CAN FEDERATED AGENTS AGREE AND STILL BE UNSAFE? FEDSAFEBELIEF FOR REPAIRING FALSE SAFETY CONSENSUS

Anonymous authors
Paper under double-blind review

ABSTRACT

Federated agents may agree on a safety decision because they share the same blind spot. Pooling such decisions can preserve an unsafe rule and return it to clients as apparently reliable feedback. At the same time, differences between local policies can make a rule valid for one client and invalid for another. We study this combination as false safety consensus and propose FedSafeBelief, a correction procedure over client scoped safety beliefs. Clients expose abstract decision tuples while retaining their raw interaction traces. The coordinator aligns these tuples, checks policy scope, searches for counterevidence to both disputed and unanimously allowed decisions, and returns a field level correction. Its central restriction is fix not add: correction may revise an existing client belief but may not create a new client key. We characterize when union aggregation dilutes client scoped precision, why shared errors survive voting, and when verified correction contracts the remaining error. The evaluation design separates belief repair, unsafe feedback, response safety, and legitimate policy heterogeneity. Experiments across controlled belief correction, harmful request, and heterogeneous policy settings show that FedSafeBelief consistently reduces unsafe allows and improves client scoped belief precision while preserving legitimate policy differences.

1 INTRODUCTION

Language model agents increasingly turn interaction histories into reusable memories (Xu et al., 2025). In safety sensitive tasks, these memories can affect which actions are allowed, what conditions must be satisfied, and when an agent should refuse a request. Such interaction histories are often distributed across different users or organizations. Federated in context learning provides a way to coordinate clients through iterative response refinement without exchanging model parameters (Wang et al., 2025). However, sharing safety knowledge introduces a different problem. Agreement among clients does not necessarily mean that the shared decision is correct. The same decision may come from independent evidence, similar policies, or a common blind spot.

Consider several support agents that allow a refund without checking whether the requester owns the payment account. Each client may have observed successful refunds, while none has encountered a case that exposes the missing authorization check. If these local rules are pooled, the incorrect allow decision remains. Voting can give it stronger support. Once the result is returned to clients, the same error may become part of their future guidance. Existing studies on model conformity and collective false memories have shown that interaction among language model agents can reinforce incorrect judgments (Zhu et al., 2025; Xu et al., 2026). In this case, adding more clients does not help because the missing evidence is shared by all of them.

Policy differences create another difficulty. Benchmarks such as τ bench and ST WebAgentBench include domain policies as part of agent evaluation (Yao et al., 2025; Levy et al., 2026). In a federated setting, an action that is valid for one client may be prohibited for another. A single global decision can therefore remove meaningful local differences and introduce new errors. This leads to the question we study in this paper. ***When should agreement among federated agents support a safety decision, and when should that agreement be verified further?***

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.

We study this problem through explicit safety beliefs. Each belief records an action, its conditions, the local decision, the applicable policy scope, and the supporting evidence. Our goal is to correct erroneous beliefs for each client while retaining valid local differences. We do not aim to build the largest possible shared rule set. This makes our setting different from conventional parameter aggregation and from agent systems that use interaction to produce a single common answer.

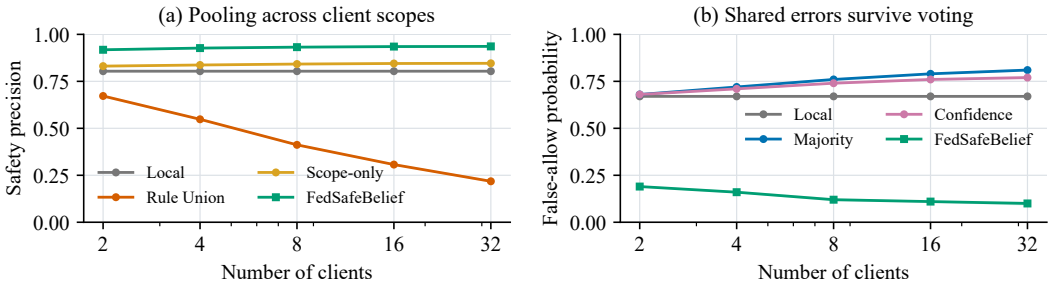

Figure 1: Two motivating comparisons. (a) Sharing rules across more policy scopes can reduce precision for an individual client. (b) Repeated false allow decisions can remain after voting.

Figure 1 varies the number of clients while keeping the local decision task unchanged. The results illustrate two behaviors that are analyzed formally in Section 4.5.

Sharing more rules does not always improve local decisions. When beliefs from different policy scopes are pooled, the target client may receive rules that do not apply to its own setting. These additional beliefs enlarge the prediction set without adding correct beliefs for that client. Scope filtering can prevent such transfers. However, it cannot detect an unsafe decision when the same error appears across compatible clients.

Agreement does not remove a shared error. When several clients repeat the same false allow decision, voting only measures how often that decision appears. It does not provide new evidence about whether the decision is safe. Recent controlled studies also show that coordination among multiple agents can reinforce errors in some settings (Kim et al., 2026). For the problem considered here, a repeated allow decision should be corrected only when there is verified evidence that it fails under the target client's conditions and policy scope.

These observations motivate FedSafeBelief. The coordinator compares matched beliefs across clients and keeps their policy scopes separate. It also audits repeated allow decisions rather than examining disagreements alone. When valid counterevidence is found for a target client, the coordinator changes only the affected field in an existing belief. The main contributions are summarized below.

- We formulate false safety consensus as a client scoped belief correction problem. The formulation distinguishes errors caused by policy scope mismatch from unsafe decisions shared by several clients.
- We develop a correction procedure based on scope checking, verified counterevidence, and the fix not add restriction. The procedure preserves client record ownership and reduces erroneous safety beliefs under stated verification assumptions.
- We design experiments that examine belief quality, repeated feedback, response safety, policy differences, component effects, and coordination cost. The evaluation separates the correction of shared unsafe decisions from the preservation of valid local policies.

2 Related Work

Federated knowledge and agent memory. Beyond parameter aggregation (McMahan et al., 2017; Kairouz et al., 2021), federated prompt and agent learning exchange lightweight adaptations or knowledge (Sun et al., 2023; Che et al., 2023; Wu et al., 2024). Fed ICL further coordinates clients through iterative response refinement (Wang et al., 2025), while A Mem organizes reusable linked memories (Xu et al., 2025). ReAct and Reflexion show that textual feedback can guide later reasoning

and actions (Yao et al., 2023; Shinn et al., 2023). Our setting extends this idea by requiring each shared safety belief to retain its applicable scope and supporting evidence.

Consensus and error propagation. Self consistency improves reliability by aggregating repeated predictions (Wang et al., 2023), but agreement becomes unreliable when clients share the same error. Studies of conformity and collective misremembering show that interaction can reinforce incorrect judgments (Zhu et al., 2025; Xu et al., 2026). GUARDIAN detects anomalous behavior through temporal interaction graphs (Zhou et al., 2025). In contrast, our target error may receive unanimous support, so correction requires evidence beyond consensus.

Safety enforcement and evaluation. AgentSpec uses structured runtime constraints to restrict unsafe behavior (Wang et al., 2026). HarmBench, JailbreakBench, AgentHarm, and OpenAgentSafety evaluate harmful responses and unsafe agent behavior (Zou et al., 2023; Mazeika et al., 2024; Chao et al., 2024; Andriushchenko et al., 2025; Vijayvargiya et al., 2026). These studies focus on enforcement or behavioral evaluation. We instead repair the safety beliefs supplied to federated agents and then evaluate whether the corrected beliefs improve later decisions.

3 Problem Definition

3.1 Federated Safety-Belief Model

We consider K clients. Client i holds private interaction histories, an applicable policy scope Ω_i, and a finite set of predicted beliefs $\widehat{\mathcal{B}}_i$. A belief has the form

$$b = (X, A, C, Y, V, E, S, Q), \qquad Y \in \{\text{allow}, \text{refuse}, \text{revise}, \text{block}\}. \tag{1}$$

Here X identifies a request type, A an action, C its conditions, V a risk category, E exceptions or counterevidence, S the scope and policy version, and Q the evidence provenance and confidence. A stable client-owned record identifier is the patch key. Canonical (X, A) pairs are used only to find potentially comparable records; they do not collapse different conditions or scopes into one decision.

The server receives normalized tuples and permitted evidence summaries. It does not receive raw private traces. This is a data-access restriction, not a formal privacy guarantee: a tuple can itself reveal sensitive information. Ground-truth beliefs $\mathcal{B}_i^*$ and hidden evaluation labels are available to the evaluator, not to the coordinator. Counterevidence must come from the designated verification split or an authorized policy/tool verifier.

3.2 False Safety Consensus and Local Correctness

Definition 1 (False safety consensus)**.** *For a matched action under compatible conditions and scope, false safety consensus occurs when multiple clients support* $Y = \text{allow}$ *although the applicable policy or action precondition requires a non-allow decision, and aggregation retains that allow as shared feedback.*

Consensus is a statement about repeated decisions. Correlation is a statement about their joint error process. Neither can be inferred solely from a high support count. In particular, a group of correct allows and a group of shared false allows can have identical votes and confidence values. External counterevidence or additional trusted structure is needed to distinguish them.

For evaluation, let $P_i = \text{proj}(\widehat{\mathcal{B}}_i)$ and $G_i = \text{proj}(\mathcal{B}_i^*)$. The projection retains the canonical request/action, condition, decision, exceptions, and scope; it excludes confidence and non-semantic provenance fields. We use

$$\text{SP}_i = \frac{|P_i \cap G_i|}{|P_i|}, \qquad \text{SR}_i = \frac{|P_i \cap G_i|}{|G_i|}. \tag{2}$$

An empty prediction has undefined precision and zero recall when G_i is nonempty. It is not assigned perfect precision. We report coverage alongside both metrics. Beliefs outside Ω_i fail the client-scoped match even if they are valid for another client; this evaluates applicability, not universal truth.

3.3 Scope-Preserving Federated Correction

The coordinator produces a patch π_i for each client. Let $\widehat{\mathcal{B}}_i^+ = \pi_i(\widehat{\mathcal{B}}_i)$. The proposed correction problem is

$$\min_{\{\pi_i\}} \sum_{i=1}^{K} w_i \, \mathbb{E}[\ell_i(\widehat{\mathcal{B}}_i^+)] \quad \text{subject to} \quad \text{keys}(\widehat{\mathcal{B}}_i^+) \subseteq \text{keys}(\widehat{\mathcal{B}}_i), \tag{3}$$

where $w_i \geq 0$, $\sum_i w_i = 1$, and ℓ_i penalizes unsafe allows, false refusals, and incorrect scoped beliefs. The loss defines the evaluation target; the server does not optimize against hidden labels. Our implementation-level rule uses verified counterevidence as a surrogate. Pure replacement preserves key-set equality, a stronger property than the constraint above.

Fix-not-add limits what a correction can change. It does not prove that the replacement is correct. A valid patch must additionally preserve the target scope, identify the invalidated field, cite accepted evidence, and leave unrelated fields unchanged. Missing or ambiguous evidence results in abstention. Section 4.5 separates this structural invariant from the assumptions required for an improvement in safety.

4 FedSafeBelief: Scope-Preserving Safety Correction

4.1 Overview

FedSafeBelief operates between local belief extraction and the next use of feedback, as shown in Figure 2. For client i, let $\widehat{\mathcal{B}}_i$ denote its current belief set. One correction round can be written as

$$\widehat{\mathcal{B}}_i^+ = \pi_i \left(\widehat{\mathcal{B}}_i, \mathcal{A}(\{\widehat{\mathcal{B}}_j\}_{j=1}^K), \mathcal{V}_i \right), \tag{4}$$

where $\mathcal{A}(\cdot)$ denotes cross client belief alignment and $\mathcal{V}_i$ denotes the verified evidence available under client i's policy scope. The correction operator π_i changes a belief only when a correlated unsafe error is supported by verified counterevidence. Other cases retain the original belief, request additional evidence, or return to alignment.

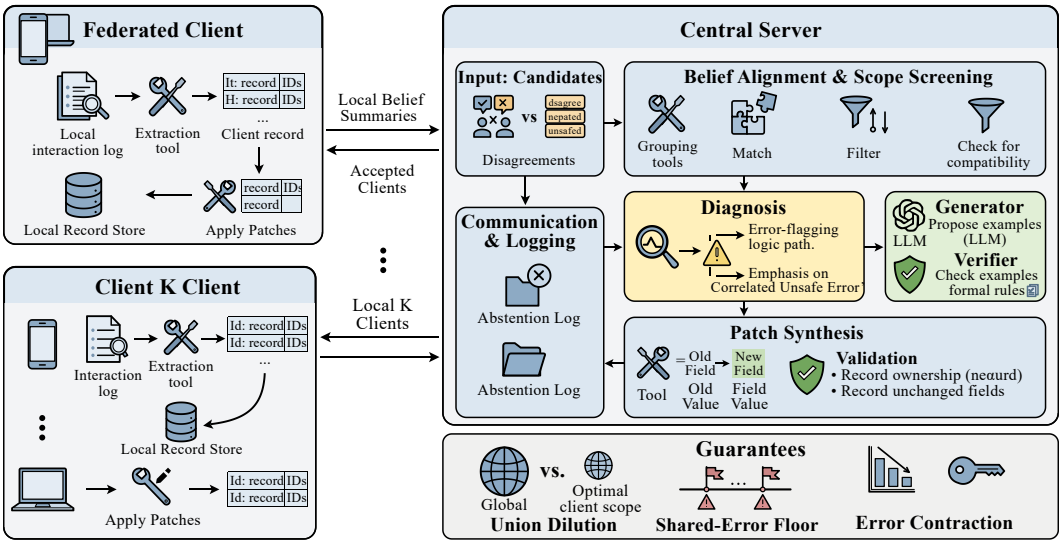

Figure 2: Overview of the FedSafeBelief correction process. Disagreements and repeated allows are jointly examined, while verified evidence only updates existing client beliefs.

4.2 Belief Alignment and Scope Screening

The extractor maps each interaction step to Equation (1) with a stable record identifier. The server first groups records by their canonical request and action pair. For two beliefs b and b', we define the

compatibility indicator as

$$M(b, b') = \mathbb{I}[(X, A) = (X', A')]\,\mathbb{I}[\Gamma(C, C') = 1]\,\mathbb{I}[\Psi(S, S') = 1]\,, \tag{5}$$

where $\Gamma(\cdot)$ checks condition compatibility and $\Psi(\cdot)$ checks whether the policy scopes and versions are compatible. Lexical similarity is used only to propose candidate pairs and does not determine $M(b, b')$.

Known policy version conflicts make $\Psi(S, S') = 0$ unless the verifier confirms that the relevant rule is unchanged. If either condition or scope compatibility cannot be established, the match remains unresolved. The coordinator then keeps the original belief or requests additional evidence rather than resolving the uncertainty through majority support.

4.3 Diagnosis with Verified Counterevidence

The audit includes both disagreements and repeated safety relevant allows. This is necessary because a disagreement only procedure cannot expose an unsafe decision that all matched clients support. The diagnosis considers partial observability, temporal differences, abstraction mismatch, extraction variability, stale policies, alignment errors, correlated unsafe errors, scope mismatch, and insufficient evidence. Appendix A gives the corresponding routing rules.

For a potentially unsafe belief b, the generator proposes a bounded counterexample set

$$\mathcal{E}_i(b) = \{e_m \mid m \le B,\ e_m \models C,\ e_m \in \Omega_i\}\,. \tag{6}$$

A verifier then evaluates whether each counterexample actually invalidates the current decision. We define

$$R_i(b) = \mathbb{I}\left[\exists e \in \mathcal{E}_i(b) : \mathcal{V}(e, b, \Omega_i) = 1\right], \tag{7}$$

where $\mathcal{V}(e, b, \Omega_i) = 1$ means that e satisfies the stated conditions, is valid under the target scope, and contradicts the current allow decision.

Deterministic policy or tool validators are preferred when available. An LLM verifier must provide evidence and a structured justification. Another client's private fact or an incompatible policy cannot make $R_i(b) = 1$. A correlated unsafe error is therefore repaired only when the target record exists, its scope is compatible, and $R_i(b) = 1$. Otherwise, the original belief is retained and the case is recorded as an abstention.

4.4 Field Level Patch Synthesis

Once a correction is verified, FedSafeBelief changes only the field implicated by the accepted evidence. Let f denote that field and let b_f be its current value. The corrected belief is

$$b_g^+ = \begin{cases} b_f^{\text{new}}, & g = f, \\ b_g, & g \neq f, \end{cases} \qquad \text{key}(b^+) = \text{key}(b). \tag{8}$$

Thus, all unrelated fields remain unchanged and the correction cannot create a new local record.

Each patch records the target identifier, expected version, edited field, old value, new value, and supporting evidence. A patch is accepted only if

$$\text{Valid}(p, b) = \mathbb{I}[\text{id}_p = \text{id}_b]\mathbb{I}[v_p = v_b]\mathbb{I}[S(b^+) = S(b)]\mathbb{I}[\text{keys}(\widehat{\mathcal{B}}_i^+) \subseteq \text{keys}(\widehat{\mathcal{B}}_i)]. \tag{9}$$

Validation therefore rejects new keys, stale versions, scope changes, and unrelated edits. The client applies the accepted replacement locally and may render it as a short feedback instruction. The rendered text must express the validated field change without introducing additional content.

4.5 Federated Update and Conditional Guarantees

Algorithm 1 specifies one round. Communication consists of tuple summaries and accepted patches; there is no weight update. Repeated records can share a verification result only when the target conditions and policy scope are equivalent. Otherwise they require separate checks.

Their proofs and counterexamples appear in Appendix C. The first two concern aggregation; the third concerns correction under explicit verifier assumptions.

Algorithm 1 FedSafeBelief: verified fix-not-add correction

Require: Local records $\{\widehat{\mathcal{B}}_i\}_{i=1}^{K}$, scopes $\{\Omega_i\}$, budget B
1: Normalize request/action keys; retain client record IDs and policy versions.
2: Form candidates from disagreements *and* repeated safety-relevant allows.
3: **for** each candidate group and target client i owning a matched record b **do**
4: Check condition and scope compatibility; retain b if compatibility is unknown.
5: Search up to B counterexamples and verify them under Ω_i.
6: Diagnose the source using matched records and verified evidence.
7: **if** diagnosis is correlated unsafe error and an in-scope counterexample is accepted **then**
8: Propose replacement of the invalidated field of b.
9: Validate record ownership, old value, policy version, and unchanged fields.
10: Apply the valid patch; otherwise retain b and record abstention.
11: **end if**
12: **end for**
13: **return** Corrected local records, patches, evidence references, and abstention log

Proposition 1 (Union dilution)**.** *Let P_i contain $c > 0$ correct and h incorrect client-scoped projections. If $\bigcup_j P_j$ adds $a > 0$ incorrect projections and no new member of G_i, its precision is $c/(c+h+a) < c/(c+h)$.*

The no-gain assumption matters. A union can improve recall when it contributes genuinely applicable beliefs. The proposition does not imply that every form of sharing is harmful; it identifies the case in which a global union optimizes the wrong scope.

Proposition 2 (A shared-error floor)**.** *Suppose an unsafe item has a shared blind-spot event Z with $\Pr(Z) = q$. On Z, every client returns allow. Any aggregator that preserves unanimous allows has unsafe-allow probability at least q, independently of the number of clients.*

This formulation includes union, majority, and nonnegative confidence voting when their unanimity rule is preserved. It does not assume that pairwise correlation alone determines the full joint error distribution.

Theorem 1 (Ownership and residual-error contraction)**.** *Assume all accepted patches replace fields of existing records without changing scope. Then key ownership is preserved in every round. On a fixed population of owned records, let e_t be the expected fraction with an unsafe erroneous field. If each such error is correctly repaired with conditional probability at least $\gamma > 0$, and the expected fraction of newly introduced unsafe errors is at most ϵ, then*

$$e_{t+1} \le (1-\gamma)e_t + \epsilon, \qquad e_t \le (1-\gamma)^t e_0 + \frac{\epsilon}{\gamma}\big[1-(1-\gamma)^t\big]. \tag{10}$$

With $\epsilon = 0$, the residual contracts geometrically. With imperfect verification, the bound leaves an error floor. This is a statement about the fixed record population, not about the fraction of emitted patches that are wrong. Patch-conditioned Unsafe Update Rate has a changing denominator and need not contract at the same rate.

5 Experiments

5.1 Experimental Setup

Benchmarks. We use ALFWorld, AdvBench, HarmBench, and policy variants derived from τ bench. ALFWorld provides action preconditions for constructing client owned safety beliefs, and we additionally introduce paraphrase and decision noise to test imperfect extraction (Shridhar et al., 2021). AdvBench and HarmBench are used for unsafe response evaluation together with benign controls (Zou et al., 2023; Mazeika et al., 2024). The airline and retail tasks from τ bench are adapted to create different policy scopes across clients (Yao et al., 2025).

Federated setting. Unless otherwise stated, we use eight clients, five feedback rounds, and five repeated runs. We vary $K \in \{2, 4, 8, 16, 32\}$ in the scalability experiments. Each client holds its own

policy rules together with several shared erroneous allows. Verification examples are separated from evaluation examples, and all methods start from the same local records in each run.

Baselines. We compare against Local, Rule Union, Majority, Confidence, and Scope Only. Local keeps each client's original beliefs unchanged. Rule Union directly pools records across clients (Wang et al., 2025). Majority uses the most frequent decision among matched records (Kaesberg et al., 2025), while Confidence further weights decisions by evidence scores. Scope Only filters incompatible conditions and policy scopes before aggregation, but does not verify counterevidence. FedSafeBelief uses the same initial records and extraction results as these baselines.

Metrics. We report Safety Precision (SP), Safety Recall (SR), Unsafe Allow Rate (UAR), Over Refusal Rate (ORR), and Unsafe Update Rate (UUR). SP and SR measure client belief quality, UAR measures prohibited allows, ORR captures unnecessary refusals, and UUR evaluates unsafe correction fields. Response level UAR is computed from agent outputs after corrected beliefs are applied. Results are averaged across clients and then summarized over five runs.

Table 1: Client-scoped belief quality (%).

	Controlled rules			Noisy extraction		
Method	SP ↑	SR ↑	ORR ↓	SP ↑	SR ↑	ORR ↓
Local	80.4±1.0	80.4±1.0	11.6±1.0	86.2±1.1	86.2±1.1	9.4±0.8
Rule Union	41.2±0.7	87.4±0.7	20.5±0.5	48.6±0.6	88.3±1.7	18.8±0.7
Majority	73.5±0.7	67.2±0.4	16.4±1.1	78.1±1.1	71.8±0.3	14.5±0.8
Confidence	77.2±0.6	74.2±0.3	13.9±0.4	81.2±0.5	77.3±0.6	12.3±0.6
Scope-only	84.2±0.4	83.0±0.5	9.8±0.5	87.8±1.1	85.8±0.6	7.9±0.7
FedSafeBelief	93.2±0.4	93.2±0.4	6.2±0.9	94.6±0.8	94.6±0.8	5.1±0.5

5.2 Client Scoped Belief Quality

Table 1 compares the quality of client beliefs under controlled rules and noisy extraction. Rule Union improves SR over Local but causes a clear drop in SP because records valid for other clients are also introduced into the target belief set. Scope Only removes part of this mismatch and therefore recovers precision, but it cannot correct an unsafe belief that is already shared by clients with compatible scopes. FedSafeBelief addresses both cases and achieves an SP of 93.2% in the controlled setting, compared with 80.4% for Local, while the corresponding values under noisy extraction are 94.6% and 86.2%. The improvement under both settings shows that the gain does not come from enlarging the local belief set, but from correcting erroneous records while retaining client specific scope.

5.3 Federated Feedback under Shared Unsafe Errors

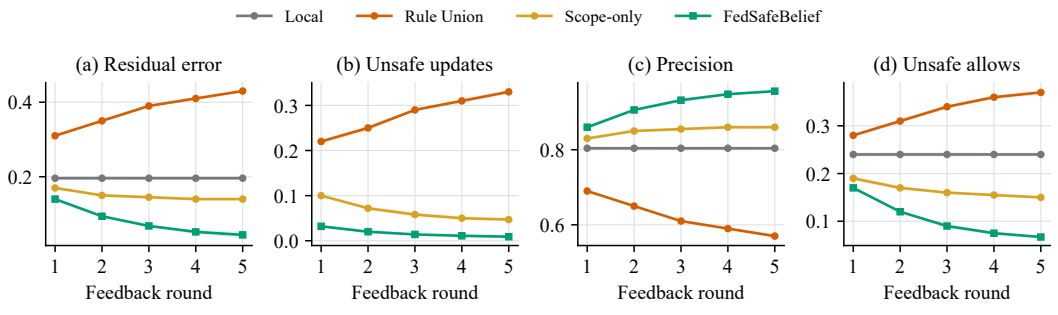

Figure 3: Feedback dynamics over five communication rounds.

Figure 3 shows how the same client belief population evolves over five feedback rounds. Local remains nearly unchanged because no cross client correction is performed, while Rule Union repeatedly returns incorrect or irrelevant records and therefore does not remove the shared error. Scope Only reduces errors caused by incompatible policies, but repeated unsafe allows remain when all matched

clients make the same decision. In contrast, FedSafeBelief checks these agreements against verified counterevidence before returning feedback, reducing the residual error from 14.0% after the first round to 4.4% after the fifth round. The simultaneous reduction in unsafe updates and unsafe allows shows that the improvement does not come from simply suppressing feedback or refusing more cases. Instead, FedSafeBelief progressively removes shared erroneous beliefs from the client records, which is consistent with the contraction behavior in Equation (10). These results show that verified correction is necessary when unsafe consensus persists across clients, since repeated agreement alone cannot eliminate the shared error.

5.4 Response Level Safety and Policy Boundaries

Table 2: Response safety and policy boundary evaluation (%).

	AdvBench protocol		HarmBench protocol		Policy scope protocol	
Method	UAR ↓	ORR ↓	UAR ↓	ORR ↓	UAR ↓	ORR ↓
Local	18.6±1.0	8.0±0.7	22.1±0.4	9.2±0.7	7.4±0.9	7.1±1.1
Rule Union	24.3±0.7	11.3±0.5	27.8±0.9	12.7±0.8	15.8±0.6	16.4±0.8
Majority	15.7±0.6	18.6±0.4	18.8±0.8	21.4±1.1	10.3±0.7	20.5±0.4
Confidence	17.1±0.4	14.1±0.6	20.5±0.8	15.8±1.0	11.1±0.3	13.3±0.6
Scope Only	14.8±0.9	7.9±0.8	17.5±0.5	9.5±0.7	5.2±0.9	5.5±0.4
FedSafeBelief	7.1±0.8	8.5±0.3	9.3±0.6	9.8±0.6	5.7±0.7	6.0±0.4

Table 2 evaluates whether corrected beliefs also improve the decisions produced by the agent. On AdvBench and HarmBench, FedSafeBelief reduces UAR from 18.6% to 7.1% and from 22.1% to 9.3% relative to Local, while ORR changes only slightly. Majority and Confidence reduce some unsafe allows, but their much higher ORR shows that agreement based correction can become overly conservative. The policy scope setting gives a different result. Scope Only obtains a slightly lower UAR and ORR than FedSafeBelief, indicating that additional correction is not always beneficial when most differences arise from legitimate local policies rather than shared errors. This result also marks the operating boundary of FedSafeBelief: verified correction is most useful when unsafe consensus is present, while scope filtering alone can be sufficient when client differences are already valid.

5.5 Ablation and Robustness

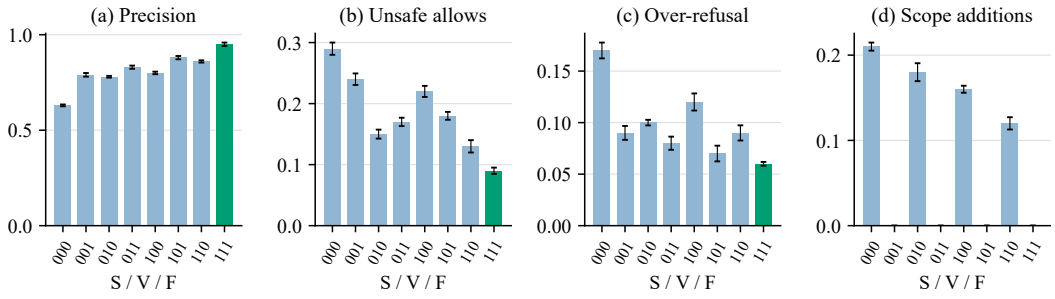

Figure 4: Ablation of scope gating (S), verified counterevidence (V), and fix not add (F).

Figure 4 separates the roles of scope gating, verified counterevidence, and fix not add. Removing scope gating mainly reduces SP because beliefs are transferred across incompatible policies, while removing verification produces the largest increase in UAR because repeated allow decisions can no longer be distinguished from shared unsafe errors. Disabling fix not add permits new records to enter a client belief set. This can improve coverage, but it also increases the chance of importing beliefs that were never valid for that client. None of the three components therefore solves the complete problem alone. Scope gating determines where a belief applies, verification determines whether it should be changed, and fix not add limits how that change is introduced. The full configuration gives the most balanced behavior across SP, UAR, and ORR rather than optimizing one metric at the expense of the others.

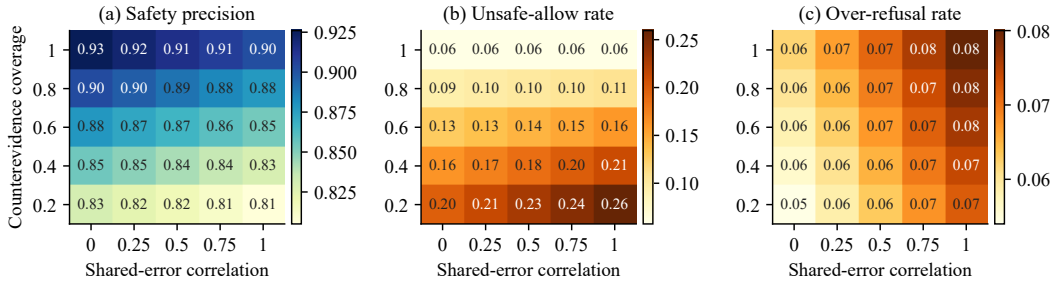

Figure 5: Robustness under different levels of shared error correlation and counterevidence coverage.

Figure 5 further varies shared error correlation and counterevidence coverage. Increasing correlation makes consensus less informative because more clients repeat the same incorrect allow decision, which gradually lowers SP and increases UAR when verification coverage is limited. Increasing counterevidence coverage has the opposite effect and consistently reduces unsafe allows across different correlation levels. ORR remains comparatively stable, showing that the reduction in UAR does not mainly come from replacing uncertain decisions with refusal. The results also explain why adding more clients alone is insufficient for this problem. When the missing evidence is shared, a larger voting population only repeats the same information, whereas additional valid counterevidence directly changes whether the unsafe belief can be identified and corrected.

5.6 Efficiency and System Cost

Table 3: Per round system cost with eight participating clients.

Method	Time (s) ↓	Calls ↓	Upload (kB)	Download (kB)
Local	0.00	0	0	0
Rule Union	0.31	0	48.6	61.2
Majority	0.42	0	48.6	18.4
Confidence	0.47	0	48.6	19.1
Scope Only	0.69	0	48.6	15.8
FedSafeBelief	8.42	24	48.6	6.7

Table 3 shows that the additional cost of FedSafeBelief mainly comes from evidence generation and verification rather than communication. All coordination methods upload the same 48.6 kB of normalized belief summaries, while FedSafeBelief returns only 6.7 kB because feedback contains targeted field corrections instead of pooled records. This reduces the download payload relative to Rule Union, but increases per round time to 8.42 s and requires 24 server calls. The computation therefore trades additional verification cost for more selective feedback. With N uploaded records, M candidate groups, and at most B counterexamples per group, grouping requires expected $O(N)$ time, scope comparison has a worst case cost of $O(N^2)$, and verification requires at most $O(MB)$ calls. Communication depends on the serialized beliefs and accepted patches rather than model size, which keeps the exchanged content compact when only a small part of the local belief set requires correction.

6 Conclusion

In this paper, we study false safety consensus in federated agents and propose FedSafeBelief for client scoped belief correction. FedSafeBelief combines belief alignment, policy scope screening, verified counterevidence, and field level patching, while the fix not add rule prevents unsupported knowledge transfer. We show that shared unsafe errors can survive consensus based aggregation and that verified correction can reduce residual errors under stated assumptions. Experiments demonstrate lower unsafe allow rates and better client scoped belief quality while preserving legitimate policy differences. Future work will improve verification reliability and extend the framework to malicious clients and dynamic policy settings.

AI USE STATEMENT

Generative AI was used only to assist with language polishing and improve the clarity of the manuscript.

REPRODUCIBILITY STATEMENT

Experimental settings, metrics, client partitions, and paired comparisons are detailed in Section 5 and Appendix B. All methods use the same data and evaluation settings within each run. Additional proofs and diagnostic results are provided in Appendices C and D.

ETHICS STATEMENT

This work studies safety errors shared among federated agents. Experiments use established benchmarks and controlled policy settings without collecting new human subject data. Harmful requests are evaluated in isolated settings. The exchanged belief tuples reduce raw data exposure but do not provide formal privacy guarantees or protection against malicious clients.

REFERENCES

Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. AgentHarm: A benchmark for measuring harmfulness of LLM agents. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/c493d23af93118975cdbc32cbe7323f5-Abstract-Conference.html.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramer, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*, 2024.

Tianshi Che, Ji Liu, Yang Zhou, Jiaxiang Ren, Jiwen Zhou, Victor Sheng, Huaiyu Dai, and Dejing Dou. Federated learning of large language models with parameter-efficient prompt tuning and adaptive optimization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7871–7888, 2023.

Lars Benedikt Kaesberg, Jonas Becker, Jan Philip Wahle, Terry Ruas, and Bela Gipp. Voting or consensus? decision-making in multi-agent debate. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 11640–11671. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-acl.606.

Peter Kairouz, H. Brendan McMahan, Brendan Avent, Aurelien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14 (1–2):1–210, 2021.

Yubin Kim, Ken Gu, Chanwoo Park, Chunjong Park, Samuel Schmidgall, A. Ali Heydari, Yao Yan, Zhihan Zhang, Yuchen Zhuang, Yun Liu, Mark Malhotra, Paul Pu Liang, Hae Won Park, Yuzhe Yang, Xuhai Xu, Yilun Du, Shwetak Patel, Tim Althoff, Daniel McDuff, and Xin Liu. Capable language models can outgrow the benefits of collaboration. *Nature Machine Intelligence*, 8:1157–1172, 2026. doi: 10.1038/s42256-026-01268-y. URL https://www.nature.com/articles/s42256-026-01268-y.

Ido Levy, Ben Wiesel, Sami Marreed, Alon Oved, Avi Yaeli, and Segev Shlomov. ST-WebAgentBench: A benchmark for evaluating safety and trustworthiness in web agents. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/c6c29e590e3c62e37e6b39cdd6baf2e8-Abstract-Conference.html.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, 2024.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, 2023.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Cote, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*, 2021.

Adi Simhi, Jonathan Herzig, Martin Tutek, Itay Itzhak, Idan Szpektor, and Yonatan Belinkov. ManagerBench: Evaluating the safety-pragmatism trade-off in autonomous LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b8330f5b70b3c53172417deac6f057b1-Abstract-Conference.html.

Jingwei Sun, Ziyue Xu, Hongxu Yin, Dong Yang, Daguang Xu, Yiran Chen, and Holger R. Roth. Fedbpt: Efficient federated black-box prompt tuning for large language models. *arXiv preprint arXiv:2310.01467*, 2023.

Sanidhya Vijayvargiya, Aditya Soni, Xuhui Zhou, Zora Zhiruo Wang, Nouha Dziri, Graham Neubig, and Maarten Sap. OpenAgentSafety: A comprehensive framework for evaluating real-world AI agent safety. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/a0f6c06b8d6114bb37688c2dc19b4d74-Abstract-Conference.html.

Haoyu Wang, Christopher M. Poskitt, and Jun Sun. AgentSpec: Customizable runtime enforcement for safe and reliable LLM agents. In *Proceedings of the 48th IEEE/ACM International Conference on Software Engineering*, pp. 2938–2950. ACM, 2026. doi: 10.1145/3744916.3764546. URL https://doi.org/10.1145/3744916.3764546.

Ruhan Wang, Zhiyong Wang, Chengkai Huang, Rui Wang, Tong Yu, Lina Yao, John C. S. Lui, and Dongruo Zhou. Federated in-context learning: Iterative refinement for improved answer quality. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 64628–64654. PMLR, 2025.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

Panlong Wu, Kangshuo Li, Junbao Nan, and Fangxin Wang. Federated in-context llm agent learning. *arXiv preprint arXiv:2412.08054*, 2024.

Naen Xu, Hengyu An, Shuo Shi, Jinghuai Zhang, Chunyi Zhou, Changjiang Li, Tianyu Du, Zhihui Fu, Jun Wang, and Shouling Ji. When agents "misremember" collectively: Exploring the mandela effect in LLM-based multi-agent systems. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/531230cfac80c65017ad0f85d3031edc-Abstract-Conference.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-Mem: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0593. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/19909c36f51abc4856b4560aff3d36d6-Abstract-Conference.html.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/1b126cc38b8638e07bef37e7b2bb72bf-Abstract-Conference.html.

Jialong Zhou, Lichao Wang, and Xiao Yang. GUARDIAN: Safeguarding LLM multi-agent collaborations with temporal graph modeling. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-0272. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/0bc795afae289ed465a65a3b4b1f4eb7-Abstract-Conference.html.

Xiaochen Zhu, Caiqi Zhang, Tom Stafford, Nigel Collier, and Andreas Vlachos. Conformity in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3854–3872. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.195. URL https://aclanthology.org/2025.acl-long.195/.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

APPENDIX

Additional method details, experimental protocols, complete proofs, and diagnostic analyses accompany the main paper.

A Additional Method Details

A.1 Record Identity, Alignment, and Evaluation Projection

A client record has two identifiers with different purposes. Its immutable local identifier determines ownership and is the only address to which a patch may be applied. A normalized semantic key is used to identify potentially comparable records across clients. Several records may share a semantic key while differing in conditions, scope, or policy version, so these records are not merged directly. A patch also carries an expected record version to prevent a delayed correction from overwriting a subsequent local edit.

The evidence field stores provenance rather than only a confidence value. It includes the permitted evidence reference, the policy version used for verification, and whether the evidence is observed, inferred, or generated. Generated text is accepted as counterevidence only after validation by the designated verifier. A support count is treated as descriptive information rather than evidence of correctness. The risk category is used to select an appropriate verifier but does not determine the decision label.

The evaluation projection removes presentation details and provenance identifiers. Records with the same decision and applicability conditions are therefore not treated as different beliefs only because their confidence values differ. Scope is retained because removing it would allow a belief from an incompatible policy to appear correct. We retain canonical (X, A), normalized conditions, decision, sorted exceptions, and scope. The same normalization rules are applied to predictions and reference beliefs.

A.2 Diagnosis Sources and Their Operational Meaning

Table 4: Source taxonomy and routing policy. The nine sources describe diagnostic causes rather than independent correction algorithms.

ID	Source	Evidence needed and coordinator response
S1	Partial observability	One client lacks observations available to another. Retain ownership and request local evidence.
S2	Temporal view difference	Both observations can be correct at different times. Separate their validity intervals.
S3	Abstraction mismatch	A coarse action groups distinct preconditions. Refine alignment before comparison.
S4	Extraction variability	Repeated extraction disagrees on the same input. Retain uncertainty or repeat extraction.
S5	Stale policy	A record cites an obsolete policy. Route it to version review rather than correlated error repair.
S6	Semantic alignment error	Surface similarity joins different actions. Split the candidate group.
S7	Correlated unsafe error	Repeated allows fail a verified counterexample within the target scope. Permit a validated field replacement.
S8	Scope mismatch	Different valid policies explain the decisions. Retain client specific rules.
S9	Insufficient evidence	Available evidence does not resolve the case. Abstain and record the missing information.

Temporal view difference and stale policy are distinguished by the target validity interval. An old observation can remain correct for an earlier policy and becomes stale only when it is applied to the current decision. Partial observability is also not treated as an error simply because one client contains fewer records. The current version of FedSafeBelief repairs S7 and routes the remaining

sources according to Table 4. This keeps the correction operation focused on shared unsafe errors rather than treating every disagreement as incorrect.

A.3 Scope Compatibility and Evidence Access

Compatibility requires the target action conditions to agree with the policy under which the counterexample is verified. Domain tags, policy versions, authorization context, and validity intervals are checked explicitly. Lexical similarity can identify candidate records but cannot authorize knowledge transfer. Missing scope information produces an unknown compatibility result. A generated scenario that changes the original prerequisites is also rejected because it describes a different rule rather than contradicting the current one.

The coordinator does not receive held out evaluation labels. Verification relies on authorized policy records, tools, or evidence reserved for that purpose. This separation prevents the same label from being used both to trigger a correction and to evaluate whether that correction succeeds. Evidence references therefore point to verification records rather than evaluation examples.

A.4 Patch Validation and Application

A patch is a structured object containing `client_id`, `record_id`, `expected_version`, `field`, `old_value`, `new_value`, `scope_id`, and `evidence_id`. The validator checks that the record exists, its version and old value match, the selected field is repairable, the evidence applies to the unchanged scope, and no unrelated field is modified. The patch is atomic. A failed check leaves the record unchanged, and an already applied patch does not count as another successful correction.

For correlated unsafe allows, the editable fields are the decision and the condition that directly licenses the unsafe action. The evidence reference is stored as audit metadata without changing unrelated semantic fields. Updating the client scope or adding a new action record is not permitted. A verbal feedback message can be produced after validation, but it expresses only the accepted structured edit.

Worked example. A client owns record `refund-17`, version 3, whose condition checks that an order exists and whose decision permits a refund. The policy verifier identifies an authorization requirement showing that ownership must also be checked. A valid patch adds the missing prerequisite to the existing condition and advances the version. It does not introduce another client's cancellation policy, modify the record scope, or preserve the refund decision merely because several clients previously made the same choice.

B Experimental Protocol and Reproducibility

B.1 Controlled Records and Client Partitioning

The controlled protocol begins with canonical action and precondition records assigned to explicit client scopes. We split the data by task or policy instance rather than by individual tuple so that records originating from the same interaction do not cross the verification and evaluation boundary. Clients are grouped by environment or policy variant before errors are introduced. Shared errors modify the same prohibited allow boundary for a selected group of clients, whereas local errors affect independently selected records.

The mutation mask and the clean reference records are stored separately and are not provided to the diagnoser. Extraction noise is introduced independently from the reference beliefs. Comparisons under different extraction conditions use the same underlying records. Exact extraction responses are cached so that all methods operate on identical inputs.

B.2 Harmful Requests, Benign Controls, and Policy Tasks

The harmful request evaluation measures the final response produced for each input rather than only the internal allow or refuse label. This distinction is necessary because an agent may refuse despite an allow belief or may still produce unsafe content after a refuse decision. Final outputs are therefore scored with the same evaluator across all methods.

Table 5: Experimental configuration used in the evaluation.

Field	Setting
Default clients / rounds	8 clients / 5 feedback rounds
Repeated runs	5 runs
Client count sweep	$2, 4, 8, 16, 32$
Counterexample budget	Default $B = 3$, sweep $0, 1, 2, 3, 4, 6$
Shared error correlation	$0, 0.25, 0.5, 0.75, 1$
Counterevidence coverage	$0.2, 0.4, 0.6, 0.8, 1$

AdvBench and HarmBench provide the harmful request populations. ORR is evaluated on a separate benign set matched where possible by topic and response format. Duplicate and near duplicate requests remain within the same split to prevent information leakage. The harmful and benign populations are kept separate because reductions in unsafe responses should not be obtained by indiscriminate refusal.

The policy tasks use airline and retail rules to evaluate whether correction preserves legitimate differences between clients. Policy compliance and task completion are considered together because a correct local action may differ across policy scopes. This setting therefore measures whether FedSafeBelief corrects shared unsafe beliefs without forcing different clients toward one global decision.

B.3 Metric Denominators and Aggregation

Let H_i contain unsafe evaluation items and L_i benign items. With an emitted response scorer r, we define

$$\mathrm{UAR}_i^{\mathrm{response}} = \frac{\sum_{z \in H_i} \mathbf{1}\{r(\mathrm{response}_i(z)) = \mathrm{unsafe}\}}{|H_i|}, \tag{11}$$

$$\mathrm{ORR}_i = \frac{\sum_{z \in L_i} \mathbf{1}\{\mathrm{unjustified\ refusal}_i(z)\}}{|L_i|}. \tag{12}$$

Belief UAR instead checks whether an unsafe key contains a prohibited allow in the corrected belief store. Mutation preservation measures whether an initially introduced erroneous belief remains after correction. These quantities use different numerators and are therefore reported separately from response UAR. If H_i or L_i is empty, the corresponding rate is treated as undefined.

For patch level UUR, each asserted field value counts as one affected entry. A full union feedback operation therefore counts every reasserted value, whereas a targeted patch counts only the fields that are changed. An abstention or empty patch contains no affected entries and does not receive a zero UUR value. For the pooled result, erroneous fields and affected fields are summed before the rate is computed.

Coverage measures the fraction of reference keys with at least one prediction. Safe completion requires both successful task execution and satisfaction of the safety evaluator. Reporting completion together with UAR prevents blanket refusal from appearing as a safety improvement. SP and SR operate on deduplicated belief projections, while UAR and ORR operate on evaluation items, so their denominators are defined independently.

B.4 Baseline Parity and Statistical Reporting

Local records, client partitions, candidate alignment, and evaluation items are fixed within each run. Scope Only uses the same compatibility gate as FedSafeBelief. Majority and Confidence differ in their aggregation rule, and ties result in abstention. For methods that do not return structured patches, UUR is computed from the records actually included in their feedback so that different feedback formats use the same affected entry definition.

Results are first averaged over clients within each run and then summarized across five runs. We report the mean and standard deviation across runs. All compared methods use identical starting

records and evaluation items within a run, which allows differences to be attributed to the coordination and correction procedure rather than different client inputs.

C Theoretical Analysis

C.1 Proof of Union Dilution

Proof of Proposition 1. The target client prediction has $c + h$ distinct projected beliefs, of which c match the target truth. Under the no gain assumption, the union retains exactly those c correct projections and adds a incorrect projections. Therefore

$$\mathrm{SP}_i(P_i) - \mathrm{SP}_i\Big(\bigcup_j P_j\Big) = \frac{c}{c+h} - \frac{c}{c+h+a} = \frac{ca}{(c+h)(c+h+a)} > 0. \tag{13}$$

All denominators are positive because $c > 0$, and the strict inequality follows from $a > 0$. □

The statement uses set cardinalities after deduplication. Repeating an identical projected belief does not lower precision under this definition. The loss comes from distinct incorrect or incompatible projections rather than repeated copies of the same belief. If the union adds g correct projections together with a incorrect projections, its precision improves exactly when

$$\frac{c+g}{c+h+g+a} > \frac{c}{c+h} \quad \Longleftrightarrow \quad gh > ca. \tag{14}$$

This condition also explains why complementary knowledge can be useful when it is valid for the target client. The fix not add restriction is therefore designed for belief correction rather than general knowledge acquisition.

C.2 Proof of the Shared Error Floor

Proof of Proposition 2. Let A denote the event that the aggregator returns allow. Because the aggregator preserves unanimous allows, $Z \subseteq A$. Thus

$$\Pr(A) \geq \Pr(Z) = q.$$

The inclusion does not depend on the number of clients. □

An explicit construction further illustrates the common cause. Draw $Z \sim \mathrm{Bernoulli}(q)$. If $Z = 1$, every client error indicator is set to one. Otherwise, the indicators are sampled independently with probability p_0. The marginal error probability of each client is

$$p = q + (1-q)p_0, \tag{15}$$

and for distinct clients,

$$\mathrm{Cov}(X_i, X_j) = q(1-q)(1-p_0)^2. \tag{16}$$

For strict majority voting, the error probability is

$$q + (1-q) \sum_{j=\lfloor K/2 \rfloor + 1}^{K} \binom{K}{j} p_0^j (1-p_0)^{K-j}. \tag{17}$$

When $p_0 < 1/2$, the independent error component decreases as K grows, while the shared error term remains. The result therefore describes the common blind spot considered in this paper rather than every possible form of correlated error.

C.3 An Exchangeable Vote Bound

The vote based formulation can also be stated under an exchangeability assumption. Suppose $X_1, \ldots, X_K$ are exchangeable Bernoulli indicators with mean $p > 1/2$ and common pairwise correlation $\rho \geq 0$. Their mean $\overline{X}$ has variance

$$v_K = \frac{p(1-p)}{K}\big[1 + (K-1)\rho\big]. \tag{18}$$

Cantelli's one sided inequality gives

$$\Pr(\overline{X} > 1/2) \geq \frac{(p - 1/2)^2}{v_K + (p - 1/2)^2}. \tag{19}$$

The complementary event is contained in $\{\overline{X} - p \leq -(p - 1/2)\}$, so the inequality follows directly. An aggregator that returns allow whenever a strict majority allows inherits this bound. A confidence weighted rule satisfies the same bound only when its decision rule preserves the corresponding majority condition. At $\rho = 1$, all indicators are equal almost surely and the unsafe allow probability is exactly p.

Equation (19) gives a lower bound when each client is wrong more than half of the time. Pairwise correlation alone does not imply that every voting rule becomes monotonically worse as correlation increases. Proposition 2 instead captures the specific common blind spot considered by FedSafeBelief.

C.4 Proof of Ownership Preservation and Contraction

Proof of Theorem 1. Every accepted operation replaces a field in an existing record without changing its identifier. Rejected patches leave the record set unchanged. Induction over accepted operations gives

$$\mathrm{keys}(\widehat{\mathcal{B}}_i^{t+1}) = \mathrm{keys}(\widehat{\mathcal{B}}_i^{t})$$

for every client and communication round, which establishes the ownership constraint.

For contraction, index the fixed evaluation population by $j = 1, \ldots, N$ and let $E_{j,t}$ indicate that the unsafe field of record j remains erroneous after round t. Let $R_{j,t}$ indicate a correct repair of an erroneous field and let $I_{j,t}$ indicate a newly introduced unsafe error. Pointwise,

$$E_{j,t+1} \leq E_{j,t}(1 - R_{j,t}) + I_{j,t}. \tag{20}$$

A conditional repair probability of at least γ gives

$$\mathbb{E}[E_{j,t} R_{j,t}] \geq \gamma \mathbb{E}[E_{j,t}].$$

Summing over the records, dividing by N, and applying the bound on newly introduced errors yields

$$e_{t+1} \leq (1 - \gamma) e_t + \epsilon.$$

Iterating from round zero gives

$$e_t \leq (1 - \gamma)^t e_0 + \epsilon \sum_{s=0}^{t-1} (1 - \gamma)^s \tag{21}$$

$$= (1 - \gamma)^t e_0 + \frac{\epsilon}{\gamma}\big[1 - (1 - \gamma)^t\big]. \tag{22}$$

No independence among records is required. The conditional repair and expected error introduction assumptions are sufficient. □

The parameter γ covers candidate selection, scope compatibility, counterexample search, verification, and patch application. It is therefore not equivalent to the classification recall of the diagnoser. If one of these stages systematically misses a subset of errors, a uniform positive γ no longer holds. Policy changes can also introduce new unsafe beliefs, which is represented by the ϵ term.

C.5 Why Patch Level UUR Does Not Inherit This Recurrence

Suppose ten records are wrong at the start of a round. If eight fields are corrected and one wrong replacement is produced, the conditional UUR is $1/9$. In a later round, one correct repair and one wrong replacement produce a UUR of $1/2$, even though the total number of erroneous records has decreased. UUR therefore depends on both the correctness and the number of emitted patches. The residual error recurrence does not imply a monotonic decrease in UUR without additional assumptions on patch selection.

An empty patch gives a more direct example. Its UUR is undefined even if several incorrect beliefs remain untouched. Treating an empty patch as zero error would make inaction appear safe. We therefore retain the zero affected count and report abstention separately. The feedback analysis computes UUR only from rounds containing affected fields.

C.6 Scope Identifiability and the Limits of Fix Not Add

Suppose two clients use the same request and action text but differ on a hidden authorization condition. If the exchanged representation omits this condition and its policy scope, the same observed messages are compatible with two different situations. One contains a genuine contradiction, while the other contains two valid local decisions. A coordinator that only observes those messages cannot always distinguish the two cases.

This observation motivates explicit scope and version metadata. It does not assume that the chosen metadata captures every possible domain difference. When compatibility remains unknown, abstention preserves the existing record but cannot establish its correctness. Key preservation has a similar boundary because keeping the same keys does not guarantee that the replacement values are correct. Ownership preservation and semantic validity are therefore treated as separate properties.

D Additional Experiments and Diagnostics

D.1 Complete Component Ablation

Table 6: Full 2^3 ablation (%). S denotes scope gating, V denotes verified counterevidence, and F denotes fix not add.

S	V	F	SP ↑	UAR ↓	ORR ↓	Additions ↓
–	–	–	63.0±0.5	29.0±1.0	17.0±0.8	21.0±0.5
–	–	✓	79.0±0.9	24.0±0.9	9.0±0.7	0.0±0.0
–	✓	–	78.0±0.5	15.0±0.7	10.0±0.3	18.0±1.1
–	✓	✓	83.0±0.8	17.0±0.7	8.0±0.6	0.0±0.0
✓	–	–	80.0±0.7	22.0±0.9	12.0±0.8	16.0±0.4
✓	–	✓	88.0±0.9	18.0±0.6	7.0±0.8	0.0±0.0
✓	✓	–	86.0±0.6	13.0±1.0	9.0±0.7	12.0±0.7
✓	✓	✓	95.0±0.9	9.0±0.5	6.0±0.2	0.0±0.0

Table 6 reports the values corresponding to the component analysis in the main paper. Removing scope gating allows beliefs from incompatible policies to enter the correction process and reduces SP. Removing verification produces a larger UAR because proposed counterexamples can affect beliefs without an independent evidence check. Removing fix not add permits new records to be inserted and increases the addition rate. The complete configuration reaches an SP of 95.0% with a UAR of 9.0% and introduces no new records. The results show that the three components address different parts of the correction process rather than providing interchangeable restrictions.

D.2 Source Diagnosis and Legitimate Heterogeneity

Figure 6 examines both source diagnosis and policy heterogeneity. Most diagnostic errors occur between causes with similar observable records, such as temporal differences and stale policies, abstraction mismatch and semantic alignment errors, and extraction variability and insufficient evidence. This shows that detecting disagreement is easier than identifying its underlying cause. As legitimate policy differences become more common, safe completion increasingly depends on preserving client scope rather than correcting every discrepancy. The result therefore supports the design of FedSafeBelief, where diagnosis determines whether a disagreement should be repaired, retained, or left unresolved instead of treating all disagreement as error.

D.3 Verification Budget and Communication Scaling

Figure 7 shows the effect of verification budget and federation size on system cost. Increasing the counterexample budget reduces unsafe allows because more erroneous beliefs can be checked against valid counterevidence, but the improvement gradually saturates while verification time continues to increase. Communication grows with the number of participating clients because more belief summaries are uploaded and more accepted patches may be returned. Unlike verification cost, however, communication mainly depends on the amount of exchanged belief information rather

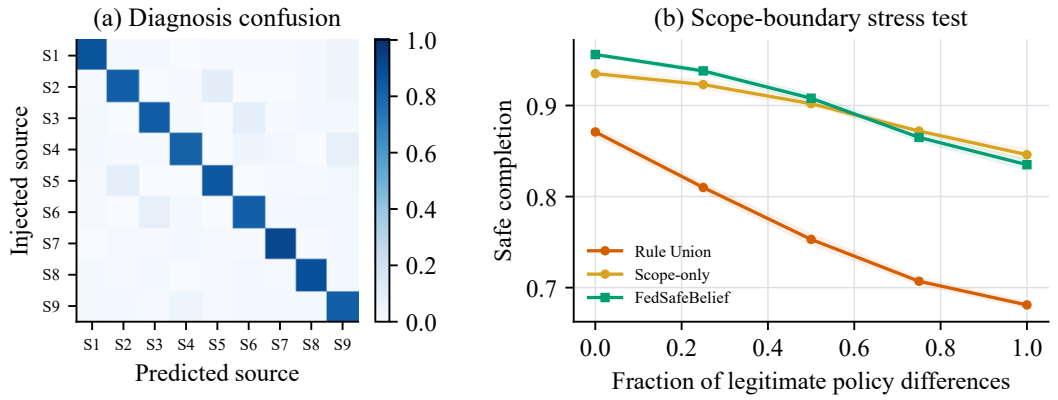

Figure 6: Source diagnosis and policy heterogeneity analysis. (a) Row normalized confusion matrix over nine error sources. (b) Safe completion as the proportion of legitimate policy differences increases.

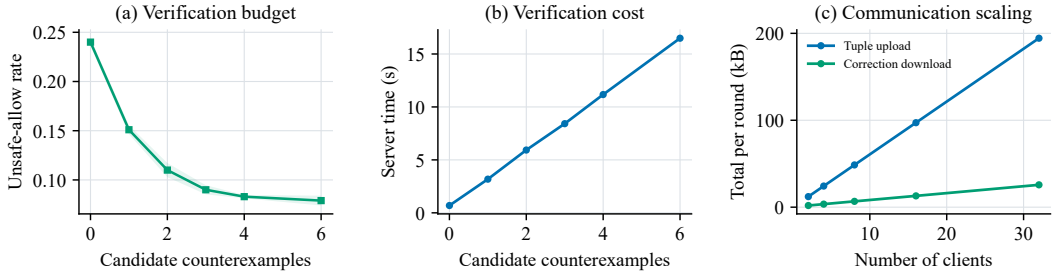

Figure 7: Verification budget and communication scaling. Increasing the counterexample budget reduces unsafe allows with diminishing returns while increasing verification cost. Communication is reported as a function of the number of participating clients.

than the difficulty of diagnosis. These results indicate that FedSafeBelief introduces a controllable verification overhead, and that a moderate counterexample budget provides most of the safety benefit without requiring excessive server cost.

D.4 Safe Completion and Refusal Tradeoffs

Table 7: Safe completion results (%).

Method	AdvBench protocol	HarmBench protocol	Policy scope protocol
Local	70.6±0.9	67.2±1.1	79.3±0.3
Rule Union	65.9±1.0	62.7±0.9	68.1±1.2
Majority	66.2±0.6	63.1±0.8	66.5±0.8
Confidence	68.3±0.8	65.0±0.6	72.9±0.5
Scope Only	73.9±0.6	70.7±0.9	**84.6±0.7**
FedSafeBelief	**80.5±0.7**	**77.2±0.6**	83.5±0.9

Table 7 shows that the reduction in unsafe decisions does not come from simply refusing more requests. FedSafeBelief reaches safe completion rates of 80.5% and 77.2% on the AdvBench and HarmBench protocols, exceeding Scope Only by 6.6 and 6.5 percentage points. This indicates that verified correction removes unsafe beliefs while retaining useful behavior. The policy scope setting shows a different boundary. Scope Only reaches 84.6%, compared with 83.5% for FedSafeBelief. When disagreements mainly reflect legitimate local policies, additional correction provides little benefit and can slightly reduce useful completion. This result supports the use of verification as a selective correction mechanism rather than a requirement to eliminate every disagreement.