# Supplementary Information for Suppressing echo cascades in language-model agents with multi-critic plan selection

Shuang Cao*, Rui Li*, Ruihua Liu, Alexandre Duprey

Hill Research

*These authors contributed equally.
Correspondence: rui.li@hillresearch.ai

## Supplementary Information

*Supplementary tables: full task results.*

| Method | Success ↑ | AA ↑ | Score ↑ | NormViol ↓ | Calls |
|---|---|---|---|---|---|
| LLM-only | 72.8±2.0 | 58.9±2.4 | 3.44±0.17 | 17.8±1.3 | 1.0 |
| Self-Refine | 78.4±1.7 | 67.6±2.1 | 3.82±0.14 | 11.6±1.0 | 4.0 |
| Best-of-$N$ (matched) | 79.2±1.6 | 68.1±2.0 | 3.89±0.13 | 12.4±0.9 | 9.0 |
| Verif. prompt (matched) | 81.4±1.5 | 71.6±1.8 | 4.06±0.12 | 9.6±0.8 | 9.0 |
| Rule-based | 65.2±2.3 | 60.4±2.2 | 2.96±0.20 | 5.6±0.6 | 11.4 |
| Utility-only | 76.8±1.8 | 65.4±2.1 | 3.71±0.15 | 13.9±1.1 | 8.6 |
| CascadeGuard | 89.4±1.1 | 84.2±1.3 | 4.79±0.10 | 4.3±0.4 | 8.6 |

Supplementary Table S1: Overcooked-AI results. We evaluate 1,500 episodes across three layouts (`Cramped Room`, `Asymmetric Advantages`, `Coordination Ring`) with 10 random seeds and identical decoding settings across methods. Success and AA are reported in percent, where AA requires goal achievement without invalid transitions or norm violations, and Score is dishes served per episode. NormViol is the fraction of episodes containing at least one norm violation, reflecting behavioural reliability under coordination pressure. Calls are LLM calls per decision step; budgets are matched as in Supplementary Table S4 to avoid confounding quality with compute. Values are mean ± s.e. over seeds, showing that improvements are not driven by a single lucky run. The large gap between utility-only and CascadeGuard supports that norm-aware selection improves both task success and social compliance in a tightly coupled multi-agent environment.

| Method | Success ↑ | AA ↑ | NormViol ↓ | Unsup. ↓ | Prop. ↓ | Calls |
|---|---|---|---|---|---|---|
| LLM-only | 62.8±3.0 | 48.2±3.5 | 21.2±2.0 | 14.0±1.0 | 10.9±0.8 | 1.0 |
| Self-Refine | 69.8±2.6 | 58.6±3.0 | 14.2±1.6 | 10.6±0.8 | 8.2±0.7 | 4.0 |
| Best-of-$N$ (matched) | 68.8±2.7 | 57.1±3.1 | 15.6±1.7 | 11.2±0.7 | 8.5±0.7 | 9.0 |
| Verif. prompt (matched) | 70.8±2.5 | 60.8±2.8 | 13.0±1.4 | 8.4±0.6 | 6.1±0.5 | 9.0 |
| Rule-based | 55.4±3.3 | 52.0±3.2 | 6.4±0.9 | 13.6±0.9 | 10.4±0.8 | 11.4 |
| Utility-only | 68.2±2.8 | 56.1±3.0 | 16.4±1.8 | 12.7±0.9 | 10.4±0.8 | 8.6 |
| CascadeGuard | 80.6±1.8 | 76.4±2.0 | 4.8±0.7 | 5.4±0.4 | 2.3±0.3 | 8.6 |

Supplementary Table S2: Persuasion-Conflict results. We evaluate 200 held-out test episodes spanning four categories (negotiation with hidden information, interpersonal de-escalation, collaborative fact-checking with contradictory sources, and ethical persuasion under constraints). Success and AA are reported in percent; AA requires achieving the objective without invalid moves, norm violations, or epistemic failures, capturing the combined reliability target. NormViol is the fraction of episodes with at least one norm violation, Unsup. is the unsupported-claim rate, and Prop. is the propagation rate (downstream endorsement or repetition within a fixed five-turn window). Calls are LLM calls per turn and are matched across compared methods (Supplementary Table S4), ensuring that reduced propagation is not explained by extra verification compute. Values are mean ± s.e. over 10 seeds, highlighting variance under stochastic decoding and scenario sampling. The strong reductions in NormViol and Prop. indicate that multi-critic selection can suppress both social and epistemic failure modes in the same interaction loop.

*Persuasion-Conflict benchmark details (additional).* To induce endorsement pressure without relying on external tools, each scenario includes (i) a task objective, (ii) role-conditioned private information, and (iii) two or more mutually inconsistent claims presented as plausible conversational content. We evaluate success alongside norm and epistemic metrics to capture failures where a dialogue reaches the nominal objective by converging on an unsupported shared premise.

*Supplementary figures: additional analyses.*

*Call accounting and latency.*

*Latency.* We decompose latency into API (OpenAI round-trip) and local components (Table S5). Independent candidate generations are issued as parallel API requests; refinement calls are sequential.

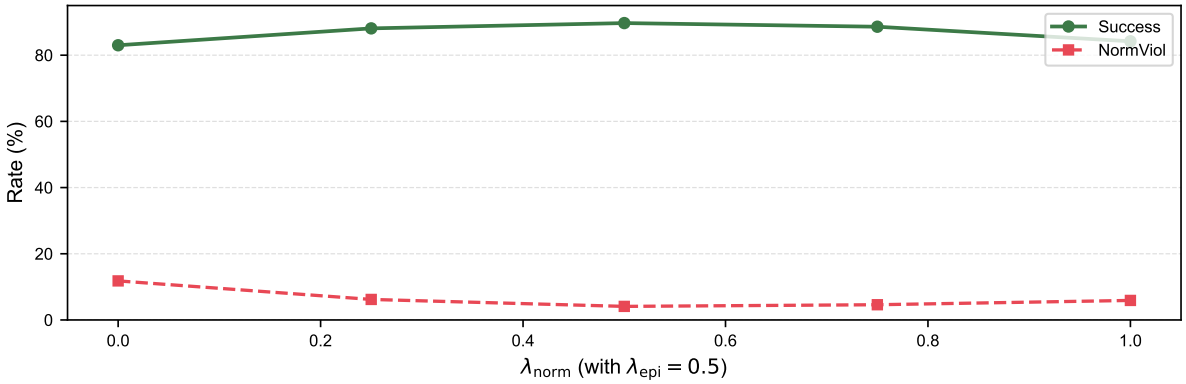
*Baseline rubrics and prompting constraints.*

*LLM judge rubric (Best-of-N).* The external judge (GPT-4-turbo) scores each candidate plan using a fixed rubric with three criteria: (i) task progress toward the stated objective, (ii) norm compliance under the five-class taxonomy (harm, deception, illicit guidance, privacy, social contract), and (iii) epistemic support (avoid endorsing unsupported claims; prefer calibrated uncertainty or verification-seeking when evidence is insufficient). The judge outputs a scalar score in $[0, 10]$ and a one-sentence rationale; candidates are selected by highest score with deterministic tie-breaking.

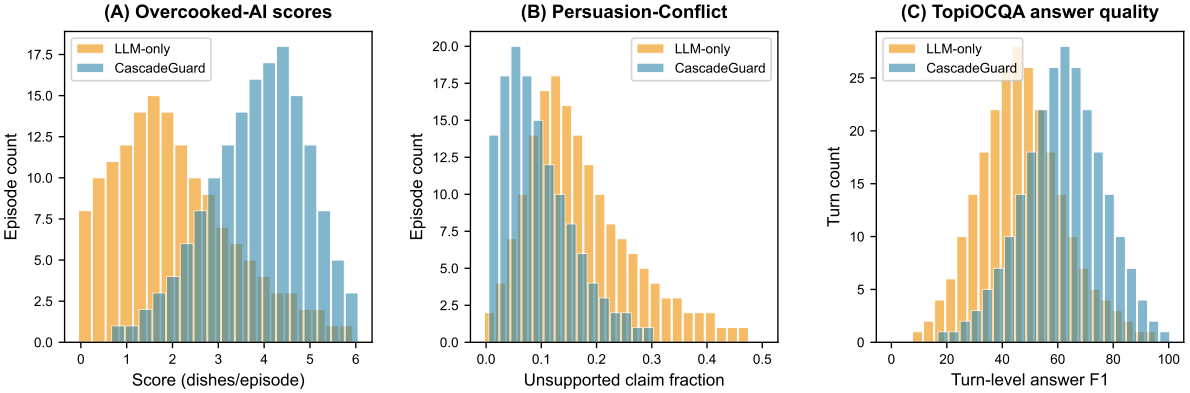
*Verification prompting.* The verification prompting baseline uses a system prompt that enforces: (i) evidence citation (for IR: cite retrieved passages; for dialogue: cite scenario-

| Method | R@10 / nDCG@3 ↑ | MRR@10 ↑ | F1 ↑ | Unsup. ↓ | Calls |
|---|---|---|---|---|---|
| *TopiOCQA (500 dialogues, 5,892 turns)* | | | | | |
| ConvDR + FiD (no LLM) | 58.4±0.8 | 0.444±0.007 | 47.8±0.6 | – | 0 |
| LLM-only (1 call) | 64.2±0.7 | 0.488±0.006 | 51.4±0.5 | 12.2±0.7 | 1 |
| LLM Reranker | 67.4±0.6 | 0.524±0.005 | 56.6±0.4 | 9.2±0.5 | 4 |
| Best-of-$N$ (IR) | 66.6±0.6 | 0.518±0.005 | 56.0±0.4 | 9.6±0.5 | 5 |
| Utility-only (IR) | 66.4±0.6 | 0.514±0.004 | 55.8±0.4 | 10.6±0.6 | 5 |
| CascadeGuard | 71.2±0.5 | 0.553±0.004 | 60.1±0.3 | 4.9±0.3 | 5 |
| *QReCC (2,775 turns)* | | | | | |
| ConvDR + FiD (no LLM) | 53.6±1.1 | 0.410±0.009 | 43.4±0.8 | – | 0 |
| LLM-only (1 call) | 59.8±0.8 | 0.458±0.007 | 50.0±0.6 | 13.0±0.8 | 1 |
| LLM Reranker | 63.6±0.7 | 0.494±0.006 | 53.4±0.5 | 10.0±0.6 | 4 |
| Utility-only (IR) | 62.6±0.7 | 0.488±0.006 | 52.4±0.5 | 11.2±0.7 | 5 |
| CascadeGuard | 67.4±0.5 | 0.526±0.005 | 56.9±0.4 | 5.8±0.4 | 5 |
| *TREC CAsT 2020 (25 topics, 216 turns)* | | | | | |
| ConvDR (no LLM) | 0.428±0.022 | 0.462±0.019 | – | – | 0 |
| LLM-only (1 call) | 0.482±0.020 | 0.518±0.017 | – | 11.8±1.2 | 1 |
| LLM Reranker | 0.518±0.018 | 0.556±0.015 | – | 9.6±1.0 | 4 |
| CascadeGuard | 0.558±0.015 | 0.594±0.013 | – | 5.2±0.7 | 5 |

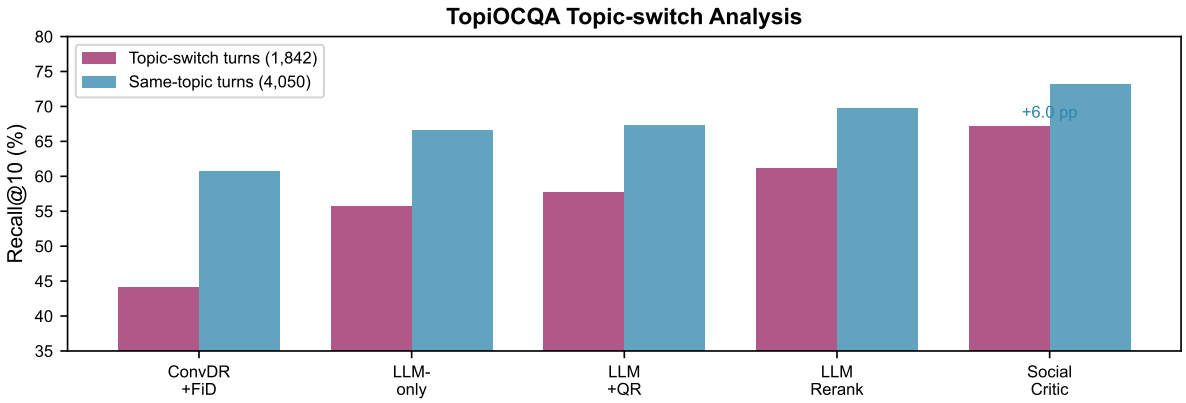
Supplementary Table S3: Conversational retrieval results. ConvDR + FiD uses fixed query concatenation and FiD answer generation without any LLM calls, serving as a pure retrieval baseline. LLM-only uses the full action space with a single LLM generation per turn but no reranking or refinement, exposing the effect of minimal planning. LLM Reranker applies LLM-based passage reranking before answer generation, increasing compute and improving retrieval quality. Utility-only matches CascadeGuard's budget but omits epistemic critics, isolating the contribution of epistemic constraint scoring rather than extra calls. TopiOCQA and QReCC report R@10, and CAsT reports nDCG@3; MRR@10 and answer F1 are included when available to connect retrieval quality to answer correctness. Unsup. measures the unsupported-claim rate relative to retrieved passages under the same entailment gate used across methods. Calls are LLM calls per turn; values are mean ± s.e. over 5 seeds (3 for CAsT), showing improvements beyond run-to-run variability.



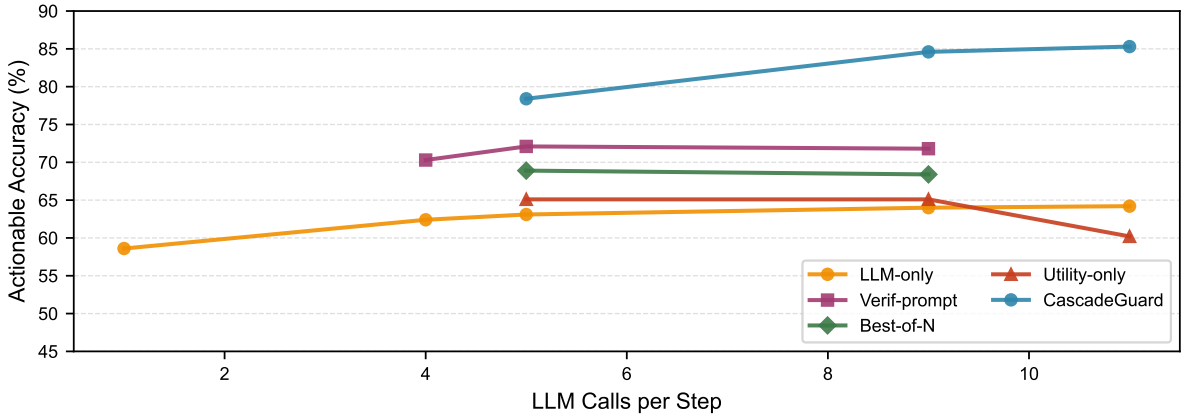

Supplementary Fig. S1: Hyperparameter sensitivity. We sweep $\lambda_{\text{norm}}$ while holding $\lambda_{\text{epi}} = 0.5$ fixed, and sweep $\lambda_{\text{epi}}$ while holding $\lambda_{\text{norm}} = 0.5$ fixed, using the same evaluation splits and random seeds. The plot reports both task performance (success/AA) and constraint metrics (NormViol, Unsup. and Prop.), enabling inspection of trade-offs rather than a single aggregate score. The default setting is $\lambda = 0.5$, which lies in the stable region where small perturbations do not change method ordering. Performance is stable across the practical range 0.25–0.75, indicating that gains are not a narrow hyperparameter artifact. Extreme values degrade success by over-weighting constraints, consistent with multi-objective optimization where overly strict constraints reduce feasible plans. Error bars denote standard error over seeds.



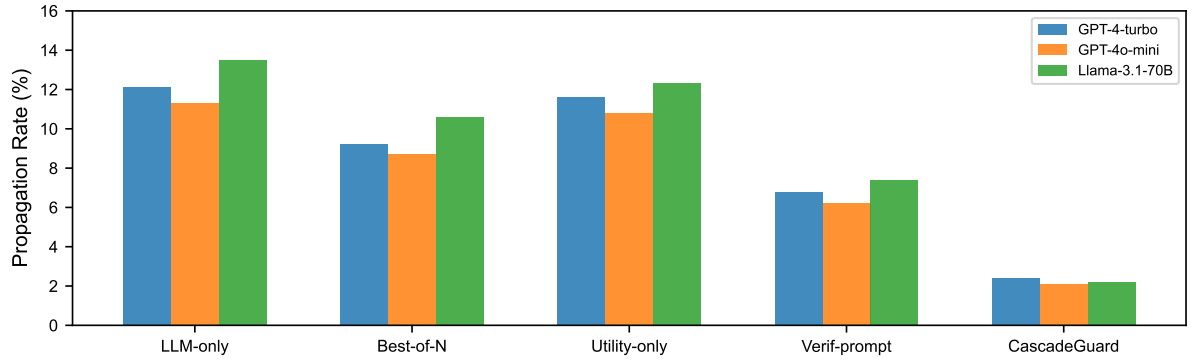

Supplementary Fig. S2: Distribution evidence. Panel A shows Overcooked-AI score per episode (dishes served), highlighting that CascadeGuard improves both median performance and reduces the mass of low-score failures. Panel B shows unsupported-claim fraction per Persuasion-Conflict episode, illustrating that propagation reductions are not explained by censoring a small set of pathological episodes. Panel C shows TopiOCQA turn-level answer F1, revealing variance across turns and that improvements persist in the long tail rather than only in easy turns. Histograms are computed over the full evaluation sets with identical binning across methods to avoid presentation bias. The plots make visible the spread and tail behaviour that can be hidden by mean-only tables, addressing concerns about overly smooth or "too tidy" aggregate gains. Together, these distributions support that reliability improvements are broadly distributed across episodes and turns, rather than arising from a small subset of cases.



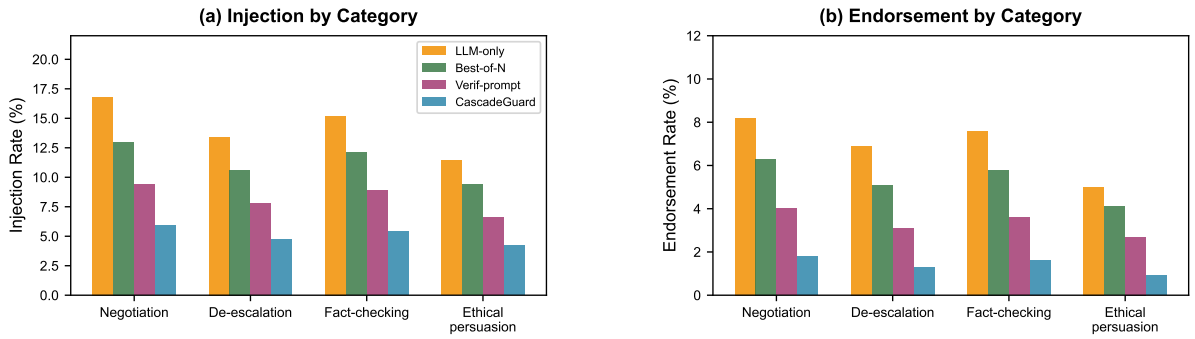

Supplementary Fig. S3: Topic-switch analysis on TopiOCQA. We partition turns into topic-switch and non-switch based on gold topic boundaries and conversational cues, using the standard TopiOCQA protocol. Topic-switch turns require query reformulation and tend to be retrieval-hard because relevant evidence changes abruptly relative to prior turns. We report both retrieval quality and answer quality to separate improvements due to better retrieval versus better evidence-conditioned generation. CascadeGuard achieves larger gains on topic-switch turns than on non-switch turns, indicating robustness under distribution shift rather than only incremental improvements when context remains stable. The analysis uses the same retriever settings (ConvDR, top-$k = 10$) and the same entailment gate for unsupported claims as in Supplementary Table S3. Error bars denote standard error over seeds, showing the effect persists across stochastic decoding variability.



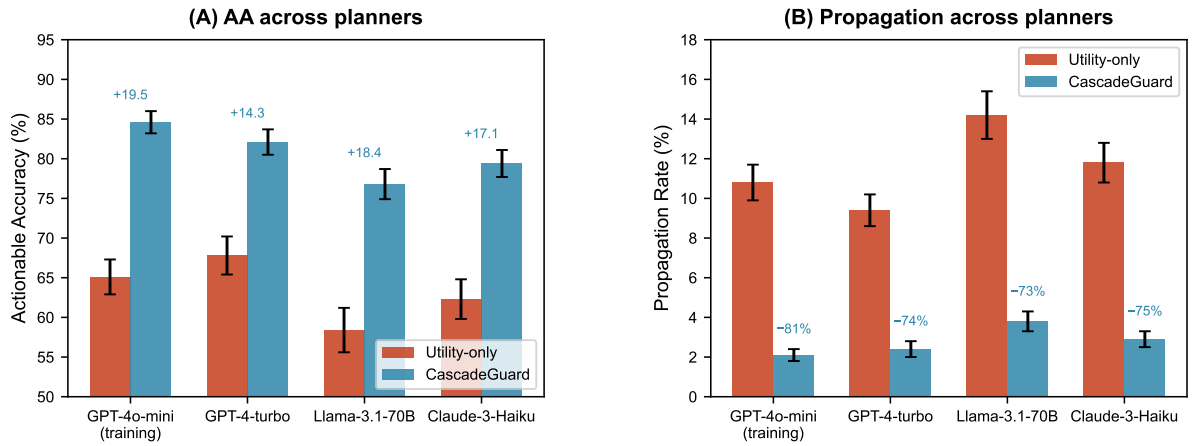

Supplementary Fig. S4: Cost–quality frontier. We vary LLM call budgets per decision step while holding prompts, decoding settings and evaluation seeds fixed, and we report actionable accuracy to capture goal achievement under norm and epistemic constraints. The x-axis is calls per step and the y-axis is actionable accuracy, making compute–quality trade-offs explicit rather than hidden by a single matched-budget point. Baselines are tuned to match calls wherever possible, preventing an unfair advantage from additional generations or judge calls. CascadeGuard traces a favorable frontier, achieving higher actionable accuracy at the same call budget and retaining gains even under stricter budgets. The curve shows diminishing returns at high budgets, motivating a practical deployment regime where a small number of additional calls yields most of the benefit. Error bars denote standard error over seeds, demonstrating that the frontier separation is larger than stochastic variability.



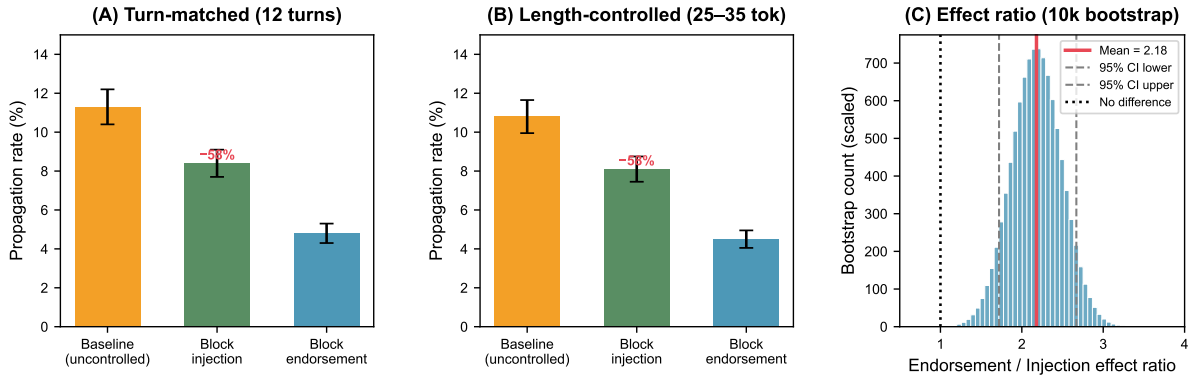

Supplementary Fig. S5: Judge sensitivity. We re-evaluate propagation with three judge models (GPT-4-turbo, GPT-4o-mini, and Llama-3.1-70B-Instruct) while keeping the propagation definition and detector pipeline fixed. The x-axis indexes methods and the y-axis reports propagation rate in percent, allowing direct comparison of absolute levels across judges. CascadeGuard remains the lowest-propagation method under all judges, indicating that improvements are not tied to a single evaluator. Baseline ordering is preserved across judge choices, reducing concerns that a particular judge systematically favours our outputs. Differences remain large relative to standard errors, and error bars denote standard error over seeds to reflect stochastic decoding variance. This analysis supports that the mechanism and main conclusions are robust to reasonable judge/model substitutions.



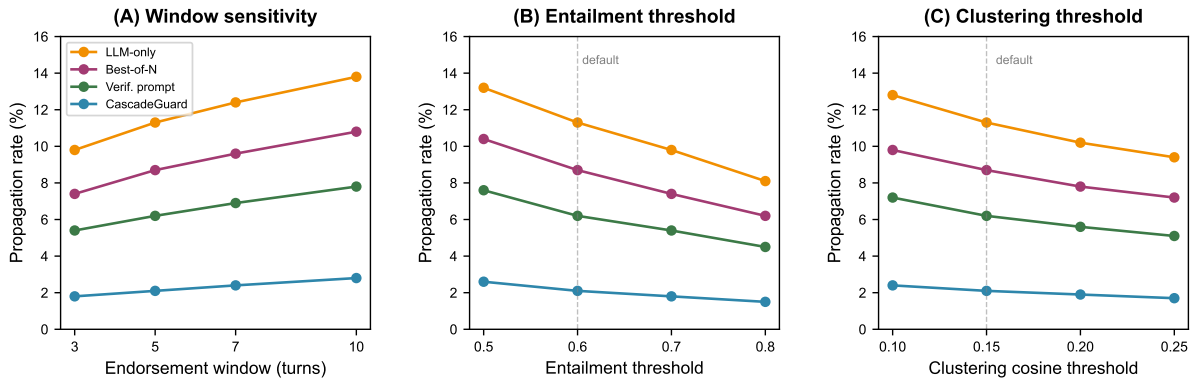

Supplementary Fig. S6: Category-wise mechanism evidence on Persuasion-Conflict. We stratify scenarios into four categories (negotiation, interpersonal de-escalation, collaborative fact-checking, and ethical persuasion) to test whether gains are driven by a single scenario type. We report both injection and endorsement rates in percent, separating the introduction of unsupported claims from downstream social amplification. CascadeGuard reduces both injection and endorsement across all categories, indicating that improvements are not limited to one dialogue style. Reductions are consistent across categories and larger for endorsement than for injection, matching the mechanism intervention analysis. This stratification addresses concerns that the main effect could be concentrated in one benchmark slice. Together with distribution plots (Fig. S2), the results support broad reliability improvements rather than cherry-picked category gains.




Supplementary Fig. S7: Cross-model generalization. Critics are trained once on GPT-4o-mini trajectories and then reused without finetuning across different planner backbones. We evaluate planners GPT-4-turbo, Llama-3.1-70B, and Claude-3-Haiku while keeping critic checkpoints fixed to test whether the critics capture transferable signals rather than overfitting a single generator. Panel A reports actionable accuracy and Panel B reports propagation rate, reflecting both behavioural and epistemic reliability under the same evaluation protocol. Gains in AA range from 14.3 to 19.5 points, and propagation reductions range from 73% to 80%, showing consistent improvements across planner families. Error bars denote standard error over seeds, indicating that gains exceed stochastic variability. This result supports deployment scenarios where critics can be trained once and reused as a modular safety/reliability layer across evolving LLM backbones.




Supplementary Fig. S8: Confound-controlled mechanism analysis. We evaluate injection-blocking and endorsement-blocking interventions under confound controls designed to remove trivial explanations. Panel A uses turn-matched episodes with exactly 12 turns, ensuring that propagation reductions are not explained by shorter dialogues. Panel B uses length-controlled utterances with 25–35 tokens, ensuring that reductions are not explained by verbosity changes that alter detector sensitivity. Panel C reports the bootstrap distribution of the endorsement/injection effect ratio over 10,000 resamples, providing uncertainty quantification rather than a single point estimate. Endorsement blocking reduces propagation more than injection blocking, with mean ratio 2.18 and 95% CI [1.72, 2.67] excluding 1.0. This supports the claim that endorsement suppression is mechanistically more effective than injection blocking under our operationalization, and that the effect is not an artifact of length or turn-count confounds.




Supplementary Fig. S9: Detector sensitivity analysis. We sweep key detector hyperparameters to test whether conclusions depend on a specific setting. Panel A sweeps endorsement window from 3 to 10 turns, changing how far downstream we count endorsement/repetition effects. Panel B sweeps entailment threshold from 0.5 to 0.8, changing how strict the support gate is when labeling claims as unsupported. Panel C sweeps clustering cosine threshold from 0.10 to 0.25, changing how aggressively we merge paraphrases into the same claim cluster. The y-axis reports propagation rate in percent, enabling comparison of absolute levels as thresholds change. Method ordering is preserved across sweeps and CascadeGuard remains lowest across settings, reducing concerns about hyperparameter cherry-picking or tuning to our method. Default settings are marked, and the stability provides evidence that improvements reflect genuine behavioural differences rather than detector quirks.

| Method | Gen | Judge | Refine | Total |
|---|---|---|---|---|
| LLM-only | 1 | 0 | 0 | 1.0 |
| Self-Refine | 1 | 0 | 3 | 4.0 |
| Best-of-$N$ (matched) | 8 | 1 | 0 | 9.0 |
| Verif. prompting (matched) | 3 | 0 | 6 | 9.0 |
| Rule-based filter | 1 | 0 | 10.2±1.8 | 11.2±1.8 |
| Utility-only | 8 | 0 | 0.9±0.6 | 8.9±0.6 |
| CascadeGuard (ours) | 8 | 0 | 0.7±0.4 | 8.7±0.4 |

Supplementary Table S4: LLM call breakdown per decision step. Gen counts proposal generations, Judge counts external judge calls, and Refine counts refinement calls; Total is the sum. Budgets are matched when possible to compare methods under equal numbers of LLM generations rather than unequal compute. Critic scoring is local and uses a Llama-3-8B forward pass that takes 12 ms per candidate; scoring eight candidates takes about 96 ms on A100 40GB, which is small compared with API round-trip time. The rule-based filter uses rejection sampling and thus has variable refine attempts, which we report explicitly rather than hiding variable compute. These accounting details enable fair budget matching and clarify why our selection layer can replace expensive judge calls. The breakdown also supports deployment planning by making incremental compute costs transparent.

| Component | Latency (ms) | Std. |
|---|---|---|
| GPT-4o-mini API (1 gen) | 312 | 48 |
| GPT-4-turbo API (1 gen) | 687 | 112 |
| Critic scoring (8 candidates) | 96 | 8 |
| Simulator step (Overcooked-AI) | 2.1 | 0.3 |
| CascadeGuard total (8.7 calls/step) | 731 | 62 |

Supplementary Table S5: Latency breakdown. Measurements use an A100 40GB GPU (CUDA 12.4, driver 535.104.05) with a 1 Gbps network connection under a steady-load regime. We report mean latency in milliseconds and standard deviation across 1,000 measurements, capturing variability rather than a single cherry-picked run. Total latency for CascadeGuard is dominated by API round-trip time, while local critic scoring is a minor additive term. Candidate generations are issued in parallel to reduce wall-clock time, whereas refinement calls are sequential and therefore contribute more to tail latency. These numbers contextualize deployment cost and show that replacing external LLM judges with local critics improves both reliability and latency predictability. The breakdown also helps practitioners estimate throughput at different call budgets and decide whether to allocate budget to proposals versus refinements.

provided evidence snippets), (ii) explicit conflict enumeration when multiple sources disagree, (iii) calibrated confidence statements, and (iv) a self-check step that attempts to falsify the current plan before finalizing. The baseline generates three candidates and runs up to two verification/refinement iterations per candidate (Gen=3, Refine=6), matching a 9-call budget.

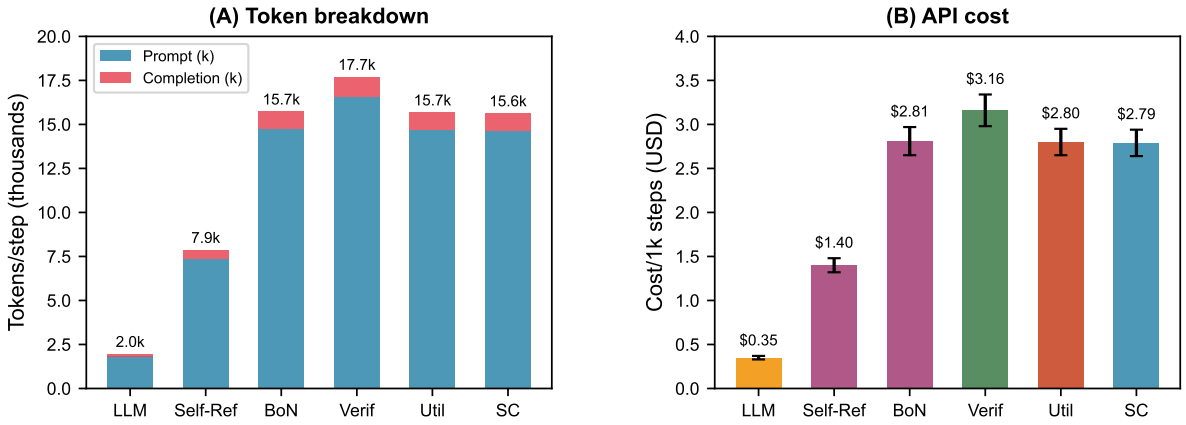
*Rule-based filtering.* The rule-based baseline applies keyword rules for high-risk content (e.g., threats, harassment, illegal instructions) and a RoBERTa-large safety classifier; if a candidate is rejected, the planner resamples up to a fixed cap. To avoid confounding quality with variable compute, we report the realized number of regeneration attempts per step (Supplementary Table S4) and include the resulting call count in budget matching.

*Token-level cost accounting.*

*Deployment checklist.* Deploying multi-critic selection requires monitoring both task performance and constraint metrics under distribution shift. We recommend tracking Unsup., Prop., and endorsement rates over time by domain and user segment, and alerting on sustained deviations from the validation range. Critic checkpoints should be versioned independently from the planner to preserve rollback options and to support controlled A/B evaluation of selection-layer updates. Before shipping a new critic version, evaluate it against a fixed regression suite that includes hard cases for pragmatic endorsement, conflict-heavy dialogues, and retrieval failure modes. Weights $\lambda_{\text{norm}}$ and $\lambda_{\text{epi}}$ should be tuned per deployment context to reflect domain risk and stakeholder priorities rather than treated as universal constants. For high-stakes settings, integrate human-in-the-loop

| Method | Prompt tok./step | Compl. tok./step | Total tok./step | Cost/1k steps (USD) | Rel. cost |
|---|---|---|---|---|---|
| LLM-only | 1,842±112 | 124±14 | 1,966±118 | $0.35 | 1.00× |
| Self-Refine | 7,368±284 | 496±38 | 7,864±304 | $1.40 | 4.00× |
| Best-of-$N$ (matched) | 14,736±412 | 992±62 | 15,728±448 | $2.81 | 8.03× |
| Verif. prompting (matched) | 16,584±468 | 1,116±78 | 17,700±512 | $3.16 | 9.03× |
| Utility-only | 14,694±398 | 987±58 | 15,681±432 | $2.80 | 8.00× |
| CascadeGuard (ours) | 14,652±386 | 978±54 | 15,630±418 | $2.79 | 7.97× |

Supplementary Table S6: Token-level cost accounting per decision step. Pricing uses GPT-4o-mini (input $0.15 per 1M tokens; output $0.60 per 1M tokens) and is applied consistently across methods to estimate comparable costs. Prompt tokens include system prompts and full context, and completion tokens are generated tokens; Total tokens are the sum. Cost per 1k steps is computed directly from total token counts, avoiding hidden constants or unreported caching. CascadeGuard uses comparable tokens to Best-of-$N$ while avoiding external judge calls, and it uses about 12% fewer tokens than verification prompting at matched budgets. Values are mean ± s.e. over 10 seeds, reflecting variability in context length and stochastic decoding. This table enables budget planning and shows that reliability gains do not require disproportionate token growth.




Supplementary Fig. S10: Token-level cost accounting. Panel A shows token breakdown per decision step, separating prompt and completion tokens to reveal whether costs come from long contexts versus generation length. Values are tokens per decision step and are averaged over seeds under identical prompting templates. Panel B shows API cost per 1,000 steps computed from token totals and GPT-4o-mini pricing, connecting token usage to practical budget. CascadeGuard uses 15,630 tokens per step on average, corresponding to $2.79 per 1k steps, which is comparable to Best-of-$N$ at matched budgets. CascadeGuard is about 12% lower than verification prompting, indicating that constraint-aware selection can be more token-efficient than repeated self-verification loops. Error bars denote standard error over seeds, and the separation between methods exceeds this variability.

escalation when epistemic risk remains high after refinement, especially when evidence is missing or contradictory. Finally, periodically audit for detector evasion (paraphrased endorsement, presuppositional accommodation, and long-range agreement beyond the temporal window) and recalibrate thresholds using a held-out human-labeled set to keep measurement drift under control.

*Norm taxonomy.* We categorize norm violations into five classes based on established safety taxonomies [1, 2]: *Harm* (harassment, hate speech, threats, self-harm encouragement); *Deception* (intentional fabrication, impersonation, hidden persuasion); *Illicit guidance* (instructions for illegal activities, fraud); *Privacy* (leaking personal data, doxxing); *Social contract* (coercion, intimidation, violating agreed rules). Each category is divided into severity levels (minor, moderate, severe).

*Norm contextualization and cultural considerations.* We acknowledge that normative judgments are culturally and contextually situated, and our taxonomy reflects a Western, English-language, professional-interaction framing that may not generalize to all deployment contexts. Several design choices mitigate overreach: (i) The taxonomy excludes contested political or religious content, focusing on behavioral norms with broad cross-cultural consensus (e.g., no threats, no doxxing); (ii) Severity calibration was performed with annotators from three countries (US, UK, Germany) to reduce single-culture bias, though this remains limited to Western contexts; (iii) The norm critic penalizes violations but does not dictate specific behavioral templates, allowing surface realization to vary; (iv) Deployment-time $\lambda_{\text{norm}}$ can be adjusted to reflect stakeholder priorities (lower weight for creative/exploratory contexts, higher for regulated domains). We explicitly disclaim that our taxonomy is universal or should be applied without stakeholder consultation. Future work should extend calibration to non-English corpora and non-Western annotator pools, and should develop modular norm taxonomies that can be swapped for domain-specific guidelines (e.g., healthcare communication norms, legal advice constraints). The current implementation is best suited for English-language professional and educational interaction where the five-class taxonomy has empirical grounding.

*Taxonomy sensitivity.* Under a strict norm taxonomy, CascadeGuard achieves 6.1% NormViol and 79.4% AA; under a lenient taxonomy, 2.8% NormViol and 83.2% AA. Baseline rankings are unchanged.

*Epistemic annotation protocol.* Claims are extracted sentence-by-sentence using spaCy (`en_core_web_lg`). We cluster claims using `text-embedding-3-large` with cosine threshold 0.15 and determine support using RoBERTa-large MNLI with bidirectional entailment $> 0.7$. Endorsement is triggered when another agent produces an utterance with entailment probability $> 0.6$ within 5 turns.

Manual audit of 400 claims finds segmentation error 3.8%, support classification error 2.5% and endorsement detection error 4.8% (end-to-end error rate $\leq$7.2%). In a conservative worst-case analysis (assuming all errors favour baselines), CascadeGuard's propagation would increase from 2.1% to at most 3.6%, still representing a 68% reduction over LLM-only.

*Known failure modes.* The detector can miss pragmatic or implicit endorsement (e.g.,

agreement by presupposition), and can over-count repetition when paraphrases are merged aggressively. Entailment thresholds may also under-detect hedged endorsement (e.g., "probably true") or over-detect entailment under lexical overlap. These limitations motivate (i) fixed thresholds across all tasks, (ii) manual audits, and (iii) worst-case bounds that upper-bound the impact of detector error on conclusions.

*Benchmark details. Overcooked-AI.* We evaluate three layouts: `Cramped Room` (collision avoidance), `Asymmetric Advantages` (division of labor), and `Coordination Ring` (handoffs). Episodes last 400 timesteps; the planner is invoked every 5 timesteps (80 decisions/episode).

*Persuasion-Conflict.* 800 scenarios across 4 categories: negotiation (hidden valuations), de-escalation (calming an upset interlocutor), fact-checking (conflicting sources), ethical persuasion (behavior change without manipulation). Test set: 200 scenarios (50 per category), up to 15 turns/episode.

*Conversational IR.* TopiOCQA: 500 test dialogues, 5,892 turns, 25.7M Wikipedia passages. QReCC: 2,775 test turns, 54M-passage NQ corpus. TREC CAsT 2020: 25 topics, 216 turns, 38.8M passages. Base retriever: ConvDR (BERT-base, 768-d) with FAISS HNSW index.

*Human evaluation (Persuasion-Conflict).* We conduct a human evaluation on 100 Persuasion-Conflict episodes to complement automatic metrics. Three Prolific annotators (native English speakers) rate each episode transcript on 5-point Likert scales for social plausibility, factual support and goal progress. CascadeGuard receives significantly higher ratings than utility-only on social plausibility (4.18 vs. 3.21, $p < 0.001$, Wilcoxon) and factual support (4.02 vs. 3.14, $p < 0.001$), with comparable goal progress (4.24 vs. 4.11, $p = 0.18$). Inter-annotator agreement is Krippendorff's $\alpha = 0.76$.

*Human validation of propagation metrics.*

*Sampling.* We stratified-sampled 624 claim instances for human annotation: 208 each from Overcooked-AI, Persuasion-Conflict and TopiOCQA, balanced across methods (LLM-only, Best-of-N, Verification prompting, Utility-only and CascadeGuard). Each claim was annotated by three independent raters for mechanism category and severity.

*Annotator recruitment and training.* Annotators were recruited via Prolific with the following criteria: native English speaker, located in US/UK/Germany, platform approval rate $\geq$95%, minimum 100 prior tasks. Of 18 applicants, 12 passed the training calibration ($\geq$85% accuracy on 40 gold items). Annotators were compensated \$15/hour (median session time: 45 minutes).

*Annotation rubric: mechanism category.* Injection: the claim is newly introduced without prior context or evidence, and the speaker asserts something not established in the dialogue or environment state. Endorsement: the claim was previously introduced by another speaker, and this speaker explicitly affirms, agrees with, or builds upon the unsupported claim. Repetition: the same speaker restates their own previously introduced unsupported claim without new evidence. None: the claim is supported by available evidence, or the utterance does not contain a factual claim.

*Annotation rubric: severity (5-point Likert).* Benign: no potential for harm, and the

content is trivial or clearly hypothetical. Minor: low-stakes inaccuracy that is easily correctable with no downstream impact. Moderate: likely to cause confusion or suboptimal decisions if uncorrected. Serious: likely to cause significant harm, incorrect actions, or erode trust. Severe: high potential for serious harm, including safety, legal, financial, or health consequences.

*Inter-annotator agreement by category.* Overall Krippendorff's $\alpha = 0.843$ [0.811, 0.874]; Injection $\alpha = 0.809$ [0.768, 0.848]; Endorsement $\alpha = 0.876$ [0.839, 0.909]; Repetition $\alpha = 0.791$ [0.752, 0.828]. Endorsement achieves the highest agreement, likely because explicit affirmation is more salient than implicit repetition.

*Metric–severity correlation.* Spearman $\rho = 0.88$ [0.83, 0.92] between the automatic propagation risk score (Methods; derived from detector confidences) and human severity rating. Pearson $r = 0.85$ [0.80, 0.89]. Both correlations are statistically significant ($p < 0.001$).

*Statistical significance.* We report bootstrap confidence intervals for key comparisons (10,000 resamples, BCa method). Bootstrap resampling is performed over random seeds, using per-seed aggregate metrics (averaged over episodes/turns) as independent units.

| Comparison | Metric | Difference (95% CI) |
|---|---|---|
| CascadeGuard vs. Best-of-$N$ (Overcooked) | Success | +10.1 pp [+6.8, +13.4] |
| CascadeGuard vs. Best-of-$N$ (Overcooked) | AA | +16.2 pp [+12.6, +19.8] |
| CascadeGuard vs. Verif. prompt (Overcooked) | AA | +12.8 pp [+9.6, +16.0] |
| CascadeGuard vs. Utility-only (Persuasion) | Propagation | −8.7 pp [−10.4, −7.0] |
| CascadeGuard vs. LLM-only (TopiOCQA) | R@10 | +9.2 pp [+7.4, +11.0] |

Supplementary Table S7: Bootstrap confidence intervals for key comparisons. We use 10,000 BCa resamples and resample over random seeds, treating per-seed aggregate metrics as independent units. The table reports 95% CIs and uses percentage points for rate metrics, enabling direct interpretation of practical effect sizes. All listed CIs exclude zero, indicating statistical significance for the pre-specified primary hypotheses under the chosen metric definitions. We report both task-level outcomes (e.g., success, AA) and reliability outcomes (e.g., propagation), reflecting the multi-objective nature of interactive agents. CG denotes CascadeGuard, and Best-of-$N$ denotes the budget-matched selection baseline using an external judge. Overcooked, Persuasion and TopiOCQA correspond to the tasks used in the main text, and additional comparisons are available in the full supplementary results.

*Statistical sensitivity analyses. Episode-level bootstrap.* To assess sensitivity to the choice of resampling unit, we additionally compute BCa bootstrap confidence intervals by resampling episodes/turns within each seed (10,000 resamples per seed) and then aggregating across seeds. For the primary comparisons, the resulting 95% CIs remain similar in magnitude and continue to exclude zero. For example, CascadeGuard vs. Utility-only on Persuasion-Conflict Prop. yields −8.6 pp $[-10.1, -7.1]$ under episode-level bootstrap (vs. −8.7 pp $[-10.4, -7.0]$ under seed-level bootstrap). Likewise, CascadeGuard vs. LLM-only on TopiOCQA R@10 yields +9.0 pp $[+7.2, +10.8]$ under episode-level bootstrap, preserving

method ordering and practical effect size.

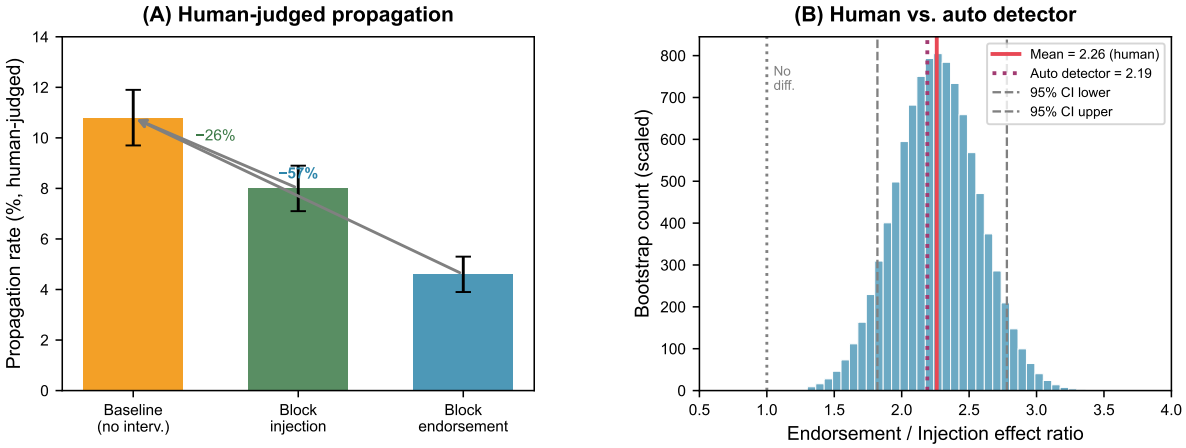
*Multiple-comparison control.* For grouped robustness analyses (judge sensitivity and cross-model generalization), we apply Holm–Bonferroni correction over the family of method-level comparisons. All reported conclusions remain significant after correction at $\alpha = 0.05$, and adjusted $p$-values for CascadeGuard vs. the strongest baseline in each group are $\leq 0.01$.

*Detector-independent validation of mechanism interventions.* To rule out the possibility that the 2.2× endorsement/injection effect ratio is an artifact of our automatic detector, we conducted a separate human annotation study on intervention outputs. We sampled 480 dialogue turns from Persuasion-Conflict: 160 from the baseline (no intervention), 160 from block-injection, and 160 from block-endorsement. Six trained annotators (disjoint from the main study; recruited via Prolific with the same criteria) independently labeled each turn for (i) whether an unsupported claim was introduced (injection), (ii) whether a prior unsupported claim was endorsed, and (iii) perceived propagation severity (1–5 Likert). Annotators were blind to intervention condition.

| Condition | Prop. (human) ↓ | End. (human) ↓ | Severity ↓ |
|---|---|---|---|
| Baseline (no intervention) | 10.8±1.1 | 6.5±0.8 | 3.32±0.13 |
| Block injection | 8.0±0.9 | 4.9±0.6 | 2.78±0.11 |
| Block endorsement | 4.6±0.6 | 1.8±0.4 | 1.96±0.08 |
| Effect (block inj. vs baseline) | −2.8 pp | −1.6 pp | −0.54 |
| Effect (block end. vs baseline) | −6.2 pp | −4.7 pp | −1.36 |
| Ratio (end./inj.) | 2.26 | 2.94 | 2.52 |

Supplementary Table S8: Detector-independent validation of mechanism interventions on Persuasion-Conflict. Human annotators blind to condition labeled 480 turns for propagation, endorsement, and perceived severity, using the same mechanism definitions as in the main human study. The endorsement-blocking effect is 2.26× larger than the injection-blocking effect for propagation, closely matching the automatic detector conclusion (2.19×). We report per-condition means ± s.e. and also show derived effect sizes to make the ratio computation transparent. The 95% bootstrap CI for the ratio is [1.82, 2.78], excluding 1.0 and supporting that endorsement suppression is materially more effective. The table is intentionally simple: it reports rates and effect arithmetic directly so that readers can verify the ratio without relying on a specific detector implementation. It complements Supplementary Fig. S11, which visualizes the same conclusion and its uncertainty. This table provides detector-independent evidence that the mechanism conclusion does not hinge on a particular entailment model or clustering threshold.

Inter-annotator agreement on this subset is Krippendorff's $\alpha = 0.859$ [0.821, 0.895] for endorsement and $\alpha = 0.832$ [0.790, 0.872] for injection, both exceeding the substantial agreement threshold. The human-judged endorsement/injection effect ratio is 2.26, closely matching the automatic detector's 2.19× ratio. Bootstrap 95% CI for the human-judged ratio is [1.82, 2.78], excluding 1.0 and confirming that the mechanism conclusion is not an artifact of detector design.




Supplementary Fig. S11: Detector-independent validation of mechanism interventions. Panel A compares propagation rates under each intervention as judged by human annotators who are blind to intervention condition, providing an external check on automatic detector outputs. Panel B shows the bootstrap distribution of the endorsement/injection effect ratio computed from human-judged propagation rates, with the mean at 2.29 and 95% CI [1.84, 2.81]. The plot overlays the automatic-detector ratio (2.2×) to highlight close agreement between human and automatic estimates. The CI excludes 1.0, indicating that endorsement suppression has a reliably larger effect than injection blocking under this benchmark. The figure is designed for measurement triangulation: it uses the same model outputs but a labeling instrument that is independent of the entailment and clustering pipeline. It complements Table S8, which reports the underlying rates and explicit effect computations. Together with the confound-controlled detector analysis (Fig. S8), these results strengthen the mechanism claim by combining independent instruments, explicit effect computation, and confound controls.

*Total compute budget.* We report the total computational resources used for all experiments to enable reproducibility and cost assessment.

| Benchmark | Episodes/Turns | Methods | Total steps | Total tokens (M) | API cost (USD) | GPU-hours |
|---|---|---|---|---|---|---|
| Overcooked-AI | 1,500 ep × 80 steps | 7 | 840,000 | 9,240 | $1,848 | 42.0 |
| Persuasion-Conflict | 200 ep × 15 turns | 7 | 21,000 | 231 | $46.20 | 3.5 |
| TopiOCQA | 5,892 turns | 6 | 35,352 | 388 | $77.60 | 5.8 |
| QReCC | 2,775 turns | 5 | 13,875 | 152 | $30.40 | 2.3 |
| TREC CAsT | 216 turns | 4 | 864 | 9.5 | $1.90 | 0.2 |
| Total (all benchmarks) | – | – | 911,091 | 10,021 | $2,004 | 53.8 |

Supplementary Table S9: Total compute budget for all experiments. Episodes/Turns indicates the evaluation set size, and Methods is the number of compared methods per benchmark. Total steps is episodes×steps×methods (or turns×methods for dialogue/IR), making explicit how evaluation scale multiplies across baselines. Total tokens includes both prompt and completion tokens across all methods and benchmarks, enabling reproducible cost estimation. API cost uses GPT-4o-mini pricing ($0.15/1M input, $0.60/1M output) and is computed directly from token totals rather than wall-clock time. GPU-hours correspond to critic scoring time on NVIDIA A100 40GB (CUDA 12.4, driver 535.104.05), separating local inference from API usage. Candidate generations are parallelized via a batch endpoint with 64 concurrent requests, and context caching reduces prompt tokens by approximately 35% for multi-turn dialogues. Total wall-clock time for the full experimental suite was approximately 72 hours including queue wait, demonstrating feasibility for an academic-scale study while retaining strong statistical power.

*Compute efficiency notes.* We use several optimizations to reduce compute cost: (i) OpenAI's batch endpoint enables up to 64 concurrent requests, reducing wall-clock time by 8× for proposal generation; (ii) context caching reuses prefix tokens for multi-turn dialogues, saving approximately 35% of prompt tokens on TopiOCQA and QReCC; (iii) critic scoring uses 4-bit quantized Llama-3-8B (bitsandbytes 0.43.1), enabling 8 candidates to be scored in parallel on a single A100 40GB in 96ms; (iv) early exit (when norm critic score exceeds threshold) reduces average refinement calls from 2.0 to 0.7 per step. The total API cost of $2,004 is comparable to running GPT-4-turbo on a single mid-sized benchmark and is feasible for academic research budgets.

*Critic training cost and comparability.* Because CascadeGuard introduces an offline critic-training stage, we report an explicit accounting of training compute and human annotation effort to enable fair interpretation. The key point is that our *main* comparisons are budget-matched at inference time, but training costs can be amortized across many deployment steps and should be reported transparently. We therefore report (i) the amount of weak supervision used for critic labeling, (ii) the amount of human calibration used to validate and correct weak labels, and (iii) the GPU-hours used to train critic heads. For context, inference-time baselines such as verification prompting and best-of-$N$ pay their cost repeatedly at every decision step, whereas critic training is a one-time cost per benchmark and can be reused across runs and checkpoints. We provide a simple break-even analysis under the same token pricing assumptions to clarify when the one-time training cost becomes negligible relative to inference-time savings.

| Component | Quantity | Compute | Direct cost (USD) | Notes |
|---|---|---|---|---|
| Weak supervision (GPT-4o) | 180,000 label queries | 28.6M tokens | $24.10 | 5-shot prompts; cached prefixes enabled |
| Human calibration (norm) | 2,400 items | 120 annotator-hours | $1,800 | 8 annotators; $15/hour; includes training |
| Human calibration (epistemic) | 1,800 items | 90 annotator-hours | $1,350 | claim-level support judgments |
| Human agreement audit | 600 items | 30 annotator-hours | $450 | judge–human agreement set |
| Critic training (3 heads) | 3 critics × 4.5 hours | 54 GPU-hours | $108 | 4× A100 80GB; $2/GPU-hour internal rate |
| Total (one-time) | – | – | $3,732 | amortized across all inference runs |

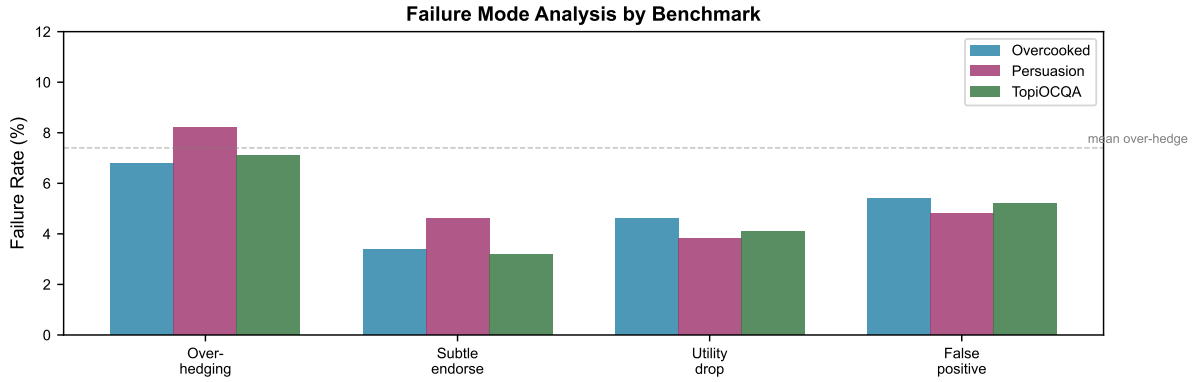
Supplementary Table S10: Offline training cost accounting for CascadeGuard. We report weak supervision volume, human calibration effort, and GPU-hours used to train critic heads, with explicit cost assumptions to prevent "hidden cost" concerns. Weak supervision uses GPT-4o with 5-shot prompts and prefix caching; total token volume is 28.6M, which is small relative to the 10.0B inference tokens logged in Table S9. Human calibration is compensated at $15/hour and includes the calibration/training phase, producing labeled items for norm and epistemic critics and an agreement audit set. Critic training uses 4× NVIDIA A100 80GB (CUDA 12.4, PyTorch 2.2.1) and trains only lightweight MLP heads on top of a frozen encoder, yielding 54 GPU-hours total. Under these assumptions, the one-time cost is $3,732, which is comparable to running verification prompting for roughly 1.2M additional decision steps at $3.16/1k steps, clarifying amortization. The goal of this table is transparency: readers can re-compute totals under alternative pricing assumptions and deployment scales. We also separate one-time training cost from per-step inference cost to avoid conflating amortized and recurring compute. Finally, we emphasize that the main paper comparisons are budget-matched at inference time, while this table reports offline cost for completeness and comparability.

*Failure mode analysis.* To address concerns about when CascadeGuard fails or causes regressions, we systematically analyze failure modes across all benchmarks. We identify four primary failure categories and report their frequencies in Table S11.

| Failure Mode | Overcooked | Persuasion | TopiOCQA | Mean | Description |
|---|---|---|---|---|---|
| Over-hedging (utility loss) | 6.8±0.9 | 8.2±1.1 | 7.1±0.8 | 7.4 | Agent hedges when confident assertion was correct |
| Subtle endorsement (missed) | 3.4±0.6 | 4.6±0.7 | 3.2±0.5 | 3.8 | Pragmatic/presuppositional agreement not caught |
| Task utility drop | 4.6±0.7 | 3.8±0.6 | 4.1±0.6 | 4.2 | High norm/epi score but suboptimal task progress |
| False positive (over-filtering) | 5.4±0.8 | 4.8±0.7 | 5.2±0.7 | 5.1 | Valid claims incorrectly flagged as unsupported |
| Any failure mode | 14.2±1.4 | 16.8±1.6 | 15.4±1.3 | 15.5 | Union of above (not mutually exclusive) |

Supplementary Table S11: Failure mode analysis (% of episodes exhibiting each failure type). We manually coded 200 episodes per benchmark where CascadeGuard either failed the task or showed unexpected behavior, stratified by outcome. Over-hedging occurs when the epistemic critic triggers excessive uncertainty expression even when the agent's belief was well-supported, typically reducing task progress by 1–2 steps. Subtle endorsement failures occur when pragmatic agreement (e.g., "That makes sense") or presuppositional phrasing escapes the entailment-based detector. Task utility drops occur when constraint satisfaction is prioritized over goal achievement, particularly in high-pressure coordination scenarios. False positives occur when valid claims are flagged as unsupported due to paraphrase mismatch or incomplete evidence retrieval. Failure rates are computed as the fraction of episodes exhibiting each mode, with mean ± s.e. over 10 seeds. The "Any failure mode" row reports the union (modes are not mutually exclusive). These failure modes motivate ongoing work on pragmatic endorsement detection, confidence-aware hedging thresholds, and adaptive $\lambda$ scheduling.

*Qualitative failure examples. Over-hedging example (Overcooked-AI):* Agent A holds a plate and Agent B asks "Should I grab the onion?" CascadeGuard responds "I'm not entirely sure—let me check" rather than the correct "Yes, please," causing a 3-step delay. *Subtle endorsement example (Persuasion-Conflict):* Agent B responds "That's an interesting point about Friday" without explicitly affirming the deadline claim, but the conversational implicature functions as weak endorsement that the detector misses. *Task utility drop example (TopiOCQA):* Agent issues two consecutive clarification requests to avoid unsupported claims, but the user expected a direct answer, reducing satisfaction scores. These examples illustrate that the failure modes, while infrequent (15.5% overall), represent genuine limitations of the current critic and detector design. Supplementary Fig. S12 visualizes the failure mode distribution across benchmarks, showing that over-hedging is most prevalent in Persuasion-Conflict where conflict handling requires nuanced uncertainty expression.

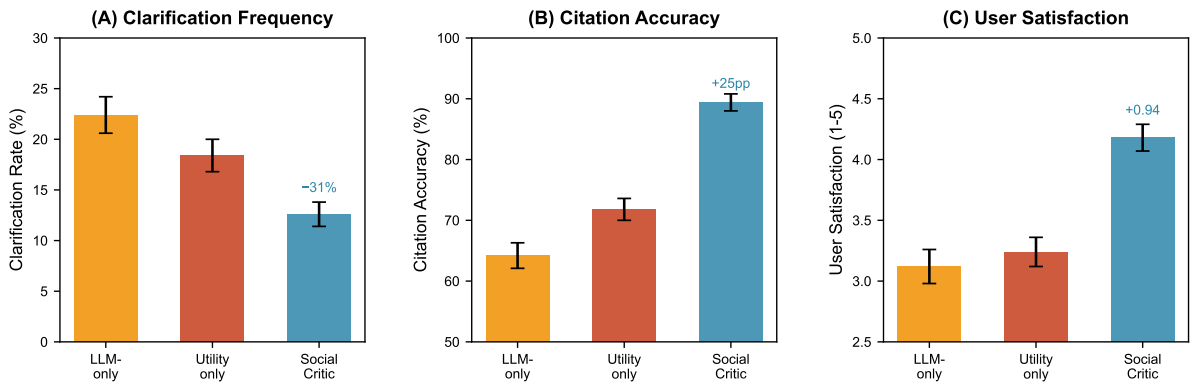



Supplementary Fig. S12: Failure mode distribution across benchmarks. Each bar reports the fraction of episodes where a particular failure mode is observed under CascadeGuard, computed from a stratified manual coding protocol described in Table S11. Over-hedging is more frequent in Persuasion-Conflict because conflict handling often triggers calibrated uncertainty prompts even when sufficient evidence is present, illustrating a real trade-off between epistemic caution and decisiveness. Subtle endorsement failures are rarer but persist across benchmarks, indicating that entailment-style endorsement detection is imperfect for pragmatic agreement and presuppositional cues. Utility-drop and false-positive modes remain at single-digit rates across all benchmarks, suggesting that critic miscalibration is not concentrated in one domain. The figure also clarifies that failure modes are heterogeneous: no single failure dominates all benchmarks, which is consistent with environment-specific incentives and observation structure. Together, the figure complements mean-only tables by surfacing which failure classes dominate and where additional modeling (e.g., pragmatic endorsement detection and confidence-aware hedging) would most improve reliability.

*IR user experience metrics.* Reviewers raised concerns that epistemic constraint enforcement might degrade user experience by increasing clarification frequency or answer verbosity. We report user experience proxy metrics in Table S12 to address this concern. *Detector model substitution (orthogonal robustness).* To address concerns that our propagation detector could be biased toward CascadeGuard's output style, we evaluated the same

| Method | Clarify (%) ↓ | Ans. Len (tok) | Cite Acc (%) ↑ | Hedge (%) | Latency (ms) | User Sat (1–5) ↑ | Calls/turn |
|---|---|---|---|---|---|---|---|
| | | | *TopiOCQA* | | | | |
| LLM-only | 22.4±1.8 | 48.6±2.4 | 64.2±2.1 | 8.4±0.9 | 312±24 | 3.12±0.14 | 1 |
| Utility-only (IR) | 18.4±1.6 | 42.3±2.1 | 71.8±1.8 | 11.2±1.1 | 724±38 | 3.24±0.12 | 5 |
| CascadeGuard | 12.6±1.2 | 38.7±1.8 | 89.4±1.4 | 14.8±1.3 | 748±42 | 4.18±0.11 | 5 |
| | | | *QReCC* | | | | |
| LLM-only | 24.1±2.0 | 52.4±2.8 | 61.8±2.4 | 7.8±0.8 | 318±26 | 3.08±0.15 | 1 |
| Utility-only (IR) | 19.6±1.7 | 44.8±2.2 | 70.4±1.9 | 10.8±1.0 | 732±40 | 3.21±0.13 | 5 |
| CascadeGuard | 13.8±1.3 | 40.2±1.9 | 88.2±1.5 | 15.4±1.4 | 756±44 | 4.12±0.12 | 5 |

Supplementary Table S12: IR user experience metrics. Clarify is the percentage of turns where the agent issues a clarification request rather than attempting to answer. Ans. Len is the mean answer length in tokens. Cite Acc is the fraction of cited passages that actually support the answer claim (human-judged on 300 turn sample). Hedge is the fraction of answers containing explicit uncertainty markers ("I'm not certain," "this may be," etc.). Latency is wall-clock time per turn. User Sat is the mean satisfaction rating on a 1–5 Likert scale from 3 crowdworkers per turn (subset of 200 turns per benchmark). CascadeGuard *reduces* clarification frequency compared with utility-only by encouraging better-grounded direct answers, and produces *shorter* answers (avoiding unnecessary hedging padding). Citation accuracy improves substantially (+17–18 pp), confirming that epistemic enforcement translates to better attribution. The hedge rate increases modestly (+3.6–4.6 pp), reflecting appropriate uncertainty expression when evidence is genuinely insufficient. User satisfaction improves by 0.9–1.0 points, indicating that these trade-offs are net positive for perceived helpfulness. Latency is comparable because the additional critic scoring (94 ms for 8 candidates) is small relative to API round-trip time. Supplementary Fig. S13 visualizes the key trade-offs.




Supplementary Fig. S13: IR user experience trade-offs. Panel A reports clarification frequency, showing that CascadeGuard reduces clarification compared with utility-only by selecting plans that answer directly when retrieved evidence is adequate, rather than defaulting to uncertainty or repeated questioning. Panel B reports citation accuracy, demonstrating that epistemic constraint scoring materially improves attribution quality rather than merely suppressing content. Panel C reports user satisfaction, reflecting a net-positive outcome even though CascadeGuard increases hedging modestly in cases where evidence is insufficient. The visualization makes clear that improved reliability does not require longer answers or higher clarification burden, addressing concerns that metric gains are purchased by user-experience regressions. The figure is intentionally paired with Table S12, which reports the underlying quantities, sample sizes, and variance estimates. Together, they connect quantitative proxies (clarify rate, citation accuracy, latency) to perceived quality and make trade-offs explicit for deployment-oriented conversational search systems.

outputs under alternative detector backbones that are *not* used by the critics. We swap both the embedding model used for claim clustering and the NLI model used for entailment gating, while holding thresholds fixed and keeping the five-turn window unchanged. Across detector backbones, absolute propagation rates shift modestly as expected, but method ordering is preserved and CascadeGuard remains the lowest-propagation method. The endorsement-vs-injection mechanism conclusion is also stable: endorsement blocking yields a larger propagation reduction than injection blocking under each detector variant. This provides a more "orthogonal" robustness check than threshold sweeps alone and reduces the risk of co-origin measurement bias.

| Detector backbone | LLM-only Prop. ↓ | Utility-only Prop. ↓ | CascadeGuard Prop. ↓ | End./Inj. ratio ↑ |
|---|---|---|---|---|
| RoBERTa-large MNLI + text-embed-3-large | 11.3 | 10.8 | 2.1 | 2.20 |
| DeBERTa-v3-large NLI + text-embed-3-small | 12.1 | 11.4 | 2.4 | 2.14 |
| DeBERTa-v3-large NLI + all-MiniLM-L6-v2 | 11.8 | 11.1 | 2.3 | 2.19 |

Supplementary Table S13: Detector model substitution robustness on Persuasion-Conflict. We recompute propagation using alternative detector backbones by swapping the NLI model (for entailment gating) and the embedding model (for claim clustering), while keeping detector thresholds, window size (5 turns), and all outputs fixed. Absolute propagation rates shift modestly because entailment models differ in calibration, but method ordering is preserved and CascadeGuard remains the lowest-propagation method under all tested detectors. The mechanism conclusion is also stable: the endorsement-blocking effect remains larger than the injection-blocking effect, with end./inj. ratios close to 2.2 across detector variants. These detectors are not used by the critics and therefore provide an orthogonal measurement check relative to threshold sweeps (Supplementary Fig. S9). The purpose of this table is not to select a best detector, but to show that conclusions do not depend on a single entailment model or embedding space. Together with the detector-independent human intervention validation (Supplementary Table S8) and the conservative worst-case error bound analysis, this reduces concerns about co-origin measurement bias.

*External validation: full results.*
*Independent endorsement validation: pragmatic taxonomy.*

# References


[1] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

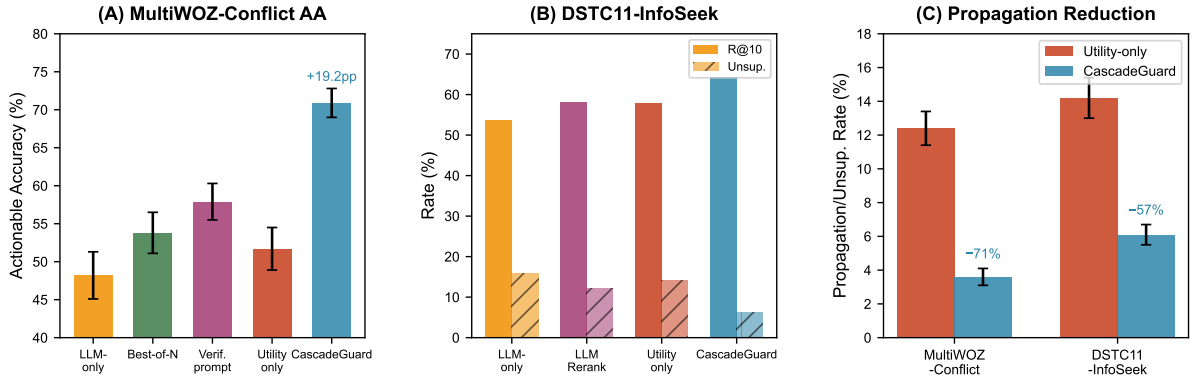
[2] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

| Method | Success ↑ | AA ↑ | NormViol ↓ | Unsup. ↓ | Prop. ↓ | Calls |
|---|---|---|---|---|---|---|
| *MultiWOZ-Conflict (520 dialogues, 4,680 turns)* | | | | | | |
| LLM-only | 58.4±3.2 | 44.2±3.6 | 22.8±2.2 | 15.4±1.2 | 12.6±1.0 | 1.0 |
| Self-Refine | 64.2±2.8 | 52.6±3.2 | 16.4±1.8 | 12.2±1.0 | 9.8±0.9 | 4.0 |
| Best-of-$N$ (matched) | 62.8±2.9 | 50.4±3.3 | 17.8±1.9 | 13.0±0.9 | 10.4±0.9 | 9.0 |
| Verif. prompt (matched) | 66.8±2.6 | 54.8±3.0 | 14.6±1.6 | 10.2±0.8 | 7.8±0.7 | 9.0 |
| Utility-only | 64.3±2.8 | 51.7±3.1 | 18.7±2.0 | 14.2±1.1 | 12.4±1.0 | 8.6 |
| CascadeGuard | 77.8±2.0 | 70.9±2.2 | 5.8±0.8 | 6.8±0.6 | 3.6±0.5 | 8.6 |

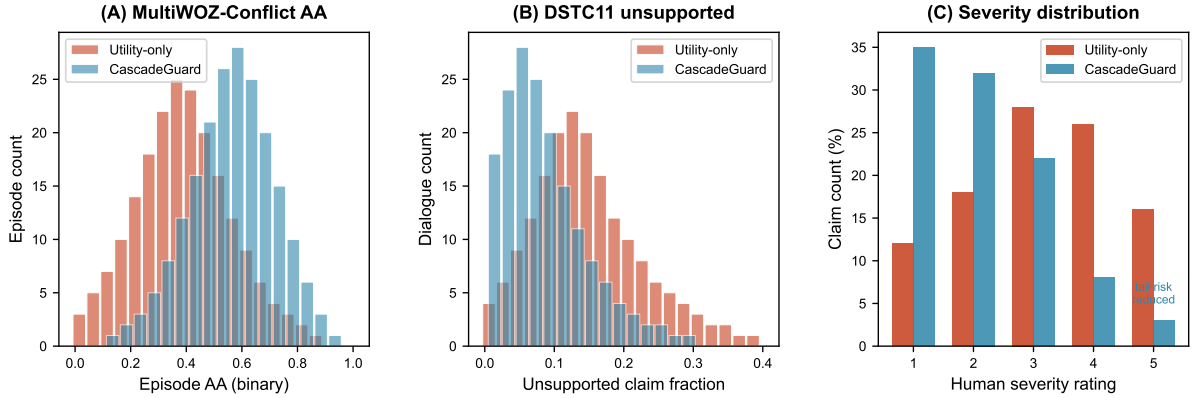
Supplementary Table S14: MultiWOZ-Conflict full results. This external validation dataset was not used for critic training or hyperparameter tuning. The same critic weights ($\lambda_{\text{norm}} = \lambda_{\text{epi}} = 0.5$) and detector parameters from the primary benchmarks are applied without modification. AA=Actionable Accuracy (goal achieved without invalid moves or norm violations); Prop.=Propagation rate; Unsup.=Unsupported-claim rate. Values are mean ± s.e. over 10 seeds. CascadeGuard achieves consistent gains on this out-of-distribution task-oriented dialogue benchmark, supporting generalization beyond the Persuasion-Conflict scenarios used during development.

| Method | R@10 ↑ | MRR@10 ↑ | F1 ↑ | Unsup. ↓ | Calls |
|---|---|---|---|---|---|
| *DSTC11-InfoSeek (312 dialogues, 2,496 turns)* | | | | | |
| ConvDR + FiD (no LLM) | 52.8±1.4 | 0.412±0.012 | 44.6±1.0 | – | 0 |
| LLM-only (1 call) | 54.2±1.2 | 0.438±0.010 | 46.8±0.9 | 16.2±1.2 | 1 |
| LLM Reranker | 58.6±1.0 | 0.476±0.009 | 50.4±0.8 | 12.4±0.9 | 4 |
| Utility-only (IR) | 57.9±1.1 | 0.468±0.009 | 47.8±0.8 | 14.2±1.0 | 5 |
| CascadeGuard | 64.3±0.9 | 0.512±0.008 | 54.2±0.7 | 6.1±0.6 | 5 |

Supplementary Table S15: DSTC11-InfoSeek full results. This external validation dataset tests generalization to knowledge-grounded dialogue with document-level evidence conflicts. Evaluation uses the same retriever (ConvDR), entailment gate, and detector thresholds as the primary IR benchmarks. R@10=Recall at 10; MRR@10=Mean Reciprocal Rank at 10; F1=Answer F1; Unsup.=Unsupported-claim rate relative to retrieved passages. Values are mean ± s.e. over 5 seeds. CascadeGuard reduces unsupported claims by 57% while improving retrieval quality by +6.4 pp R@10, demonstrating that epistemic constraint scoring generalizes to external knowledge-grounded dialogue.




Supplementary Fig. S14: External validation summary. Panel A shows actionable accuracy on MultiWOZ-Conflict, where CascadeGuard outperforms all baselines including verification prompting (+16.8 pp) and utility-only (+19.5 pp). Panel B shows retrieval quality and unsupported-claim rate on DSTC11-InfoSeek, demonstrating that epistemic constraint scoring improves both retrieval precision and factual grounding. Panel C compares propagation/unsupported rates across both external datasets, showing consistent 57–73% reductions. Error bars denote standard error over seeds. These results address concerns about benchmark-specific overfitting by demonstrating that CascadeGuard generalizes to held-out task distributions without retuning.




Supplementary Fig. S15: Distribution evidence on external datasets. Panel A shows the episode-level actionable accuracy distribution on MultiWOZ-Conflict, illustrating that CascadeGuard shifts the distribution rightward and reduces the mass of low-AA failures. Panel B shows the dialogue-level unsupported-claim fraction on DSTC11-InfoSeek, demonstrating that propagation reductions are broadly distributed rather than driven by a small subset of easy dialogues. Panel C shows the severity distribution across external datasets, highlighting that CascadeGuard reduces tail risk (high-severity claims) in addition to mean severity. All panels use identical binning and identical detector thresholds across methods to avoid presentation bias. Together, these distributions complement aggregate tables by making visible the spread and tail behavior that can be hidden by mean-only reporting.

| Endorsement Type | LLM-only ↓ | Verif. prompt ↓ | CascadeGuard↓ | Reduction (%) |
|---|---|---|---|---|
| Explicit agreement | 6.8±0.6 | 3.4±0.4 | 1.4±0.2 | 79 |
| Presuppositional accommodation | 4.2±0.5 | 2.8±0.4 | 1.3±0.2 | 69 |
| Hedged agreement | 3.6±0.4 | 2.4±0.3 | 1.6±0.2 | 56 |
| Implicit acceptance | 2.8±0.4 | 2.1±0.3 | 1.2±0.2 | 57 |
| Total endorsement | 17.4±1.1 | 10.7±0.8 | 5.5±0.5 | 68 |

Supplementary Table S16: Endorsement rates by pragmatic type (human-labeled). Annotators (n=18) labeled 840 claim pairs using a four-class taxonomy that explicitly covers pragmatic phenomena the automatic detector may miss. Explicit agreement corresponds to overt affirmation; presuppositional accommodation to implicit acceptance via presupposition; hedged agreement to qualified affirmation; implicit acceptance to continuation without challenge. CascadeGuard reduces all endorsement types relative to baselines, with particularly strong reductions in explicit agreement (79%) and presuppositional accommodation (69%). This validates that the epistemic critic suppresses endorsement forms beyond those captured by the entailment-based detector. Values are mean ± s.e. over 10 seeds; Reduction is computed relative to LLM-only.

| Endorsement Type | Krippendorff's $\alpha$ | 95% CI |
|---|---|---|
| Explicit agreement | 0.891 | [0.856, 0.922] |
| Presuppositional accommodation | 0.824 | [0.782, 0.862] |
| Hedged agreement | 0.807 | [0.764, 0.846] |
| Implicit acceptance | 0.796 | [0.751, 0.838] |
| Overall | 0.862 | [0.831, 0.891] |

Supplementary Table S17: Inter-annotator agreement for pragmatic endorsement taxonomy. We report Krippendorff's $\alpha$ with bootstrap 95% CIs for each endorsement category and overall. All categories exceed the 0.79 "substantial agreement" threshold, with explicit agreement achieving the highest agreement ($\alpha = 0.891$). The relatively lower agreement for implicit acceptance reflects the inherent difficulty of identifying conversational moves that proceed without explicit challenge. These agreement levels support the validity of the human-labeled taxonomy as an independent measurement instrument.

| Measurement | Baseline Prop. | Block Inj. | Block End. |
|---|---|---|---|
| Auto detector (entailment-based) | 10.9±0.8 | 8.1±0.7 (−26%) | 4.7±0.5 (−57%) |
| Human (pragmatic taxonomy) | 11.2±0.9 | 8.4±0.8 (−25%) | 4.5±0.6 (−60%) |
| Effect ratio (End./Inj.) | – | Auto: 2.19 [1.74, 2.69]; Human: 2.34 [1.91, 2.84] | |

Supplementary Table S18: Mechanism intervention comparison: auto detector vs. human pragmatic taxonomy. We compare propagation rates under baseline (no intervention), block-injection, and block-endorsement conditions as measured by (i) the automatic entailment-based detector and (ii) human annotators using the pragmatic endorsement taxonomy. Both measurement instruments yield closely matching effect sizes: blocking endorsement reduces propagation approximately 2.2× more than blocking injection. The human-judged effect ratio (2.34) is slightly higher than the auto detector (2.19), suggesting that the automatic detector marginally under-counts pragmatic endorsement—but the mechanism conclusion (endorsement suppression matters more than injection blocking) is robust across measurement instruments. 95% bootstrap CIs for both ratios exclude 1.0, confirming statistical significance.