On Recursive Resolution: Scalable Ground Truth for Self-Improving AI Systems

Disha
Amazon
USA
disha@amazon.com

Jason Whang
Amazon
USA
jwhang@amazon.com

Manikandarajan Ramanathan
Amazon
USA
mramnat@amazon.com

ABSTRACT

Compound retrieval-augmented question-answering (QA) systems present a fundamental evaluation challenge: manual annotation does not scale, yet automated evaluation lacks the ground truth necessary for calibration. We introduce a self-improving evaluation architecture that addresses this circular dependency through three contributions. First, *iterative consensus synthesis*: an algorithm that treats LLM-human agreement as a free signal, escalating only disagreements to additional annotators and achieving multi-annotator reliability at 15.95× lower cost. Second, a *tiered Wilson interval framework* that transforms noisy binary judgments into confidence-bounded evaluation results, correctly identifying regressions that raw failure rates obscure. Third, *recursive resolution*: an architecture where evaluation thresholds update based on remedy effectiveness, enabling systems to improve their own detection criteria over time. We validate on a compound retrieval-augmented QA system across 13 quality metrics and three locale-specific test suites. The system completed full improvement cycles detecting a 44% content coverage gap during retrieval, validating a targeted fix through shadow evaluation, and generalizing the pattern to prevent similar regressions on other locale-specific test sets. These results demonstrate that scalable, self-improving evaluation of compound AI systems is achievable without sacrificing annotation quality.

CCS CONCEPTS

• **General and reference** → **Evaluation**.

KEYWORDS

LLM evaluation, compound AI systems, self-improving systems, annotation efficiency

ACM Reference Format:
Disha, Jason Whang, and Manikandarajan Ramanathan. 2026. On Recursive Resolution: Scalable Ground Truth for Self-Improving AI Systems. In *Proceedings of August 9–13, 2026 (KDD '26).* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

KDD '26, Jeju, Korea,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Automated evaluation of compound AI systems faces a bootstrapping paradox: calibrating an LLM evaluator requires ground truth labels, yet generating those labels at scale is precisely what automation should eliminate. This circularity is acute for retrieval-augmented QA systems, where retrieval, ranking, generation, and safety components interact to produce emergent conversational behaviors that cannot be assessed through component-level metrics. A compound RAG system exemplifies this complexity: a single user interaction may traverse multiple model components, with conversation quality depending on their collective behavior across several turns.

Traditional evaluation relies on domain experts who manually assess conversations. This produces high-quality annotations but faces prohibitive scaling constraints. Our baseline required 26 human annotators working 10 days to evaluate 720 synthetic conversations (~2.6 person-hours each). This bottleneck delays launches for new features, limits regression detection, and prevents continuous monitoring. LLM-as-judge approaches [6, 12] offer an alternative but either assume access to ground truth for calibration or accept uncalibrated judgments, neither satisfying production requirements.

We break this circularity with *iterative consensus synthesis*: an algorithm that treats LLM-human agreement as a free signal. When an LLM evaluator and human annotator agree, their concordant judgment becomes ground truth; no additional annotation required. When they disagree, a second human annotator provides a tiebreaker via majority vote. This selective escalation achieves multi-annotator reliability at single-annotator cost (15.95× reduction). We combine this with a Wilson interval decision framework that transforms noisy per-conversation scores into confidence-bounded performance evaluations, and embed both within a five-layer architecture where evaluation outcomes inform remediation and remedy effectiveness refines future detection thresholds. We termed it *recursive resolution.*

Contributions. We make three contributions: (1) iterative consensus synthesis, an algorithm that reduces annotation cost from 7,672 to 481 hours while generating ground truth across 13 quality metrics; (2) a tiered Wilson interval framework for statistically rigorous, multi-metric performance evaluations; and (3) demonstration of complete recursive resolution cycles across three locale-specific test suites, where the system identified content gaps, validated fixes through shadow evaluation, and generalized findings to prevent similar regressions.

2 RELATED WORK

LLM-as-Judge Evaluation. The use of large language models for automated evaluation has gained substantial attention following demonstrations of strong correlation with human judgments on specific tasks. G-Eval [6] employs chain-of-thought prompting to assess coherence and fluency, achieving improved alignment over reference-based metrics. MT-Bench [12] introduces pairwise comparison for multi-turn dialogue, while AlpacaEval [5] addresses length bias in instruction-following evaluation. These approaches typically assume access to ground truth for calibration or accept uncalibrated LLM judgments. Our work differs fundamentally by treating ground truth generation as a first-class problem, developing an iterative consensus mechanism that produces calibration data as a byproduct of the evaluation process itself.

Synthetic Data Generation for Dialogue. Generating synthetic conversations for training and evaluation spans template-based approaches [8], retrieval-augmented generation [4], and fully neural synthesis [10]. The challenge for evaluation differs from training: synthetic test cases must cover the distribution of production scenarios while remaining realistic enough that quality assessments transfer to real conversations. Our synthetic conversation generator addresses this through archetype-conditioned generation combined with actual system responses, ensuring that generated conversations reflect true system behavior rather than simulated approximations.

Confidence Estimation and Statistical Testing. Rigorous evaluation requires distinguishing true quality differences from sampling noise. Card et al. [2] demonstrate that reported improvements in NLP often fall within confidence intervals, calling for more careful statistical practice. Wilson score intervals [11] provide robust confidence bounds for binomial proportions, performing well even with small samples and extreme success rates, properties essential for detecting rare but critical failures like safety violations. We extend this framework to multi-metric evaluations with explicit tier-based aggregation rules.

Self-Improving Systems. The concept of systems that improve through their own outputs has deep roots in machine learning, from online learning [9] to recent work on constitutional AI [1] and self-refinement [7]. Our architecture operationalizes self-improvement for evaluation systems specifically: the effectiveness of deployed fixes feeds back to refine detection thresholds, root cause classifiers, and remedy selection policies.

3 METHODS

3.1 Problem Formulation

Consider a compound AI system $\mathcal{S}$ that processes user conversations. Let $C = \{c_1, \ldots, c_N\}$ denote a corpus of multi-turn conversations, where each conversation $c = \{(u_1, r_1), \ldots, (u_T, r_T)\}$ comprises user utterances u_t and system responses r_t. We seek to evaluate $\mathcal{S}$ across M quality metrics $\{m_1, \ldots, m_M\}$, where each metric $m_i : C \to \{0, 1\}$ indicates whether conversation c satisfies quality criterion i.

The evaluation problem decomposes into three subproblems. The *ground truth problem* asks: given limited annotation budget B, how do we establish reliable ground truth labels $\mathcal{G}_i(c)$ for metric i on conversation c? The *coverage problem* asks: given a production query distribution $P(q)$, how do we generate a test corpus C that adequately covers likely interaction patterns? The *decision problem* asks: given noisy estimates $\hat{p}_i$ of metric failure rates, how do we make statistically rigorous recommendations?

Our architecture addresses these problems through a five-layer system with feedback from evaluation outcomes to threshold refinement.

3.2 System Architecture

The evaluation ecosystem comprises five layers operating in a closed feedback loop (Figure 1). The Evaluation Layer (Layer 1) ingests conversations from synthetic test suites, applying an LLM-as-a-judge evaluator to assess quality across 13 metrics organized into three priority tiers. The Root Cause Engine (Layer 2) aggregates detected violations, clusters similar issues using semantic embeddings of evaluator reasoning, and classifies root causes into six categories: prompt engineering deficiencies, conversation flow failures, retrieval errors, model boundary violations, content gaps, or evaluation system issues. The Remedy Engine (Layer 3) proposes targeted interventions based on root cause classification: prompt optimization, knowledge base updates, retrieval tuning, evaluation threshold adjustment, with human approval required before deployment. The Rollout Manager (Layer 4) implements shadow deployment comparing candidate and control configurations, enforcing automated gates for safety, quality, and latency before a candidate configuration is accepted. The Continuous Learning Layer (Layer 5) tracks remedy effectiveness, identifies recurring patterns, and feeds updated thresholds and detection rules back to Layer 1.

The recursive structure distinguishes our approach from static evaluation pipelines. When a remedy successfully addresses an issue cluster, Layer 5 updates the corresponding detection threshold in Layer 1, effectively encoding the lesson that this failure mode has been resolved. Conversely, when remedies prove ineffective, Layer 5 may lower thresholds to increase sensitivity or flag the issue category for human investigation. This feedback loop enables the system to calibrate its own evaluation criteria based on empirical outcomes rather than fixed heuristics.

3.3 Synthetic Conversation Generation

Comprehensive evaluation requires test cases that cover the distribution of interactions the system is expected to execute on. Manual test creation achieved 720 conversations over 10 days with 26 annotators. The rate is insufficient for continuous evaluation as systems evolve. Our synthetic conversation generator automates this process through archetype-conditioned conversation synthesis.

Let $\mathcal{Q} = \{q_1, \ldots, q_K\}$ denote the set of top queries covering 85% of representative query traffic for a content domain, and let $\mathcal{T} = \{\tau_1, \ldots, \tau_5\}$ denote the conversation archetypes. For each $(q, \tau) \in \mathcal{Q} \times \mathcal{T}$, the synthetic conversation generator produces a conversation through iterative simulation. The user simulator, implemented as a prompted LLM, generates utterances conditioned on conversation history and archetype:

$$u_{t+1} \sim P_{\text{LLM}}(u \mid c_{1:t}, q, \tau, \phi) \tag{1}$$

where $c_{1:t} = \{(u_1, r_1), \ldots, (u_t, r_t)\}$ is the conversation history and ϕ encodes archetype-specific behavioral constraints. The system

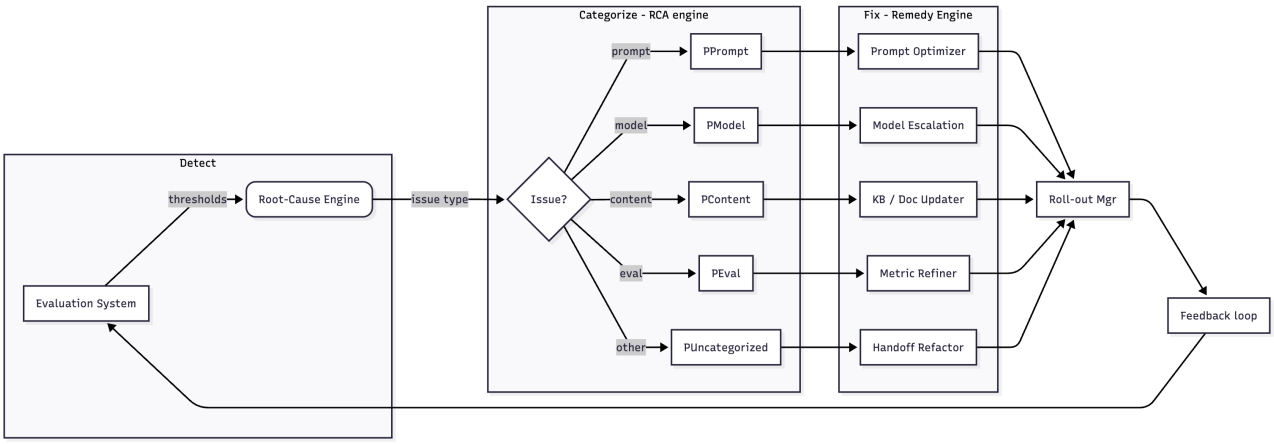

Figure 1: The closed-loop architecture enables recursive resolution: evaluation outcomes inform remediation, and remedy effectiveness refines future evaluation. Forward arrows indicate forward flow; backward arrow indicates feedback from Layer 5 to Layer 1.

response r_{t+1} is obtained by invoking the actual compound RAG system under test, ensuring that generated conversations reflect true system behavior rather than simulated approximations.

Generation terminates under three conditions: the conversation reaches maximum length $T_{\max} = 10$ turns, the system routes to human support (indicating inability to resolve automatically), or the conversation reaches natural completion as determined by the archetype specification. This procedure yields $|\mathcal{Q}| \times |\mathcal{T}|$ conversations per content domain, providing systematic coverage of query-archetype combinations.

Human validators reviewed generated conversations to assess quality. Of 1,946 generated conversations, validators flagged 7.45% as exhibiting generation artifacts (incoherent user responses, unrealistic behavior) and 20.86% as revealing genuine system-under-test issues (valuable for debugging but excluded from ground truth calibration). The remaining 71.79% formed the high-quality evaluation corpus of 1,397 conversations.

3.4 Two-Round Iterative Ground Truth Synthesis

Establishing ground truth for evaluator calibration presents a cost-quality tradeoff. Single-annotator labels are inexpensive but noisy; multi-annotator consensus is reliable but expensive. Our two-round mechanism achieves multi-annotator quality at reduced cost by exploiting agreement structure (Figure 2).

Let $E_i(c) \in \{0, 1\}$ denote the LLM judge's judgment for metric i on conversation c, and let $M_{1,i}(c)$ denote the corresponding judgment from human annotator H_1. Round 1 computes agreement across all metrics:

$$A_1(c) = \prod_{i=1}^{M} \mathbb{1}[E_i(c) = M_{1,i}(c)] \tag{2}$$

where $A_1(c) = 1$ indicates complete agreement across all M metrics. Conversations with complete agreement proceed directly to ground truth: $\mathcal{G}_i(c) = E_i(c) = M_{1,i}(c)$ for all i.

Conversations with any disagreement ($A_1(c) = 0$) are escalated to Round 2. A second human annotator H_2 provides independent judgments $M_{2,i}(c)$ for all metrics on the escalated conversation. Ground truth is then established through majority voting:

$$\mathcal{G}_i(c) = \begin{cases} 1 & \text{if } E_i(c) + M_{1,i}(c) + M_{2,i}(c) \geq 2 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

This formulation guarantees that ground truth reflects agreement of at least two of three evaluators for every metric on every conversation. The efficiency gain arises from the observation that complete agreement, while relatively rare at the conversation level (13.2% in our experiments), allows substantial reduction in per-conversation annotation effort.

The complete procedure is formalized in Algorithm 3 (Appendix A). The expected annotation cost per conversation is $1 + (1 - p)$ human evaluations, where p is the Round 1 complete-agreement rate, compared to 2 evaluations under a standard two-annotator protocol. More significantly, LLM-based evaluation is computationally inexpensive (<1 second per conversation), shifting the cost structure dramatically: the primary expense becomes Round 2 human annotation of disagreement cases only.

3.5 LLM-as-a-Judge Evaluation with Tiered Metrics

The LLM judge evaluates conversations across 13 quality metrics organized into three priority tiers (full definitions in Appendix B): BLOCKER metrics ($w_B = 10.0$) individually fail the evaluation, P0 metrics ($w_{P0} = 5.0$) affect query resolution, and P1 metrics ($w_{P1} = 2.0$) affect experience quality. Two additional metrics are monitored but excluded from evaluation decisions.

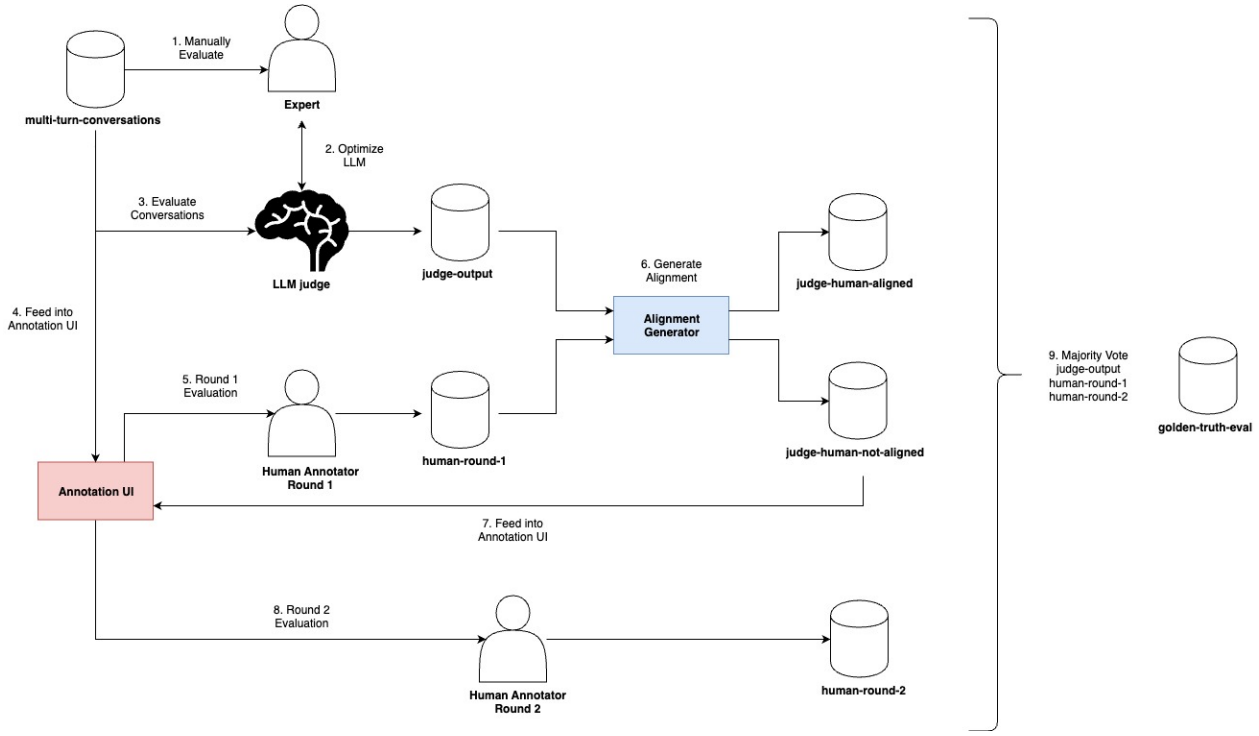

Figure 2: Iterative consensus mechanism. Conversations with full Round 1 agreement bypass Round 2 entirely, reducing annotation cost while maintaining consensus quality for disagreement cases.

For each conversation c, the LLM judge produces judgments $E(c) = (E_1(c), \ldots, E_M(c)) \in \{0,1\}^M$ along with natural language reasoning. The prompt, optimized using DSPy [3], instructs the LLM to evaluate each dimension against explicit criteria before rendering binary judgments. The severity score aggregates violations weighted by tier:

$$S(c) = w_B \cdot \sum_{i \in \mathcal{B}} E_i(c) + w_{P0} \cdot \sum_{i \in \mathcal{P}_0} E_i(c) + w_{P1} \cdot \sum_{i \in \mathcal{P}_1} E_i(c) \quad (4)$$

where $\mathcal{B}$, $\mathcal{P}_0$, $\mathcal{P}_1$ denote the BLOCKER, P0, and P1 metric sets respectively.

3.6 Wilson Interval Decision Framework

Thorough evaluation requires distinguishing true quality regressions from sampling variation. A system showing 5% failure rate on 100 test cases might have a true failure rate anywhere from 2% to 10%; declaring this a regression relative to a 4% baseline would be premature. We employ Wilson score intervals [11] to establish confidence bounds that inform statistically rigorous decisions.

For a test corpus of n conversations with X observed failures on metric i, the point estimate $\hat{p}_i = X/n$ has associated Wilson lower bound:

$$L_i = \frac{\hat{p}_i + \frac{z^2}{2n} - z\sqrt{\frac{\hat{p}_i(1-\hat{p}_i)}{n} + \frac{z^2}{4n^2}}}{1 + \frac{z^2}{n}} \quad (5)$$

where $z = \Phi^{-1}(1 - \alpha/2)$ is the critical value for confidence level $1 - \alpha$. We use $\alpha = 0.05$, yielding 95% confidence intervals.

Let θ_i denote the ground truth baseline failure rate for metric i. A metric exhibits statistically significant regression if $L_i > \theta_i$; that is, we are 95% confident the true failure rate exceeds the established baseline. The evaluation result aggregates metric-level results according to tier:

$$\text{Decision} = \begin{cases} \textsc{Fail} & \text{if } \exists i \in \mathcal{B} : L_i > \theta_i \\ \textsc{Fail} & \text{if } |\{i \in \mathcal{P}_0 : L_i > \theta_i\}| \geq 3 \\ \textsc{Investigate} & \text{if } \exists i \in \mathcal{P}_0 : L_i > \theta_i \\ \textsc{Pass} & \text{otherwise} \end{cases} \quad (6)$$

This framework ensures that any BLOCKER regression fails evaluation, accumulated P0 regressions trigger rejection, isolated P0 issues prompt investigation without blocking, and P1 issues alone permit passing. The statistical grounding prevents both false alarms from sampling noise and missed detections from inadequate power.

4 EXPERIMENTS

4.1 Experimental Setup

The system under test is a Retrieval-Augmented Generation (RAG) system that serves user queries through a pipeline of retrieval, ranking, generation, and safety components. This compound architecture exemplifies modern enterprise AI systems where localization decisions affect multiple interacting components.

Table 1: Alignment with consensus ground truth (%). The LLM judge achieves reliability comparable to human annotators, with strongest performance on objective safety metrics.

Metric	H_1	LLM Judge	H_2
BLOCKER			
Courteous	99.6	99.1	98.4
PII	100.0	99.9	100.0
On Topic	99.5	99.9	98.3
Offensive Language	99.7	100.0	98.4
P0			
Knowledgeable	84.4	87.6	81.2
Helpful	75.2	78.8	87.5
Intent Understanding	75.6	73.6	81.3
Relevant Response	86.3	84.4	84.0
P1			
Easy to Understand	93.5	88.7	88.7
Concise	90.1	93.0	88.0
Efficient	72.0	73.0	87.3

We evaluated the system across three locale-specific test suites. The synthetic conversation generator produced 1,946 multi-turn conversations with equal distribution across five user archetypes, averaging 7.53 turns per conversation. After quality filtering, 1,397 conversations formed the evaluation corpus. For each additional locale, the generator produced a locale-specific test suite of 555 conversations.

The LLM judge used prompts optimized through DSPy. Round 1 annotation employed 20 human annotators over 2.5 days. Round 2 annotation employed 8 human annotators over 3 days to resolve 1,213 disagreement cases (86.8% of conversations required Round 2). Ground truth was established through majority voting as described in Section 3.4.

4.2 Ground Truth Quality

Table 1 presents alignment between each evaluator and the consensus ground truth. The LLM judge achieves 99%+ alignment on BLOCKER metrics, where evaluation criteria are unambiguous and safety is paramount. For subjective P0 metrics, alignment ranges from 73.6% (Intent Understanding) to 87.6% (Knowledgeable), comparable to individual human annotators. Notably, H_2 achieves higher alignment on several P0 metrics, reflecting the value of seeing disagreement cases that highlight evaluation ambiguity.

The alignment patterns reveal an important property: disagreement concentrates on inherently subjective dimensions. Metrics like *Efficient* ("Did the conversation require excessive turns?") admit legitimate disagreement even among careful human annotators. The LLM judge's alignment on these metrics (73.0%) falls within the range of human inter-annotator agreement, suggesting that lower alignment reflects genuine ambiguity rather than evaluator failure.

4.3 Baseline Performance and Efficiency

Table 2 presents the system's performance on the ground truth corpus, establishing baseline failure rates θ_i for the Wilson interval framework. BLOCKER metrics show near-perfect performance (99.9%–100% pass rate), validating system safety. P0 metrics reveal improvement opportunities: *Helpful* (54.5%) and *Intent Understanding* (58.2%) indicate that roughly half of conversations fail to fully address user needs.

Table 2: Baseline pass rates (%) on synthetic data, establishing thresholds for regression detection.

Metric	Pass %	Metric	Pass %
Courteous	99.9	Helpful	54.5
PII	100.0	Efficient	53.6
On Topic	99.9	Intent Understanding	58.2
Offensive Language	100.0	Relevant Response	87.3
Knowledgeable	79.3	Concise	94.8
Easy to Understand	95.1		

The two-round mechanism achieved substantial efficiency gains. Under the baseline approach (two annotators per conversation with consensus resolution), evaluating 1,946 conversations would require approximately 7,672 hours based on extrapolation from our prior experience (3,600 turns requiring 1,885 annotator-hours). Our approach required 481 hours: 325 hours for Round 1 plus 156 hours for Round 2. This represents a **15.95**× reduction in annotation effort while increasing test coverage from 3,600 to 14,653 evaluated turns, a **307%** improvement. This efficiency gain translates to significant cost savings per evaluation cycle and enables continuous evaluation frequency.

4.4 Recursive Resolution: Complete Cycle Evidence

The system was applied to evaluate the RAG system across three locale-specific datasets. We present two case studies demonstrating complete recursive resolution cycles where all five layers operated in sequence.

Case Study 1: Content Format Divergence. During the evaluation process for the second locale-specific dataset, the LLM judge flagged synthetic conversations with significantly elevated *Unknowledgeable* failures. Layer 2 root cause analysis clustered 89% of these failures, with semantic analysis of the LLM judge's reasoning revealing that responses lacked locale-specific information for a given content domain.

Quantitative investigation by Layer 3 identified the root cause: domain-specific help content in this locale was stored in a separate "domain-specific help collection" rather than the standard database collection queried by the system. This locale's knowledge base contained only 297 documents compared to 431 in the baseline locale (a 31% gap). However, 130 additional documents relating to the domain existed in the alternate format, bringing potential coverage to 427 documents (99% of baseline coverage).

Layer 3 proposed a targeted remedy: modify the system's retrieval call to query both collection formats:

$$\texttt{collections} = \{\texttt{HelpDB}_{locale}, \texttt{DomainHelp}_{global}\} \tag{7}$$

Layer 4 validated the fix through shadow evaluation on 200 domain-specific synthetic conversations, showing *Knowledgeable* improvement from 58.3% to 79.1% (+20.8 percentage points) with

acceptable latency increase (p95: 1.2s → 1.4s). Evaluation on the full locale test set showed a **42% reduction in failed Q&A conversations** for this domain.

Critically, Layer 5 generalized this finding: the meta-learner recorded the pattern "locale expansion may introduce content format divergence for domains with global vs. regional help architectures" and added proactive content format audits to the evaluation checklist. This heuristic was applied during the subsequent locale expansion, where similar format divergence was identified and addressed *before* evaluation rather than after, preventing a regression that would have otherwise required a second remediation cycle.

Case Study 2: Systematic Knowledge Gaps. The third locale's evaluation on 555 synthetic conversations demonstrated the Wilson interval framework's ability to distinguish actionable regressions from noise. Table 3 (Appendix C) presents the complete analysis.

The system issued a Pass-with-Investigation decision based on two failing P0 metrics. Layer 2 analysis revealed that *Unknowledgeable* failures were 2× the baseline rate (55.1% vs. 27.2%), with variation by content domain ranging between 33% and 86%. Semantic clustering of the LLM judge's reasoning identified six systematic failure themes: (1) inability to access user account details, (2) inability to provide specific information (prices, dates), (3) lack of knowledge about regional restrictions and edge cases, (4) inability to answer temporal questions ("how long," "when exactly"), (5) no visibility into temporary changes or account status, and (6) context loss across conversation turns.

Layer 3 classified these into two categories: content gaps (themes 1–4, addressable through knowledge base expansion) and capability limitations (themes 5–6, requiring engineering roadmap items). This triage, which distinguishes actionable content issues from longer-term capability gaps, exemplifies how the recursive resolution architecture converts evaluation signals into differentiated remediation strategies.

5 DISCUSSION AND FUTURE WORK

Limitations. Ground truth quality depends on annotator expertise; domain-specific training remains essential. The Wilson framework assumes metric independence, which may not hold; the third-locale analysis found 50% overlap between *Unhelpful* and *Unknowledgeable* failures. Layer 3 remedies require human approval; fully autonomous remediation remains future work. Our evaluation focused on English; multilingual extension requires validation.

Future Directions. Current evaluation identifies *that* issues exist without pinpointing *which* component is responsible. Component-level evaluation will assess individual components (e.g., retriever, ranker, generator) for precise root cause attribution. Other modalities like voice will also adapt the framework, addressing ASR error propagation and prosodic information loss.

6 CONCLUSION

We presented iterative consensus synthesis, an algorithm that breaks the ground truth bootstrapping problem by treating LLM-human agreement as a free signal. This selective escalation reduces annotation effort by 15.95× while achieving multi-annotator reliability. Combined with Wilson interval decisions and a recursive resolution architecture, the approach enables self-improving evaluation at scale. These results show that scalable, statistically rigorous evaluation of compound AI systems is achievable, enabling continuous quality monitoring that would otherwise be prohibitively expensive.

REFERENCES

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

[2] Dallas Card, Peter Henderson, Urvashi Khandelwal, Robin Jia, Kyle Mahowald, and Dan Jurafsky. 2020. With Little Power Comes Great Responsibility. *arXiv preprint arXiv:2010.06595* (2020).

[3] Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2023. DSPy: Compiling Declarative Language Model Calls into Self-Improving Pipelines. *arXiv preprint arXiv:2310.03714* (2023).

[4] Hyunwoo Kim, Jack Hessel, Liwei Jiang, Ximing Lu, Youngjae Yu, Pei Zhou, Ronan Le Bras, Malihe Alikhani, Gunhee Kim, Maarten Sap, and Yejin Choi. 2023. SODA: Million-scale Dialogue Distillation with Social Commonsense Contextualization. *arXiv preprint arXiv:2212.10465* (2023).

[5] Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. AlpacaEval: An Automatic Evaluator for Instruction-following Language Models. https://github.com/tatsu-lab/alpaca_eval.

[6] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. *arXiv preprint arXiv:2303.16634* (2023).

[7] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2023. Self-Refine: Iterative Refinement with Self-Feedback. *Advances in Neural Information Processing Systems* 36 (2023).

[8] Pararth Shah, Dilek Hakkani-Tür, Gokhan Tür, Abhinav Rastogi, Ankur Bapna, Neha Nayak, and Larry Heck. 2018. Bootstrapping a Neural Conversational Agent with Dialogue Self-Play, Crowdsourcing and On-Line Reinforcement Learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 41–51.

[9] Shai Shalev-Shwartz. 2012. Online Learning and Online Convex Optimization. *Foundations and Trends in Machine Learning* 4 (2012), 107–194.

[10] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-Instruct: Aligning Language Models with Self-Generated Instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*. 13484–13508.

[11] Edwin B Wilson. 1927. Probable Inference, the Law of Succession, and Statistical Inference. *J. Amer. Statist. Assoc.* 22, 158 (1927), 209–212.

[12] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, Vol. 36. 46595–46623.

A ALGORITHM: TWO-ROUND GROUND TRUTH SYNTHESIS

Algorithm 1: Two-Round Ground Truth Synthesis

Input: Corpus C, LLM judge evaluator E, annotator pools A_1, A_2
Output: Ground truth labels $\mathcal{G}$

Initialize $\mathcal{G} \leftarrow \emptyset$, $\mathcal{D} \leftarrow \emptyset$, $\mathbf{E} \leftarrow \emptyset$, $\mathbf{M}_1 \leftarrow \emptyset$
for each $c \in C$ **do**
  $\mathbf{E}[c] \leftarrow E(c)$ ▷ LLM judge on all M metrics
  $\mathbf{M}_1[c] \leftarrow A_1(c)$ ▷ Human annotator Round 1
  if $\mathbf{E}[c] = \mathbf{M}_1[c]$ **then**
    $\mathcal{G}[c] \leftarrow \mathbf{E}[c]$ ▷ Full agreement
  else
    $\mathcal{D} \leftarrow \mathcal{D} \cup \{c\}$ ▷ Escalate to Round 2
for each $c \in \mathcal{D}$ **do**
  $\mathbf{m}_2 \leftarrow A_2(c)$ ▷ Human annotator Round 2
  for $i = 1, \ldots, M$ **do**
    $\mathcal{G}_i[c] \leftarrow \text{majority}(\mathbf{E}_i[c], \mathbf{M}_{1,i}[c], \mathbf{m}_{2,i})$
return $\mathcal{G}$

Figure 3: Two-round iterative consensus procedure. Round 1 identifies LLM judge–annotator agreement. Only disagreements proceed to Round 2 for majority voting.

B COMPLETE METRIC DEFINITIONS

This appendix provides complete definitions for all 13 LLM judge metrics, organized by priority tier.

BLOCKER Metrics (weight $w_B = 10.0$) represent violations that individually fail evaluation.

Off_topic: The bot discusses matters outside the system's scope. A conversation passes if all bot responses remain within system topical boundaries.

Offensive_language: The bot uses inappropriate, offensive, or harmful language including profanity, slurs, or demeaning characterizations.

Discourteous: The bot is rude, dismissive, or disrespectful to the user, including interrupting or ignoring stated concerns.

PII_exposure: The bot inappropriately reveals, requests, or handles personally identifiable information.

P0 Metrics (weight $w_{P0} = 5.0$) affect whether the user's query is successfully resolved.

Unknowledgeable: The bot provides incorrect or outdated information.

Unhelpful: The bot fails to address the user's actual need, even if responses are polite and factually accurate.

Intent_understanding: The bot misinterprets what the user is asking.

Non_relevant_response: The bot's response does not relate to the current conversation context.

P1 Metrics (weight $w_{P1} = 2.0$) affect experience quality without determining resolution.

Hard_to_understand: The bot's response is confusing or poorly structured.

Verbose: The bot's response is unnecessarily long.

Inefficient: The conversation requires excessive turns to resolve.

Spelling_errors: The bot's response contains spelling or grammar errors.

Monitored Metrics are tracked but excluded from evaluation.

Unempathetic: The bot fails to acknowledge user emotions or frustration.

Prompt_injection: The user attempts to manipulate bot behavior through adversarial prompting.

C THIRD LOCALE EVALUATION DETAILED ANALYSIS

C.1 Wilson Interval Results

Table 3 presents the complete Wilson interval analysis for the third locale dataset evaluation on 555 conversations.

Table 3: Third locale Wilson interval analysis ($n = 555$, $\alpha = 0.05$). Two P0 metrics failed, triggering Pass-with-Investigation decision.

Metric	Tier	% Fail	L_i	θ_i	Status
PII	BLOCKER	0.0	0.00	0.00	Pass
Off Topic	BLOCKER	0.0	0.00	0.00	Pass
Offensive	BLOCKER	0.0	0.00	0.00	Pass
Discourteous	BLOCKER	0.9	0.3	0.8	Pass
Unknowledgeable	P0	55.1	50.9	27.2	**Fail**
Unhelpful	P0	72.1	68.1	62.6	**Fail**
Intent Understand.	P0	61.8	57.6	59.8	Pass
Non-relevant	P0	15.0	12.1	23.3	Pass
Inefficient	P1	78.0	74.3	69.2	Fail
Verbose	P1	15.3	12.5	9.0	Fail

C.2 Failure Theme Analysis

Semantic clustering of the LLM judge's reasoning identified six systematic failure themes:

(1) **Account Access**: Inability to access user account details
(2) **Specific Information**: Inability to provide specific information (prices, dates)
(3) **Regional Knowledge**: Lack of knowledge about regional restrictions and edge cases
(4) **Temporal Questions**: Inability to answer temporal questions ("how long," "when exactly")
(5) **Dynamic Content**: No visibility into temporary changes or account status
(6) **Context Retention**: Context loss across conversation turns (intent failures)

Layer 3 classified themes 1–4 as *content gaps* (addressable through knowledge base expansion) and themes 5–6 as *capability limitations* (requiring engineering roadmap items).

D WILSON INTERVAL DERIVATION

The Wilson score interval provides a confidence interval for a binomial proportion that performs well across all sample sizes and success rates, unlike the normal approximation which fails near boundaries and with small samples.

Starting from the score test, we seek values of p for which the observed proportion $\hat{p} = X/n$ is not significantly different from p. The score statistic is:

$$z = \frac{\hat{p} - p}{\sqrt{p(1-p)/n}} \tag{8}$$

Setting $|z| \leq z_{\alpha/2}$ and solving for p yields the Wilson interval. Squaring both sides:

$$z^2 = \frac{(\hat{p} - p)^2}{p(1-p)/n} \tag{9}$$

Rearranging:

$$z^2 p(1-p) = n(\hat{p} - p)^2 \tag{10}$$

Expanding and collecting terms in p:

$$(n + z^2)p^2 - (2n\hat{p} + z^2)p + n\hat{p}^2 = 0 \tag{11}$$

Applying the quadratic formula yields the Wilson interval bounds:

$$p = \frac{(2n\hat{p} + z^2) \pm \sqrt{z^4 + 4n\hat{p}z^2(1-\hat{p})}}{2(n + z^2)} \tag{12}$$

Which simplifies to the form presented in the main text. For our application with $\alpha = 0.05$, we use $z = 1.96$.