reference Trustable Decoding: A Training-Free Augmentation Paradigm for Large Language Models. *ArXiv* **2024**, *abs/2409.20181*.
104. Dhamala, J.; Kumar, V.; Gupta, R.; Chang, K.W.; Galstyan, A.G. An Analysis of The Effects of Decoding Algorithms on Fairness in Open-Ended Language Generation. *2022 IEEE Spoken Language Technology Workshop (SLT)* **2022**, pp. 655–662.
105. Das, M.; Balke, W.T. Quantifying Bias from Decoding Techniques in Natural Language Generation. In Proceedings of the International Conference on Computational Linguistics, 2022.
106. Wang, H.; Xu, X.; Huang, B.; Shu, K. Privacy-Aware Decoding: Mitigating Privacy Leakage of Large Language Models in Retrieval-Augmented Generation. 2025.
107. Ok, H.; Ryu, J.; Lee, J. Decoding with Limited Teacher Supervision Requires Understanding When to Trust the Teacher. *ArXiv* **2024**, *abs/2406.18002*.
108. Wei, J.; Abdulrazzag, A.; Zhang, T.; Muursepp, A.; Saileshwar, G. Privacy Risks of Speculative Decoding in Large Language Models. *ArXiv* **2024**, *abs/2411.01076*.
109. Borec, L.; Sadler, P.; Schlangen, D. The Unreasonable Ineffectiveness of Nucleus Sampling on Mitigating Text Memorization. In Proceedings of the International Conference on Natural Language Generation, 2024.
110. Majmudar, J.; Dupuy, C.; Peris, C.; Smaili, S.; Gupta, R.; Zemel, R.S. Differentially Private Decoding in Large Language Models. *ArXiv* **2022**, *abs/2205.13621*.
111. Isonuma, M.; Titov, I. What's New in My Data? Novelty Exploration via Contrastive Generation. *ArXiv* **2024**, *abs/2410.14765*.
112. Wang, P.; Zhang, D.; Li, L.; Tan, C.; Wang, X.; Ren, K.; Jiang, B.; Qiu, X. InferAligner: Inference-Time Alignment for Harmlessness through Cross-Model Guidance. *arXiv preprint arXiv:2401.11206* **2024**.
113. O'Brien, K.; Majercak, D.; Fernandes, X.; Edgar, R.; Bullwinkel, B.; Chen, J.; Nori, H.; Carignan, D.; Horvitz, E.; Poursabzi-Sangdeh, F. Steering Language Model Refusal with Sparse Autoencoders. *arXiv preprint arXiv:2411.11296* **2024**.
114. Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Panickssery, N.; Gurnee, W.; Nanda, N. Refusal in Language Models Is Mediated by a Single Direction. *arXiv preprint arXiv:2406.11717* **2024**.
115. Lee, B.W.; Padhi, I.; Natesan Ramamurthy, K.; Miehling, E.; Dognin, P.; Nagireddy, M.; Dhurandhar, A. Programming Refusal with Conditional Activation Steering. *arXiv preprint arXiv:2409.05907* **2024**.
116. Sheng, L.; Shen, C.; Zhao, W.; Fang, J.; Liu, X.; Liang, Z.; Wang, X.; Zhang, A.; Chua, T.S. AlphaSteer: Learning Refusal Steering with Principled Null-Space Constraint. *arXiv preprint arXiv:2506.07022* **2025**.
117. Zhou, Z.; Yu, H.; Zhang, X.; Xu, R.; Huang, F.; Wang, K.; Liu, Y.; Fang, J.; Li, Y. On the role of attention heads in large language model safety. *arXiv preprint arXiv:2410.13708* **2024**.

 doi:10.20944/preprints202605.1041.v1

118. Zheng, C.; Yin, F.; Zhou, H.; Meng, F.; Zhou, J.; Chang, K.W.; Huang, M.; Peng, N. On prompt-driven safeguarding for large language models. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning, 2024, pp. 61593–61613.
119. Han, P.; Qian, C.; Chen, X.; Zhang, Y.; Zhang, D.; Ji, H. SafeSwitch: Steering Unsafe LLM Behavior via Internal Activation Signals. *arXiv preprint arXiv:2502.01042* **2025**.
120. Gao, L.; Geng, J.; Zhang, X.; Nakov, P.; Chen, X. Shaping the Safety Boundaries: Understanding and Defending Against Jailbreaks in Large Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 25378–25398.
121. Zhao, J.; Huang, J.; Wu, Z.; Bau, D.; Shi, W. LLMs Encode Harmfulness and Refusal Separately. *arXiv preprint arXiv:2507.11878* **2025**.
122. Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-Time Intervention: Eliciting Truthful Answers from a Language Model, 2024, [arXiv:cs.LG/2306.03341].
123. Hościłowicz, J.; Wiacek, A.; Chojnacki, J.; Cieślak, A.; Michon, L.; Urbanevych, V.; Janicki, A. Non-Linear Inference Time Intervention: Improving LLM Truthfulness, 2024, [arXiv:cs.CL/2403.18680].
124. Bayat, F.F.; Liu, X.; Jagadish, H.V.; Wang, L. Enhanced Language Model Truthfulness with Learnable Intervention and Uncertainty Expression. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, 2024.
125. Wang, T.; Jiao, X.; Zhu, Y.; Chen, Z.; He, Y.; Chu, X.; Gao, J.; Wang, Y.; Ma, L. Adaptive Activation Steering: A Tuning-Free LLM Truthfulness Improvement Method for Diverse Hallucinations Categories, 2024, [arXiv:cs.CL/2406.00034]. Also appears at TheWebConf (WWW) 2025.
126. Liu, J.; Chen, S.; Cheng, Y.; He, J. On the Universal Truthfulness Hyperplane Inside LLMs. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2024.
127. Li, H.; Cao, Y.; Yu, Y.; Suchow, J.W.; Zhu, Z. Truth Neurons, 2025, [arXiv:cs.CL/2505.12182].
128. Li, Y.; Fan, Z.; Chen, R.; Gai, X.; Gong, L.; Zhang, Y.; Liu, Z. FairSteer: Inference-Time Debiasing for LLMs with Dynamic Activation Steering. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 11293–11312.
129. Ravfogel, S.; Elazar, Y.; Gonen, H.; Twiton, M.; Goldberg, Y. Null It Out: Guarding Protected Attributes by Iterative Nullspace Projection. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7237–7256.
130. Belrose, N.; Schneider-Joseph, D.; Ravfogel, S.; Cotterell, R.; Raff, E.; Biderman, S. LEACE: Perfect Linear Concept Erasure in Closed Form. *arXiv preprint arXiv:2306.03819* **2023**.
131. Jourdan, F.; Béthune, L.; Picard, A.; Risser, L.; Asher, N. TaCo: Targeted Concept Erasure Prevents Non-Linear Classifiers From Detecting Protected Attributes. *arXiv preprint arXiv:2312.06499* **2024**.
132. Liang, P.P.; Li, I.M.; Zheng, E.; Lim, Y.C.; Salakhutdinov, R.; Morency, L.P. Towards Debiasing Sentence Representations. In Proceedings of the Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020.
133. Nakka, K.K.; Jiang, X.; Usynin, D.; Zhou, X. PII Jailbreaking in LLMs via Activation Steering Reveals Personal Information Leakage, 2025, [arXiv:cs.CR/2507.02332]. Preprint.
134. Suri, M.; Anand, N.; Bhaskar, A. Mitigating Memorization in LLMs using Activation Steering, 2025, [arXiv:cs.CL/2503.06040]. Preprint.
135. Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.K.; et al. Representation engineering: A top-down approach to ai transparency. *arXiv arXiv:2310.01405* **2023**.
136. Wehner, J.; Abdelnabi, S.; Tan, D.; Krueger, D.; Fritz, M. Taxonomy, opportunities, and challenges of representation engineering for large language models. *arXiv preprint arXiv:2502.19649* **2025**.
137. Jolliffe, I.T. *Principal Component Analysis*; Springer Verlag, 1986.
138. Golub, G.H.; Reinsch, C. Singular value decomposition and least squares solutions. *Linear Algebra* **1971**, pp. 134–151.
139. Jiang, X.; Zhang, L.; Zhang, J.; Yang, Q.; Hu, G.; Wang, D.; Hu, L. MSRS: Adaptive Multi-Subspace Representation Steering for Attribute Alignment in Large Language Models. *arXiv arXiv:2508.10599* **2025**.
140. Turner, A.M.; Thiergart, L.; Leech, G.; Udell, D.; Vazquez, J.J.; Mini, U.; MacDiarmid, M. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248* **2023**.
141. Zhou, A. Compositional Subspace Representation Fine-tuning for Adaptive Large Language Models. *arXiv preprint arXiv:2503.10617* **2025**.
142. Deng, Z.; Liu, C.Y.; Pang, Z.; He, X.; Feng, L.; Xuan, Q.; Zhu, Z.; Wei, J. Guard: Generation-time llm unlearning via adaptive restriction and detection. *arXiv preprint arXiv:2505.13312* **2025**.

 doi:10.20944/preprints202605.1041.v1

143. Zhang, Z.; Yang, J.; Lu, Y.; Ke, P.; Cui, S.; Zheng, C.; Wang, H.; Huang, M. From Theft to Bomb-Making: The Ripple Effect of Unlearning in Defending Against Jailbreak Attacks. *arXiv preprint arXiv:2407.02855* **2024**.
144. Takashiro, S.; Kojima, T.; Gambardella, A.; Cao, Q.; Iwasawa, Y.; Matsuo, Y. Answer When Needed, Forget When Not: Language Models Pretend to Forget via In-Context Knowledge Unlearning, 2025, [arXiv:cs.CL/2410.00382].
145. Sanyal, D.; Mandal, M. Agents are all you need for LLM unlearning. *arXiv preprint arXiv:2502.00406* **2025**.
146. Pawelczyk, M.; Neel, S.; Lakkaraju, H. In-context unlearning: Language models as few shot unlearners. *arXiv preprint arXiv:2310.07579* **2023**.
147. Muresanu, A.I.; Thudi, A.; Zhang, M.R.; Papernot, N. Fast Exact Unlearning for In-Context Learning Data for LLMs. In Proceedings of the ICML, 2025.
148. Zhou, Y.; Li, X.; Wang, Q.; Shen, J. Visual in-context learning for large vision-language models. *arXiv preprint arXiv:2402.11574* **2024**.
149. Voigt, P.; Von dem Bussche, A. The eu general data protection regulation (gdpr). *A practical guide, 1st ed., Cham: Springer International Publishing* **2017**.
150. Bonta, R. California consumer privacy act (CCPA). *Retrieved from State of California Department of Justice: https://oag. ca. gov/privacy/ccpa* **2022**.
151. Lang, Y.; Guo, K.; Huang, Y.; Zhou, Y.; Zhuang, H.; Yang, T.; Su, Y.; Zhang, X. Beyond Single-Value Metrics: Evaluating and Enhancing LLM Unlearning with Cognitive Diagnosis. *arXiv preprint arXiv:2502.13996* **2025**.
152. Wei, B.; Huang, K.; Huang, Y.; Xie, T.; Qi, X.; Xia, M.; Mittal, P.; Wang, M.; Henderson, P. Assessing the brittleness of safety alignment via pruning and low-rank modifications. *arXiv preprint arXiv:2402.05162* **2024**.
153. Zhang, J.; Chen, K.; He, L.; Lou, J.; Li, D.; Feng, Z.; Song, M.; Liu, J.; Ren, K.; Yang, X. Activation approximations can incur safety vulnerabilities even in aligned llms: Comprehensive analysis and defense. *arXiv preprint arXiv:2502.00840* **2025**.
154. Chen, B.; Lyu, X.; Gao, L.; Song, J.; Shen, H.T. SafePTR: Token-Level Jailbreak Defense in Multimodal LLMs via Prune-then-Restore Mechanism. *arXiv preprint arXiv:2507.01513* **2025**.
155. Zayed, A.; Mordido, G.; Shabanian, S.; Baldini, I.; Chandar, S. Fairness-aware structured pruning in transformers. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, pp. 22484–22492.
156. Ma, S.; Salinas, A.; Nyarko, J.; Henderson, P. Breaking Down Bias: On The Limits of Generalizable Pruning Strategies. In Proceedings of the Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency, 2025.
157. Pochinkov, N.; Schoots, N. Dissecting language models: Machine unlearning via selective pruning. *arXiv preprint arXiv:2403.01267* **2024**.
158. Liu, Z.; Dou, G.; Yuan, X.; Zhang, C.; Tan, Z.; Jiang, M. Modality-Aware Neuron Pruning for Unlearning in Multimodal Large Language Models. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025.
159. Guo, P.; Wang, Y.; Li, W.; Liu, M.; Li, M.; Zheng, J.; Qu, L. Exploring Federated Pruning for Large Language Models. *arXiv preprint arXiv:2505.13547* **2025**.
160. Chrysostomou, G.; Zhao, Z.; Williams, M.; Aletras, N. Investigating hallucinations in pruned large language models for abstractive summarization. *Transactions of the Association for Computational Linguistics* **2024**, *12*, 1163–1181.
161. Zeng, Y.; Wu, Y.; Zhang, X.; Wang, H.; Wu, Q. AutoDefense: Multi-Agent LLM Defense against Jailbreak Attacks. In Proceedings of the Neurips Safe Generative AI Workshop 2024, 2024.
162. Li, H.; Wang, A.; kunquan li.; Wang, Z.; Zhang, L.; Qiu, D.; Liu, Q.; Su, J. A Multi-Agent Framework with Automated Decision Rule Optimization for Cross-Domain Misinformation Detection, 2025, [arXiv:cs.AI/2503.23329].
163. Fan, F.; Li, X. PeerGuard: Defending Multi-Agent Systems Against Backdoor Attacks Through Mutual Reasoning, 2025, [arXiv:cs.MA/2505.11642].
164. Zhou, J.; Wang, L.; Yang, X. GUARDIAN: Safeguarding LLM Multi-Agent Collaborations with Temporal Graph Modeling, 2025, [arXiv:cs.AI/2505.19234].
165. Ebrahimi, S.; Dehghankar, M.; Asudeh, A. An Adversary-Resistant Multi-Agent LLM System via Credibility Scoring, 2025, [arXiv:cs.MA/2505.24239].
166. Asad, A.; Obadinma, S.; Shayanfar, R.; Zhu, X. RedDebate: Safer Responses through Multi-Agent Red Teaming Debates, 2025, [arXiv:cs.CL/2506.11083].

 doi:10.20944/preprints202605.1041.v1

167. Zhu, Y.; Zhang, C.; Shi, X.; Zhang, X.; Yang, Y.; Luo, Y. MASTER: Multi-Agent Security Through Exploration of Roles and Topological Structures – A Comprehensive Framework, 2025, [arXiv:cs.MA/2505.18572].
168. Zhang, Z.; Zhang, Y.; Li, L.; Gao, H.; Wang, L.; Lu, H.; Zhao, F.; Qiao, Y.; Shao, J. PsySafe: A Comprehensive Framework for Psychological-based Attack, Defense, and Evaluation of Multi-agent System Safety. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 15202–15231.
169. He, P.; Lin, Y.; Dong, S.; Xu, H.; Xing, Y.; Liu, H. Red-Teaming LLM Multi-Agent Systems via Communication Attacks. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 6726–6747.
170. Du, Y.; Li, S.; Torralba, A.; Tenenbaum, J.B.; Mordatch, I. Improving factuality and reasoning in language models through multiagent debate. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning, 2024.
171. Khan, A.; Hughes, J.; Valentine, D.; Ruis, L.; Sachan, K.; Radhakrishnan, A.; Grefenstette, E.; Bowman, S.R.; Rocktäschel, T.; Perez, E. Debating with more persuasive LLMs leads to more truthful answers. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning, 2024.
172. Fang, Y.; Li, M.; Wang, W.; Hui, L.; Feng, F. Counterfactual Debating with Preset Stances for Hallucination Elimination of LLMs. In Proceedings of the COLING, 2025, pp. 10554–10568.
173. Sun, X.; Li, J.; Zhong, Y.; Zhao, D.; Yan, R. Towards Detecting LLMs Hallucination via Markov Chain-based Multi-agent Debate Framework. In Proceedings of the ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2025, pp. 1–5.
174. Wan, D.; Chen, J.; Stengel-Eskin, E.; Bansal, M. MAMM-Refine: A Recipe for Improving Faithfulness in Generation with Multi-Agent Collaboration. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2025, pp. 9882–9901.
175. Kim, K.; Lee, S.; Huang, K.H.; Chan, H.P.; Li, M.; Ji, H. Can LLMs Produce Faithful Explanations For Fact-checking? Towards Faithful Explainable Fact-Checking via Multi-Agent Debate, 2024, [arXiv:cs.CL/2402.07401].
176. Tang, Z.; Wang, R.; Chen, W.; Zheng, Y.; Chen, Z.; Liu, Y.; Wang, K.; Chen, T.; Lin, L. Towards CausalGPT: A Multi-Agent Approach for Faithful Knowledge Reasoning via Promoting Causal Consistency in LLMs, 2025, [arXiv:cs.AI/2308.11914].
177. Feng, S.; Shi, W.; Wang, Y.; Ding, W.; Balachandran, V.; Tsvetkov, Y. Don't Hallucinate, Abstain: Identifying LLM Knowledge Gaps via Multi-LLM Collaboration. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2024, pp. 14664–14690.
178. Yang, R.; Rajagopal, D.; Hayati, S.A.; Hu, B.; Kang, D. Confidence Calibration and Rationalization for LLMs via Multi-Agent Deliberation, 2024, [arXiv:cs.CL/2404.09127].
179. Ki, D.; Rudinger, R.; Zhou, T.; Carpuat, M. Multiple LLM Agents Debate for Equitable Cultural Alignment. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 24841–24877.
180. Wan, Y.; Chang, K.W. White Men Lead, Black Women Help? Benchmarking and Mitigating Language Agency Social Biases in LLMs. In Proceedings of the Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2025, pp. 9082–9108.
181. Owens, D.M.; Rossi, R.A.; Kim, S.; Yu, T.; Dernoncourt, F.; Chen, X.; Zhang, R.; Gu, J.; Deilamsalehy, H.; Lipka, N. A multi-llm debiasing framework. *arXiv preprint arXiv:2409.13884* **2024**.
182. Borah, A.; Mihalcea, R. Towards Implicit Bias Detection and Mitigation in Multi-Agent LLM Interactions. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 9306–9326.
183. Wan, Y.; Chen, X.; Chang, K.W. Which Cultural Lens Do Models Adopt? On Cultural Positioning Bias and Agentic Mitigation in LLMs, 2025, [arXiv:cs.CL/2509.21080].
184. Wang, L.; Wang, W.; Wang, S.; Li, Z.; Ji, Z.; Lyu, Z.; Wu, D.; Cheung, S.C. IP Leakage Attacks Targeting LLM-Based Multi-Agent Systems, 2025, [arXiv:cs.CR/2505.12442].
185. Li, W.; Sun, L.; Guan, Z.; Zhou, X.; Sap, M. 1-2-3 Check: Enhancing Contextual Privacy in LLM via Multi-Agent Reasoning. In Proceedings of the Proceedings of the The First Workshop on LLM Security (LLMSEC), 2025, pp. 115–128.
186. Huang, Y.; Jiang, Y.; Wang, W.; Zhuang, H.; Luo, X.; Ma, Y.; Xu, Z.; Chen, Z.; Moniz, N.; Lin, Z.; et al. Emergent Social Intelligence Risks in Generative Multi-Agent Systems. *arXiv preprint arXiv:2603.27771* **2026**.

 doi:10.20944/preprints202605.1041.v1

187. Huang, Y.; Zhang, Q.; Yu, P.S.; Sun, L. TrustGPT: A Benchmark for Trustworthy and Responsible Large Language Models. *arXiv preprint arXiv:2306.11507* **2023**.
188. Wang, Y.; Ye, J.; Wu, S.; Gao, C.; Huang, Y.; Chen, X.; Zhao, Y.; Zhang, X. TrustEval: A Dynamic Evaluation Toolkit on Trustworthiness of Generative Foundation Models. In Proceedings of the Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations), 2025.
189. Hasan, A.; Rugina, I.; Wang, A. Pruning for Protection: Increasing Jailbreak Resistance in Aligned LLMs Without Fine-Tuning. In Proceedings of the The 7th BlackboxNLP Workshop, 2024.
190. Yan, H.; Liu, Z.; Jiang, M. Dual-Space Smoothness for Robust and Balanced LLM Unlearning. *arXiv preprint arXiv:2509.23362* **2025**.
191. Meng, K.; Bau, D.; Andonian, A.; Belinkov, Y. Locating and editing factual associations in GPT. *Advances in Neural Information Processing Systems* **2022**, *35*, 17359–17372.
192. Meng, K.; Sharma, A.S.; Andonian, A.; Belinkov, Y.; Bau, D. Mass-editing memory in a transformer. *ArXiv preprint* **2022**, *abs/2210.07229*.
193. Cai, Y.; Cao, D.; Guo, R.; Wen, Y.; Liu, G.; Chen, E. Locating and mitigating gender bias in large language models. In Proceedings of the International Conference on Intelligent Computing, 2024, pp. 471–482.
194. Dasu, V.A.; Gupta, V.; Tizpaz-Niari, S.; Tan, G.; et al. Attention Pruning: Automated Fairness Repair of Language Models via Surrogate Simulated Annealing. *arXiv preprint arXiv:2503.15815* **2025**.
195. Qin, Z.; Wang, H.; Wang, Z.; Liu, D.; Fan, C.; Lv, Z.; Tu, Z.; Chu, D.; Sui, D. Mitigating gender bias in code large language models via model editing. *arXiv preprint arXiv:2410.07820* **2024**.
196. Ye, J.; Wang, Y.; Huang, Y.; Chen, D.; Zhang, Q.; Moniz, N.; Gao, T.; Geyer, W.; Huang, C.; Chen, P.Y.; et al. Justice or Prejudice? Quantifying Biases in LLM-as-a-Judge. *arXiv preprint arXiv:2410.02736* **2024**.
197. Liu, Z.; Dou, G.; Chien, E.; Zhang, C.; Tian, Y.; Zhu, Z. Breaking the trilemma of privacy, utility, and efficiency via controllable machine unlearning. In Proceedings of the Proceedings of the ACM Web Conference 2024, 2024, pp. 1260–1271.
198. Li, T.; et al. Revisiting Jailbreaking for Large Language Models. In Proceedings of the COLING, 2025.
199. U.S. AI Safety Institute at NIST. Managing Misuse Risk for Dual-Use Foundation Models (NIST AI 800-1, 2nd Public Draft). Technical report, NIST, 2025.
200. Amayuelas, A.; Yang, X.; Antoniades, A.; Hua, W.; Pan, L.; Wang, W.Y. MultiAgent Collaboration Attack: Investigating Adversarial Attacks in Large Language Model Collaborations via Debate. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 6929–6948.