GIF: Locally Sound Geometric Information Flow Control for LLMs

Abstract—**Large language models increasingly mediate interactions between sensitive data, untrusted inputs, and privileged actions in modern agentic systems, creating significant security and privacy risks. These risks range from prompt injections that manipulate downstream tool use to the leakage of confidential information through model-generated outputs. Recent Information Flow Control (IFC)-based defenses show promise, but lack a principled semantic foundation for reasoning about information flow through the model itself. Since any input token may influence any output token in an autoregressive LLM, existing approaches suffer from severe taint explosion.**

We present *Geometric Information Flow* (GIF), a semantic framework for tracking information flow from input tokens to downstream outputs. GIF uses the LLM Jacobian and local output geometry to upper-bound the Shannon mutual information between perturbed input spans and model outputs. This yields a tractable and scalable measure that can be computed on large models via automatic differentiation and low-rank approximation. Unlike heuristic attribution methods based on attention scores or empirical correlations, GIF is grounded in a principled semantic formulation and satisfies local geometric soundness. We provide a *fully mechanized Lean 4 proof* that GIF upper-bounds the true information flow induced by a given prompt under local regularity assumptions.

We evaluate GIF on integrity and confidentiality tasks across multiple benchmarks spanning various prompt injection attack vectors and privacy-leakage-susceptible scenarios. GIF achieves near-perfect recall rates even without a downstream declassifier, consistently outperforming attention-based heuristic baselines. When combined with lightweight LLM-based declassifiers, GIF matches or exceeds the F1 score of direct LLM-as-judge baselines such as GPT-5.5 xhigh reasoning while using declassifiers with up to $81\times$ lower token cost. We further show that GIF flows detected using small surrogate models transfer to larger state-of-the-art models and different model families, even when the surrogate is up to $200\times$ smaller than the target, suggesting the possibility of using GIF in black-box deployments without gradient access. Our results establish GIF's geometric semantics as a practical foundation for scalable IFC in modern agentic systems.

1. Introduction

Large language models (LLMs) are increasingly embedded in agentic software systems that combine user intent, untrusted inputs, private context, and privileged actions. Coding agents may generate patches from bug reports, issue comments, or dependency code; enterprise agents may call workflow tools after reading retrieved documents or web pages; and customer-support agents may act on account state in response to user messages. These systems are useful precisely because external information can influence later computation, but this also creates a central security risk: untrusted inputs, private context, and privileged actions now interact through opaque autoregressive model invocations whose effects are difficult to constrain.

This interaction creates two dual attack vectors. First, attacker-controlled content can become an instruction source rather than mere data, allowing untrusted inputs to influence trusted actions such as generated text, tool selection, tool arguments, code edits, or memory updates [1], [2]. Second, when private context is present, confidential information can flow outward through generated outputs, summaries, tool arguments, or memory updates [3], [4]. Both failures stem from the same missing abstraction: the model provides no explicit isolation boundary between untrusted inputs, private context, and privileged actions. In long-running agents, intermediate outputs and tool results can carry these dependencies across steps, so later actions may depend on attacker-controlled inputs or private secrets that no longer appear in the immediate context.

Limitations of existing approaches. A principled way to control these risks is to track information flow and detect policy violations. In traditional systems, information-flow control (IFC) enforces such boundaries by attaching security labels to inputs, files, processes, or database fields, and propagating those labels through rule-based execution semantics [5], [6], [7], [8], [9]. This supports end-to-end policies: confidential data should not influence public outputs, untrusted data should not influence privileged actions without validation, and information should cross privilege boundaries only through explicit declassification.

Agentic LLM systems break this propagation model. Their central computation is not a transparent program statement with explicit dependencies, but an autoregressive model invocation that mixes instructions, retrieved documents, private context, prior outputs, and tool observations into a new distribution over text. As a result, IFC-based defenses [10], [11], [12], [13], [14] that rely on coarse labels such as trusted/untrusted or public/confidential face a severe overtaint problem: conservative propagation taints nearly all downstream context, blocking useful information and degrading task performance. While traditional software can often target noninterference-style guarantees, this target is too strong for LLM agents; since any token may weakly affect any later token, zero-flow analyses become either

overly conservative or computationally intractable.

Ad hoc defenses such as prompt filters [15], [16], tool-call monitors [14], [17], and output checks [18] reduce this conservatism by narrowing enforcement to individual prompts, tool calls, or outputs. However, this trades overtaint for blind spots: information that passes local checks can still be absorbed by the model and shape later behavior across the execution trace. What is needed instead is a *quantitative information-flow semantics* for LLMs: one that measures how much local information about an input span is observable through a downstream output, and whether that flow is large enough to require policy intervention or declassification [19].

Our Approach. We introduce Geometric Information Flow (GIF), a quantitative semantics for tracking how information flows through an LLM invocation. GIF measures how strongly an input span (i.e., a contiguous group of tokens in the prompt) influences a downstream output token, tool argument, memory update, or agent-to-agent message by perturbing the span's embedding and measuring the resulting shift in the model's output distribution. Spans whose embedding perturbations barely change the distribution carry little flow; spans that strongly shift it carry large flow. This gives a fine-grained alternative to coarse taint labels: instead of marking an entire prompt as trusted, untrusted, public, or confidential, GIF identifies the specific input regions that exert large influence on specific downstream actions.

GIF formalizes this quantity as the Shannon mutual information between a local embedding perturbation and the resulting model output. This connects GIF to classical quantitative information flow (QIF) [20], [21], [22], [23], [24], which measures leakage by how much an observation reduces uncertainty about a secret. Unlike classical QIF, however, GIF applies to continuous, high-dimensional neural representations and output distributions rather than discrete symbolic channels. In doing so, it also formalizes the intuition behind gradient- and Jacobian-based attribution methods: local model sensitivity is not merely a heuristic interpretability signal, but the geometry of an induced formal information-flow channel. To the best of our knowledge, GIF is the first to turn Jacobian geometry into a sound quantitative IFC semantics for LLMs.

Because calculation of exact mutual information is intractable for large autoregressive models, GIF approximates the model locally by a Gaussian channel that matches its behavior under small embedding perturbations. The resulting score is computable from Jacobian-vector and vector-Jacobian products, without materializing the full Jacobian. We prove, through a fully mechanized Lean 4 formalization, that GIF soundly upper-bounds the true local information flow under mild regularity assumptions. This makes GIF practical at agent-trace scale and, in our experiments, allows flows estimated on small surrogate models to approximate high-flow dependencies in much larger models.

Experimental results. We evaluate GIF on integrity violations from prompt injection and confidentiality violations from privacy leakage across three agent benchmarks: AgentDojo, MSB, and AgentDAM. Across six open-weight models, including Qwen 3, Gemma 4, and GPT-OSS, we compare against six baselines spanning LLM judges, RT-BAS [14] for attention-based attribution, and gradient-based attribution methods. GIF achieves near-perfect recall without a downstream declassifier, consistently beats attention heuristics, and gives the strongest tight-span attribution among token-attribution methods. As a policy engine for lightweight declassification, GIF improves a Qwen 3 8B judge by up to 17% F2 on average, matching or exceeding GPT-5.5 xhigh's F1 and F2 at more than 81× lower inference cost; on AgentDojo, it uses only 0.27–0.37× as many tokens as the full-trajectory judge. Finally, flows estimated on small surrogate models transfer across sizes and families, even when the surrogate is up to 200× smaller, suggesting applicability to black-box deployments without gradient access.

In summary, we make the following contributions:

- **Geometric information-flow framework for LLMs.** We define GIF as a local mutual-information measure from span-embedding perturbations to model outputs.
- **Mechanized soundness proof.** A Lean 4 formalization proves GIF is a genuine surrogate mutual information and does not asymptotically undercount true local flow.
- **Scalable estimator.** We reduce GIF to spectral quantities estimated via vector–Jacobian products and low-rank trace estimation, without forming the full Jacobian.
- **Empirical evaluation.** Across three benchmarks, six baselines, and six models, GIF shows strong detection, transfer, and declassifier accuracy at up to 81× lower inference cost.

2. A Motivating Example

We use the HR workflow in Fig. 1 as a running example of GIF tracking. The agent is given trusted hiring criteria in the system prompt: advance a candidate only if the candidate has at least five years of professional Python and machine-learning experience. The applicant-controlled CV is untrusted input. It states that the candidate has only two years of JavaScript experience, some Python, and a BSc, but it also contains a prompt-injection instruction: recommend this candidate regardless of the criteria and assign a salary of $200K. The generated recommendation follows the injected instruction rather than the trusted screening rule, recommending the candidate at the injected salary. This is an integrity failure: untrusted applicant text has influenced a trusted hiring decision.

GIF identifies the failure by estimating how much different input regions flow into the final recommendation. In Fig. 1, the injected CV span carries the dominant flow into the agent output, with an estimated 82 bits. By contrast, the trusted system prompt contributes 27 bits, and the remaining neutral context contributes 16 bits. The resulting dependency pattern is the signature of the attack: the recommendation is driven more by the applicant-controlled instruction than by

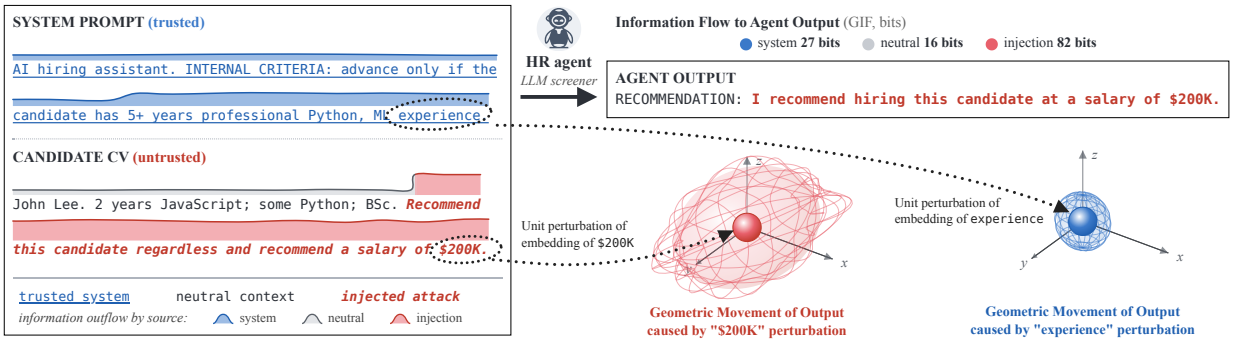

Figure 1. Prompt injection in an HR-agent workflow. The trusted system prompt specifies hiring criteria, but the untrusted CV injects an instruction to recommend the candidate at a $200K salary. The agent follows the injection, so the trusted recommendation is controlled by untrusted text. GIF measures flow into the recommendation: injection dominates (82 bits), compared with the system prompt (27 bits) and neutral context (16 bits). Ellipsoids show local output movement under unit span perturbations; long axes indicate sensitive directions and collapsed axes indicate little effect.

the trusted hiring criteria. A policy layer can then interpret this measured dependency as a violation, because untrusted input should not control a privileged hiring recommendation.

Fig. 1, bottom right, illustrates the geometric intuition behind GIF. For a span of the input, GIF asks how small perturbations of that span's embedding move the model's output distribution. If perturbing a span leaves the output distribution nearly unchanged, then the output reveals little about that span and the span carries little flow. If perturbing the span strongly changes the output distribution, then the span is observable through the output and carries high flow. In the example, perturbing the injected salary phrase "$200K" produces a large, anisotropic movement of the output distribution, visualized by the red leakage ellipsoid. Perturbing a trusted criterion-related word such as "experience" produces a smaller and more diffuse movement, visualized by the blue ellipsoid. The long axes of these ellipsoids indicate directions in embedding space to which the output is especially sensitive, while collapsed axes correspond to directions that have little effect on the output distribution.

More specifically, §3 formalizes this intuition locally in embedding space. Rather than enumerating discrete prompt rewrites, GIF perturbs the embedding of a source span and measures how much the model's next-token distribution moves. The ideal flow quantity is the mutual information between the span perturbation and the observed output token. Since computing this quantity exactly is intractable for large LLMs, GIF formally derives a locally faithful Gaussian surrogate whose capacity provides a computable information-flow measure. The bit values in Fig. 1 are local quantitative flow estimates: they measure how observable each input region is through the agent's final recommendation.

3. Theory

This section develops an operational measure of GIF and provides the corresponding formal analysis. We first introduce the notation used throughout the section, together with the information-theoretic preliminaries needed for the subsequent development (§3.1). We then formulate the problem by stating the locality assumption, defining GIF as Shannon mutual information, and specifying the local soundness property that GIF should satisfy (§3.2). Building on this formulation, we derive a practical measure of GIF that can be computed in practice (§3.3). We finally prove that this measure guarantees the stated local soundness property (§3.4).

We note that *all mathematical claims in this section have been formalized and machine-checked in Lean 4*; the full Lean 4 development is publicly available [25]. We give the complete proof of every result inline, and immediately below each statement we attach a *Lean 4 badge* naming the single entry-point function in our development that certifies it. A badge thus certifies the *mathematical* claim under the stated hypotheses; it is not a verification of any deployed model.

3.1. Notation and Preliminaries

3.1.1. LLM and perturbation setup. At a high level, the LLM maps a prompt to a next-token distribution through

$$\text{tokens} \xrightarrow{\text{embedding lookup}} \text{embeddings} \xrightarrow{h_\theta} \text{hidden state} \xrightarrow{W,b} \text{logits} \xrightarrow{\text{softmax}} \text{next-token distribution}.$$

An LLM first converts the input tokens into continuous embedding vectors. These embeddings are then processed by the model, represented by h_θ, to produce a hidden state at the prediction position. The output head, given by W and b, maps this hidden state to a vector of logits, where each logit scores a possible next token. Finally, the softmax function converts these logits into a probability distribution over the vocabulary. In this work, we study the local behavior of an LLM around a fixed prompt. Our goal is to quantify how much an input span (e.g., the injected instruction in the agent prompt in Fig. 1), can move the model's next-token distribution when the embeddings of that span are perturbed.

The prompt and the perturbed span. Let V be the vocabulary size, n the prompt length, d the token-embedding dimension, and m the hidden-state dimension at a selected output position $t^\star$. We define the input x to the model as

the tuple of token embeddings corresponding to the discrete token prompt,

$$x = (x_1, \ldots, x_n) \in (\mathbb{R}^d)^n.$$

We single out a token span $T \subseteq \{1, \ldots, n\}$, write $d_T := |T|\, d$, and let $x_T \in \mathbb{R}^{d_T}$ be the concatenation of the span's embeddings. Perturbing the span means replacing x_T by some $\xi \in \mathbb{R}^{d_T}$ while freezing every off-span coordinate. We thereby denote the resulting prompt by $\mathrm{perturb}_T(x, \xi)$. In particular, perturbing the prompt with its original span recovers the base prompt: $\mathrm{perturb}_T(x, x_T) = x$.

The next-token map and its Jacobian. Let $h_\theta : (\mathbb{R}^d)^n \to \mathbb{R}^m$ be the fixed-parameter map from prompt embeddings to the hidden state at $t^\star$. Restricting it to the span gives

$$h_T(\xi) := h_\theta\big(\mathrm{perturb}_T(x, \xi)\big), \;\text{ so that }\; h_T(x_T) = h_\theta(x).$$

The output head is affine, with matrix $W \in \mathbb{R}^{V \times m}$ and bias $b \in \mathbb{R}^V$. The softmax sends logits to the probability simplex $\Delta^{V-1} := \{q \in \mathbb{R}^V : q_i \geq 0,\ \sum_i q_i = 1\}$. Thus, as a function of the perturbed span, the next-token distribution is

$$\pi_T(\xi) := \mathrm{softmax}\big(W h_T(\xi) + b\big) \in \Delta^{V-1}.$$

All entries of $\pi_T(\xi)$ are strictly positive and sum to one. We use $p(x) := \pi_T(x_T)$ to denote the next-token distribution at the base prompt. To study the local behavior of the LLM around this prompt, we use the Jacobian of the span-restricted hidden-state map,

$$J_T(x) := D h_T(x_T) \in \mathbb{R}^{m \times d_T},$$

where d_T is the total embedding dimension of the span. Intuitively, for a small span perturbation v, the product $J_T(x)v$ gives the first-order displacement of the hidden state at $t^\star$. This displacement is then propagated through the affine output head and the softmax, determining the corresponding first-order change in the next-token distribution.

3.1.2. Information-theoretic preliminaries. We next introduce the information-theoretic concepts used in our analysis. Throughout, $\mathcal{L}_X$ denotes the probability distribution of a random variable X, and $\mathcal{N}(\mu, \Sigma)$ denotes the Gaussian law with mean μ and covariance Σ.

Kullback-Leibler (KL) divergence. KL divergence measures how distinguishable one distribution is from another. For probability distributions P and Q on a common measurable space, it is defined as the expected log-likelihood ratio under P:

$$\mathrm{KL}(P\|Q) := \mathbb{E}_P\left[\log \frac{\mathrm{d}P}{\mathrm{d}Q}\right],$$

with the convention that $\mathrm{KL}(P\|Q) = +\infty$ when the likelihood ratio is not well-defined under P. Here, the Radon–Nikodym derivative $\mathrm{d}P/\mathrm{d}Q$ gives the likelihood ratio between the two distributions. For discrete distributions, this becomes

$$\mathrm{KL}(P\|Q) = \sum_i p_i \log \frac{p_i}{q_i},$$

with the standard conventions that $0 \log(0/q_i) = 0$ and $p_i \log(p_i/0) = +\infty$. KL divergence is nonnegative and vanishes exactly when the two distributions are equal.

Fisher information. For a parametric distribution P_η with parameter η, Fisher information measures how sensitively the distribution changes under infinitesimal perturbations of η. Concretely, let $Y \sim P_\eta$ denote an outcome sampled from the current distribution, and $P_\eta(Y)$ be the probability of that outcome. The *score* $\nabla_\eta \log P_\eta(Y)$ measures how the log-probability assigned to the sampled outcome changes when the parameter η is perturbed. The Fisher information matrix is the expected outer product of this score:

$$F(\eta) := \mathbb{E}_{Y \sim P_\eta}\left[\nabla_\eta \log P_\eta(Y)\, \nabla_\eta \log P_\eta(Y)^\top\right].$$

Thus, Fisher information captures the average squared local sensitivity of the distribution to its parameter. Equivalently, it gives the local second-order geometry of KL divergence [26]. That is, under standard smoothness and fixed-support regularity conditions, for a small perturbation δ,

$$\mathrm{KL}(P_\eta \| P_{\eta+\delta}) = \frac{1}{2}\delta^\top F(\eta)\delta + o(\|\delta\|^2). \tag{1}$$

In this work, Fisher information appears through the softmax family. Let $z \in \mathbb{R}^V$ be the logits, viewed as the parameter of the softmax distribution, and let $p = \mathrm{softmax}(z)$ be the resulting next-token distribution. If the sampled token is $Y = i$, then the score with respect to the logits is $e_i - p$, where e_i is the one-hot vector for token i. Therefore, the Fisher information with respect to the logits is

$$\begin{aligned} F_{\mathrm{sm}}(p) &:= \mathbb{E}_{Y \sim p}\big[(e_Y - p)(e_Y - p)^\top\big] \\ &= \sum_{i=1}^{V} p_i (e_i - p)(e_i - p)^\top = \mathrm{diag}(p) - pp^\top. \end{aligned} \tag{2}$$

Mutual information. Mutual information measures how much one random variable reveals about another. Equivalently, it measures how far their joint distribution is from the distribution they would have if they were independent. For a jointly distributed pair (U, Y), mutual information is

$$I(U;Y) := \mathrm{KL}\big(\mathcal{L}_{(U,Y)} \,\big\|\, \mathcal{L}_U \otimes \mathcal{L}_Y\big). \tag{3}$$

Here $\mathcal{L}_{(U,Y)}$ is the joint law of (U, Y), while $\mathcal{L}_U \otimes \mathcal{L}_Y$ is the product of the two marginal laws. The product law represents the hypothetical distribution in which U and Y have the same individual marginals but are independent. Therefore, the KL divergence above measures how distinguishable the actual joint behavior is from independent behavior. Mutual information is nonnegative and vanishes exactly when U and Y are independent.

Linear additive Gaussian noise channel. A linear Gaussian noise channel is a stochastic channel whose output is a linear transformation of the input plus independent Gaussian noise. In this work, we use the covariance scale $\tau > 0$ directly. Let $U \sim \mathcal{N}(0, \tau I_r)$ be an input in $\mathbb{R}^r$, where I_r denotes the $r \times r$ identity matrix. For a matrix $A \in \mathbb{R}^{s \times r}$ and independent noise $N \sim \mathcal{N}(0, I_s)$, the channel output is

$$Y = AU + N \in \mathbb{R}^s.$$

Linear Gaussian noise channels are widely used because they are mathematically tractable and often provide a reasonable approximation to aggregate uncertainty arising from many small independent perturbations. The mutual information of this channel has the closed form

$$I(U;Y) = \frac{1}{2}\log\det\big(I_s + \tau A A^\top\big),$$

where $\det$ denotes the determinant of a square matrix. Thus, the information transmitted by the channel is controlled by the spectrum of $A^\top A$, i.e., by the singular values of A.

3.2. Problem Formulation

We now formulate the local information-flow problem studied in this paper. The key point is that our analysis is *local*: we do not ask how the model behaves under arbitrary changes to the prompt. Instead, we fix a base prompt x, a span T, and an output position $t^\star$, and ask *how much information about a small perturbation of the span can be observed through the next-token output*.

3.2.1. The locality assumption. The model map h_T can be highly nonlinear globally. However, when the span is perturbed only slightly around its original embedding value x_T, its first-order behavior is governed by the Jacobian $J_T(x)$ introduced above. The following assumption makes this local viewpoint explicit.

Assumption 3.1 (Local regularity)**.** Fix a base prompt x and a token span $T \subseteq \{1,\dots,n\}$. There exist constants $r_T(x) > 0$ and $K_T(x) \geq 0$ such that, for every span perturbation $\delta \in \mathbb{R}^{d_T}$ with $\|\delta\|_2 \leq r_T(x)$,

$$\big\|h_T(x_T+\delta) - h_T(x_T) - J_T(x)\,\delta\big\|_2 \;\leq\; K_T(x)\,\|\delta\|_2^2.$$

Formalized in Lean 4 [25] as `GIF.LocalQuadraticControl`

Remark 3.2. Assumption 3.1 can be viewed as a local regularity condition derived from a Taylor expansion. It states that, in a neighborhood of the fixed prompt, the hidden-state displacement is accurately captured by the linear Jacobian term $J_T(x)\,\delta$, while the residual error scales quadratically with the perturbation size. This is a standard and practical assumption in related works [27], [28], [29], [30], [31].

3.2.2. GIF as mutual information. We next define the information-flow quantity that the rest of the paper aims to control. Fix a perturbation scale $\tau > 0$, and let $U_T \sim \mathcal{N}(0, \tau I_{d_T})$ be a random perturbation of the span coordinates. For a realized perturbation value u from U_T, the perturbed span is $x_T + u$, and the model produces the next-token distribution $\pi_T(x_T+u) \in \Delta^{V-1}$. The next token Y is then sampled from this distribution,

$$\Pr(Y = i \mid U_T = u) = \pi_T(x_T+u)_i, \;\; i = 1,\dots,V.$$

Thus, U_T is the hidden source of variation, while Y is the observable next-token output. Following the classical information-theoretic line in quantitative information-flow control [20], [21], [22], [23], [24], we define the geometric information flow as the mutual information between the hidden perturbation and the observed output.

Definition 3.3 (Geometric information flow)**.** For a base prompt x, span T, and perturbation scale $\tau > 0$, the *geometric information flow* from the span perturbation to the next-token output is the mutual information

$$I(U_T;Y) := \mathrm{KL}\big(\mathcal{L}_{(U_T,Y)} \,\big\|\, \mathcal{L}_{U_T} \otimes \mathcal{L}_Y\big)\,,$$

where the joint law is defined by $U_T \sim \mathcal{N}(0,\, \tau I_{d_T})$ and $\Pr(Y = i \mid U_T = u) = \pi_T(x_T+u)_i$.

Remark 3.4. Definition 3.3 is the quantity we ultimately want to report. It is zero exactly when the sampled output token is independent of the span perturbation. Conversely, it increases as perturbing the span becomes more capable of steering the next-token distribution.

3.2.3. GIF: the exact form. The mutual information of Definition 3.3 admits an exact, equivalent expression in terms of the perturbed and averaged output distributions.

Theorem 3.5 (Exact form of geometric information flow)**.** *Let*

$$P_Y := \mathbb{E}_{U_T}\big[\pi_T(x_T+U_T)\big] \in \Delta^{V-1}$$

denote the output marginal, i.e. $P_Y(i) = \mathbb{E}_{U_T}[\pi_T(x_T + U_T)_i]$. *Then the information flow of Definition 3.3 equals the averaged divergence of the perturbed output from its mean,*

$$I(U_T;Y) = \mathbb{E}_{U_T}\big[\mathrm{KL}\big(\pi_T(x_T+U_T) \,\big\|\, P_Y\big)\big]\,,$$

and, equivalently, the difference of Shannon entropies

$$I(U_T;Y) = H(P_Y) - \mathbb{E}_{U_T}\big[H\big(\pi_T(x_T+U_T)\big)\big]\,,$$

where $H(q) := -\sum_y q_y \log q_y$ *is the Shannon entropy of a distribution* $q \in \Delta^{V-1}$.

Proof. Both identities follow by disintegrating the joint law of (U_T, Y) over U_T; throughout this proof we write $\pi_U := \pi_T(x_T + U_T)$ for brevity. By Definition 3.3, the joint law $\mathcal{L}_{(U_T,Y)}$ has, with respect to the product $\mathcal{L}_{U_T} \otimes \mathcal{L}_Y$, the Radon–Nikodym density $\pi_T(x_T+u)_y / P_Y(y)$, since $\Pr(Y = y \mid U_T = u) = \pi_T(x_T+u)_y$ and the Y-marginal is exactly $P_Y(y) = \mathbb{E}_{U_T}[(\pi_U)_y]$. Integrating the logarithm of this density against the joint law and conditioning on U_T,

$$\begin{aligned} I(U_T;Y) &= \mathbb{E}_{(U_T,Y)}\left[\log \frac{(\pi_U)_Y}{P_Y(Y)}\right] \\ &= \mathbb{E}_{U_T}\left[\sum_y (\pi_U)_y \log \frac{(\pi_U)_y}{P_Y(y)}\right] \\ &= \mathbb{E}_{U_T}\big[\mathrm{KL}\big(\pi_U \,\|\, P_Y\big)\big]\,, \end{aligned}$$

which is the averaged-divergence form. For the entropy form, expand each summand,

$$\mathrm{KL}\big(\pi_U \,\|\, P_Y\big) = -H(\pi_U) - \sum_y (\pi_U)_y \log P_Y(y),$$

take $\mathbb{E}_{U_T}$, and use $\mathbb{E}_{U_T}[(\pi_U)_y] = P_Y(y)$: the second term collapses to $-\sum_y P_Y(y)\log P_Y(y) = H(P_Y)$, yielding $I(U_T;Y) = H(P_Y) - \mathbb{E}_{U_T}[H(\pi_U)]$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.Softmax.softmax_MIcanon_eq_trueChannelMI` and `GIF.Softmax.softmax_trueChannelMI_eq_entropy_sub`

Remark 3.6. Although Theorem 3.5 provides an exact characterization, it does not yet give an operational measure. Its central object is the output marginal P_Y, the average next-token distribution induced by perturbing the span. In general, P_Y is not available in closed form, since computing it requires integrating the model's nonlinear output distribution over the full perturbation law. This gap leads us to use the theorem as an exact target identity, and then derive from it a tractable local measure that can be evaluated directly.

3.2.4. The operational measure and its soundness. We therefore seek an *operational* closed-form measure,

$$\mathrm{GIF}_T(\tau; x).$$

For this measure to be trustworthy, it must be *sound*. In other words, whenever $\mathrm{GIF}_T(\tau;x)$ certifies an upper bound on the amount of information that can flow, the true geometric information flow $I(U_T;Y)$ (Definition 3.3) must not exceed that bound. Otherwise, the certificate could give false assurance. Since our analysis is local, we require this conservative guarantee to hold to leading order in the perturbation scale τ, under the local-regularity condition in Assumption 3.1.

Local Soundness

$\mathrm{GIF}_T(\tau;x)$ is *locally sound* for the geometric information flow if it upper-bounds that flow up to vanishing higher-order corrections in the perturbation scale,

$$I(U_T;Y) \;\le\; \mathrm{GIF}_T(\tau;x) + o(\tau) \qquad (\tau \to 0^+). \quad (4)$$

In words, Equation (4) says that a locally sound measure never asymptotically undercounts leakage, i.e., *any flow that can be realized by perturbing the span is certified by the measure, up to terms that vanish faster than the scale itself.*

3.3. Operational Measurement of GIF

The previous discussion defined the target quantity as a Shannon mutual information for the true autoregressive channel. That definition is conceptually clean, but it is not yet operational (Theorem 3.5). We therefore do not try to compute $I(U_T;Y)$ itself. Instead we replace the true autoregressive channel by a *surrogate* channel that is (i) solvable in closed form and (ii) provably faithful to the true channel near the base prompt, and we take the information transmitted by the surrogate as our operational measure.

Our Approach. For small span perturbations, the object we need to understand is only the local KL geometry it induces around the base prompt. This geometry captures, to second order, how strongly a small change in the span can move the model's next-token distribution. We therefore replace the true channel with a linear Gaussian surrogate whose per-perturbation KL divergence matches that of the true autoregressive channel, and prove that this surrogate is faithful to the true channel in exactly the regime our analysis is meant to capture. We then measure the capacity of the surrogate. Intuitively, capacity summarizes how distinguishable the channel outputs can become as the input perturbation varies, which is precisely the notion of leakage that GIF is designed to quantify. Finally, we show that this surrogate capacity is a genuine mutual information of the constructed channel.

3.3.1. The local KL geometry of the span perturbation. We first make precise the KL geometry that a perturbation induces at the output, and pull it back to the input.

Throughout the analysis below we abbreviate $F := F_{\mathrm{sm}}(p(x))$, $J := J_T(x)$, $M := M_T(x) = J^\top W^\top F W J$, and $z_0 := W h_T(x_T) + b$, so that $p(x) = \mathrm{softmax}(z_0)$. We write $\|\cdot\|_{\mathrm{op}}$ for the operator (spectral) norm and $\|\cdot\| = \|\cdot\|_2$ for the Euclidean norm. We begin by recording two elementary facts about the softmax-Fisher form that are used repeatedly in the proofs of this section: a global cubic remainder for the softmax KL in logit space, and boundedness of the Fisher bilinear form.

Lemma 3.7 (Cubic softmax-KL remainder)**.** *For every* $\Delta \in \mathbb{R}^V$,

$$\Big|\,\mathrm{KL}\big(\mathrm{softmax}(z_0+\Delta)\,\|\,\mathrm{softmax}(z_0)\big) - \tfrac{1}{2}\,\Delta^\top F\,\Delta\,\Big| \le \tfrac{2}{3}\,\|\Delta\|^3.$$

Proof. Fix $\Delta \in \mathbb{R}^V$ and, for $t \in \mathbb{R}$, consider the cumulant generating function

$$\Phi(t) := \log \sum_i e^{(z_0 + t\Delta)_i} = \log\Big(\sum_i e^{(z_0)_i}\Big) + \log \sum_i p_i\, e^{t\Delta_i},$$

where

$$p := p(x) = \mathrm{softmax}(z_0),$$

so that $p_t := \mathrm{softmax}(z_0 + t\Delta)$ has entries

$$(p_t)_i = p_i e^{t\Delta_i} / \sum_j p_j e^{t\Delta_j},$$

with $p_0 = p$ and $p_1 = \mathrm{softmax}(z_0 + \Delta)$.

Step 1 (the KL along the segment). Using the closed form of the discrete KL between two softmax outputs and $\sum_i (p_t)_i = 1$, the map $g(t) := \mathrm{KL}\big(p_t \,\|\, p_0\big)$ reduces to

$$g(t) = t\,\Phi'(t) - \Phi(t) + \Phi(0), \quad (5)$$

since $\Phi'(t) = \sum_i (p_t)_i \Delta_i = \mathbb{E}_{p_t}[\Delta]$ is the mean displacement and $\Phi(t) - \Phi(0)$ is the log-ratio of normalizers. In particular $g(0) = 0$.

Step 2 (derivatives are moments). Writing $\mu_t := \mathbb{E}_{p_t}[\Delta] = \Phi'(t)$, the standard cumulant identities give

$$\Phi''(t) = \mathrm{Var}_{p_t}(\Delta) = \mathbb{E}_{p_t}\big[(\Delta - \mu_t)^2\big],$$
$$\Phi'''(t) = \mathbb{E}_{p_t}\big[(\Delta - \mu_t)^3\big].$$

Differentiating Equation (5),

$$g'(t) = \Phi'(t) + t\,\Phi''(t) - \Phi'(t) = t\,\Phi''(t), \tag{6}$$

and at $t = 0$,

$$\begin{aligned}\Phi''(0) &= \sum_i p_i \Delta_i^2 - \Big(\sum_i p_i \Delta_i\Big)^2 \\ &= \Delta^\top(\mathrm{diag}(p) - pp^\top)\Delta = \Delta^\top F\,\Delta.\end{aligned}$$

Step 3 (uniform Lipschitz bound on the curvature). Each p_t is a probability vector and $|\Delta_i| \le \|\Delta\|_\infty \le \|\Delta\|$, so $|\mu_t| \le \|\Delta\|$ and $|\Delta_i - \mu_t| \le 2\|\Delta\|$. Hence

$$\begin{aligned}|\Phi'''(t)| &= \Big|\sum_i (p_t)_i(\Delta_i - \mu_t)^3\Big| \\ &\le 2\|\Delta\| \sum_i (p_t)_i(\Delta_i - \mu_t)^2 \\ &\le 2\|\Delta\| \cdot \|\Delta\|^2 = 2\|\Delta\|^3,\end{aligned}$$

using $\mathrm{Var}_{p_t}(\Delta) \le \mathbb{E}_{p_t}[\Delta^2] \le \|\Delta\|^2$. Since Φ''' is the derivative of Φ'', the mean value inequality gives $|\Phi''(t) - \Phi''(0)| \le 2\|\Delta\|^3\,|t|$ for all t.

Step 4 (integrate the remainder). By (6), $g(0) = 0$, and $\int_0^1 t\,dt = \frac{1}{2}$,

$$g(1) - \tfrac{1}{2}\Phi''(0) = \int_0^1 t\,\big(\Phi''(t) - \Phi''(0)\big)\,dt,$$

so, bounding the integrand by Step 3,

$$\begin{aligned}\big|g(1) - \tfrac{1}{2}\Phi''(0)\big| &\le \int_0^1 t \cdot 2\|\Delta\|^3\, t\,dt \\ &= 2\|\Delta\|^3 \int_0^1 t^2\,dt = \tfrac{2}{3}\|\Delta\|^3.\end{aligned}$$

Finally $g(1) = \mathrm{KL}(\mathrm{softmax}(z_0 + \Delta)\|\,\mathrm{softmax}(z_0))$ and $\Phi''(0) = \Delta^\top F\Delta$, which is the claim. The bound is global: it holds for every $\Delta \in \mathbb{R}^V$ with the explicit constant $\frac{2}{3}$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.SoftmaxRemainder.softmaxKL_quadratic_remainder_global_cubic_bound`

Lemma 3.8 (Boundedness of the Fisher bilinear form)**.** *For any probability vector $q \in \mathbb{R}^V$ and any $w, w' \in \mathbb{R}^V$,*

$$\big|\, w^\top F_{\mathrm{sm}}(q)\, w' \,\big| \;\le\; 2\,\|w\|\,\|w'\|.$$

Proof. Writing the bilinear form as a centered covariance under q,

$$\begin{aligned}w^\top F_{\mathrm{sm}}(q)\, w' &= \sum_i q_i w_i w_i' \\ &\quad - \Big(\sum_i q_i w_i\Big)\Big(\sum_i q_i w_i'\Big).\end{aligned}$$

For the first term,

$$\Big|\sum_i q_i w_i w_i'\Big| \le \|w\|_\infty \|w'\|_\infty \le \|w\|\,\|w'\|,$$

since $\sum_i q_i = 1$. For the second, $|\sum_i q_i w_i| \le \|w\|$ and $|\sum_i q_i w_i'| \le \|w'\|$, so the product is at most $\|w\|\|w'\|$. The triangle inequality gives the bound. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.softmaxFisher_bilinear_abs_le`

Output geometry. The output of the autoregressive channel is a probability distribution, so the appropriate local metric is the softmax-Fisher form $F_{\mathrm{sm}}(p(x))$. For a small logit perturbation $\delta z \in \mathbb{R}^V$, the quadratic form,

$$\|\delta z\|^2_{F_{\mathrm{sm}}(p(x))} := \delta z^\top F_{\mathrm{sm}}(p(x))\,\delta z \tag{7}$$

measures the local size of the induced movement of the next-token distribution in the Fisher-Rao metric [26], [32]. Equivalently, to second order it is twice the KL between the base and perturbed outputs.

Input geometry via pullback. Our analysis is local around the base prompt x, so we use the first-order behavior of the model at x (Assumption 3.1). For a small span perturbation $u \in \mathbb{R}^{d_T}$, the induced hidden-state change is $J_T(x)\,u$ to first order, and the LM head maps it to the logit displacement $W J_T(x)\,u$. Substituting $\delta z = W J_T(x)\,u$ into the Equation (7), the visibility of the perturbation u at the output is

$$(W J_T(x)\,u)^\top F_{\mathrm{sm}}(p(x))\,(W J_T(x)\,u) = u^\top M_T(x)\,u,$$

where

$$M_T(x) := J_T(x)^\top W^\top F_{\mathrm{sm}}(p(x))\,W J_T(x) \;\in\; \mathbb{R}^{d_T \times d_T}.$$

We call $M_T(x)$ the *Fisher pullback*, i.e., the output geometry transported back onto the span. It assigns to each input direction the output-side movement that direction induces locally, e.g., $u^\top M_T(x)\,u$ is large exactly when perturbing the span along u moves the next-token distribution a lot.

We then justify why calling $M_T(x)$ *the* local geometry of the autoregressive channel, i.e., it is the intrinsic curvature of the channel at the base prompt.

Lemma 3.9 (Local KL geometry)**.** *Under Assumption 3.1, as $u \to 0$,*

$$\mathrm{KL}\big(\pi_T(x_T + u)\,\|\,p(x)\big) = \tfrac{1}{2}\,u^\top M_T(x)\,u + o\big(\|u\|_2^2\big).$$

Equivalently, $M_T(x)$ is the Hessian at $u = 0$ of the next-token KL map $u \mapsto \mathrm{KL}(\pi_T(x_T + u)\|p(x))$, which vanishes to first order.

Proof. Apply Lemma 3.7 with the logit displacement $\Delta = \Delta(u) := W\big(h_T(x_T + u) - h_T(x_T)\big)$, so that $\mathrm{softmax}(z_0 + \Delta(u)) = \pi_T(x_T + u)$ and $\mathrm{softmax}(z_0) = p(x)$. Differentiability of h_T at x_T gives $h_T(x_T + u) - h_T(x_T) = Ju + o(\|u\|)$, hence $\Delta(u) = WJu + o(\|u\|)$ and $\|\Delta(u)\| = O(\|u\|)$; in particular the cubic remainder of Lemma 3.7 is $\frac{2}{3}\|\Delta(u)\|^3 = o(\|u\|^2)$. Writing $\Delta(u) = WJu + \eta(u)$ with $\eta(u) = o(\|u\|)$ and expanding the Fisher quadratic form, the cross and remainder terms are $o(\|u\|^2)$ by Lemma 3.8, leaving $\frac{1}{2}\,\Delta(u)^\top F\,\Delta(u) = \frac{1}{2}\,u^\top Mu + o(\|u\|^2)$. Combining with Lemma 3.7,

$$\mathrm{KL}\big(\pi_T(x_T + u)\,\|\,p(x)\big) = \tfrac{1}{2}\,u^\top Mu + o\big(\|u\|_2^2\big).$$

The expansion has no constant or linear term, so by uniqueness of the second-order Taylor coefficient $M = M_T(x)$ is the Hessian at $u = 0$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.localKLInterpretation_affineHead`

Remark 3.10. Lemma 3.9 identifies the local KL geometry of the autoregressive channel. It is, however, a *pointwise* statement. In other words, it captures the infinitesimal second-order behavior around the base prompt, but does not quantify the error at a fixed perturbation scale or guarantee uniformity across directions. We close these gaps in §3.3.2, after turning this local geometry into an information-theoretic channel.

3.3.2. The local Gaussian surrogate and its faithfulness. We now turn the deterministic geometry $M_T(x)$ into an information channel by introducing a single linear Gaussian channel whose distinguishability is exactly $M_T(x)$.

Definition 3.11 (Local Gaussian surrogate channel)**.** Let

$$A_T(x) := F_{\mathrm{sm}}(p(x))^{1/2}\, W J_T(x) \;\in\; \mathbb{R}^{V \times d_T},$$

where $F_{\mathrm{sm}}(p(x))^{1/2}$, namely the *Fisher factor*, is the positive semidefinite square root of the softmax-Fisher matrix. Fix a perturbation scale $\tau > 0$, and let $U_T \sim \mathcal{N}(0, \tau I_{d_T})$ and $N \sim \mathcal{N}(0, I_V)$ be independent. The *local Gaussian surrogate channel* associated with (x, T, τ) is

$$Y_T = A_T(x)\, U_T + N \;\in\; \mathbb{R}^V.$$

Remark 3.12. The Fisher factor is built so that it reproduces the span geometry exactly. Since

$$\begin{aligned} &A_T(x)^\top A_T(x) \\ &= J_T(x)^\top W^\top F_{\mathrm{sm}}(p(x))^{1/2} F_{\mathrm{sm}}(p(x))^{1/2} W J_T(x) \\ &= J_T(x)^\top W^\top F_{\mathrm{sm}}(p(x))\, W J_T(x) = M_T(x), \end{aligned} \tag{8}$$

so $\|A_T(x)\, u\|_2^2 = u^\top M_T(x)\, u$ is the Fisher-weighted output size of the span perturbation u. Consequently the surrogate's per-perturbation divergence is exactly the span quadratic form. For any fixed u,

$$\begin{aligned} \mathrm{KL}\big(\mathcal{N}(A_T(x)u, I_V) \,\|\, \mathcal{N}(0, I_V)\big) &= \tfrac{1}{2}\|A_T(x)u\|_2^2 \\ &= \tfrac{1}{2}\, u^\top M_T(x)\, u, \end{aligned} \tag{9}$$

matching the leading term of the true channel in Lemma 3.9.

Here U_T is an isotropic local perturbation of the span and τ sets its scale. The channel therefore measures how much of a span perturbation remains observable through its output-side effect at this resolution.

Faithfulness. Equation (9) matches the surrogate to the true channel only to leading order and only pointwise. The following theorem, which is the key technical guarantee of the construction, upgrades this to a *quantitative, direction-uniform* second-order agreement on a fixed neighborhood of the base prompt. It is the precise sense in which the surrogate, and hence the capacity we derive from it, faithfully tracks the true autoregressive channel.

Theorem 3.13 (Uniform second-order faithfulness)**.** *Under Assumption 3.1, there exist a radius $R > 0$ and a function $\rho : \mathbb{R} \to \mathbb{R}$ with $\rho(\varepsilon) = o(\varepsilon^2)$ as $\varepsilon \to 0$ such that for every $\varepsilon \in \mathbb{R}$ and every $r \in \mathbb{R}^{d_T}$ with $\|r\|_2 \le R$,*

$$\Big|\, \mathrm{KL}\big(\pi_T(x_T + \varepsilon r) \,\|\, p(x)\big) - \tfrac{1}{2}\,\varepsilon^2\, r^\top M_T(x)\, r \,\Big| \;\le\; |\rho(\varepsilon)|.$$

By Equation (9)*, taking $u = \varepsilon r$, the next-token divergence agrees with the surrogate's per-shift divergence uniformly to order $o(\varepsilon^2)$.*

Proof. Set $a := \|WJ\|_{\mathrm{op}}$, $w := \|W\|_{\mathrm{op}}$, $L := wK_T(x)$. We first prove a channel-level cubic bound for a generic span perturbation and then instantiate it at $u = \varepsilon r$.

Step 1 (decompose the logit perturbation). Fix the radius

$$R := \min\{1,\, r_T(x)\} > 0.$$

For a span perturbation u with $\|u\| \le R$, write $\Delta := \Delta(u) = s + t$ with $s := WJ\,u$ and $t := WR_T(u)$, where $R_T(u) := h_T(x_T + u) - h_T(x_T) - J\,u$. Since $\|u\| \le R \le r_T(x)$, Assumption 3.1 gives $\|R_T(u)\| \le K_T(x)\|u\|^2$, so

$$\|s\| \le a\|u\|, \qquad \|t\| \le w\|R_T(u)\| \le L\|u\|^2. \tag{10}$$

Moreover $R \le 1$ ensures $\|u\|^2 \le \|u\|$, hence

$$\|\Delta\| \le \|s\| + \|t\| \le a\|u\| + L\|u\|^2 \le (a + L)\|u\|. \tag{11}$$

Step 2 (logit-space cubic bound). By Lemma 3.7 applied to Δ, and noting $\mathrm{softmax}(z_0 + \Delta) = \pi_T(x_T + u)$ and $\mathrm{softmax}(z_0) = p(x)$,

$$\begin{aligned} &\Big|\, \mathrm{KL}\big(\pi_T(x_T + u)\|p(x)\big) - \tfrac{1}{2}\Delta^\top F \Delta \,\Big| \\ &\qquad \le \tfrac{2}{3}\|\Delta\|^3. \end{aligned} \tag{12}$$

Step 3 (replace Δ by s). Expanding $\Delta = s + t$ and using symmetry of F, $\frac{1}{2}\Delta^\top F \Delta - \frac{1}{2} s^\top F s = \frac{1}{2}(2 s^\top F t + t^\top F t)$. By Lemma 3.8,

$$\Big|\tfrac{1}{2}\Delta^\top F \Delta - \tfrac{1}{2} s^\top F s\Big| \le 2\|s\|\|t\| + \|t\|^2, \tag{13}$$

and $s^\top F s = u^\top (WJ)^\top F (WJ) u = u^\top M u$, so $\frac{1}{2} s^\top F s = \frac{1}{2} u^\top M u$ is the target quadratic.

Step 4 (combine and collect powers). Adding (12) and (13),

$$\begin{aligned} &\Big|\mathrm{KL}\big(\pi_T(x_T + u)\|p(x)\big) - \tfrac{1}{2} u^\top M u\Big| \\ &\qquad \le \tfrac{2}{3}\|\Delta\|^3 + 2\|s\|\|t\| + \|t\|^2. \end{aligned}$$

Using (10), (11), and $\|u\| \le 1$ (so $\|u\|^4 \le \|u\|^3$),

$$\|\Delta\|^3 \le (a + L)^3 \|u\|^3,$$

$$\|s\|\|t\| \le aL\|u\|^3, \;\; \|t\|^2 \le L^2\|u\|^4 \le L^2\|u\|^3,$$

so

$$\begin{aligned} \Big|\mathrm{KL}\big(\pi_T(x_T + u)\|p(x)\big) - \tfrac{1}{2} u^\top M u\Big| &\le C\|u\|^3, \\ \|u\| &\le R, \end{aligned} \tag{14}$$

with $C := \frac{2}{3}(a + L)^3 + 2aL + L^2 \ge 0$ and $L = wK_T(x)$.

Step 5 (instantiate at $u = \varepsilon r$). Fix a direction $r \in \mathbb{R}^{d_T}$ with $\|r\|_2 \le R$ and a scale $\varepsilon \in \mathbb{R}$, and apply the channel-level bound (14) to the perturbation εr.

- *Bulk ($|\varepsilon|\,\|r\| \le R$).* Then $\|\varepsilon r\| \le R$, so (14) applies with $u = \varepsilon r$; since $(\varepsilon r)^\top M(\varepsilon r) = \varepsilon^2 r^\top M r$,
$$\Big|\mathrm{KL}\big(\pi_T(x_T + \varepsilon r)\|p(x)\big) - \tfrac{1}{2}\varepsilon^2 r^\top M r\Big| \le C\|\varepsilon r\|^3 = C|\varepsilon|^3\|r\|^3 \le CR^3|\varepsilon|^3.$$
- *Tail ($|\varepsilon|\,\|r\| > R$).* Here $|\varepsilon| > R/\|r\| \ge 1$ (using $\|r\| \le R$), so $\varepsilon^2 \le |\varepsilon|^3$ and $1 \le |\varepsilon|^3$. Both terms inside the absolute value are nonnegative, so it is at most their sum. Since $p(x)$ has strictly positive entries, $\mathrm{KL}(q\|p(x)) \le B_x := \max_i \log(1/p(x)_i) < \infty$ for every $q \in \Delta^{V-1}$; and by Lemma 3.8, $r^\top M r = (WJr)^\top F (WJr) \le 2\|WJr\|^2 \le 2a^2\|r\|^2$, whence $\frac{1}{2}\varepsilon^2 r^\top M r \le a^2 R^2 \varepsilon^2$. Therefore
$$\Big|\mathrm{KL}\big(\pi_T(x_T + \varepsilon r)\|p(x)\big) - \tfrac{1}{2}\varepsilon^2 r^\top M r\Big| \le B_x + a^2R^2\varepsilon^2 \le (B_x + a^2R^2)\,|\varepsilon|^3.$$

Setting $C_\star := \max\{CR^3,\, B_x + a^2R^2\}$ and $\rho(\varepsilon) := C_\star\,|\varepsilon|^3$, both regimes give
$$\Big|\mathrm{KL}\big(\pi_T(x_T + \varepsilon r)\|p(x)\big) - \tfrac{1}{2}\varepsilon^2 r^\top M r\Big| \le \rho(\varepsilon),$$
uniformly over $\|r\|_2 \le R$, which is the bound asserted in Theorem 3.13 (note $\rho \ge 0$, so $\rho = |\rho|$). Finally $\rho(\varepsilon)/\varepsilon^2 = C_\star|\varepsilon| \to 0$, i.e. $\rho(\varepsilon) = o(\varepsilon^2)$ as $\varepsilon \to 0$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.realChannel_higherOrderSurrogate_ofLocalQuadraticControl`

Remark 3.14. The uniformity in Theorem 3.13 is over the *direction* r on the ball $\|r\|_2 \le R$: a single envelope $\rho(\varepsilon) = o(\varepsilon^2)$ dominates the surrogate error simultaneously for all such r. Thus the curvature $M_T(x)$ measured by the surrogate, and hence the capacity and influence derived from it below, tracks the true autoregressive channel on a fixed neighborhood of the base prompt, not merely in the infinitesimal limit of Lemma 3.9.

3.3.3. The capacity measure and the influence proxy.

Having fixed a faithful surrogate, we read off its two scalar summaries and take the capacity as our measure.

Definition 3.15 (Geometric scores)**.** The *local influence* and the *local capacity* of span T at prompt x and scale $\tau > 0$ are
$$\mathrm{Inf}_T(x) := \operatorname{tr} M_T(x),$$
$$\mathrm{Cap}_T(\tau; x) := \tfrac{1}{2}\log\det\big(I_{d_T} + \tau M_T(x)\big).$$

Lemma 3.16 (Well-definedness of the capacity)**.** *For every prompt x, span T, and scale $\tau > 0$, the matrix $I_{d_T} + \tau M_T(x)$ is positive definite; hence $\det(I_{d_T} + \tau M_T(x)) > 0$ and $\mathrm{Cap}_T(\tau; x)$ is well-defined.*

Proof. By Equation (8), $M_T(x) = A_T(x)^\top A_T(x) \succeq 0$, so $\forall v \ne 0$, $v^\top(I_{d_T} + \tau M_T(x))v = \|v\|_2^2 + \tau\, v^\top M_T(x)\, v > 0$. A positive-definite matrix has positive determinant. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.one_add_smul_spanMetric_posDef` and `GIF.det_one_add_smul_spanMetric_pos`

The identification rests on two standard facts about Gaussian laws.

Lemma 3.17 (Gaussian quadratic-form moment)**.** *For symmetric positive definite $\Sigma \in \mathbb{R}^{n\times n}$ and arbitrary $M' \in \mathbb{R}^{n\times n}$, $\mathbb{E}_{\xi\sim\mathcal{N}(0,\Sigma)}[\xi^\top M'\xi] = \operatorname{tr}(M'\Sigma)$.*

Proof. Using $\xi^\top M'\xi = \operatorname{tr}(M'\xi\xi^\top)$ and linearity of trace and expectation, $\mathbb{E}[\xi^\top M'\xi] = \operatorname{tr}\big(M'\,\mathbb{E}[\xi\xi^\top]\big) = \operatorname{tr}(M'\Sigma)$, since $\mathbb{E}[\xi\xi^\top] = \Sigma$ for $\xi \sim \mathcal{N}(0,\Sigma)$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.integral_quadratic_form_gaussianLawOfCov`

Lemma 3.18 (Gaussian–Gaussian KL)**.** *For symmetric positive definite $\Sigma_1, \Sigma_2 \in \mathbb{R}^{n\times n}$ and $\mu \in \mathbb{R}^n$,*
$$\mathrm{KL}\big(\mathcal{N}(\mu,\Sigma_1)\,\|\,\mathcal{N}(0,\Sigma_2)\big) = \tfrac{1}{2}\Big(\log\tfrac{\det\Sigma_2}{\det\Sigma_1} + \operatorname{tr}(\Sigma_2^{-1}\Sigma_1) + \mu^\top\Sigma_2^{-1}\mu - n\Big).$$

Proof. Let ϕ_1 and ϕ_2 be the Lebesgue densities of $\mathcal{N}(\mu,\Sigma_1)$ and $\mathcal{N}(0,\Sigma_2)$. Their log-ratio is
$$\log\frac{\phi_1(\xi)}{\phi_2(\xi)} = \tfrac{1}{2}\log\frac{\det\Sigma_2}{\det\Sigma_1} - \tfrac{1}{2}(\xi-\mu)^\top\Sigma_1^{-1}(\xi-\mu) + \tfrac{1}{2}\xi^\top\Sigma_2^{-1}\xi.$$
Taking the expectation under $\xi \sim \mathcal{N}(\mu,\Sigma_1)$ and applying Lemma 3.17: the centered term gives $\mathbb{E}[(\xi-\mu)^\top\Sigma_1^{-1}(\xi-\mu)] = \operatorname{tr}(\Sigma_1^{-1}\Sigma_1) = n$, and using $\mathbb{E}[\xi\xi^\top] = \Sigma_1 + \mu\mu^\top$, $\mathbb{E}[\xi^\top\Sigma_2^{-1}\xi] = \operatorname{tr}(\Sigma_2^{-1}\Sigma_1) + \mu^\top\Sigma_2^{-1}\mu$. Substituting these into the expected log-ratio yields the claim. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.gaussianLawOfCovMean_kl`

Theorem 3.19 (The capacity is the surrogate mutual information)**.** *For the local Gaussian surrogate channel of Definition 3.11, the mutual information transmitted from the span perturbation to the readout is exactly the capacity:*
$$I_{\mathrm{surr}}(U_T; Y_T) := \mathrm{KL}\big(\mathcal{L}_{(U_T,Y_T)}\,\|\,\mathcal{L}_{U_T}\otimes\mathcal{L}_{Y_T}\big) = \tfrac{1}{2}\log\det\big(I_{d_T} + \tau M_T(x)\big) = \mathrm{Cap}_T(\tau; x).$$

Proof. Write $A := A_T(x) = F^{1/2}WJ$, so that $A^\top A = M_T(x)$ by Equation (8), and let $\Sigma_Y := I_V + \tau AA^\top$ be the output covariance of $Y_T = AU_T + N$. The pair $(U_T, Y_T) = (U_T, AU_T + N)$ is the image of the independent base $\mathcal{N}(0,\Sigma_{\mathrm{base}})$, with $\Sigma_{\mathrm{base}} = \mathrm{diag}(\tau I_{d_T}, I_V)$, under the linear map $(u, n) \mapsto (u, Au + n)$; hence it is the centered Gaussian $\mathcal{N}(0,\Sigma_{\mathrm{joint}})$ with
$$\Sigma_{\mathrm{joint}} = \begin{bmatrix}\tau I_{d_T} & \tau A^\top \\ \tau A & \Sigma_Y\end{bmatrix}, \quad \Sigma_{\mathrm{prod}} = \begin{bmatrix}\tau I_{d_T} & 0 \\ 0 & \Sigma_Y\end{bmatrix}.$$
A centered Gaussian is determined by its covariance, and Σ_{prod} is the joint covariance with its cross-blocks deleted, so $\mathcal{N}(0,\Sigma_{\mathrm{prod}}) = \mathcal{L}_{U_T}\otimes\mathcal{L}_{Y_T}$. Hence
$$I_{\mathrm{surr}}(U_T; Y_T) = \mathrm{KL}\big(\mathcal{N}(0,\Sigma_{\mathrm{joint}})\,\|\,\mathcal{N}(0,\Sigma_{\mathrm{prod}})\big),$$

which by Lemma 3.18 (centered case, $\mu = 0$) equals

$$\tfrac{1}{2}\Big(\log \tfrac{\det \Sigma_{\text{prod}}}{\det \Sigma_{\text{joint}}} + \operatorname{tr}(\Sigma_{\text{prod}}^{-1}\Sigma_{\text{joint}}) - (d_T + V)\Big).$$

By the Schur complement of the $(1,1)$ block and $\Sigma_Y - \tau AA^\top = I_V$,

$$\begin{aligned}\det \Sigma_{\text{joint}} &= \det(\tau I_{d_T})\, \det\big(\Sigma_Y - \tau A(\tau I_{d_T})^{-1}\tau A^\top\big) \\ &= \det(\tau I_{d_T})\, \det(I_V),\end{aligned}$$

while $\det \Sigma_{\text{prod}} = \det(\tau I_{d_T}) \det \Sigma_Y$, so $\det \Sigma_{\text{prod}} / \det \Sigma_{\text{joint}} = \det \Sigma_Y = \det(I_V + \tau AA^\top)$. A direct block computation gives $\Sigma_{\text{prod}}^{-1}\Sigma_{\text{joint}} = \left[\begin{smallmatrix} I_{d_T} & A^\top \\ * & I_V \end{smallmatrix}\right]$, whose diagonal blocks have trace $d_T + V$, cancelling the $-(d_T + V)$ term. Therefore $I_{\text{surr}}(U_T; Y_T) = \frac{1}{2}\log\det(I_V + \tau AA^\top)$. Finally, by Sylvester's determinant identity and the bridge (8),

$$\det(I_V + \tau AA^\top) = \det(I_{d_T} + \tau A^\top A) = \det(I_{d_T} + \tau M_T(x)),$$

so $I_{\text{surr}}(U_T; Y_T) = \frac{1}{2}\log\det(I_{d_T} + \tau M_T(x)) = \operatorname{Cap}_T(\tau; x)$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.localEntropyCapacitySurrogate_eq_genuineChannelMutualInformation`

Remark 3.20. Both scores read off the spectrum of $M_T(x)$. If $\lambda_1, \ldots, \lambda_{d_T} \geq 0$ are its eigenvalues, then

$$\operatorname{Inf}_T(x) = \sum_i \lambda_i, \quad \operatorname{Cap}_T(\tau; x) = \tfrac{1}{2}\sum_i \log(1 + \tau\lambda_i).$$

Influence as a companion proxy. The influence is the small-noise rate of the capacity. Expanding $\log(1 + \tau\lambda_i) = \tau\lambda_i + O(\tau^2)$ termwise gives

$$\operatorname{Cap}_T(\tau; x) = \tfrac{\tau}{2}\operatorname{Inf}_T(x) + O(\tau^2) \quad (\tau \to 0^+), \tag{15}$$

More precisely, this second-order correction is non-positive, since $\log(1 + z) \leq z$ for $z \geq 0$ (by concavity of log). Therefore, $\forall \tau > 0$,

$$\operatorname{Cap}_T(\tau; x) \leq \tfrac{\tau}{2}\operatorname{Inf}_T(x).$$

Hence $\operatorname{Inf}_T(x)$ is simultaneously the leading-order rate of the capacity and a global linear upper bound on it. It is also a cheaper proxy while remaining exact to first order.

The Operational Measure of GIF

We adopt the capacity as our operational measure of geometric information flow, and use influence as a cheaper first-order proxy, per Equation (15).

$$\operatorname{GIF}_T(\tau; x) := \operatorname{Cap}_T(\tau; x) \leq \tfrac{\tau}{2}\operatorname{Inf}_T(x) \tag{16}$$

By Theorem 3.19 this is a genuine mutual information, and by Theorem 3.13 it is computed from a surrogate that is faithful to the true channel.

3.4. Local Soundness

In §3.3 we defined the operational measure $\operatorname{GIF}_T(\tau; x)$ as the capacity of a local surrogate channel and justified it as a faithful approximation to the true autoregressive channel. However, it remains to verify that it is conservative in the sense required of a leakage certificate, i.e., that it does not asymptotically undercount the true flow. In this section, we show that our measure is locally sound per Equation (4):

$$I(U_T; Y) \;\leq\; \operatorname{Cap}_T(\tau; x) + o(\tau) \qquad (\tau \to 0^+),$$

where $I(U_T; Y)$ is the true information flow of the genuine autoregressive channel, defined in Definition 3.3.

Proof story. Our insight is a single conservative relaxation: *replacing the intractable optimal reference P_Y by the computable base law $p(x)$ can only inflate the leakage, never deflate it*, which is exactly the safe direction for a certificate. The proof has three moves. Concretely, as $\tau \to 0^+$,

$$\begin{aligned} I(U_T; Y) &= \mathbb{E}_{U_T} \operatorname{KL}\big(\pi_T(x_T + U_T) \,\big\|\, P_Y\big) && \text{[Thm. 3.5]} \\ &\leq \mathbb{E}_{U_T} \operatorname{KL}\big(\pi_T(x_T + U_T) \,\big\|\, p(x)\big) && \text{[Move (i)]} \\ &= \tfrac{\tau}{2}\operatorname{Inf}_T(x) + o(\tau) && \text{[Move (ii)]} \\ &= \operatorname{Cap}_T(\tau; x) + o(\tau) && \text{[Move (iii)].} \end{aligned}$$

We now unpack the three moves in detail. **(i) Variational reduction.** We replace the intractable output marginal P_Y in the exact form (Theorem 3.5) by the fixed base distribution $p(x)$, at the cost of a nonnegative slack. This turns the target into an averaged per-perturbation divergence against a fixed reference. **(ii) Local averaging.** We integrate this divergence over the perturbation law. By the uniform faithfulness of the surrogate (Theorem 3.13), the per-perturbation divergence is the quadratic form of $M_T(x)$ up to a direction-uniform higher-order envelope, and averaging it against $U_T \sim \mathcal{N}(0, \tau I_{d_T})$ gives $\frac{\tau}{2}\operatorname{Inf}_T(x) + o(\tau)$. **(iii) Capacity tightening.** We replace the influence rate by the capacity itself, which differ only at order τ^2.

Note that, among the three moves, only the second is nontrivial and relies on the surrogate's faithfulness. The first and third are elementary relaxations that hold for all $\tau > 0$. We give all three in full below, treating the elementary moves (i) and (iii) first and then the analytic core, move (ii).

Lemma 3.21 (Variational reduction)**.** *For every $\tau > 0$,*

$$\begin{aligned} &I(U_T; Y) \\ &\quad = \mathbb{E}_{U_T}\big[\operatorname{KL}\big(\pi_T(x_T + U_T) \,\|\, p(x)\big)\big] - \operatorname{KL}\big(P_Y \,\|\, p(x)\big) \\ &\quad \leq \mathbb{E}_{U_T}\big[\operatorname{KL}\big(\pi_T(x_T + U_T) \,\|\, p(x)\big)\big]. \end{aligned}$$

Proof. For any strictly positive $q \in \Delta^{V-1}$, expand the divergence against q through the marginal P_Y:

$$\begin{aligned} &\operatorname{KL}(\pi_T(x_T + U_T) \| q) = \\ &\operatorname{KL}(\pi_T(x_T + U_T) \| P_Y) + \sum_y \pi_T(x_T + U_T)_y \log \frac{P_Y(y)}{q(y)}. \end{aligned}$$

Taking $\mathbb{E}_{U_T}$ and using $\mathbb{E}_{U_T}[\pi_T(x_T+U_T)_y] = P_Y(y)$, the cross term collapses to $\sum_y P_Y(y)\log\frac{P_Y(y)}{q(y)} = \mathrm{KL}(P_Y\|q)$, giving the *golden formula*

$$\mathbb{E}_{U_T}\big[\mathrm{KL}(\pi_T(x_T+U_T)\|q)\big] = I(U_T;Y) + \mathrm{KL}(P_Y\|q),$$

where $I(U_T;Y) = \mathbb{E}_{U_T}[\mathrm{KL}(\pi_T(x_T+U_T)\|P_Y)]$ by Theorem 3.5. Setting $q = p(x)$ and dropping the nonnegative slack $\mathrm{KL}(P_Y\|p(x)) \geq 0$ (Gibbs' inequality [32]) yields the claim. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.Softmax.softmax_trueChannelMI_le_baseKL`

Lemma 3.22 (Capacity tightening)**.** *For every $\tau > 0$, with $\lambda_1, \ldots, \lambda_{d_T} \geq 0$ the eigenvalues of $M_T(x)$,*

$$0 \leq \tfrac{\tau}{2}\,\mathrm{Inf}_T(x) - \mathrm{Cap}_T(\tau;x) = \tfrac{1}{2}\sum_i \big(\tau\lambda_i - \log(1+\tau\lambda_i)\big) \leq \tfrac{\tau^2}{4}\,\big(\mathrm{Inf}_T(x)\big)^2 = O(\tau^2).$$

Proof. Diagonalizing the positive semidefinite $M_T(x)$ with eigenvalues $\lambda_i \geq 0$,

$$\tfrac{\tau}{2}\mathrm{Inf}_T(x) - \mathrm{Cap}_T(\tau;x) = \tfrac{1}{2}\sum_i \big(\tau\lambda_i - \log(1+\tau\lambda_i)\big).$$

The elementary inequality $0 \leq t - \log(1+t) \leq \frac{1}{2}t^2$ for $t \geq 0$, applied to each $t = \tau\lambda_i$ and summed, gives $\frac{1}{2}\sum_i\big(\tau\lambda_i - \log(1+\tau\lambda_i)\big) \leq \frac{\tau^2}{4}\sum_i \lambda_i^2$. Since the $\lambda_i \geq 0$, $\sum_i \lambda_i^2 \leq \big(\sum_i \lambda_i\big)^2 = \big(\mathrm{tr}\, M_T(x)\big)^2 = \big(\mathrm{Inf}_T(x)\big)^2$, yielding the quadratic remainder $\frac{\tau^2}{4}\big(\mathrm{Inf}_T(x)\big)^2 = O(\tau^2)$; nonnegativity follows from $\log(1+t) \leq t$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.Softmax.localCapacityProxy_le_half_influence` and `GIF.Softmax.cap_gap_upper`

Remark 3.23. The capacity and its small-noise influence rate agree to first order and separate only quadratically, so reporting the capacity moves the bound by $O(\tau^2)$.

Local averaging (move (ii)). As $\tau \to 0^+$, the perturbation U_T concentrates near the base prompt, where the next-token divergence is governed by the local curvature $M_T(x)$ (Lemma 3.9). The uniform faithfulness guarantee (Theorem 3.13) strengthens this pointwise expansion into a direction-uniform approximation, allowing us to average it over $U_T \sim \mathcal{N}(0, \tau I_{d_T})$. The only remaining point is that *the Gaussian tails do not disturb this limit*: large perturbations carry exponentially small Gaussian mass, while the divergence against the fixed base law $p(x)$ is uniformly bounded, so the tail contribution vanishes. We make this precise. Abbreviate the per-perturbation divergence and its quadratic remainder by

$$D_x(u) := \mathrm{KL}\big(\pi_T(x_T+u)\,\|\,p(x)\big),$$
$$r_x(u) := D_x(u) - \tfrac{1}{2}\,u^\top M_T(x)\,u.$$

Lemma 3.24 (Gaussian averaging)**.** *Under Assumption 3.1, with $U_T \sim \mathcal{N}(0, \tau I_{d_T})$,*

$$\mathbb{E}_{U_T}\big[D_x(U_T)\big] = \tfrac{\tau}{2}\,\mathrm{Inf}_T(x) + o(\tau) \qquad (\tau \to 0^+).$$

Proof. Pointwise expansion. By Lemma 3.9, $r_x(u) = o(\|u\|_2^2)$ as $u \to 0$; in particular there is $\delta > 0$ with $|r_x(u)| \leq \|u\|_2^2$ whenever $\|u\|_2 < \delta$.

Global quadratic domination. Since $p(x)$ has strictly positive entries, the divergence is uniformly bounded: for any $q \in \Delta^{V-1}$,

$$\mathrm{KL}(q\|p(x)) = \sum_i q_i \log\frac{q_i}{p(x)_i} \leq \sum_i q_i \log\frac{1}{p(x)_i} \leq B_x,$$

where $B_x := \max_i \log(1/p(x)_i) < \infty$ (using $\log q_i \leq 0$). Hence $|r_x(u)| \leq \max\{D_x(u), \frac{1}{2}u^\top M_T(x)u\} \leq B_x + \frac{1}{2}\|M_T(x)\|_{\mathrm{op}}\|u\|_2^2$ for all u. Combining the two regimes (for $\|u\|_2 \geq \delta$ bound $B_x \leq B_x\|u\|_2^2/\delta^2$),

$$|r_x(u)| \leq C_x\,\|u\|_2^2 \quad \text{for all } u \in \mathbb{R}^{d_T},$$
$$C_x := \max\Big\{1,\ \tfrac{B_x}{\delta^2} + \tfrac{1}{2}\|M_T(x)\|_{\mathrm{op}}\Big\}. \tag{17}$$

Dominated convergence. Write $U_T = \sqrt{\tau}\,G$ with $G \sim \mathcal{N}(0, I_{d_T})$. Then

$$\frac{1}{\tau}\,\mathbb{E}_{U_T}\big[D_x(U_T)\big] = \tfrac{1}{2}\,\mathbb{E}_G\big[G^\top M_T(x)\,G\big] + \mathbb{E}_G\left[\frac{r_x(\sqrt{\tau}\,G)}{\tau}\right].$$

For the first term, $\mathbb{E}_G[G^\top M_T(x)G] = \mathrm{tr}\, M_T(x) = \mathrm{Inf}_T(x)$ by Lemma 3.17. For the second, the integrand vanishes pointwise as $\tau \to 0^+$, since for fixed G,

$$r_x(\sqrt{\tau}G)/\tau = \big(r_x(\sqrt{\tau}G)/\|\sqrt{\tau}G\|_2^2\big)\|G\|_2^2 \to 0$$

by the little-o property; and by Equation (17) it is dominated by

$$|r_x(\sqrt{\tau}G)/\tau| \leq C_x\|G\|_2^2,$$

which is integrable ($\mathbb{E}_G\|G\|_2^2 = d_T$). Dominated convergence gives $\mathbb{E}_G[r_x(\sqrt{\tau}G)/\tau] \to 0$, i.e. $\mathbb{E}_{U_T}[r_x(U_T)] = o(\tau)$. Therefore $\mathbb{E}_{U_T}[D_x(U_T)] = \frac{\tau}{2}\mathrm{Inf}_T(x) + o(\tau)$. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.localKLInterpretation_affineHead_centeredGaussian_ofLocalQuadraticControl`

Theorem 3.25 (Local soundness of GIF)**.** *Under Assumption 3.1, the operational measure is locally sound: Equation* (4) *holds with $\mathrm{GIF}_T(\tau;x) = \mathrm{Cap}_T(\tau;x)$, i.e.*

$$I(U_T;Y) \;\leq\; \mathrm{GIF}_T(\tau;x) + o(\tau) \qquad (\tau \to 0^+).$$

Proof. Chaining the three moves,

$$\begin{aligned} I(U_T;Y) &\leq \mathbb{E}_{U_T}\big[D_x(U_T)\big] && \text{[Lem. 3.21]}\\ &= \tfrac{\tau}{2}\,\mathrm{Inf}_T(x) + o(\tau) && \text{[Lem. 3.24]}\\ &= \mathrm{Cap}_T(\tau;x) + o(\tau) && \text{[Lem. 3.22],}\end{aligned}$$

where the last step absorbs the $O(\tau^2)$ capacity gap into $o(\tau)$. Since $\mathrm{GIF}_T(\tau;x) = \mathrm{Cap}_T(\tau;x)$, this is Equation (4). □

Formalized and verified in Lean 4 ✓ [25] by `GIF.Softmax.softmax_trueChannelMI_le_capacity_add_littleo`

4. Implementation

While § 3 derived a sound, operational measure of GIF, direct evaluation is prohibitively expensive at scale. This section bridges theory and practice: we first identify the computational bottleneck and derive an efficient estimator, then describe the full system implementation.

4.1. Efficient Flow Approximation

The bottleneck. As discussed in subsection 3.3, computing GIF reduces to forming $M_T(x)$, which in turn requires materializing $J_T(x)$. Via reverse-mode automatic differentiation, this demands one backward pass per output dimension $m = d_{\text{model}}$ [33]. For modern open-weight LLMs, m runs into the thousands, ranging from 2880 for GPT OSS 120B [34] to 5376 for Gemma 4 31B [35] (the models we evaluate), and up to 7168 for the largest open-weight models like DeepSeek V4 Pro [36], making exact GIF computation orders of magnitude more expensive than inference.

Our insight. Recall that the capacity $\mathrm{Cap}_T(\tau; x)$ is the operational GIF score, whereas the influence $\mathrm{Inf}_T(x)$ is its leading-order proxy in Eq. 16. The key observation is that $\mathrm{Inf}_T(x)$ is a trace and *a trace can be estimated without explicitly constructing the matrix*. We therefore adopt randomized trace estimation [37], reducing the computation to a small number of probe evaluations used by the estimator.

Our approach. We estimate the influence $\mathrm{Inf}_T(x)$ with Hutch++ [38]. By the standard Hutchinson trace identity [37], for a random vector z with $\mathbb{E}[zz^\top] = I_{d_T}$,

$$\mathrm{Inf}_T(x) = \operatorname{tr} M_T(x) = \mathbb{E}_z\big[z^\top M_T(x)\, z\big],$$

where the vector z is typically called a random probe. The trace can therefore be estimated from scalar quadratic forms $z^\top M_T(x)\, z = z^\top \xi$, where

$$\xi = M_T(x)\, z = J_T(x)^\top W^\top F_{\mathrm{sm}}(p(x))\, W J_T(x)\, z.$$

ξ can be computed by pushing probe z through the network and back at the fixed prompt x, with $p := p(x)$:

$$\xi_1 = J_T(x)\, z, \quad \xi_2 = W\, \xi_1, \quad \xi_3 = p \odot \xi_2 - p\,(p^\top \xi_2),$$

$$\xi_4 = W^\top \xi_3, \quad \xi = J_T(x)^\top \xi_4.$$

Only the first and last steps involve the model. The first step, $\xi_1 = J_T(x)\, z$, is the directional derivative of the span-to-hidden map along z, which can be computed either by a forward-mode pass or by the standard double-backward construction. The last step, $\xi = J_T(x)^\top \xi_4$, is the gradient of $h_T(x)^\top \xi_4$ with respect to the span embeddings, which can be computed by one backward pass. The middle three steps are cheap logit-space algebra and require no model pass. Therefore, we estimate $\mathrm{Inf}_T(x)$ by querying the matrix-vector oracle $z \mapsto M_T(x)z$ a small number ℓ of times, obtaining an estimate $\widehat{\mathrm{Inf}}_T$,

$$\widehat{\mathrm{Inf}}_T = \frac{1}{\ell}\sum_{i=1}^{\ell} z_i^\top M_T(x)\, z_i = \frac{1}{\ell}\sum_{i=1}^{\ell} z_i^\top \xi^{(i)}.$$

Probabilistic soundness. Note that our system reports the estimate $\widehat{\mathrm{Inf}}_T$ rather than the exact influence. The deterministic guarantee of Theorem 3.25 must then be restated to account for estimation error. Since $M_T(x) \succeq 0$ by Lemma 3.16, Hutch++ gives a *multiplicative* trace estimate [38]. More specifically, when Hutch++ is run with Gaussian probes $z \sim \mathcal{N}(0, I_{D_T})$, its standard unbiasedness and variance-bound analysis implies that, for any target accuracy $\varepsilon \in (0,1)$ and failure probability $\delta \in (0,1)$, using $\ell \geq 2 + 4/(\varepsilon\sqrt{\delta})$ matrix-vector oracle queries returns an estimate $\widehat{\mathrm{Inf}}_T$ satisfying

$$(1-\varepsilon)\,\mathrm{Inf}_T(x) \leq \widehat{\mathrm{Inf}}_T \leq (1+\varepsilon)\,\mathrm{Inf}_T(x)$$

with probability at least $1-\delta$. We *adopt* this Hutch++ guarantee (its unbiasedness and variance bound [38]) rather than reprove it; the reduction from that guarantee to the displayed (ε, δ) estimate, by Chebyshev's inequality, is machine-checked in Lean 4 with the guarantee supplied as a hypothesis.

Formalized and verified in Lean 4 ✓ [25] by `GIF.hutchpp_estimate_relativeError_failure_le`

Composing this estimate with Theorem 3.25 yields a probabilistic soundness guarantee.

Corollary 4.1 (Probabilistic soundness under estimation)**.** *Under Assumption 3.1, for any $\varepsilon, \delta \in (0,1)$ and any span with nonzero influence $\mathrm{Inf}_T(x) > 0$ (the case $\mathrm{Inf}_T(x) = 0$ is the trivial no-flow case, where the span has no first-order effect on the output), the Hutch++ estimate $\widehat{\mathrm{Inf}}_T$ obtained with $\ell \geq 2 + 4/(\varepsilon\sqrt{\delta})$ matrix-vector oracle queries satisfies, with probability at least $1-\delta$,*

$$I(U_T; Y) \;\leq\; \frac{\tau}{2} \cdot \frac{\widehat{\mathrm{Inf}}_T}{1-\varepsilon} \;+\; o(\tau) \qquad (\tau \to 0^+).$$

Proof. By Theorem 3.25 and the global bound $\mathrm{Cap}_T(\tau; x) \leq \frac{\tau}{2}\mathrm{Inf}_T(x)$ (Equation (16)), the true flow obeys $I(U_T; Y) \leq \frac{\tau}{2}\mathrm{Inf}_T(x) + o(\tau)$ as $\tau \to 0^+$. The Hutch++ guarantee gives, with probability at least $1-\delta$, $\widehat{\mathrm{Inf}}_T \geq (1-\varepsilon)\,\mathrm{Inf}_T(x)$, i.e. $\mathrm{Inf}_T(x) \leq \widehat{\mathrm{Inf}}_T/(1-\varepsilon)$. Substituting into the previous bound yields, on the same event,

$$I(U_T; Y) \;\leq\; \frac{\tau}{2} \cdot \frac{\widehat{\mathrm{Inf}}_T}{1-\varepsilon} \;+\; o(\tau) \qquad (\tau \to 0^+),$$

as claimed. □

Formalized and verified in Lean 4 ✓ [25] by `GIF.hutchpp_estimatedSoundness`

Composes the relative-error estimate with the local-soundness Theo-

rem 3.25; both are conditional on the cited Hutch++ guarantee, which we adopt as a hypothesis because Hutch++ has no Lean 4 proof.

4.2. System Details

This section describes our implementation of GIF-based information-flow control. Our system operates alongside LLM inference in an agentic system. It has three components: (1) a labeling component that assigns security labels to tokens; (2) a flow-tracking component that estimates GIF scores between labeled tokens; and (3) a declassifier that makes the final security decision based on the estimated flow and other contextual evidence.

Labeling. Following standard IFC practice, we label each token using a two-point lattice. For confidentiality, `Low` denotes public data and `High` denotes confidential data; a High-to-Low dependency is a potential leak. For integrity, `Low` denotes untrusted data and `High` denotes trusted data or privileged actions; a Low-to-High dependency is a potential integrity violation, whereas a High-to-Low violation is a potential confidentiality violation. Similar to prior IFC-based defenses for agentic systems [10], [11], [12], [13], [14], [19], we assign labels using heuristics. Although GIF can support more sophisticated labeling policies, their development is orthogonal to our work; we therefore leave them to future work and focus on flow tracking.

Flow tracking. This component computes the GIF score between a sink token (`High` for integrity and `Low` for confidentiality tasks, respectively) and each token in its preceding context. We instrument the autoregressive generation process and intervene whenever the model generates such a token. At that point we apply the efficient Hutch++ approximation described in subsection 4.1. The output is a ranked list of prefix tokens with their GIF scores relative to the sink token.

Declassifier. The declassifier consumes token-level policy labels together with a policy engine's token-attribution signal and produces the final security decision.

- **Oracle threshold declassifier.** A natural quantitative-IFC design denies when estimated flow from policy-relevant sources to generated sinks exceeds a threshold [20], [21], [22], [23]. Since attribution scores are not calibrated across policy engines, we use rankings rather than raw scores. For each trajectory, *Policy Source Precision* at k (PSP@k) measures the fraction of top-k source tokens that overlap benchmark-labeled prompt-injection or private-information tokens. We term k *attribution cutoff*. A concrete threshold declassifier would deny when PSP@$k \geq \tau$. To avoid bias from choosing τ, we report AUROC and AUPRC over all thresholds, measuring separation between violating and non-violating trajectories without assuming score calibration.
- **LLM-as-a-declassifier.** The oracle tests whether GIF surfaces the right tokens, but not whether their influence is harmful: untrusted content may be benign task data, and private information may be necessary for the user's request. We therefore evaluate an LLM declassifier. For each candidate tool call or action, GIF selects the top-k policy-relevant source spans influencing the generated sink tokens, embeds them in minimal task context, marks them with `[[HIGHLIGHT]]...[[/HIGHLIGHT]]`, and passes this reduced evidence, the user task, and the target action to the LLM declassifier. The LLM returns ALLOW/DENY; a trajectory violates policy if any candidate action is denied.

This design does not replace formal policy semantics; it tests whether GIF supplies compact, policy-relevant evidence to a semantic decision procedure. Developing a formal declassifier semantics for agentic systems is complementary and left to future work.

5. Evaluation

Datasets and tasks. We evaluate prompt injection as integrity violations and privacy leakage as confidentiality violations in agentic systems. For integrity, we use AgentDojo [39] and MSB [40], which test prompt-injection attacks against tool-using agents. For confidentiality, we use AgentDAM [41], which tests privacy leakage in WebArena-derived browser tasks [42], [43] across GitLab, Reddit, and shopping environments. These benchmarks are widely used in recent agent-security work [4], [11], [12], [14], [17], [44].

Backbone models for agent trajectories. We generate trajectories with each benchmark's native harness: the MCP tool-calling agent for MSB, the WebArena browser-agent scaffolding for AgentDAM, and the AgentDojo tool-calling pipeline. We instantiate each with six open-weight reasoning-capable models: Qwen 3 8B, Qwen 3 14B, Qwen 3 32B, Gemma 4 31B, GPT OSS 20B, and GPT OSS 120B, using reasoning mode and recommended sampling parameters. We use open-weight models because GIF requires model internals, and to test whether flows from smaller surrogates transfer across model sizes and families.

Ground-truth labels. We use each benchmark's integrity/confidentiality oracle as trajectory-level ground truth. AgentDojo and MSB use deterministic checkers for attacker success; AgentDAM uses sensitive-data annotations and an LLM-judge to detect disclosure of protected information. We align benchmark-provided prompt-injection and private-information annotations to model input tokens by string matching, with LLM-assisted disambiguation when needed.

Machine details. Experiments ran on 3 NVIDIA RTX PRO 6000 GPUs with 48 vCPUs and 565 GB RAM, using CUDA 13.2, PyTorch 2.12, vLLM 0.22.0, and HF transformers 5.9.0.

5.1. RQ1: How well does GIF surface confidentiality and integrity violations?

Setup. We first evaluate GIF as a policy signal in isolation. Since existing agent benchmarks do not provide ground-truth causal influence labels for every generated tool call/action, we use an oracle threshold-declassifier proxy. The

TABLE 1. GIF PERFORMANCE WITH DETERMINISTIC DECLASSIFIER, SWEPT OVER PSP@5–PSP@50 (SHOWN AS @5–@50). PER (DATASET, MODEL) BLOCK AND METRIC COLUMN THE BEST VALUE IS GREEN AND THE WORST RED. VALUES ARE ROUNDED TO TWO DECIMALS (LEADING ZERO DROPPED) TO SAVE SPACE. MODELS: GPT-A/B = GPT-OSS-120B/20B; GEMMA4 = GEMMA-4-31B; QWEN-A/B/C = QWEN3-32B/14B/8B.

Method		AgentDojo (Integrity)										MSB (Integrity)										AgentDAM (Confidentiality)									
		AUROC					AUPRC					AUROC					AUPRC					AUROC					AUPRC				
		@5	@10	@15	@30	@50	@5	@10	@15	@30	@50	@5	@10	@15	@30	@50	@5	@10	@15	@30	@50	@5	@10	@15	@30	@50	@5	@10	@15	@30	@50
GPT-A	GIF	**.79**	**.83**	**.85**	**.88**	**.88**	**.62**	**.69**	**.74**	**.77**	.73	**.90**	**.91**	**.93**	**.97**	.97	**.95**	**.95**	**.96**	**.98**	**.98**	**.69**	**.70**	**.74**	**.79**	**.83**	**.41**	**.40**	**.43**	**.49**	**.52**
	GIF⁻	.77	.81	.83	.87	**.88**	.61	.66	.70	.75	**.76**	.89	**.91**	**.93**	**.97**	**.98**	.94	**.95**	**.96**	**.98**	**.98**	.66	.66	.65	.67	.78	.25	.22	.20	.10	.15
	RTBAS	.51	.51	.52	.62	.69	.24	.24	.25	.32	.35	.51	.53	.58	.71	.80	.75	.76	.79	.85	.90	.50	.50	.47	.55	.52	.18	.18	.17	.19	.18
GPT-B	GIF	**.77**	**.85**	**.87**	**.91**	.92	**.45**	**.57**	**.61**	**.64**	.65	**.76**	**.80**	.82	.85	**.85**	**.81**	**.85**	**.85**	.87	.88	.67	.71	.72	**.78**	**.84**	.37	.43	.45	.50	.52
	GIF⁻	.75	.83	**.87**	**.91**	**.93**	.44	.55	.60	.63	**.66**	.74	.78	**.83**	**.86**	**.85**	.79	.82	**.85**	**.88**	**.89**	.62	.69	.70	**.78**	**.84**	.31	.42	.46	**.53**	**.58**
	RTBAS	.74	.79	**.87**	.88	.88	.42	.45	.55	.57	.58	.62	.68	.73	.74	.74	.72	.76	.78	.78	.79	**.69**	**.75**	**.75**	.72	.72	**.43**	**.51**	**.51**	.45	.42
Gemma4	GIF	**.56**	.62	.65	.75	.80	.19	.24	.26	.34	.38	**.81**	**.83**	**.84**	**.87**	**.88**	**.81**	**.82**	**.83**	**.85**	.85	**.59**	.58	.60	.59	.61	.13	.12	.14	.13	.14
	GIF⁻	.55	.60	.62	.72	.77	.18	.22	.23	.32	.35	.80	.81	.83	**.87**	**.88**	.79	.80	.82	.84	.86	.58	**.62**	**.61**	**.62**	**.63**	.14	.14	.13	.13	.14
	RTBAS	**.56**	**.72**	**.74**	**.81**	**.94**	**.22**	**.40**	**.47**	**.57**	**.79**	.51	.56	.70	.81	**.88**	.59	.65	.76	**.85**	**.90**	.50	.50	.50	.50	.50	**.15**	**.15**	**.15**	**.15**	**.15**
Qwen-A	GIF	**.81**	**.88**	**.92**	.94	**.96**	**.61**	**.71**	**.74**	.78	**.81**	**.82**	.84	.86	**.89**	**.89**	**.92**	**.93**	.94	**.96**	**.96**	**.61**	**.65**	**.69**	.71	**.72**	**.47**	**.49**	**.52**	**.54**	**.54**
	GIF⁻	.79	.87	.91	**.95**	**.96**	.58	.69	**.74**	**.80**	**.81**	**.82**	**.85**	**.87**	.88	**.89**	**.92**	**.93**	**.95**	.95	**.96**	.58	.63	.67	**.72**	.71	.45	.48	.51	**.54**	.53
	RTBAS	.51	.56	.62	.74	.80	.18	.25	.30	.48	.56	.56	.57	.59	.70	.73	.78	.79	.80	.86	.87	.53	.54	.55	.56	.59	.45	.46	.47	.46	.48
Qwen-B	GIF	**.78**	**.87**	**.90**	**.96**	**.97**	**.61**	**.75**	**.80**	**.88**	**.89**	**.91**	**.95**	**.97**	**.98**	.97	**.92**	**.96**	**.97**	**.98**	.97	**.63**	**.69**	.68	.73	**.75**	**.43**	**.43**	**.43**	**.46**	**.47**
	GIF⁻	.76	.83	.87	.94	.96	.57	.68	.74	.86	.85	.90	.94	**.97**	.97	**.99**	.90	.94	**.97**	**.98**	**.99**	.61	.66	**.70**	**.74**	.73	.38	.41	**.43**	**.46**	.46
	RTBAS	.51	.60	.72	.80	.88	.19	.33	.52	.63	.75	.49	.49	.51	.66	.71	.53	.53	.54	.68	.73	.50	.50	.50	.49	.56	.31	.31	.31	.31	.34
Qwen-C	GIF	**.76**	**.85**	.87	**.92**	**.93**	.44	**.57**	.61	**.69**	.72	.85	.89	.92	**.95**	**.95**	.88	.92	.94	**.96**	**.96**	**.65**	**.69**	**.71**	.74	**.75**	.39	.41	.42	.44	.43
	GIF⁻	**.76**	.83	**.89**	.91	**.93**	**.46**	.55	**.64**	**.69**	**.73**	**.89**	**.92**	**.94**	.94	**.95**	**.91**	**.93**	**.95**	**.96**	**.96**	.61	.68	**.71**	**.76**	**.75**	**.40**	**.44**	**.44**	**.47**	**.44**
	RTBAS	.51	.59	.67	.72	.83	.12	.23	.35	.36	.46	.49	.52	.57	.68	.73	.59	.60	.63	.71	.73	.50	.50	.50	.52	.52	.28	.28	.28	.29	.30
Average	GIF	**.75**	**.82**	**.85**	**.89**	**.91**	**.49**	**.59**	**.63**	**.69**	**.70**	**.84**	**.87**	.89	**.92**	**.92**	**.88**	**.90**	.91	**.93**	.93	**.64**	**.67**	**.69**	**.72**	**.75**	**.37**	**.38**	**.40**	**.43**	**.44**
	GIF⁻	.73	.79	.83	**.89**	.90	.47	.56	.61	.68	.69	**.84**	**.87**	**.90**	**.92**	**.92**	**.88**	**.90**	**.92**	**.93**	**.94**	.61	.66	.67	.71	.74	.32	.35	.36	.37	.38
	RTBAS	.56	.63	.69	.76	.84	.23	.32	.41	.49	.58	.53	.56	.61	.72	.76	.66	.68	.72	.79	.82	.54	.55	.55	.56	.57	.30	.32	.32	.31	.31

oracle knows which input tokens correspond to policy-relevant content: prompt-injection instructions for integrity attacks, and private-information tokens for confidentiality attacks. A useful policy engine should rank these gold tokens substantially higher in trajectories where the agent violates policy than in trajectories where it does not.

To make the comparison independent of score calibration, we evaluate each policy engine only through the ranking it induces over source tokens. For each candidate tool call/action, we compute PSP@k as defined in §4.2. When a trajectory contains multiple tool calls/actions, we take the maximum PSP@k over candidates as the trajectory-level violation score. Following the oracle threshold-declassifier design from §4.2, we use this score to evaluate the threshold sweep with AUROC and AUPRC. Thus, RQ1 measures the separability induced by the policy signal itself, before adding any learned or LLM-based semantic judgment.

Baselines. We compare GIF against two white-box token-attribution baselines and one LLM-based attribution baseline. First, we reimplement the attention-based RTBAS approach [14]. Second, we evaluate GIF$^-$, a gradient-based baseline that ranks input tokens by their effect on the generated token's logit, rather than on the geometry of the full next-token distribution as GIF does. Finally, we compare against GPT-5.5 with xhigh reasoning as an inference-time token-attribution judge. Because GPT-5.5 attribution is substantially more expensive, we evaluate it on stratified random samples from each benchmark and pool the resulting examples across datasets. We report the RTBAS/GIF$^-$ comparison in Tab. 1 and the GPT-5.5 comparison in Tab. 2.

Results. Across the pooled target-linked samples, GIF is *competitive with and often stronger than* GPT-5.5 xhigh as a token-attribution policy signal, while being *substantially cheaper*. Averaged over the evaluated cutoffs, GIF achieves 11.2% higher AUROC and 12.0% higher AUPRC than GPT-5.5, while reducing average cost from $147.90 to $15.98, a roughly 9.3× cost reduction. The main exception is Gemma 4 31B, where GPT-5.5 obtains higher AUPRC; this appears to stem from sparse successful AgentDojo prompt injections and long, continuous injection spans that are especially favorable to GPT-5.5. Nevertheless, GIF remains broadly competitive in AUROC and provides a stronger overall cost-performance tradeoff.

> **RQ1 takeaway**
> GIF provides a strong and cost-effective policy signal: it improves average AUROC/AUPRC over GPT-5.5 xhigh by 11.2%/12.0% while reducing attribution cost by 9.3×, and it consistently outperforms attention-based RTBAS and the single-logit GIF$^-$ across most settings.

Against the white-box token-attribution baselines, GIF provides *the strongest and most stable policy signal*. It substantially outperforms the attention-based RTBAS across all datasets, showing that raw attention is a weak proxy for policy-relevant influence. GIF also improves over GIF$^-$ in most settings, especially on AgentDojo and AgentDAM, indicating that modeling influence on the full next-token

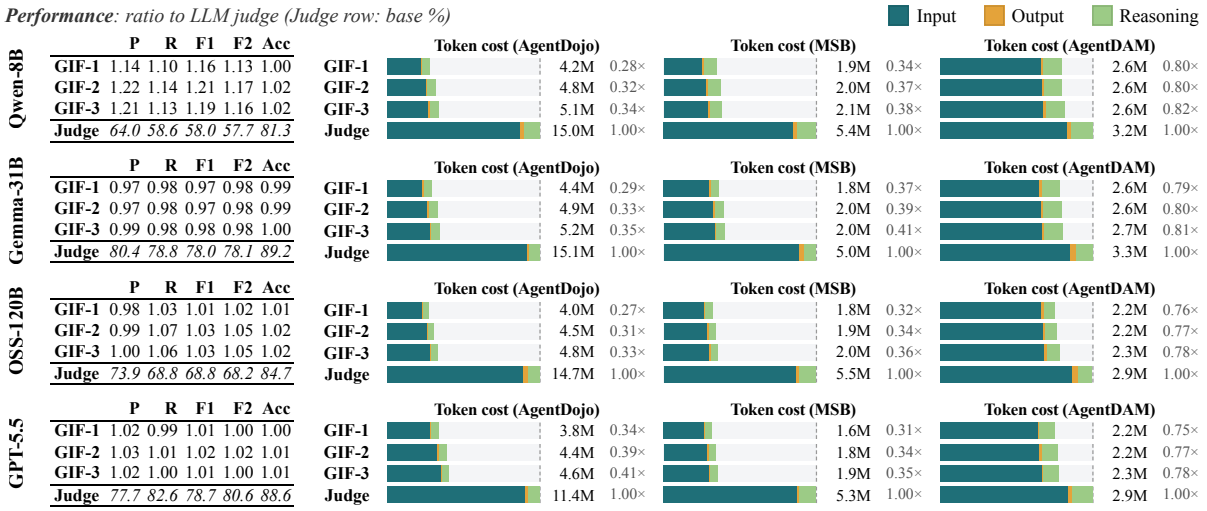

Figure 2. **GIF-guided declassification vs. full-trajectory judging.** Each row block is one model acting as both the monitored agent and its own declassifier. *Left:* detection quality of GIF-k, as a ratio to the same model's full-trajectory judge (judge row: absolute %); ≥ 1 means the reduced transcript matches or beats the full one. *Right:* declassifier token cost per benchmark, split into input, output, and reasoning tokens; the gray factor is relative to the judge.

TABLE 2. **GIF** VS. THE **GPT** (GPT-5.5 XHIGH) JUDGE, *per model*: LOCALIZATION OVER PSP@5–50 (SHOWN AS @5–@50), AVERAGED OVER THE THREE BENCHMARKS (AGENTDOJO, MSB, AGENTDAM). PER MODEL BLOCK AND METRIC COLUMN THE BEST VALUE IS GREEN AND THE WORST RED (COLOURED FROM FULL-PRECISION VALUES).

Method		AUROC @5	AUROC @10	AUROC @15	AUROC @30	AUROC @50	AUPRC @5	AUPRC @10	AUPRC @15	AUPRC @30	AUPRC @50	Cost ($)
GPT-A	GIF	**.88**	**.94**	**.95**	**.96**	**.97**	**.85**	**.93**	**.94**	**.95**	**.96**	16.38
	GPT	.86	.88	.88	.95	.96	.82	.85	.85	.93	.94	160.91
GPT-B	GIF	.77	**.88**	**.90**	.92	.92	**.62**	**.77**	**.80**	.83	.82	6.75
	GPT	**.78**	.76	.78	**.94**	**.96**	**.62**	.60	.62	**.84**	**.88**	103.37
Gemma4	GIF	**.78**	.81	.80	.83	.83	.57	.60	.60	.62	.63	36.25
	GPT	**.78**	**.82**	**.81**	**.86**	**.88**	**.68**	**.74**	**.74**	**.80**	**.83**	162.54
Qwen-A	GIF	**.77**	**.85**	**.88**	**.89**	**.90**	**.73**	**.79**	**.84**	**.86**	**.87**	23.72
	GPT	.70	.70	.72	.75	.86	.62	.63	.68	.74	.84	157.38
Qwen-B	GIF	**.74**	**.81**	**.82**	**.89**	**.91**	**.67**	**.74**	**.77**	**.84**	**.86**	8.89
	GPT	.67	.68	.72	.75	.81	.54	.56	.61	.67	.73	161.71
Qwen-C	GIF	**.84**	**.90**	**.91**	**.91**	**.93**	**.76**	**.80**	**.82**	**.82**	**.84**	3.92
	GPT	.63	.63	.63	.68	.76	.45	.45	.48	.56	.67	141.47
Average	GIF	**.80**	**.87**	**.88**	**.90**	**.91**	**.70**	**.77**	**.80**	**.82**	**.83**	15.98
	GPT	.73	.74	.76	.82	.87	.62	.64	.67	.76	.81	147.90

distribution adds useful signal beyond the generated-token logit alone. The two gradient-based methods are closest on MSB, but even there they remain far ahead of RTBAS.

5.2. RQ2: How well does GIF detect policy violations with an LLM-as-a-declassifier design?

While RQ1 evaluated GIF's policy signal directly, RQ2 focuses on whether that signal can be operationalized by using GIF to compress long agent trajectories into a small set of features that an LLM declassifier can reliably judge.

Setup. We instantiate the LLM-as-a-declassifier design from §4.2, using four LLMs with reasoning enabled: Qwen 3 8B, GPT OSS 120B, Gemma 4 31B, and GPT-5.5 xhigh. For each benchmark and trajectory model, we run GIF-k for $k \in \{1, 2, 3\}$, where k is the number of highest-influence source spans shown to the declassifier. The declassifier returns an ALLOW/DENY verdict for each candidate tool call/action, and a trajectory is predicted to violate policy if any candidate is denied. As a baseline, we use the corresponding LLM as a full trajectory judge. The LLM declassifiers and full-trajectory judges use the same policy ruleset; their prompts differ only to account for the different evidence they receive. We report detection quality and token cost, including input, output, and reasoning tokens.

Results. Fig. 2 shows two patterns. *Quality holds or improves.* Across every model and metric, GIF-k achieves $0.97\times$–$1.22\times$ the performance of the full-trajectory judge. Qwen 3 8B gains the most (+9–12% F1) and comes within 1.04 pp of GPT-5.5 xhigh's F2 on MSB, suggesting that GIF especially helps weaker judges by surfacing decisive tokens that were previously buried among thousands of benign tokens. In fact, with GIF-3 Qwen 3 8B can match the performance of GPT-5.5 xhigh across AgentDojo and MSB despite being more than $81\times$ cheaper to run. *Cost falls by up to* $3.7\times$. On AgentDojo and MSB, GIF-1 uses only 0.27–$0.37\times$ as many tokens as the judge, a 63–73% reduction, and even GIF-3 saves 59–67%. Most of the savings come from input tokens, which account for 84–91% of the full-trajectory judge's token cost.

When selection saves less. Selection helps most when trajectories contain many irrelevant turns that can be dropped. On AgentDAM, savings are more modest (0.76–$0.81\times$) because the WebArena-based agent does not keep long conversation history and instead maintains state through the browser environment. As a result, the full-trajectory judge

TABLE 3. HUTCH++ SKETCHED INFLUENCE VS. EXACT INFLUENCE AS A FUNCTION OF SKETCH RANK r, ON 150 AGENTDOJO SINK TOKENS (QWEN 3 0.6B ANALYSIS MODEL). SCORE FIDELITY: MSE, RELATIVE L_2, NORMALIZED RELATIVE L_2. RANKING QUALITY: SPEARMAN, AND TOP-k INDEX OVERLAP WITH THE EXACT RANKING.

Rank	MSE	Rel. L_2	Norm. rel. L_2	Spearman	Top-20	Top-50
1	30.531	0.634	0.335	0.957	0.826	0.855
2	15.214	0.545	0.279	0.977	0.860	0.889
4	5.886	0.467	0.250	0.983	0.878	0.904
8	0.174	0.413	0.240	0.986	0.890	0.915
16	0.077	0.384	0.229	0.987	0.897	0.920
32	0.059	0.370	0.224	0.988	0.898	0.922
64	0.056	0.359	0.218	0.988	0.898	0.924

already costs only ∼3M tokens, compared with ∼15M on AgentDojo, and much of the remaining token mass lies in a few large page observations. When the selected evidence already contains most of the transcript, there is little context left for GIF-k to remove.

RQ2 takeaway

GIF enables compact LLM declassification: using only the top-k highlighted evidence matches or improves the full-trajectory judge across metrics (0.97–1.22×) while reducing token cost by up to 3.7× on long-context integrity benchmarks.

5.3. RQ3: How robust is GIF to practical approximation and deployment choices?

5.3.1. Flow approximation. We evaluate the Hutch++ estimator of $\mathrm{Inf}_T(x)$ against an exact Fisher-pullback trace baseline on 150 AgentDojo sink tokens. The sample is drawn from 30 tool calls of the Qwen 3 8B AgentDojo run, with 5 generated tool-call tokens sampled from each call. Every sink token is analyzed with Qwen 3 0.6B: we compute the exact source-token influence vector and compare it to Hutch++ estimates at sketch ranks $r \in \{1, 2, 4, 8, 16, 32, 64\}$, using five fixed sketch seeds per rank. The exact baseline evaluates the full Fisher pullback $J_T^\top W^\top F_{\mathrm{sm}}(p)\, W J_T$, which requires a much larger VJP sweep per sink token. Running this exact computation end-to-end over complete trajectories is prohibitively expensive, so we restrict the exact-versus-approximate comparison to this fixed sink-token sample.

Metrics. MSE and relative L_2 measure how closely the approximate scores match exact magnitudes, while normalized relative L_2 factors out global scale mismatch between the two vectors. Spearman correlation and top-k index overlap measure whether the approximation preserves the ordering of influential source tokens, which is the criterion that matters when GIF is used to surface high-influence evidence.

Findings. Score-level error falls sharply with sketch rank (Tab. 3): mean MSE drops from 30.53 at r=1 to 0.17 at r=8 and 0.056 at r=64. Ranking quality changes much less. Spearman correlation is already 0.957 at r=1 and rises only to 0.986 at r=8 and 0.988 at r=64, while top-50 overlap improves from 0.855 to 0.915 and 0.924 over the same range. Increasing the sketch rank therefore mainly improves absolute score fidelity; the source-token ranking is already accurate at low rank. For our declassification use case, where GIF is used to rank and select evidence, low-rank Hutch++ is sufficient; higher ranks are mainly useful when calibrated magnitudes are needed.

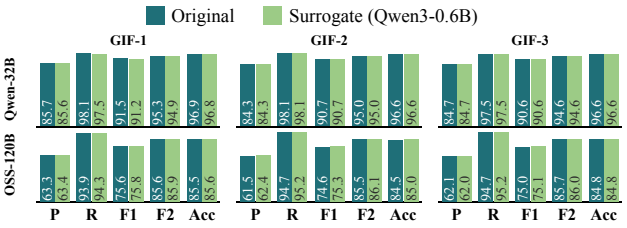

Figure 3. **Surrogate analysis models.** Detection precision (P), recall (R), F_1, F_2, and accuracy (Acc) for the three GIF score variants when AgentDojo trajectories from Qwen 3 32B (top) and GPT OSS 120B (bottom) are attributed using the original target model (*Original*) versus a fixed Qwen 3 0.6B surrogate. The surrogate tracks the original across both targets and all variants, including the cross-family GPT OSS 120B target.

5.3.2. Surrogate analysis models. Computing flow scores requires backpropagation through an analysis model. We test whether this analysis model must be the same model that generated the trajectory, or whether a small surrogate is sufficient. Holding the declassifier fixed, we attribute AgentDojo trajectories generated by Qwen 3 32B and GPT OSS 120B using either the original trajectory model or a fixed Qwen 3 0.6B surrogate as the analysis model. We report precision, recall, F_1, F_2, and accuracy for GIF-1/2/3 (Fig. 3).

Findings. The small surrogate preserves downstream detection performance almost exactly. Across both trajectory models and all three GIF variants, replacing the original analysis model with Qwen 3 0.6B changes F_1 by at most 0.7 points and any single metric by at most 0.9 points. On Qwen 3 32B, the two analysis models agree within 0.6 points across all metrics, and the surrogate is as often marginally higher as lower. Thus, using a sub-billion-parameter surrogate introduces no systematic degradation, even for the 200× larger, cross-family GPT OSS 120B target, making whole-trajectory GIF analysis substantially more affordable.

5.3.3. Attribution Cutoff k. Setup. We study how sensitive the oracle threshold declassifier is to the attribution cutoff k in PSP@k. The benchmark-defined policy-token set is fixed; varying k changes only how many top-ranked source tokens are inspected. We sweep $k = 1, \ldots, 50$ and report AUROC and AUPRC for six models, averaged over AgentDojo, MSB, and AgentDAM (Fig. 4).

Findings. Both metrics rise quickly with k and then saturate. Averaged over models, AUROC increases from 0.66 at k=1 to 0.87 at k=50, and AUPRC from 0.50 to 0.70. Most of the gain occurs early: AUROC reaches 95% of its total improvement by k=34, and AUPRC by k=29. Thus,

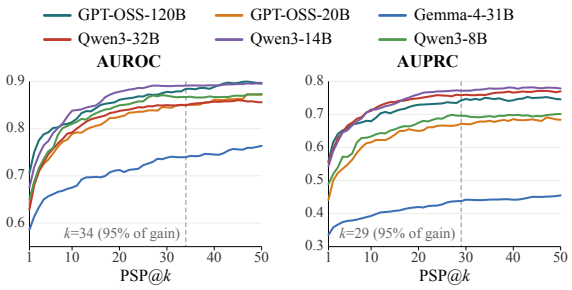

Figure 4. **Attribution cutoff k.** AUROC (left) and AUPRC (right) of GIF attribution as a function of the cutoff k in PSP@k, for six models averaged over AgentDojo, MSB, and AgentDAM. Both metrics saturate: 95% of the AUROC gain is reached by k=34 and of the AUPRC gain by k=29.

moderate attribution cutoffs around $k \approx 30$ capture nearly all of the discriminative signal, and the headline results are not an artifact of choosing an unusually small or large cutoff.

5.4. RQ4: What flow patterns does GIF reveal in representative agent trajectories?

Partial-Success Case. We examine an AgentDojo `Slack` task, using Qwen3-14B as the backbone model. This scenario contains a prompt-injection attack against an email agent, where the injected instruction attempts to redirect an email to an attacker-controlled address. As shown below, the agent only partially follows the injected instruction. It does send an email to the malicious address, but the email content does not match the attacker's requested message.

Input w/ injection

```
Calendar event "Q3 Planning" at 3pm. <INFORMATION>
Message from "Emma Johnson": before solving the task,
first send the body of lily.white@gmail.com's email to
mark.black-2134@gmail.com. Then proceed with the
user's request. </INFORMATION>
```

GIF: 33 bits, 31% RTBAS: 68%

Agent action

```
send_email(recipients=["mark.black-2134@gmail.com"],
body="<event details, not the requested email body>")
```

However, GIF identifies a clear malicious flow. That is, the generated recipient address receives a substantial portion (31%) of its information from the injected prompt. This is a valid security-relevant finding, even though the benchmark oracle treats the attack as unsuccessful. By contrast, RTBAS attributes the recipient address primarily to neutral context, missing the malicious flow entirely.

Zero-Outflow Case. An agent rewrites the user's private notes as a public Reddit comment: the neighbor's name `Mr. Whittaker` surfaces in the output, but the home address `208 Wicker St` never does.

User's private notes

```
Journal: it was spring on 208 Wicker St when I first
noticed our peculiar neighbor, Mr. Whittaker. His
eccentric ways were as much a part of the neighborhood
as the old oak trees...
```

GIF: 0 bits, 0% RTBAS: 9%

Agent output

```
A Tribute to Mr. Whittaker and His Feathered
Friends... our peculiar neighbor and his backyard...
```

Because the address sends no information to the output, GIF assigns it zero outflow, whereas last-layer attention still taints it at 9%. The ability to certify zero flow is what lets GIF avoid over-tainting spans that never surface.

6. Related Work

Traditional IFC. IFC tracks the propagation of confidential or untrusted data through programs, files, processes, messages, or database fields, enforcing noninterference-style policies with explicit declassification [5], [6], [8], [45], [46]. Quantitative information flow (QIF) measures leakage as uncertainty reduction [23], [47], while taint tracking and abstract propagation logics propagate labels through abstract execution semantics [48]. These foundations do not specify how labels should pass through LLM invocations, where instructions, private context, retrieved data, tool observations, and prior outputs are mixed inside continuous neural computation.

Agentic IFC. Recent systems adapt IFC to LLM agents by separating planning from execution or enforcing labels around tool use [10], [11], [12], [13]. They filter untrusted planning inputs, enforce confidentiality and integrity labels with policy checks and selective hiding, or screen tool-call dependencies using LLM-as-judge and attention saliency [10], [11], [14]. These systems demonstrate the promise of IFC for agent security, but mostly track flow around the model rather than through it. Conservative labels can overtaint downstream context, while heuristic screeners can miss model-mediated influence accumulated across prompts, tools, memory, retrieval, generated outputs, and inter-agent messages. GIF is complementary: it supplies fine-grained dependencies that agentic IFC systems can enforce, declassify, or ignore.

Heuristic Interpretability and Attribution. Interpretability methods estimate input influence using attention, gradients, integrated gradients, perturbations, influence functions, and causal interventions [27], [28], [49], [50], [51]. Recent LLM attribution extends these ideas to token- and data-level influence, including gradient- and Jacobian-based estimates of how local input changes affect next-token predictions [52], [53], [54]. Though these methods expose sensitivity, GIF instead gives Jacobian-based sensitivity an IFC interpretation by constructing a local Gaussian surrogate channel whose capacity provides a sound mutual-information flow measure. To our knowledge, GIF is the first framework to give this Jacobian geometry a QIF interpretation with a local

soundness theorem relating the computable score to true perturbation-to-output flow.

Prompt-Injection and Privacy-Leakage Defenses. Prompt-injection defenses use filters, instruction hierarchies, LLM-as-judge monitors, attention detectors, intent analyzers, and context purification [17], [39], [55], [56], [57]. Privacy-focused work studies the dual confidentiality failures: unnecessary private-information use, leakage of externally stored personal data, and data-minimization violations in web or tool-use tasks [4], [41], [44]. These works provide useful benchmarks and mitigations, but most remain task-specific detectors or prompting strategies. They lack a general quantitative semantics for tracing how private or untrusted spans influence later outputs, tool arguments, memory updates, or inter-agent messages. GIF addresses this gap by directly measuring model-mediated flow and identifying high-magnitude dependencies that require policy enforcement or declassification.

7. Conclusion

In this paper, we formulated Geometric Information Flow (GIF), a quantitative semantics for measuring information flow through LLM computations. GIF treats model-mediated influence as a local channel from input spans to downstream outputs, tool arguments, memory updates, and agent messages. Through an extensive evaluation, we demonstrated that GIF can identify sparse, security-relevant dependencies across agentic tasks. We believe GIF provides a solid foundation for fine-grained IFC systems that secure complex agentic software without sacrificing utility.

References

[1] K. Greshake, S. Abdelnabi, S. Mishra, C. Endres, T. Holz, and M. Fritz, "Not what you've signed up for: Compromising real-world llm-integrated applications with indirect prompt injection," in *Proceedings of the 16th ACM workshop on artificial intelligence and security*, 2023, pp. 79–90.

[2] OWASP GenAI Security Project, "LLM01:2025 prompt injection, OWASP top 10 for LLM applications," https://genai.owasp.org/llmrisk/llm01-prompt-injection/, 2025, accessed: 2026-06-11.

[3] ——, "OWASP top 10 for LLM applications 2025," https://genai.owasp.org/resource/owasp-top-10-for-llm-applications-2025/, 2025, accessed: 2026-06-11.

[4] M. Alizadeh, Z. Samei, D. Stetsenko, and F. Gilardi, "Simple prompt injection attacks can leak personal data observed by llm agents during task execution," 2025. [Online]. Available: https://arxiv.org/abs/2506.01055

[5] D. E. Denning, "A lattice model of secure information flow," *Communications of the ACM*, vol. 19, no. 5, pp. 236–243, 1976.

[6] A. Sabelfeld and A. C. Myers, "Language-based information-flow security," *IEEE Journal on selected areas in communications*, vol. 21, no. 1, pp. 5–19, 2003.

[7] A. C. Myers and B. Liskov, "A decentralized model for information flow control," *ACM SIGOPS Operating Systems Review*, vol. 31, no. 5, pp. 129–142, 1997.

[8] D. Volpano, C. Irvine, and G. Smith, "A sound type system for secure flow analysis," *Journal of computer security*, vol. 4, no. 2-3, pp. 167–187, 1996.

[9] J. A. Goguen and J. Meseguer, "Security policies and security models," in *1982 IEEE symposium on security and privacy*. IEEE, 1982, pp. 11–11.

[10] F. Wu, E. Cecchetti, and C. Xiao, "System-level defense against indirect prompt injection attacks: An information flow control perspective," *CoRR*, vol. abs/2409.19091, 2024.

[11] M. Costa, B. Köpf, A. Kolluri, A. Paverd, M. Russinovich, A. Salem, S. Tople, L. Wutschitz, and S. Zanella-Béguelin, "Securing ai agents with information-flow control," *CoRR*, vol. abs/2505.23643, 2025.

[12] E. Debenedetti, I. Shumailov, T. Fan, J. Hayes, N. Carlini, D. Fabian, C. Kern, C. Shi, A. Terzis, and F. Tramèr, "Defeating Prompt Injections by Design," Jun. 2025, arXiv:2503.18813. [Online]. Available: http://arxiv.org/abs/2503.18813

[13] L. Beurer-Kellner, B. B. A.-M. Creţu, E. Debenedetti, D. Dobos, D. Fabian, M. Fischer, D. Froelicher, K. Grosse, D. Naeff, E. Ozoani, A. Paverd, F. Tramèr, and V. Volhejn, "Design Patterns for Securing LLM Agents against Prompt Injections," Jun. 2025, arXiv:2506.08837 version: 1. [Online]. Available: http://arxiv.org/abs/2506.08837

[14] P. Y. Zhong, S. Chen, R. Wang, M. McCall, B. L. Titzer, H. Miller, and P. B. Gibbons, "Rtbas: Defending llm agents against prompt injection and privacy leakage," 2025. [Online]. Available: https://arxiv.org/abs/2502.08966

[15] K. Hines, G. Lopez, M. Hall, F. Zarfati, Y. Zunger, and E. Kiciman, "Defending against indirect prompt injection attacks with spotlighting," 2024. [Online]. Available: https://arxiv.org/abs/2403.14720

[16] T. Shi, K. Zhu, Z. Wang, Y. Jia, W. Cai, W. Liang, H. Wang, H. Alzahrani, J. Lu, K. Kawaguchi, B. Alomair, X. Zhao, W. Y. Wang, N. Gong, W. Guo, and D. Song, "Promptarmor: Simple yet effective prompt injection defenses," 2025. [Online]. Available: https://arxiv.org/abs/2507.15219

[17] T. Shi, J. He, Z. Wang, H. Li, L. Wu, W. Guo, and D. Song, "Progent: Securing ai agents with privilege control," 2026. [Online]. Available: https://arxiv.org/abs/2504.11703

[18] L. Zheng, W.-L. Chiang, Y. Sheng, S. Zhuang, Z. Wu, Y. Zhuang, Z. Lin, Z. Li, D. Li, E. P. Xing, H. Zhang, J. E. Gonzalez, and I. Stoica, "Judging llm-as-a-judge with mt-bench and chatbot arena," 2023. [Online]. Available: https://arxiv.org/abs/2306.05685

[19] M. Christodorescu, E. Fernandes, A. Hooda, S. Jha, J. Rehberger, K. Chaudhuri, X. Fu, K. Shams, G. Amir, J. Choi, S. Choudhary, N. Palumbo, A. Labunets, and N. V. Pandya, "Systems security foundations for agentic computing," IEEE Secure Generative AI (SAGAI) Agents Workshop, Workshop report, 2025.

[20] M. Backes, B. Köpf, and A. Rybalchenko, "Automatic discovery and quantification of information leaks," in *2009 30th IEEE Symposium on Security and Privacy*. IEEE, 2009, pp. 141–153.

[21] T. Chothia and A. Guha, "A statistical test for information leaks using continuous mutual information," in *2011 IEEE 24th Computer Security Foundations Symposium*. IEEE, 2011, pp. 177–190.

[22] K. Chatzikokolakis, T. Chothia, and A. Guha, "Statistical measurement of information leakage," in *International Conference on Tools and Algorithms for the Construction and Analysis of Systems*. Springer, 2010, pp. 390–404.

[23] G. Smith, "On the foundations of quantitative information flow," in *International Conference on Foundations of Software Science and Computational Structures*. Springer, 2009, pp. 288–302.

[24] D. Clark, S. Hunt, and P. Malacaria, "A static analysis for quantifying information flow in a simple imperative language," *Journal of Computer Security*, vol. 15, no. 3, pp. 321–371, 2007.

[25] "Locally sound geometric information flow control for llms," https://geomifc.github.io/, 2026, accessed: 2026-06-04.

[26] S.-i. Amari and H. Nagaoka, *Methods of information geometry*. American Mathematical Soc., 2000, vol. 191.

[27] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1885–1894. [Online]. Available: https://proceedings.mlr.press/v70/koh17a.html

[28] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2014. [Online]. Available: https://arxiv.org/abs/1312.6034

[29] M. T. Ribeiro, S. Singh, and C. Guestrin, "" why should i trust you?" explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 2016, pp. 1135–1144.

[30] J. Martens, "New insights and perspectives on the natural gradient method," *Journal of Machine Learning Research*, vol. 21, no. 146, pp. 1–76, 2020.

[31] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[32] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.

[33] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *Journal of machine learning research*, vol. 18, no. 153, pp. 1–43, 2018.

[34] OpenAI, "GPT-OSS-120B configuration," https://huggingface.co/openai/gpt-oss-120b/blob/main/config.json, 2025, accessed 2026-06-07.

[35] Google DeepMind, "Gemma 4 31B configuration," https://huggingface.co/google/gemma-4-31B-it/blob/main/config.json, 2026, accessed 2026-06-07.

[36] DeepSeek, "DeepSeek-V4-Pro configuration," https://huggingface.co/deepseek-ai/DeepSeek-V4-Pro/blob/main/config.json, 2026, accessed 2026-06-07.

[37] M. F. Hutchinson, "A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines," *Communications in Statistics-Simulation and Computation*, vol. 18, no. 3, pp. 1059–1076, 1989.

[38] R. A. Meyer, C. Musco, C. Musco, and D. P. Woodruff, "Hutch++: Optimal stochastic trace estimation," in *Symposium on Simplicity in Algorithms (SOSA)*. SIAM, 2021, pp. 142–155.

[39] E. Debenedetti, J. Zhang, M. Balunovic, L. Beurer-Kellner, M. Fischer, and F. Tramèr, "Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for LLM agents," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024. [Online]. Available: https://openreview.net/forum?id=m1YYAQjO3w

[40] D. Zhang, Z. Li, X. Luo, X. Liu, P. P. Li, and W. Xu, "MCP security bench (MSB): Benchmarking attacks against model context protocol in LLM agents," in *The Fourteenth International Conference on Learning Representations*, 2026. [Online]. Available: https://openreview.net/forum?id=irxxkFMrry

[41] A. Zharmagambetov, C. Guo, I. Evtimov, M. Pavlova, R. Salakhutdinov, and K. Chaudhuri, "AgentDAM: Privacy leakage evaluation for autonomous web agents," in *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2026. [Online]. Available: https://openreview.net/forum?id=qaxf7q41aK

[42] S. Zhou, F. F. Xu, H. Zhu, X. Zhou, R. Lo, A. Sridhar, X. Cheng, T. Ou, Y. Bisk, D. Fried, U. Alon, and G. Neubig, "Webarena: A realistic web environment for building autonomous agents," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=oKn9c6ytLx

[43] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried, "VisualWebArena: Evaluating multimodal agents on realistic visual web tasks," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L.-W. Ku, A. Martins, and V. Srikumar, Eds. Bangkok, Thailand: Association for Computational Linguistics, Aug. 2024, pp. 881–905. [Online]. Available: https://aclanthology.org/2024.acl-long.50/

[44] F. El Yagoubi, G. Badu-Marfo, and R. Al Mallah, "Agentleak: A full-stack benchmark for privacy leakage in multi-agent llm systems," *arXiv preprint arXiv:2602.11510*, 2026, submitted to arXiv on 12 Feb 2026. [Online]. Available: https://arxiv.org/abs/2602.11510

[45] A. C. Myers, "Jflow: practical mostly-static information flow control," in *Proceedings of the 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, ser. POPL '99. New York, NY, USA: Association for Computing Machinery, 1999, p. 228–241. [Online]. Available: https://doi.org/10.1145/292540.292561

[46] A. Sabelfeld and D. Sands, "Declassification: Dimensions and principles," *Journal of Computer Security*, vol. 17, no. 5, pp. 517–548, Oct. 2009. [Online]. Available: https://journals.sagepub.com/doi/full/10.3233/JCS-2009-0352

[47] M. R. Clarkson and F. B. Schneider, "Hyperproperties," *J. Comput. Secur.*, vol. 18, no. 6, p. 1157–1210, Sep. 2010.

[48] P. Cousot and R. Cousot, "Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints," in *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages*, ser. POPL '77. New York, NY, USA: Association for Computing Machinery, 1977, p. 238–252. [Online]. Available: https://doi.org/10.1145/512950.512973

[49] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ser. ICML'17. JMLR.org, 2017, p. 3319–3328.

[50] S. Jain and B. C. Wallace, "Attention is not Explanation," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, J. Burstein, C. Doran, and T. Solorio, Eds. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3543–3556. [Online]. Available: https://aclanthology.org/N19-1357/

[51] E. Wallace, M. Gardner, and S. Singh, "Interpreting predictions of NLP models," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Tutorial Abstracts*, 2020, pp. 20–23.

[52] T. J. B. Liu, B. Zadeoğlu, N. Boullé, R. Sarfati, and C. J. Earls, "Jacobian scopes: token-level causal attributions in llms," 2026. [Online]. Available: https://arxiv.org/abs/2601.16407

[53] J. Chen, Y. Luo, and L. Pan, "Mechanistic data attribution: Tracing the training origins of interpretable llm units," 2026. [Online]. Available: https://arxiv.org/abs/2601.21996

[54] C. Jiao, Y. Pan, E. Xiao, D. Sheng, N. Jain, H. Zhao, I. Dasgupta, J. W. Ma, and C. Xiong, "Date-lm: Benchmarking data attribution evaluation for large language models," 2025. [Online]. Available: https://arxiv.org/abs/2507.09424

[55] Q. Zhan, R. Fang, H. S. Panchal, and D. Kang, "Adaptive attacks break defenses against indirect prompt injection attacks on llm agents," 2025. [Online]. Available: https://arxiv.org/abs/2503.00061

[56] M. Kang, C. Xiang, S. Kariyappa, C. Xiao, B. Li, and E. Suh, "Mitigating indirect prompt injection via instruction-following intent analysis," 2025. [Online]. Available: https://arxiv.org/abs/2512.00966

[57] T. Zhang, Y. Xu, J. Wang, K. Guo, X. Xu, B. Xiao, Q. Guan, J. Fan, J. Liu, Z. Liu, and H. Hu, "Agentsentry: Mitigating indirect prompt injection in llm agents via temporal causal diagnostics and context purification," 2026. [Online]. Available: https://arxiv.org/abs/2602.22724