CINFA: ACTION-CONDITIONED CONTEXTUAL PRIVACY CHECKING FOR LLM AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Contextual privacy in large language model (LLM) agents depends on the recipient, payload, and conditions of a proposed disclosure. We present CINFA, a learned pre-execution checking framework built around context-action commits. Controlled edits of candidate actions provide global and partial recipient, attribute, and transmission labels. A frozen LLM encodes each commit, and logistic heads use these labels to learn risk scores. The runtime combines context-guided generation with action checking, one-step revision, and re-checking; unresolved actions are withheld. Across CI-RL-Synth, PrivacyLens, and ConfAIde, the reported three-benchmark averages are 12.4% versus 5.4% leakage rate and 17.3% versus 7.8% adjusted leakage rate for CI-GRPO and CINFA, respectively, with a 2.6-percentage-point higher task Utility for CINFA. Adjusted leakage is computed on helpful outputs. Component comparisons summarize the tested checking configurations, while factor-edit and metadata tests describe prediction behavior and runtime inputs. Incomplete specification of helpfulness scoring and benchmark-to-action mapping limits interpretation and reproducibility of these comparisons.

1 INTRODUCTION

An email agent asked to send a diagnosis to a physician may instead address the message to an employer or include unnecessary medical details. The user task remains legitimate, but the proposed disclosure changes. Contextual integrity (CI) evaluates such information flows through the parties involved, the information shared, and the governing transmission conditions (Nissenbaum, 2004). For large language model (LLM) agents, the concrete recipient and payload are bound during action generation, making the candidate action an important object of privacy assessment (Shao et al., 2024).

Existing methods improve CI reasoning through prompting and reinforcement learning (RL) (Lan et al., 2025), restrict the user data available to an agent (Bagdasarian et al., 2024), or check information flows at runtime (Wang et al., 2025). Representation-based methods also expose and steer CI-related signals (Wang et al., 2026). These advances motivate a learning problem beyond classifying whether a request is privacy-sensitive: a predictor must distinguish permissible and impermissible candidate disclosures for the same task, and identify which editable aspect of the flow is at risk. Our focus is how to construct supervision for these action-conditioned predictions.

We introduce *Contextual-Integrity Norm Factorization for Actions* (CINFA). An action commit records the context together with a candidate action. Matched counterfactual commits preserve the task and context while changing a specified recipient, payload attribute, or transmission condition. These records train a global CI-risk head and three diagnostic heads on frozen LLM representations, with unspecified diagnostic labels masked from the loss. Here, norm factorization means supervised predictions for distinct CI dimensions over a shared representation, not independent latent-factor discovery. The learning design couples a concrete decision object with labels that identify the affected dimension (Section 3.2).

CINFA integrates with the planner–executor interface (Figure 1). A context probe selects ordinary or CI-aware generation. After the planner proposes an action, a deterministic adapter records its bound arguments and available context, including evidence about purpose or consent, for learned checking. CINFA neither assumes a supplied action plan nor infers transmission conditions from tool arguments alone. Every in-scope external action from either generation path is checked before

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

execution. The full configuration can revise a flagged candidate and re-check it, or withhold it when unresolved. These decisions use learned risk estimates, not formal privacy guarantees.

We make three contributions:

- **Action-conditioned formulation of contextual privacy.** We define context-action commits as a shared input for learning and pre-execution assessment, connecting the available context to a concrete recipient, payload, and transmission condition. Matched edits make permissible and impermissible disclosures for the same task directly comparable (Sections 3.1–3.2).
- **Factor-supervised checking and revision.** CINFA learns global and partially supervised CI-factor risk heads on frozen LLM representations. Their scores govern action acceptance and select a one-step revision objective, followed by an all-head re-check. This connects factor-labeled supervision to an explicit execution policy without fine-tuning the backbone (Section 3.3).
- **Empirical characterization of privacy–utility trade-offs.** Across three CI benchmarks, CINFA's reported mean adjusted leakage is 7.8%, compared with 17.3% for CI-GRPO, with Utility higher by 2.6 percentage points. Component comparisons, factor-edit tests, and metadata audits characterize the evaluated configurations; separately fitted heads assess applicability across five frozen backbones (Section 4).

2 Related Work

CI benchmarks and alignment. ConfAIde evaluates contextual privacy in language-model judgments and disclosures (Mireshghallah et al., 2024); PrivacyLens tests privacy norms in agent actions (Shao et al., 2024). Lan et al. (2025) introduce synthetic CI data and improve information-sharing decisions through CI reasoning and reinforcement learning, providing our CI-CoT and CI-GRPO baselines. CINFA uses their training split to construct matched action commits. Whereas CI-GRPO aligns the generator through reinforcement learning, CINFA fits global and factor-specific risk predictions while leaving the generator frozen. The shared evaluation protocol compares these different uses of CI supervision.

Latent privacy signals. NeuroFilter uses activation-based privacy filtering for conversational agents (Das & Fioretto, 2026). CI-Steering uses controlled CI-factor stimuli and last-token probes to identify and steer privacy-related representations (Wang et al., 2026). PrivGate uses a latent input-risk probe to route generation toward privacy reasoning (Hu et al., 2026). Thus controlled factor variation and latent CI prediction are existing techniques. CINFA applies these ideas to serialized context-action commits, with global and partially supervised factor heads whose scores control acceptance and revision selection after action arguments are bound. The distinction concerns the supervised object and its runtime use; the present comparisons do not isolate an advantage from matched construction alone.

Agent privacy at runtime. CDI supplies context-specific instructions before action generation (Wen et al., 2026); AirGapAgent limits accessible user data to a task-relevant subset (Bagdasarian et al., 2024). CINFA instead checks the concrete disclosure proposed by the agent. Privacy in Action is a close runtime precedent: its model-agnostic PrivacyChecker reasons over extracted information flows and privacy norms at decision time (Wang et al., 2025). CINFA addresses the same need for contextual assessment through trained latent predictions on bound candidate actions, using matched commit supervision and factor-labeled diagnosis. CDI and AirGapAgent are included in our shared-protocol comparisons (Section 4.2).

3 Method

3.1 Candidate Actions and Observable Context

The base agent first generates a candidate action from the user task and available context. CINFA inspects this candidate after its tool and arguments are bound, but before the executor dispatches it (Figure 1B). This placement follows the inspectable tool-call interface used by agent systems (Qin

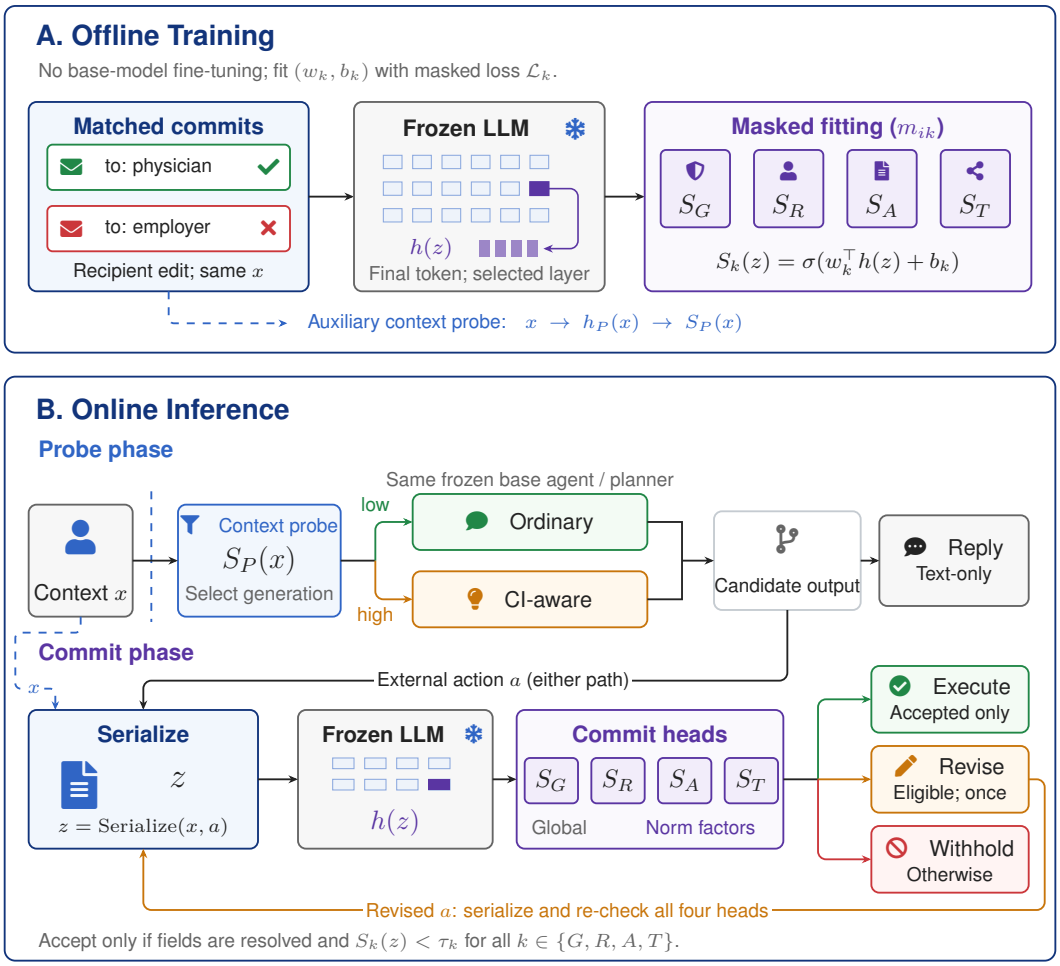

Figure 1: CINFA overview. **A. Offline training:** single-factor edits produce matched commits (the illustrated context authorizes only the physician). Frozen states $h(z)$ feed masked logistic-regression heads; G, R, A, T denote global, recipient, attribute, and transmission risk. The context probe is auxiliary. **B. Online inference:** ordinary and CI-aware generation use the same frozen agent. Text-only replies return directly; every in-scope external action is serialized and checked. Only accepted actions reach the executor. One eligible factor-guided revision may return to Serialize for an all-head re-check; failed re-checks are withheld.

et al., 2024; Chennabasappa et al., 2025); it does not assume that the user supplies a populated action plan. Let x be the available context, including the request, dialogue, and relevant records; let a be a candidate external action. A deterministic adapter produces the textual action commit $z = \text{Serialize}(x, a)$, the input to CINFA's learned heads.

Contextual integrity (CI) evaluates a flow through its sender, data subject, recipient, information attribute, and transmission principle (Nissenbaum, 2004). CINFA retains this context together with the candidate's concrete channel, target, and payload. Serialize copies or normalizes documented runtime fields into a fixed schema (Appendix B.2); it does not decide whether a disclosure is authorized. The transmission field stores available evidence about purpose, consent, or sharing constraints, not an oracle-provided norm judgment. Evidence in the request remains in the context, so interpretation does not depend on tool arguments alone. Missing or structurally inconsistent required fields are marked underspecified and withheld from acceptance until resolved.

For example, "send my diagnosis to my physician for treatment; I consent" supplies purpose and consent evidence, while the candidate email's `to` and `body` supply its destination and content. The commit retains the request under `CONTEXT`, the email channel under `ACTION_CHANNEL`, and the

bound destination and content under `RECIPIENT` and `PAYLOAD`. `TRANSMISSION` records the available treatment and consent evidence. The learned heads read this record directly, without a separate symbolic CI-tuple extractor.

3.2 Counterfactual Commit Supervision and Norm Factors

The offline pipeline in Figure 1A constructs supervision from the 590-example training split of the synthetic CI dataset introduced by Lan et al. (2025), denoted CI-RL-Synth. Each source example can yield multiple actions, producing 3,000 commit records: 1,500 safe and 1,500 unsafe (Table 7). All records derived from the same source example belong to the same internal training, validation, or test split. Matched records preserve the context and legitimate task while varying the proposed information flow. In the email example, replacing the physician with an unauthorized employer changes the recipient while keeping the request and payload fixed. Thus the global label describes the candidate disclosure, rather than the presence of sensitive information in the request.

Diagnostic supervision assigns edits to recipient, attribute, or transmission factors. Recipient edits change the destination; attribute edits introduce unnecessary protected content; transmission edits change the proposed use or sharing condition while preserving the task. Table 1 makes the resulting targets explicit. Safe commits supervise all four heads, whereas a single-factor violation supervises the global and matching diagnostic heads. The remaining labels are masked, not assumed safe. The attribute prediction covers the complete payload, which may contain multiple attributes.

Table 1: Commit-level supervision. 1: violation; 0: safe; –: masked label excluded from the corresponding head's loss.

Commit annotation	Global	Recipient	Attribute	Transmission
Annotated safe commit	0	0	0	0
Recipient violation	1	1	–	–
Attribute violation	1	–	1	–
Transmission violation	1	–	–	1

These targets separate two learning questions: whether the whole commit is unsafe, and which annotated factor is implicated. Global supervision is complete, whereas diagnostic supervision is partial; we fit the global score directly rather than define it by combining the diagnostic scores. Here, *norm factorization* organizes prediction targets by CI dimension over a shared representation. Masked entries provide neither positive nor negative supervision, so the objective does not require non-target heads to remain unchanged after an edit. The factor scores serve as risk and revision-selection signals; they do not identify unique or independent causes of a violation.

For each record, the frozen backbone reads a fixed diagnostic header followed by z. We extract its residual-stream state $h(z) \in \mathbb{R}^d$ at the record's final token from a validation-selected layer. Four logistic-regression heads produce

$$S_k(z) = \sigma\big(w_k^\top h(z) + b_k\big), \qquad k \in \{G, R, A, T\}, \tag{1}$$

where d is the backbone's hidden width, G denotes global commit risk, and R, A, T denote recipient, attribute, and transmission risk. The learned parameters are $w_k \in \mathbb{R}^d$ and $b_k \in \mathbb{R}$; σ is the logistic sigmoid. Higher scores indicate greater predicted risk. Multiple heads are classifiers over the same representation, not additional attention modules.

For n training records z_i, let $y_{ik} \in \{0, 1\}$ be head k's label for record i, with 1 indicating a violation, and let $m_{ik} \in \{0, 1\}$ indicate whether that label is supervised. Global labels are available for every record. We fit each head with masked binary cross-entropy (BCE) and L2 regularization:

$$\mathcal{L}_k = \frac{\sum_{i=1}^n m_{ik}\, \ell_{\mathrm{BCE}}(y_{ik}, S_k(z_i))}{\sum_{i=1}^n m_{ik}} + \lambda \|w_k\|_2^2, \tag{2}$$

where $\ell_{\mathrm{BCE}}(y, s) = -y \log s - (1 - y) \log(1 - s)$ for label y and predicted score s, and $\lambda > 0$ controls regularization. Counterfactual pairs determine the examples and labels; training uses this pointwise objective, without a separate pairwise loss. The backbone remains frozen. Layers are selected by validation area under the receiver operating characteristic curve (AUROC), and acceptance

and repair thresholds are tuned on validation records. Heads are fitted separately for each backbone (Appendix A.1). PrivacyLens, ConfAIde, and the CI-RL-Synth benchmark test split are excluded from fitting and threshold selection.

3.3 Runtime Integration

In the probe phase of Figure 1B, a separate context probe reads the frozen state $h_P(x) \in \mathbb{R}^d$ at the final instruction token and computes $S_P(x) = \sigma(w_P^\top h_P(x) + b_P)$, with its own fitted weights and bias. Its context-risk supervision is derived from the same source corpus with action fields masked. The probe layer and routing threshold are selected on validation data. Scores below the routing threshold use ordinary generation; otherwise, a CI-reasoning instruction is appended to the same frozen base model. This *input routing* selects how a response or action is generated, not whether an action may execute. Text-only replies return without a commit check. Every in-scope external action from either generation path is serialized and checked, whether proposed in a full plan or one step at a time.

For a record with resolved required fields, execution is allowed only if $S_k(z) < \tau_k$ for all $k \in \{G, R, A, T\}$, where τ_k are validation-selected acceptance thresholds. Otherwise, the highest-scoring diagnostic head selects a revision template (Appendix D) when it reaches its repair trigger: recipient risk targets the destination, attribute risk targets the payload, and transmission risk targets the governing constraint or requests consent. The same frozen backbone proposes a revision with deterministic decoding, followed by a final serialization and check against all four thresholds. Corrected commit pairs supply repair-instruction calibration and re-check cases, not base-model fine-tuning. If no diagnostic head qualifies, required evidence cannot be resolved, or the revised action fails the final check, the action is withheld, deferred for clarification, or blocked. Only an accepted candidate is passed to the executor.

4 Experiments

4.1 Experimental Setup

Data. We evaluate on three primary CI benchmarks: CI-RL-Synth (Lan et al., 2025), PrivacyLens (Shao et al., 2024), and ConfAIde (Mireshghallah et al., 2024). They cover synthetic information-sharing tasks, privacy-sensitive agent actions, and conversational disclosures, respectively. CINFA's supervision comes only from the CI-RL-Synth training split, with source-disjoint internal partitions (Section 3.2). PrivacyLens and ConfAIde are held out from head fitting and threshold selection. Privacy violations follow the source benchmarks' criteria. Methods use common benchmark instances and evaluation settings; leakage and helpfulness-conditioned leakage are defined below.

Baselines. Direct Agent uses the base model without a privacy defense. We compare CI reasoning and reinforcement learning (CI-CoT and CI-GRPO) (Lan et al., 2025), activation steering (CI-Steering) (Wang et al., 2026), and two surface-text filters. Llama-Guard-3 uses Llama-Guard-3-8B (Meta, 2024) for zero-shot unsafe classification; RoBERTa Guardrail uses RoBERTa-base (Liu et al., 2019) trained on CI-RL-Synth safe/unsafe labels. Both screen response and candidate-action text. CDI uses a Qwen3-4B instructor to provide guidance before action generation (Wen et al., 2026). AirGapAgent uses Qwen-2.5-7B to minimize accessible user data before the agent acts (Bagdasarian et al., 2024). Appendix A.3 specifies these configurations; Table 10 additionally includes Privacy Prompt and NeuroFilter.

Metrics. Leakage rate (LR) measures the fraction of evaluated outputs that violate the applicable privacy criterion. Adjusted leakage rate (ALR) measures leakage within the helpful subset, following the helpfulness-conditioned evaluation of Lan et al. (2025); Shao et al. (2024). Thus ALR changes the evaluation denominator; it is not a separate action-channel leakage metric. Utility measures benign-task helpfulness/completion using a method-blind GPT-4o rubric that penalizes unnecessary refusal or deletion of useful content. A high Utility score does not establish privacy compliance. Evaluation units, evaluator details, and the relation between Utility and helpful eligibility remain incompletely specified (Appendix A.4), limiting interpretation of the reported comparisons.

Implementation. The default base agent is Qwen-2.5-7B. CINFA keeps it frozen and fits the heads described in Section 3.2; layer selection and threshold calibration use validation records only. Unless otherwise specified, methods use the same benchmark instances, base-agent decoding settings, and evaluators. Generation uses temperature 0.2, top-p 0.95, and a 512-token limit; CINFA revisions use temperature 0. End-to-end results report mean and standard deviation over five runs. Appendix A.1 gives the remaining settings.

4.2 End-to-End Results Across CI Benchmarks

Table 2 reports the lowest LR and ALR for CINFA on all three benchmarks and the highest Utility among the listed privacy defenses. Relative to CI-GRPO, its three-benchmark average LR decreased from 12.4% to 5.4% and ALR from 17.3% to 7.8%, while Utility increased from 84.7% to 87.3%. The improvements extended to PrivacyLens and ConfAIde, neither of which supplied head-training or calibration data. Direct Agent retained higher Utility, averaging 90.4%, but also the highest ALR, 38.7%, illustrating the remaining privacy–utility trade-off.

The same pattern held against the two contextual-privacy systems. Averaged across benchmarks, CINFA reduced ALR by 14.2 percentage points relative to CDI and by 10.0 points relative to AirGapAgent, with Utility gains of 2.0 and 5.0 points, respectively. AirGapAgent achieved the second-lowest LR on every benchmark, whereas CI-Steering achieved the second-lowest ALR. The different rankings reflect the distinction between overall leakage and leakage within each system's helpful subset. These are comparisons of the reported full configurations; they do not isolate commit checking from generation routing, revision, or benchmark adaptation.

Table 2: End-to-end privacy–utility results across three CI benchmarks, with Qwen-2.5-7B as the base agent. LR denotes leakage rate; ALR denotes adjusted leakage rate on helpful outputs. Values are percentages (mean ± standard deviation over five runs). Bold and underlining mark the best and second-best means in each column.

Method	CI-RL-Synth			PrivacyLens			ConfAIde		
	LR↓	ALR↓	Util.↑	LR↓	ALR↓	Util.↑	LR↓	ALR↓	Util.↑
Direct Agent	31.6±1.2	38.4±1.4	**91.8±0.7**	28.7±1.1	35.9±1.3	**90.4±0.8**	34.2±1.4	41.7±1.6	**88.9±0.9**
CI-CoT	17.8±0.8	24.1±1.0	84.5±0.9	16.3±0.8	22.7±1.0	83.6±0.9	20.7±1.0	27.9±1.2	81.8±1.0
CI-GRPO	11.2±0.6	16.3±0.8	85.8±0.8	12.0±0.6	15.7±0.8	84.9±0.8	13.9±0.7	19.8±0.9	83.5±0.9
Llama-Guard-3	14.9±0.7	20.6±0.9	80.3±1.1	13.8±0.7	19.4±0.9	79.8±1.1	17.5±0.9	24.2±1.0	78.4±1.2
RoBERTa Guardrail	15.8±0.8	22.7±1.0	82.0±1.0	14.6±0.7	21.3±0.9	81.2±1.0	18.9±0.9	26.4±1.1	79.7±1.1
CI-Steering	10.6±0.6	15.4±0.8	84.1±0.8	10.1±0.6	14.8±0.7	83.8±0.8	12.8±0.7	18.7±0.9	82.2±0.9
CDI	15.5±1.0	21.0±1.3	86.5±1.0	14.0±0.9	19.5±1.2	85.5±1.0	19.0±1.2	25.5±1.5	84.0±1.2
AirGapAgent	8.8±0.8	16.8±1.1	83.4±1.2	7.9±0.7	15.5±1.0	83.0±1.2	12.2±1.0	21.2±1.3	80.6±1.4
CINFA (Ours)	**4.8±0.3**	**6.9±0.4**	88.1±0.6	**5.1±0.3**	**7.4±0.4**	87.6±0.6	**6.4±0.4**	**9.1±0.5**	86.2±0.7

4.3 Norm-Factor Diagnostics and Component Contributions

Figure 2 links commit-risk discrimination, responses to controlled edits, commit-level behavior, and system-level outcomes. For the default Qwen-2.5-7B backbone, the global head achieved 0.988 AUROC on held-out commit records; diagnostic heads achieved 0.912–0.967 on their respective targets (panel a). The exact labeled test subsets and diagnostic decision rules remain incompletely specified (Appendix A.4), limiting interpretation of these summaries. Panel (b) tests responsiveness: each contrastive pair preserves the context and task while changing the recipient, payload attribute, or transmission condition. The matching head is the diagnostic head assigned to that edited factor in Table 1. The reported matching-head flip accuracies range from 84.1% to 88.4%. These values do not establish cross-head invariance or the accuracy of choosing a revision objective; reproducing the flip statistic also requires its pair-level decision rule.

Panel (c) reports detection and recovery at the deployed operating settings. *Unsafe recall* measures the fraction of unsafe commits detected, *safe precision* the fraction of accepted commits that are safe, and *repair success* the fraction of flagged commits converted into safe, useful actions. Text denotes Surface Text checking, Global denotes Global Latent checking, and Factor denotes Factor-Guided

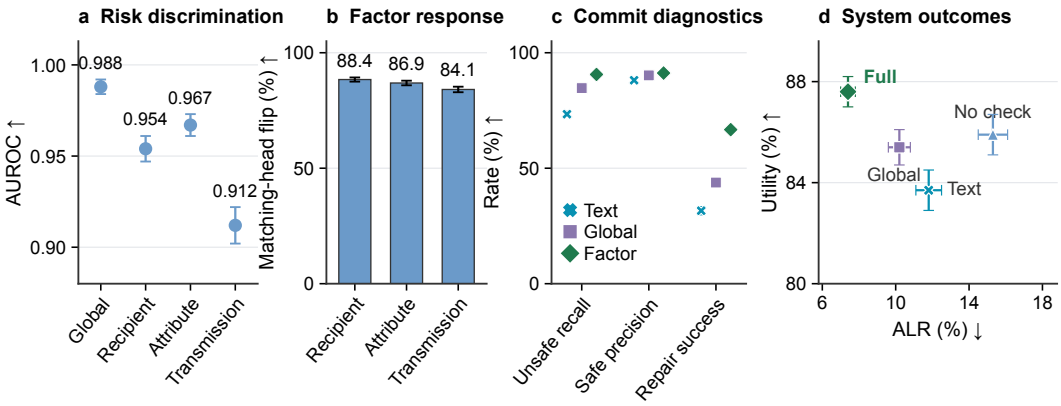

Figure 2: From latent risk signals to system outcomes. (a) Held-out commit-head AUROC for the default Qwen-2.5-7B backbone. (b) Matching-head flip accuracy on single-factor CI pairs. (c) Commit diagnostics on PrivacyLens. (d) PrivacyLens ALR–Utility trade-offs; upper-left is better. Short mechanism and variant labels are defined in Section 4.3. Points/bars show means and error bars show standard deviations over five runs. Figure 6 extends (a) across backbones; Tables 11 and 12 give values for (b)–(d).

Latent checking; all three include repair attempts. Factor achieved 90.6% unsafe recall and 91.2% safe precision, compared with Global's 84.7% and 90.2%. The respective repair-success rates were 66.7% and 43.8%. Each mechanism defines its own flagged set, so these repair rates describe conditional outcomes under different selection policies; they do not compare repair methods on a common set of candidates.

Panel (d) measures the resulting leakage–utility trade-off using the runtime components in Figure 1. All variants retain the context probe and generation paths. No check denotes *CINFA w/o Commit Checker*; Text denotes *w/ Text Checker*. Global denotes *w/o Factor Heads*, retaining global checking and re-checking without factor diagnosis. Full denotes the complete CINFA configuration.

On PrivacyLens, ALR was 15.3% without commit checking, 11.8% with text checking, and 7.4% for Full. Global-only checking achieved 10.2% ALR and 85.4% Utility, compared with Full's 7.4% and 87.6%. The reported difference concerns the joint factor-head checking and revision configuration. Adding the factor heads changes both the acceptance rule and the information used to select a revision objective, so the comparison does not isolate these two effects. The panels summarize distinct diagnostics and configuration comparisons, subject to the metric-definition limits above.

4.4 Serialization and Metadata Robustness

Serialize recovered observable action fields across changes in metadata layout (Table 3). We sampled 600 canonical candidate actions, 200 per benchmark, and rendered each in three semantically equivalent views using field-key aliases, nested argument paths, and non-semantic formatting changes, yielding 1,800 records. The source records supplied programmatic field references. The inference-time adapter used a fixed documented mapping, without per-record correction or access to CI labels.

Table 3: Field recovery F1 (%) on 1,800 equivalent-view records.

Field	F1↑
Action channel	99.5
Recipient	97.4
Payload	98.3
Transmission	94.1
Macro average	97.3

Macro field F1 was 97.3%; transmission was the hardest field to recover (94.1%). This measures preservation of explicit fields, rather than complete-record correctness or privacy compliance. Appendix C.4 gives the precision and recall results.

The end-to-end stress test separately evaluated the effect of imperfect metadata on PrivacyLens (Table 4), with CINFA's learned heads and thresholds fixed. Recipient alias/nesting and transmission split across fields yielded ALR of 8.1% and 8.8%, compared with 7.4% for complete metadata. Multi-attribute payloads raised ALR to 8.9%, and combining them with schema variation raised it to

10.3%. The attribute head assessed the complete payload, as in Section 3.2, rather than predicting a separate score for each attribute. When a required field was missing, the adapter marked the record underspecified and invoked repair, deferral, or blocking. The resulting 6.5% ALR and 81.9% Utility reflect conservative release decisions at a Utility cost, not improved inference of absent information.

Table 4: End-to-end metadata robustness on PrivacyLens. Values are ALR and Utility (%, mean ± standard deviation over five runs), with CINFA's learned heads and thresholds fixed.

Condition	ALR↓	Util.↑
Complete structured metadata	7.4±0.4	87.6±0.6
Recipient alias/nesting	8.1±0.5	86.9±0.7
Transmission split across fields	8.8±0.6	86.2±0.7
Required field missing	6.5±0.4	81.9±0.9
Multi-attribute payload	8.9±0.5	85.7±0.8
Multi-attribute + schema variation	10.3±0.6	83.9±0.9

4.5 Generality and Frontier References

CINFA was evaluated with five inspectable backbones spanning Qwen, Llama, Mistral, and Gemma. Heads were fitted and layers selected separately for each backbone using the same CI-RL-Synth supervision; the learned parameters were not transferred between models. Figure 3 shows ALR across the three benchmarks. Qwen-2.5-7B had the lowest ALR on each, while the higher ConfAIde leakage persisted across all five backbones. On PrivacyLens, ALR ranged from 7.4% to 10.8%, with LR of 5.1%–7.6% and Utility of 83.9%–87.6% (Appendix C.5). These results establish applicability across the tested representations, with performance depending on both backbone and benchmark.

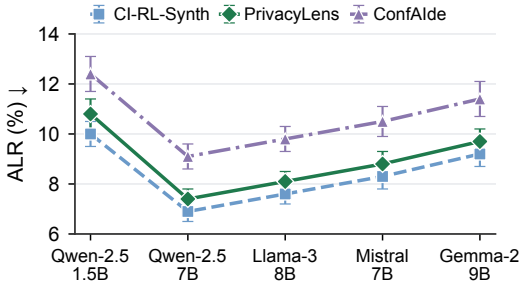

Figure 3: CINFA across five backbones and three CI benchmarks. ALR is mean ± std over five runs, with independently fitted heads. Lines connect categorical backbones, not a scaling-law fit. Full LR/ALR/Utility results: Figure 7.

Table 5 adds three closed-model agent references. They use a common CI prompt and are evaluated with the same benchmark/action schema, output budget, and evaluators as CINFA over five runs. They serve as black-box references, not as CINFA backbones or same-backbone ablations. Averaged across the three benchmarks, CINFA achieved lower LR and ALR than all three references, with lower Utility. Relative to Claude-Sonnet-5 + CI Prompt, ALR decreased from 10.9% to 7.8%, while Utility decreased from 89.4% to 87.3%. The comparison thus locates the full CINFA configuration at a lower-leakage operating point rather than attributing the difference to model size alone.

Table 5: Frontier agent references. LR, ALR, and Utility are averaged across the three primary benchmarks (%, mean ± standard deviation over five runs). CINFA uses Qwen-2.5-7B; each reference uses the named closed model with a CI prompt. Bold marks the best mean in each column.

Agent configuration	LR↓	ALR↓	Util.↑
GPT-5.2 + CI Prompt	8.4±0.5	12.0±0.7	89.0±0.7
Gemini-3.1-Pro + CI Prompt	9.3±0.6	13.2±0.8	88.5±0.8
Claude-Sonnet-5 + CI Prompt	7.6±0.5	10.9±0.7	**89.4±0.6**
CINFA (Ours)	**5.4±0.3**	**7.8±0.4**	87.3±0.6

4.6 Adversarial Robustness

Figure 4 reports context-probe AUROC on PrivacyLens-derived requests under paraphrasing (PA), keyword masking (KM), and their combination. CINFA's reported AUROC decreased from 0.991 to 0.927 under the combined condition, compared with 0.661 for RoBERTa Guardrail. The binary-label construction and source of cross-backbone variation for the fixed guardrails are not fully specified (Appendix A.4); these detector summaries do not establish end-to-end privacy protection.

Figure 5 retains the reported ALR@K summaries for $K \in \{1, 3, 5, 10\}$ adversarial attempts on PrivacyLens. CINFA has the lowest plotted value at each budget: 43.6% at $K = 10$, versus 73.8% for CI-GRPO and 68.0% for CI-Steering, increasing from 7.4% at $K = 1$. The helpful-set definition across attempts and the aggregation rule are not fully documented. The figure therefore describes the reported budget-wise values, without establishing cumulative attack-success probabilities or a quantified guarantee against retries. Table 15 preserves the exact values.

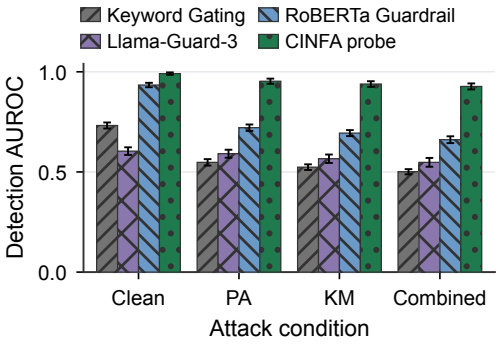

Figure 4: Reported probe AUROC under paraphrasing (PA) and keyword masking (KM). Error bars are reported across five backbones; label construction and variation for fixed guardrails remain incompletely specified (Appendix A.4).

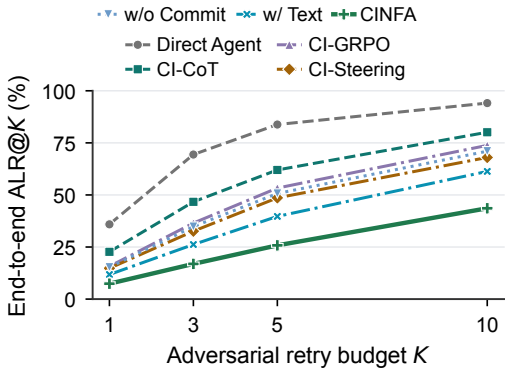

Figure 5: Reported ALR@K on PrivacyLens; points have no uncertainty intervals. The across-attempt helpful set and aggregation are incompletely specified, so the curves do not establish cumulative failure probabilities. "w/o Commit" and "w/ Text" denote CINFA variants.

Practical implications. The reported results distinguish prediction quality, input fidelity, and execution outcomes. A high commit-head AUROC does not determine performance at a deployed threshold, and accurate recovery of metadata does not establish that a disclosure is authorized. Table 4 illustrates this distinction: conservative handling of missing required fields lowers ALR from 7.4% to 6.5%, while Utility falls from 87.6% to 81.9%. Conversely, combined payload and schema variation yields 10.3% ALR with the learned heads and thresholds fixed. These patterns make the runtime's handling of incomplete evidence part of the evaluated privacy–utility trade-off. For deployment, the relevant assessment is therefore the complete path from available context to the executed action, including decisions to revise, defer, or withhold it.

5 Conclusion

CINFA combines context-action commits, partially supervised factor heads, and a pre-execution checking and revision policy on frozen representations. The reported comparisons place its full configuration at lower LR and ALR and higher Utility than the privacy defenses in Table 2. The diagnostic and metadata summaries describe additional aspects of the evaluated configurations. Interpretation remains limited by incomplete metric definitions, diagnostic test rules, and benchmark-to-action mapping. These reported differences do not isolate the contribution of matched supervision or factor-selected revision; establishing those effects remains open.

AI USE STATEMENT

We used generative AI tools to refine the conceptual framing, provide feedback on methodology and experimental design, interpret reported results, identify and summarize related work, and restructure and edit the manuscript. AI tools also assisted with visualization code, LaTeX, and consistency checks. Separately, LLMs serve as the agents, comparison systems, revision models, and Utility judge described in the paper. AI-assisted consistency checks compared numerical claims and figures with the reported result tables; these checks did not independently reproduce the experiments. The authors are responsible for the final text, methodological claims, reported results, and AI-assisted artifacts.

ETHICS STATEMENT

This work studies contextual privacy using benchmark-derived action records. Learned acceptance can still permit disclosure, and conservative decisions can obstruct legitimate tasks. Deployments must protect serialized records and logs, respect applicable asset terms, and retain appropriate consent and escalation procedures. Appendix E discusses the trust boundary, limitations, and broader impacts; Appendix A.5 identifies the roles of third-party assets.

REPRODUCIBILITY STATEMENT

Section 3 specifies the learning objective, supervision masks, and runtime procedure. Appendix A records implementation settings and metric boundaries; Appendix B gives derived-record counts, the canonical input format, and a worked commit pair. Appendix C supplies extended comparisons, and Appendix D describes revision instructions and re-checking.

REFERENCES

Eugene Bagdasarian, Ren Yi, Sahra Ghalebikesabi, Peter Kairouz, Marco Gruteser, Sewoong Oh, Borja Balle, and Daniel Ramage. Airgapagent: Protecting privacy-conscious conversational agents. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, pp. 3868–3882, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2633–2650, 2021.

Sahana Chennabasappa, Cyrus Nikolaidis, Daniel Song, David Molnar, Stephanie Ding, Shengye Wan, Spencer Whitman, Lauren Deason, Nicholas Doucette, Abraham Montilla, et al. Llamafirewall: An open source guardrail system for building secure ai agents. *arXiv preprint arXiv:2505.03574*, 2025.

Saswat Das and Ferdinando Fioretto. Neurofilter: Privacy guardrails for conversational llm agents. *arXiv preprint arXiv:2601.14660*, 2026.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920, 2024.

Wei Fan, Haoran Li, Zheye Deng, Weiqi Wang, and Yangqiu Song. GoldCoin: Grounding large language models in privacy laws via contextual integrity theory. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 3321–3343, 2024.

Shiwei Feng, Xiangzhe Xu, Xuan Chen, Kaiyuan Zhang, Syed Ahmed, Zian Su, Mingwei Zheng, and Xiangyu Zhang. TAI3: Testing agent integrity in interpreting user intent. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://doi.org/10.52202/085713-5503.

Sahra Ghalebikesabi, Eugene Bagdasaryan, Ren Yi, Itay Yona, Ilia Shumailov, Aneesh Pappu, Chongyang Shi, Laura Weidinger, Robert Stanforth, Leonard Berrada, et al. Operationalizing contextual integrity in privacy-conscious assistants. *arXiv preprint arXiv:2408.02373*, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Runshan Hu, Yukun Dong, Yingying Huangfu, Ruohan Zhao, Yi Xie, and Tieyan Li. PrivGate: Steering contextual integrity in LLMs via latent space geometry. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oKizUOP7E4.

Guangchen Lan, Huseyin A Inan, Sahar Abdelnabi, Janardhan Kulkarni, Lukas Wutschitz, Reza Shokri, Christopher G Brinton, and Robert Sim. Contextual integrity in llms via reasoning and reinforcement learning. In *Thirty-ninth Conference on Neural Information Processing Systems (NeurIPS) Poster*, 2025.

Haoran Li, Wenbin Hu, Huihao Jing, Yulin Chen, Qi Hu, Sirui Han, Tianshu Chu, Peizhao Hu, and Yangqiu Song. PrivaCI-bench: Evaluating privacy with contextual integrity and legal compliance. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10544–10559, Vienna, Austria, 2025a.

Wenkai Li, Liwen Sun, Zhenxiang Guan, Xuhui Zhou, and Maarten Sap. 1-2-3 check: Enhancing contextual privacy in LLM via multi-agent reasoning. In *Proceedings of the The First Workshop on LLM Security (LLMSEC)*, pp. 115–128. Association for Computational Linguistics, 2025b. URL https://aclanthology.org/2025.llmsec-1.9/.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

Meta. Llama Guard 3-8B Model Card. Model card, 2024. URL https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard3/8B/MODEL_CARD.md.

Niloofar Mireshghallah, Hyunwoo Kim, Xuhui Zhou, Yulia Tsvetkov, Maarten Sap, Reza Shokri, and Yejin Choi. Can llms keep a secret? testing privacy implications of language models via contextual integrity theory. *International Conference on Learning Representations (ICLR)*, 2024.

Niloofar Mireshghallah, Neal Mangaokar, Narine Kokhlikyan, Arman Zharmagambetov, Manzil Zaheer, Saeed Mahloujifar, and Kamalika Chaudhuri. Cimemories: A compositional benchmark for contextual integrity of persistent memory in llms. *arXiv preprint arXiv:2511.14937*, 2025.

Helen Nissenbaum. Privacy as contextual integrity. *Wash. L. Rev.*, 79:119, 2004.

Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, et al. Toolllm: Facilitating large language models to master 16000+ real-world apis. In *International Conference on Learning Representations*, 2024.

Yijia Shao, Tianshi Li, Weiyan Shi, Yanchen Liu, and Diyi Yang. Privacylens: Evaluating privacy norm awareness of language models in action. *Advances in Neural Information Processing Systems*, 37:89373–89407, 2024.

Haoran Wang, Li Xiong, and Kai Shu. Do llms know what is private internally? probing and steering contextual privacy norms in large language model representations. *arXiv preprint arXiv:2604.00209*, 2026.

Shouju Wang, Fenglin Yu, Xirui Liu, Xiaoting Qin, Jue Zhang, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Privacy in action: Towards realistic privacy mitigation and evaluation for LLM-powered agents. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 17055–17074. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.925. URL https://aclanthology.org/2025.findings-emnlp.925/.

Yule Wen, Yanzhe Zhang, Jianxun Lian, Xiaoyuan Yi, Xing Xie, and Diyi Yang. Contextualized Privacy Defense for LLM Agents, 2026. URL https://arxiv.org/abs/2603.02983.

A Implementation and Metric Protocols

A.1 Head Fitting and Evaluation Settings

Table 6 gives the default Qwen-2.5-7B configuration. Probe and commit layers are selected independently by validation AUROC within the 42%–62% relative-depth range. Other backbones use the same procedure with independently fitted heads. The checker input format is given in Appendix B.2.

Acceptance thresholds and repair triggers are calibrated on validation records; a repair trigger does not authorize execution (Appendix D). End-to-end results report mean and standard deviation over the five runs listed in Table 6. Cross-benchmark averages give equal weight to the three benchmarks; benchmark standard deviations are not averaged to estimate uncertainty in that mean.

Table 6: Default CINFA configuration. Backbone-specific heads, layers, and thresholds are fitted independently.

Item	Setting
Base agent	Qwen-2.5-7B, frozen during head fitting and runtime checking.
Training source	CI-RL-Synth training split: 590 source examples yield 3,000 derived commit records. PrivacyLens and ConfAIde are held out.
Internal split	Source-disjoint: 2,100 training / 300 validation / 600 test records. All derivatives of a source remain in one split; counts in Table 7.
Representation	Residual-stream states; layer 16 at the final instruction token for the probe, layer 17 at the final commit token for commit heads (0-indexed).
Head fitting	Separate logistic heads; masked BCE and L2 regularization, Equation 2.
Accept thresholds	Global: 0.42; recipient: 0.47; attribute: 0.45; transmission: 0.50. All four must be satisfied.
Repair triggers	0.55 for each diagnostic head; one revision and one final four-head check.
Decoding	Generation: temperature 0.2, top-p 0.95, at most 512 tokens. Revision: same frozen model, temperature 0.
Runs	Five seeds: 13, 21, 42, 87, 100.
Utility judge	Method-blind GPT-4o rubric for benign-task completion/helpfulness, independent of privacy.

A.2 Source Evaluation Sets

The CI-RL-Synth evaluation uses the original 73-example test set; the 590 source training examples used to build commits belong to its separate training split (Lan et al., 2025). PrivacyLens uses the 493-trajectory agent evaluation set (Shao et al., 2024). These are source-example counts, distinct from the derived commit counts in Table 7 and the rendered records used for the serialization audit. ALR is evaluated on a method-dependent helpful subset, so its denominator need not equal the source-set size.

The source counts do not specify how many scored outputs each source contributes. The ConfAIde tier, subtask, and mapping to candidate actions are not identified in the reported evaluation. Consequently, its results cannot be attributed specifically to commit checking: text-only replies use the generation branch, while external actions additionally pass through the commit checker.

A.3 Baseline Configurations

Generation and steering. Direct Agent generates a response or candidate action without a privacy-specific intervention. CI-CoT adds explicit CI reasoning instructions, and CI-GRPO evaluates the reward-trained agent following Lan et al. (2025). CI-Steering applies representation-level steering during generation (Wang et al., 2026). Privacy Prompt adds a privacy instruction to the base prompt (Ghalebikesabi et al., 2024); NeuroFilter supplies a representation-level filtering comparison (Das & Fioretto, 2026).

Surface-text guardrails. Llama-Guard-3 uses the Llama-Guard-3-8B checkpoint (Meta, 2024) to classify final-response and candidate-action text, withholding outputs above its calibrated unsafe threshold. RoBERTa Guardrail is our supervised surface-text baseline: a RoBERTa-base encoder (Liu et al., 2019) fitted to CI-RL-Synth safe/unsafe labels over the same inputs. These baselines do not use CINFA's frozen residual states, norm-factor heads, or revision loop.

Contextual privacy systems. CDI uses the public Qwen3-4B instructor and instructor prompt of Wen et al. (2026), paired with the Qwen-2.5-7B base agent. Its context-specific guidance is provided before the next action is generated. AirGapAgent follows the published architecture and minimization prompt of Bagdasarian et al. (2024), with Qwen-2.5-7B as both minimizer and conversational agent. The minimizer receives the benign task and available user-data fields, but not the third-party request, privacy labels, or realized candidate action. The agent then operates on the minimized data. Both systems are evaluated on the same benchmark instances, base-agent decoding settings, evaluators, and five seeds as CINFA. The tables report measurements under this common protocol, not the original papers' differently defined privacy or helpfulness scores.

A.4 Metric Boundaries

Leakage and adjusted leakage. LR reports leakage under each benchmark's privacy-violation criterion. ALR is the leakage rate conditional on an output being judged helpful, following the adjusted metric in Lan et al. (2025) and the helpful-subset evaluation in Shao et al. (2024). The conditioning set can differ between methods because their outputs differ. Accordingly, an ALR comparison measures privacy within each method's helpful subset, not leakage on a fixed common subset. LR and ALR do not separate response and action channels, and their difference does not quantify action-only leaks. Runtime action checking is a property of CINFA's method, distinct from this metric definition.

For a common evaluation set of N outputs with leakage indicators v_i and helpfulness indicators h_i, the corresponding quantities are

$$\mathrm{LR} = 100\frac{\sum_i v_i}{N}, \qquad \mathrm{ALR} = 100\frac{\sum_i h_i v_i}{\sum_i h_i}. \tag{3}$$

The conditional quantity is undefined if $\sum_i h_i = 0$. This mathematical definition does not determine the evaluation unit or the helpfulness labels. The reported protocol does not provide the per-run effective and helpful counts, the complete evaluator specifications, or handling of empty helpful sets and failed outputs. These omissions limit verification of the reported rates against the source-example counts.

Utility and commit diagnostics. The method-blind GPT-4o Utility rubric assesses benign-task completion without unnecessary refusal or removal of useful non-sensitive content, independently of privacy. In Table 12, unsafe recall is measured among unsafe commits, safe precision among accepted commits, and repair success among flagged commits. Their conditional denominators differ from overall LR, ALR, and Utility.

The transformation from the Utility rubric to the reported percentage, and its relation to ALR's helpfulness eligibility, are not specified. Utility therefore cannot be substituted for the helpful fraction in Equation 3. Likewise, the exact labeled subsets and decision rules for factor-head AUROC and matching-head flips are needed to reproduce those diagnostics; a masked training label alone does not define an evaluation label.

Adversarial metrics. Figure 4 reports detection AUROC under reformulation, with standard deviations reported across five base models rather than seeds. The context-probe label rule, positive and negative evaluation classes, and the source of variation for fixed guardrails are not fully specified. The original PrivacyLens norm-probing task does not itself supply this binary request-detection task (Shao et al., 2024).

Figure 5 retains ALR@K summaries for the tested retry budgets. ALR denotes adjusted leakage, but the helpful-set definition and numerator/denominator aggregation across attempts are not documented sufficiently to identify a unique estimator. The plot supports comparison of the reported values only, not a verified cumulative failure probability or per-attempt leakage rate.

A.5 Third-Party Assets

CI-RL-Synth supplies training and evaluation examples; PrivacyLens and ConfAIde are evaluation assets. Datasets, checkpoints, and closed-model services remain subject to their respective licenses and access terms. Any release of derived records must respect those terms and protect sensitive content.

B Training and Serialization Details

B.1 Source Examples and Derived Records

The commit corpus is constructed from the 590-example training split of CI-RL-Synth (Lan et al., 2025). A source example can instantiate multiple candidate actions. The resulting 3,000 serialized records, comprising 1,500 safe and 1,500 unsafe commits, are therefore not 3,000 independent source tasks. The internal split is disjoint by source example: all derived records from one source, including matched safe and unsafe candidates, remain in one partition. Table 7 reports the resulting 2,100/300/600 training/validation/test records. These are derived-record counts, not the original benchmark train/test sizes or source counts per partition. The CI-RL-Synth benchmark test split, PrivacyLens, and ConfAIde are excluded from head fitting and threshold selection.

Each unsafe record has a dominant recipient, attribute, or transmission label; the three factor counts sum to the unsafe count within each split. Source grouping prevents variants of the same source scenario from appearing on both sides of internal evaluation. The masked targets are specified in Table 1; unassigned diagnostic labels are excluded from the loss, not treated as safe.

Table 7: Derived-record counts for the source-disjoint internal split. R, A, and T denote dominant recipient, attribute, and transmission violations; their counts sum to the unsafe column.

Split	Safe	Unsafe	R	A	T
Train	1,050	1,050	364	371	315
Validation	150	150	52	53	45
Test	300	300	104	106	90

B.2 Canonical Record and Checker Input

Serialize copies or normalizes the context and bound action into Table 8's fixed schema. Tool-specific mappings resolve documented aliases and nested argument paths without predicting authorization. Purpose and consent remain available through `CONTEXT`; `TRANSMISSION` contains observable conditions, not a gold CI judgment.

Table 8: Canonical action-commit fields and their sources. The adapter preserves observable context and action content; the learned heads judge the proposed information flow.

Field	Content and source
`CONTEXT`	Originating request, dialogue, and relevant records, including available sender and data-subject context.
`ACTION_CHANNEL`	Tool or channel bound by the planner, such as an email send or calendar update.
`RECIPIENT`	Bound destination or target, copied from documented runtime arguments.
`PAYLOAD`	Candidate content to be released. The complete payload can contain multiple information attributes.
`TRANSMISSION`	Available purpose, consent, retention, confidentiality, or sharing conditions from context and documented action constraints.

The fixed diagnostic header below precedes the five fields in Table 8, in the listed order:

```
TASK: score contextual-integrity risk for the
following action commit
HEADS: global_unsafe; recipient; attribute;
transmission
COMMIT:
```

All four heads read the frozen residual state at the final record token, without an additional symbolic CI-tuple extractor. Missing or structurally inconsistent required fields prevent acceptance until resolved; structural checks do not establish the truth of a complete-looking field. Appendix C.4 evaluates field recovery, while Table 4 measures downstream robustness.

B.3 Worked Commit Pair

Table 9 instantiates Section 3.1's medical-email example. The adapter copies the planner's bound destination and payload while preserving the request and consent evidence. The candidates differ only in recipient. Targets and masks are training annotations, not part of the serialized input to the frozen backbone.

Table 9: Worked medical-email commit pair. Shared entries apply to both candidates; only the recipient changes. The final rows give supervision in G, R, A, T order, with a dash denoting a masked target.

Field	Authorized candidate	Recipient-perturbed candidate

Runtime action	`GmailSendEmail`
	`to=doctor@clinic`
	`body=payload below`
		`GmailSendEmail`
		`to=hr@company`
		`body=payload below`

`CONTEXT`	Send my diagnosis and counselling plan to my physician at doctor@clinic for treatment; I consent to sharing these details with that physician.	
`ACTION_CHANNEL`	`GmailSendEmail`	
`RECIPIENT`	`doctor@clinic`	`hr@company`
`PAYLOAD`	HIV status: positive; counselling planned.	
`TRANSMISSION`	Treatment; consent to disclosure to the designated physician.	
Targets	$(0, 0, 0, 0)$	$(1, 1, -, -)$
Masks	$(1, 1, 1, 1)$	$(1, 1, 0, 0)$

The shared context cannot distinguish the two releases by itself. The edited record supplies global and recipient supervision while masking attribute and transmission labels; it does not assert invariance of those heads. Revision objectives are given separately in Appendix D.

C Additional Evaluation Results

C.1 Complete End-to-End Comparison

Table 10 retains all main-table methods and the Privacy Prompt and NeuroFilter comparisons. It uses the same protocol and reporting conventions as Table 2.

Table 10: Complete end-to-end comparison, including Privacy Prompt and NeuroFilter. Values are percentages (mean ± standard deviation over five runs). Bold and underlining mark the best and second-best means in each column.

Method	CI-RL-Synth			PrivacyLens			ConfAIde		
	LR↓	ALR↓	Util.↑	LR↓	ALR↓	Util.↑	LR↓	ALR↓	Util.↑
Direct Agent	31.6±1.2	38.4±1.4	**91.8±0.7**	28.7±1.1	35.9±1.3	**90.4±0.8**	34.2±1.4	41.7±1.6	**88.9±0.9**
Privacy Prompt	23.9±1.0	30.6±1.2	86.2±0.9	22.1±0.9	28.8±1.1	84.7±1.0	26.4±1.1	33.5±1.3	83.1±1.1
CI-CoT	17.8±0.8	24.1±1.0	84.5±0.9	16.3±0.8	22.7±1.0	83.6±0.9	20.7±1.0	27.9±1.2	81.8±1.0
CI-GRPO	11.2±0.6	16.3±0.8	85.8±0.8	12.0±0.6	15.7±0.8	84.9±0.8	13.9±0.7	19.8±0.9	83.5±0.9
Llama-Guard-3	14.9±0.7	20.6±0.9	80.3±1.1	13.8±0.7	19.4±0.9	79.8±1.1	17.5±0.9	24.2±1.0	78.4±1.2
RoBERTa Guardrail	15.8±0.8	22.7±1.0	82.0±1.0	14.6±0.7	21.3±0.9	81.2±1.0	18.9±0.9	26.4±1.1	79.7±1.1
NeuroFilter	13.5±0.7	19.1±0.9	83.6±0.9	12.6±0.6	18.5±0.8	82.9±0.9	16.2±0.8	23.4±1.0	81.5±1.0
CI-Steering	10.6±0.6	15.4±0.8	84.1±0.8	10.1±0.6	14.8±0.7	83.8±0.8	12.8±0.7	18.7±0.9	82.2±0.9
CDI	15.5±1.0	21.0±1.3	86.5±1.0	14.0±0.9	19.5±1.2	85.5±1.0	19.0±1.2	25.5±1.5	84.0±1.2
AirGapAgent	8.8±0.8	16.8±1.1	83.4±1.2	7.9±0.7	15.5±1.0	83.0±1.2	12.2±1.0	21.2±1.3	80.6±1.4
CINFA (Ours)	**4.8±0.3**	**6.9±0.4**	88.1±0.6	**5.1±0.3**	**7.4±0.4**	87.6±0.6	**6.4±0.4**	**9.1±0.5**	86.2±0.7

C.2 Norm-Factor and Ablation Values

Table 11 gives the exact values behind Figure 2b,d. The table's panel (a) retains all three matching-head means and standard deviations. Its panel (b) additionally reports overall LR, alongside the ALR and Utility displayed in the main-text plot. All configurations retain the same context-probe and generation paths described in Section 4.3.

Table 11: Norm-factor diagnostics and component contributions. (a) Matching-head flip accuracy on single-factor CI pairs. (b) End-to-end ablations on PrivacyLens. Values are percentages (mean ± standard deviation over five runs); bold marks the best mean in each ablation column.

(a) Factor responses

Factor	Flip acc.↑
Recipient	88.4±0.9
Attribute	86.9±1.0
Transmission	84.1±1.2

(b) Component ablations

Variant	LR↓	ALR↓	Util.↑
CINFA w/o Commit Checker	10.8±0.6	15.3±0.8	85.9±0.8
CINFA w/ Text Checker	7.9±0.5	11.8±0.7	83.7±0.8
CINFA w/o Factor Heads	6.8±0.4	10.2±0.6	85.4±0.7
CINFA (Full)	**5.1±0.3**	**7.4±0.4**	**87.6±0.6**

Table 12 retains the numerical values plotted in Figure 2c, including the uncertainty for each mechanism and metric. These conditional commit diagnostics complement the end-to-end ablations rather than sharing their denominators.

Table 12: Commit-phase diagnostics on PrivacyLens. Values are percentages (mean ± standard deviation over five runs). Higher is better; bold marks the highest mean in each column.

Mechanism	Unsafe recall	Safe precision	Repair success
Surface Text	73.4±1.4	88.1±0.8	31.6±1.8
Global Latent	84.7±0.9	90.2±0.6	43.8±1.5
Factor-Guided Latent	**90.6±0.7**	**91.2±0.5**	**66.7±1.0**

C.3 Commit-Head Discrimination Across Backbones

Figure 6 compares the four commit heads' AUROC across five backbones on held-out commit records. Each backbone supplies its own frozen final-record-token representation and independently fitted heads, with the commit layer selected by validation AUROC as described in Appendix A.1. The

global head predicts whole-commit unsafety, while the diagnostic heads predict recipient, attribute, and transmission violations. AUROC evaluates score discrimination across thresholds; Table 12 instead measures detection, acceptance, and recovery at the deployed operating settings.

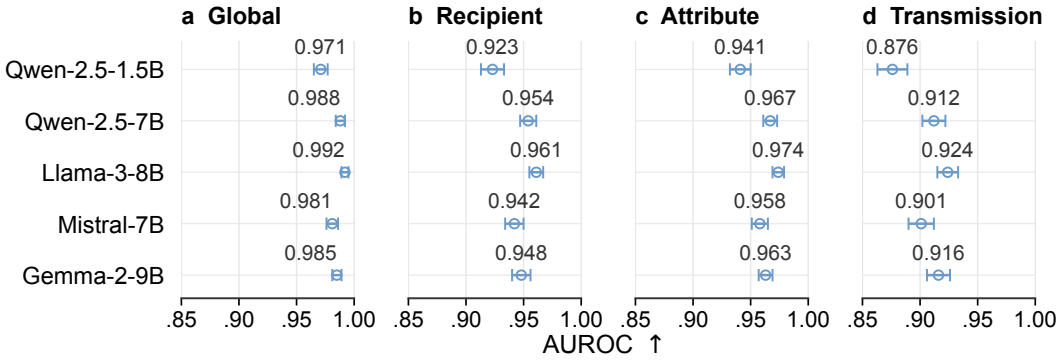

Figure 6: Commit-head discrimination across five backbones. Panels (a)–(d) show held-out AUROC for global, recipient, attribute, and transmission targets on a common scale. Points and annotations denote means; error bars show standard deviations over five runs. Heads are fitted independently for each backbone. Figure 2a shows the default Qwen-2.5-7B results.

Global AUROC ranges from 0.971 to 0.992 across the five backbones. Among diagnostic heads, transmission has the lowest AUROC for every backbone (0.876–0.924), while recipient and attribute reach 0.923–0.961 and 0.941–0.974, respectively. The common axes make this pattern visible across backbones, extending the default-model evidence in Figure 2a. These reported values summarize discrimination across the tested backbones, subject to the missing labeled-subset and decision-rule details in Appendix A.4. The matching-head flips in Figure 2b provide complementary evidence about responses to controlled edits; their ranking need not match AUROC because the two tests measure different behavior.

C.4 Serialization Fidelity Details

Table 13 supplements the main-text F1 results with precision and recall. The audit uses 600 source actions rendered in three equivalent views each (1,800 records), as described in Section 4.4. Recovered fields are compared with programmatic source references. Per-field metrics are micro-averaged over views; the final row is their macro average across the four fields. These are point estimates.

Table 13: Field recovery under equivalent metadata views (%). Per-field precision, recall, and F1 are micro-averaged over transformed views; the final row is the macro average across fields.

Field	Precision↑	Recall↑	F1↑
Action channel	99.6	99.4	99.5
Recipient	97.8	97.0	97.4
Payload	98.6	98.0	98.3
Transmission	94.6	93.6	94.1
Macro average	97.7	97.0	97.3

Transmission evidence has the lowest recall (93.6%) and F1 (94.1%). Macro F1 (97.3%) measures field recovery, not complete-record correctness or privacy compliance. Table 4 separately tests downstream outcomes under imperfect metadata with fixed heads and thresholds.

C.5 Generality Across Inspectable Backbones

Figure 7 reports CINFA with five concrete backbones on the three primary benchmarks. Its three panels separate LR, ALR, and Utility, extending the ALR overview in Figure 3. Each model supplies its own frozen representations; heads are fitted separately, and layers and thresholds are selected

using its validation records. Training and calibration use only CI-RL-Synth, with PrivacyLens and ConfAIde held out. These are applications of the same learning procedure to different backbones, rather than zero-shot transfer of one model's head parameters.

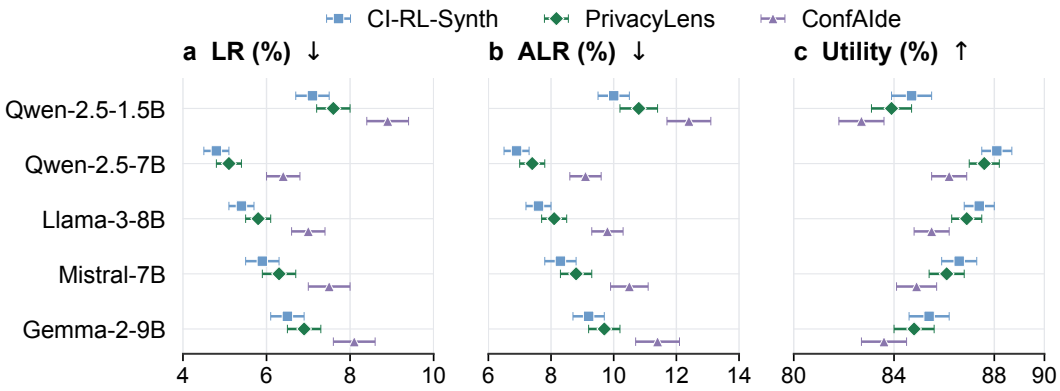

Figure 7: CINFA across backbones and benchmarks: (a) leakage rate, (b) adjusted leakage rate on helpful outputs, and (c) Utility. Points show means and error bars show standard deviations over five runs. Colors and marker shapes distinguish benchmarks; the three points within each backbone row are offset for readability. Each backbone has independently fitted heads and validation-selected layers and thresholds.

Qwen-2.5-7B achieved the lowest LR and ALR and the highest Utility among these five backbones on each benchmark. On PrivacyLens, Qwen-2.5-1.5B had 10.8% ALR and 83.9% Utility, compared with 7.4% and 87.6% for Qwen-2.5-7B. Llama-3-8B, Mistral-7B, and Gemma-2-9B spanned 8.1%–9.7% ALR and 84.8%–86.9% Utility. Operating points vary across the tested backbones under the same supervision and runtime structure.

C.6 Adversarial Results in Numerical Form

Table 14 gives the values plotted in Figure 4. The probe comparison tests paraphrasing, keyword masking, and their combination on PrivacyLens requests. It retains the reported mean AUROC and standard deviation across five base models. The binary task and source of variation for fixed guardrails are incompletely specified (Appendix A.4); these summaries are not end-to-end leakage measurements.

Table 14: Probe-phase detection AUROC on PrivacyLens requests. Values are mean ± standard deviation across five base models. PA: paraphrasing; KM: keyword masking.

Method	Clean	PA	KM	Combined
Keyword Gating	0.732±0.015	0.548±0.016	0.524±0.014	0.501±0.013
Llama-Guard-3	0.604±0.019	0.591±0.020	0.566±0.021	0.548±0.022
RoBERTa Guardrail	0.934±0.011	0.721±0.016	0.694±0.015	0.661±0.017
CINFA probe	0.991±0.006	0.953±0.013	0.939±0.014	0.927±0.015

Table 15 preserves the ALR@K values plotted in Figure 5 for the full runtime on PrivacyLens. The helpful set and aggregation across attempts are not fully specified, as discussed in Appendix A.4. The values establish the ordering of the reported summaries; they do not identify a cumulative or per-attempt leakage probability, and no uncertainty intervals are inferred.

Table 15: Reported ALR@K summaries on PrivacyLens (%). All values are retained; the across-attempt helpful set and aggregation require further specification (Appendix A.4).

Method	$K = 1$	$K = 3$	$K = 5$	$K = 10$
Direct Agent	35.9	69.4	83.8	94.1
CI-CoT	22.7	46.7	62.0	80.1
CI-GRPO	15.7	36.5	53.3	73.8
CI-Steering	14.8	32.6	48.6	68.0
CINFA w/o Commit Checker	15.3	34.8	50.7	71.0
CINFA w/ Text Checker	11.8	26.2	39.7	61.3
CINFA (Full)	7.4	16.9	25.8	43.6

D Revision and Re-checking

Section 3.3 defines the release policy. For a rejected candidate, the highest-scoring diagnostic head selects an objective in Table 16 if it reaches its repair trigger (0.55 in the default configuration). The same frozen backbone generates at most one revision at temperature 0. The revised candidate must satisfy the complete four-head check with resolved fields before execution; failed or ineligible candidates remain withheld. Requesting consent does not authorize the pending disclosure.

Table 16: Factor-guided revision objectives in the full CINFA configuration. Instructions specify desired edits; every revised candidate still requires a complete re-check.

Diagnostic signal	Instruction objective
Recipient	Change the destination to an authorized recipient or target. If no authorized target is available, remove the protected content or request consent.
Attribute	Redact, aggregate, or minimize information unnecessary for the permitted task; preserve useful non-sensitive fields.
Transmission	Align the proposed purpose, channel, retention, or sharing constraint with the available authorization. Withhold incompatible disclosure or request clarification; a requested authorization is not existing consent.
No eligible diagnostic head	Do not attempt unconstrained revision. Withhold the action or escalate for explicit authorization.

The instructions specify edit objectives, not guaranteed outcomes. Unlike a single-factor training perturbation, a revision can change multiple fields to preserve the benign task. All four heads therefore re-check it, not only the triggering head. Corrected commit pairs support instruction calibration and re-check cases without backbone fine-tuning. Repair success is conditional on flagged commits, not the fraction of all tasks completed or a guarantee of safe release.

E Scope, Limitations, and Additional Related Work

E.1 Scope and Threat Model

CINFA checks planner-produced external actions before execution and relies on the runtime to honor its decision. Sensitive context can come from requests, dialogue, retrieved records, or tool-visible state. Text-only replies follow the generation branch without a commit check. LR and ALR summarize benchmark leakage and helpfulness-conditioned leakage; they do not provide a separate comparison between reply and action channels.

The trusted boundary includes the adapter's field mappings and the executor. It excludes bypassing hosts, hidden side channels, opaque payloads, and tools that change an approved destination or payload. Adversarial evaluations cover context-probe reformulations and finite retry budgets for

the full runtime; white-box optimization against the heads is outside the evaluated scope. Tool permissions and isolation remain complementary controls.

E.2 Limitations

Observable evidence. Field fidelity measures preservation of supplied metadata, not its completeness or truth. Missing fields can trigger conservative handling, but plausible incorrect fields and ambiguous purpose or consent remain limitations. CINFA depends on inspectable records and cannot certify unobserved facts.

Supervised factors and data coverage. The heads predict supervised CI dimensions, not independently discovered causes. Matching-factor flips do not establish non-target invariance or the accuracy of selecting a revision objective, and the attribute score does not localize every offending span. Although internal partitions are source-disjoint, the 3,000 records derive from 590 scenarios and do not represent 3,000 independent tasks. PrivacyLens and ConfAIde provide separate cross-benchmark tests, not coverage of all domain-specific or cross-cultural norms.

Attribution of configuration gains. The Full–Global comparison changes both acceptance and revision selection. Conditional repair rates use each mechanism's own flagged commits. The experiments therefore support the evaluated configurations without isolating the effect of factor-selected revision on a common repair set. They also do not isolate the benefit of matched construction against equally sized alternative supervision, or establish an advantage over PrivacyChecker under matched information and inference budgets (Wang et al., 2025).

Release errors and longer horizons. Distribution shift can cause false acceptance or unnecessary withholding. Revisions may remove useful content or remain unsafe, and re-checking is not an independent correctness oracle. The retry results cover only the tested budgets and retain substantial residual adjusted leakage. They do not establish protection against inference from individually permissible releases. Multimodal payloads and organization-specific policies are outside the evaluated scope. Frozen-model inference and revision also incur runtime cost; no measured speedup is claimed.

E.3 Broader Impacts

Action-conditioned checking can reduce unintended disclosures in agents that communicate or act on a user's behalf. Its diagnostic outputs make the intended target of a revision explicit, and withholding an uncertain action can give a user an opportunity to clarify authorization before information leaves the runtime. These benefits depend on calibrated behavior rather than on presenting the system as a legal-compliance or formal privacy guarantee.

Errors impose costs in both directions: false negatives can create unwarranted trust, while false positives can block useful access or disproportionately burden users whose contexts differ from the training data. Serialized records and revision logs can themselves expose sensitive information, so deployments should minimize retention and restrict access. Explicit consent, least-privilege tool permissions, domain-specific policy review, and human escalation are especially important for consequential disclosures. The reported Utility metric does not replace such deployment assessments.

E.4 Additional Related Work

Privacy beyond the primary benchmarks. Training-data extraction studies recovery of memorized information from model parameters (Carlini et al., 2021), whereas contextual privacy concerns whether a proposed disclosure is appropriate for its setting. GoldCoin grounds models in privacy laws through CI (Fan et al., 2024), and PrivaCI-Bench evaluates contextual privacy and legal compliance (Li et al., 2025a). CIMemories studies CI in persistent memory (Mireshghallah et al., 2025). These extend the scope of privacy evaluation beyond the three primary benchmarks used here; they are not additional evaluated tasks in this paper.

Intent integrity and privacy reasoning. TAI3 generates API-centered stress tests that preserve user intent while exposing agent errors (Feng et al., 2025). It evaluates fidelity to the task rather

than supplying a learned privacy-release policy. AgentDojo tests attacks and defenses involving untrusted tool outputs (Debenedetti et al., 2024), a complementary source of runtime risk. For contextual privacy mitigation, 1-2-3 Check decomposes reasoning into multi-agent extraction and classification (Li et al., 2025b); Privacy in Action supplies a model-agnostic PrivacyChecker over information flows and norms (Wang et al., 2025). CINFA instead fits factor-labeled predictors over a frozen representation of the bound context-action record. The distinction concerns the learned decision interface and its supervision, not an exclusive claim to action-boundary privacy checking.

Interpreting latent evidence. Probe accuracy can reflect the probe's learning capacity (Hewitt & Liang, 2019); decodability does not establish causal use by the backbone (Belinkov, 2022). CINFA's factor-response tests assess learned predictions, while component comparisons evaluate the combined checking and revision policies, without claiming causal disentanglement.