

# Agent-in-the-Loop: A Data Flywheel for Continuous Improvement in LLM-based Customer Support

Cen (Mia) Zhao Tiantian Zhang Hanchen Su Yufeng (Wayne) Zhang  
Shaowei Su Mingzhi Xu Yu (Elaine) Liu Wei Han  
Jeremy Werner Claire Na Cheng Yashar Mehdad  
Airbnb, Inc., USA

{mia.zhao, tiantian.zhang, hanchen.su, wayne.zhang, shaowei.su, mingzhi.xu, elaine.liu, wei.han, jeremy.werner, claire.cheng, yashar.mehdad}@airbnb.com

## Abstract

We introduce an *Agent-in-the-Loop* (AITL) framework that implements a continuous *data flywheel* for iteratively improving an LLM-based customer support system. Unlike standard offline approaches that rely on batch annotations, AITL integrates four key types of annotations directly into live customer operations: (1) pairwise response preferences, (2) agent adoption decisions and rationales, (3) knowledge relevance checks, and (4) identification of missing knowledge. These feedback signals seamlessly feed back into model updates, reducing retraining cycles from months to weeks. Our production pilot involving US-based customer support agents demonstrated significant improvements in retrieval accuracy (+11.7% recall@75, +14.8% precision@8), generation quality (+8.4% helpfulness), and agent adoption rates (+4.5%). These results underscore the effectiveness of embedding human feedback loops directly into operational workflows to continuously refine LLM-based customer support systems.

## 1 Introduction

Retrieval-augmented generation (RAG) improves large language models (LLMs) by grounding responses to external knowledge, overcoming static limitations, and improving transparency through evidence-based outputs (Lewis et al., 2020). However, traditional LLMs, typically trained in a fixed dataset with static knowledge cut-off points, inherently struggle to adapt to evolving real-world interactions without interventions such as continuous learning or retrieval enhancement (Shah et al., 2023).

Recent research emphasizes the importance of a *data flywheel*, an iterative feedback loop that continuously leverages new interaction data to enhance model performance (Luo et al., 2024). In customer

support scenarios, such a data flywheel is particularly valuable due to evolving product features, shifting user preferences, and continuously updated policies and procedures. Dai et al. (2025)’s daily oracle benchmark demonstrates that static models, even when paired with retrieval, lose more than 20 percentage points of accuracy on news questions within a few years, indicating that continuous feedback loops are crucial for preventing drift and maintaining relevance in real-world systems.

**Our Contributions.** To maintain a continuous human-driven data flywheel for accurate and relevant customer support, we (1) develop an *annotation interface* capturing response preferences, adoption rationales, knowledge relevance, and missing knowledge during live conversations, and (2) implement a *continuous learning pipeline* that integrates these annotations into training datasets, reducing model update cycles from months to weeks. A US-based pilot confirms significant improvements in retrieval accuracy, response helpfulness, citation correctness, and agent adoption rates. To further optimize annotation efficiency at scale, we recommend delaying annotations for preference, adoption, and knowledge relevance, while immediately annotating missing knowledge when SLAs permit.

## 2 Related Work

To address critical issues such as preference drift and knowledge decay, recent research has integrated human or AI-simulated feedback within reinforcement learning frameworks.

**Human-in-the-Loop and Preference Optimization.** Human-in-the-loop (HITL) approaches enhance LLM alignment by directly optimizing outputs toward explicit human preferences, moving beyond traditional supervised learning metrics (Stienon et al., 2020). Reinforcement Learning

with Human Feedback (RLHF), pioneered by OpenAI (Ouyang et al., 2022a), aligns models with human preferences through pairwise comparisons obtained via offline annotations. Anthropic subsequently introduced iterative online human feedback loops, continuously incorporating real-time human annotations to significantly enhance conversational agent helpfulness and harmlessness (Bai et al., 2022a). Further advancing alignment at scale, Anthropic proposed Constitutional AI, a method employing Reinforcement Learning from AI Feedback (RLAIF) guided by explicit human-defined principles (Bai et al., 2022b).

**Data Flywheel and Continuous Learning Pipelines.** Recently, Luo et al. (2024) introduced *Arena Learning*, an automated data flywheel using simulated self-play between LLMs and AI judges to generate offline preference labels. While highly scalable, *Arena Learning* predominantly addresses open-domain dialogues and lacks mechanisms for domain-specific knowledge retrieval or incorporating human feedback. Consequently, it does not directly tackle real-world preference drift issues inherent in dynamic environments.

Our approach integrates and extends these methodologies not only by collecting online human preference feedback, but also explicitly gathering feedback on knowledge relevance and missing knowledge. Similarly to *Arena Learning*, we incorporate an LLM-based virtual judge (VJ) to filter data quality. By carefully aligning annotation methods with our training pipeline, we achieve update cycles comparable to *Arena Learning*’s automated data flywheel, with key differences summarized in Table 1.

### 3 Method

Figure 1 illustrates the interactive workflow in the following key steps: (1) *Customer Input*: A customer sends a query or message. (2) *LLM-Based Interactive System*: The system retrieves relevant knowledge (Sect. 3.1) and uses an LLM to generate response candidates. (3) *Suggested Responses*: The system presents two alternative responses, potentially originating from different models. (4) *Agent Annotation*: A support agent evaluates these suggestions while serving customers, indicating their preferred response, adoption decision, critical feedback, assessment of the relevance of the knowledge used by LLM, and adding any necessary missing information (Sect. 3.2). (5) *Review Annotation*: Both

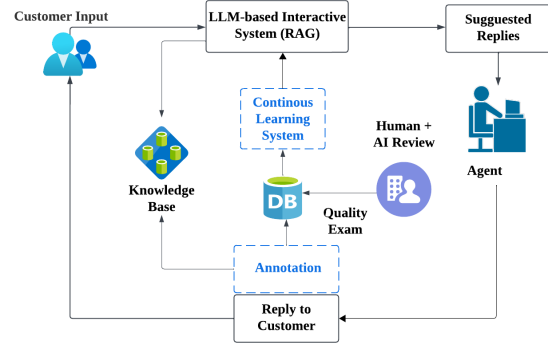


Figure 1: Overview of the agent-in-the-loop architecture.

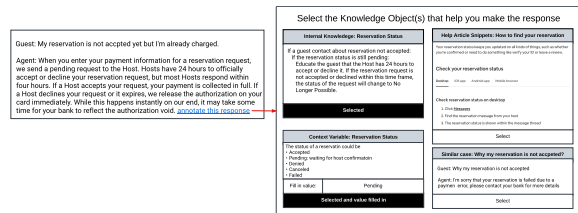


Figure 2: Example of selecting knowledge references

a human expert and the LLM-based verifier review the annotation and the agent–customer interaction to flag any conflicts. (Sect. 3.3). (6) *Continuous Learning*: The annotations and feedback collected are reintegrated into the training pipeline for continuous model improvements (Sect. 3.4).

#### 3.1 Unified Knowledge Base

We consolidate diverse domain resources, customer guides, FAQs, internal policies, workflows, dynamic context (e.g., reservation status) and historical cases into a *Unified Knowledge Base* (Figure 2). The resources are enriched with detailed metadata in a centralized content management system, which facilitates the annotation and retrieval of agents in real time. We can also modularize policies and workflows into LLM-friendly units with versioning and temporal validity following Su et al. (2025) to improve annotation granularity.

#### 3.2 Agent Annotation

Figure 3 illustrates our online annotation workflow, which comprises four main steps:

**Step 1: Pairwise Response Preference** Agents compare randomly ordered candidate responses and annotate degrees of preference as *significantly better*, *better*, or *slightly better*. These signals inform preference learning and help to improve generation models.

Aspect	Arena Learning	AITL (ours)
Feedback Source	AI-generated annotations (simulated self-play)	Human-generated annotations (real-world interaction)
Annotation Mode	Offline annotation	Real-time (or near-real-time) annotation
Update Frequency	Weekly	Weekly
Target Modules	Generation module only	Retrieval, Ranking, and Generation modules
Handling of Human Preference Drift	Indirect (no real-time human feedback)	Direct (real-time human feedback integration)
Verification Process	LLM-based virtual judge	LLM-based virtual judge combined with sampled human verification

Table 1: Comparison between Arena Learning and AITL on feedback mechanisms and training pipelines.

Figure 3: Online annotation interface.

## Step 2: Rationale for Response Selection

Agents provide the adoption decision and rationales in free text (critiques), which supports the improvement of the generation model and a broader evaluation.

## Step 3: Relevance of knowledge resources

Agents assess and score the relevance of the knowledge resource used in the prompt, performing an additional verification in real time. These annotations directly enhance the retrieval process across diverse support topics.

## Step 4: Missing Knowledge Identification

During this stage, agents use a dedicated Knowledge Resources Selection Interface to flag missing information, such as policies or unrecorded best practices, that they rely on to help customers. By integrating these newly identified gaps and references back into the training pipeline, the system can improve existing retrieval recall and continuously adapt to the evolving knowledge landscape of live customer support.

## 3.3 Review Annotation

Human and LLM-based verifiers assess the consistency between agent annotations and actual interactions, identifying common errors: preference mismatches, incorrect knowledge relevance, adoption discrepancies, and omitted knowledge (examples in Appendix A). The human and LLM verifier evaluations show a strong correlation (Appendix G).

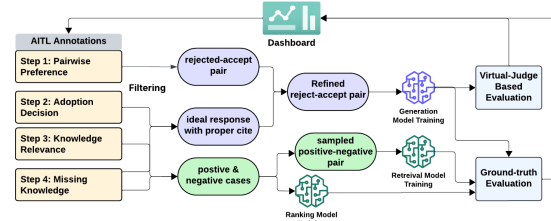


Figure 4: Data in continuous learning pipeline

## 3.4 Continuous Learning Pipeline

Figure 4 outlines our automated pipeline for periodic model retraining and evaluation using agent feedback. It utilizes generalized LLM offline workflow (GLOW) modules, optimizing resource usage with parameter-efficient fine-tuning (PEFT) and model partitioning (Appendix F).

### (1) Data Aggregation and Filtering

Annotations collected from the four-step evaluation process are filtered using both rules-based and model-driven approaches. The rule-based method applies thresholds based on review scores and selects annotations that meet significant preference criteria. The model-driven approach employs an LLM-based virtual judge to filter annotations that exhibit low prompt adherence scores (Zheng et al., 2023). These filtering strategies mitigate data inconsistencies and hallucinations, which are critical to improving model performance (Section 4.3).

### (2) Automated Model Retraining

Retrieval, ranking, and generation models are periodically retrained using GLOW-managed Ray clusters, with automated resource handling and synchronization. Parameter-efficient fine-tuning (e.g., LoRA/QLoRA) optimizes GPU usage, and built-in monitoring ensures stable progress.

### (3) Evaluation

Retrained models go with batch inference runs on curated evaluation datasets, evaluated using both ground-truth-based evaluation and virtual judges (Appendix B) that simulate human evaluations. Performance improvements act as a

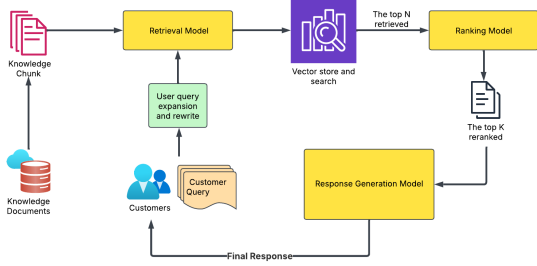


Figure 5: RAG example on knowledge document

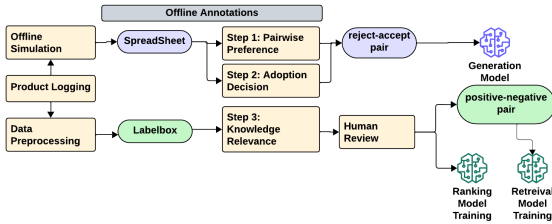


Figure 6: Offline Annotation Workflow

*proxy for annotation quality*, strengthening the cycle of enhancement.

**(4) Feedback Loop** Retrained models are deployed back into the RAG system, completing the feedback loop (Figure 5).

## 4 Experiments

Our baseline offline workflow (Figure 6) preprocesses production logs and simulation data into preference and adoption annotations using spreadsheets, and knowledge relevance annotations via Labelbox, both validated through human review. These annotations produce rejection-accept pairs for generation tasks and positive-negative pairs for retrieval and ranking models, establishing the baseline performance prior to AITL deployment. Updating models with this offline annotation pipeline took three months.

Subsequently, we deployed the AITL annotation system into a production environment with 40 agents supporting US-based customers via an asynchronous messaging channel, where SLA requires responses within hours. Over the course of the experiment, we collected annotations from more than 5,000 customer support cases. Each agent annotated approximately 11 cases daily alongside their regular customer assistance tasks, maintaining productivity levels comparable to agents not participating in annotation tasks. The primary goal of our experiment was to compare the new AITL

framework with the existing setup, evaluating its impact on annotation quality, model development cycle efficiency, and overall performance of our LLM-based customer support system.

### 4.1 Evaluation Metrics

We evaluated the effectiveness of our AITL pipeline based on annotation quality and model performance.

**Annotation Quality.** The quality of the annotations was evaluated by averaging human experts and LLM verifiers on detecting inconsistencies between the annotations of the agents and their responses to the customers (Section 3.3). Reliability was measured by agreement scores, where a higher agreement indicates fewer annotation-response conflicts. The reviewers were blinded to the agent labels to reduce bias.

**Model Performance.** We measure retrieval with **recall @ 75** (proportion of relevant documents among the top 75 given a total of ten thousand documents) and **precision @ 8** (relevance of the top eight ranked documents) as empirically optimal. (Appendix C). For generation, we evaluate:

- **Helpfulness:**

1. **Point-wise Helpfulness (Model-based):** Combines scores from a trained preference model and an LLM-based evaluation aligned with business criteria (Appendix B).
2. **Pair-wise Helpfulness (Human-based):** Human annotators perform pairwise comparisons between responses, validating model-based assessments. (outlined in online annotation step 1)

- **Citation Correctness:** Measured as Jaccard overlap between model ( $M$ ) and human ( $H$ ) cited references:  $\frac{|M \cap H|}{|M \cup H|}$ . E.g., if  $M = \{a, b\}$ ,  $H = \{a, c, d\}$ , score is  $\frac{|\{a\}|}{|\{a, b, c, d\}|} = 0.25$ .

- **Response Correctness:** Checks the factual accuracy and the adherence to the policies through the review of agents.

### 4.2 Annotation Quality Comparison

High-quality annotations drive a robust data flywheel, enhancing model performance and producing richer data for future cycles. Table 2 shows higher annotation agreement rates in the online

workflow compared to offline across three steps: preference, adoption, and knowledge relevance. Step 4 (missing knowledge) was not evaluated offline due to the limitations of the annotation tool in annotating all the potential missing knowledge.

	Offline	Online
<b>Step 1 (Preference Judgment)</b>	0.635	0.832
<b>Step 2 (Adoption Judgment)</b>	0.721	0.775
<b>Step 3 (Knowledge Relevancy)</b>	0.436	0.923

Table 2: Agreement for offline vs. online setup

### 4.3 Impact on Model Performance

We evaluated how AITL annotations and optimized fine-tuning impact retrieval-augmented generation (RAG) system performance against our baseline (Appendix D). Models were trained using a 90%/10% temporal split for training and evaluation, respectively, comparing AITL annotations to offline annotations (Steps 1-3).

**Retrieval Accuracy.** AITL annotations significantly outperform offline annotations (Table 3). Precision@8 improved from 0.357 to 0.410 (+14.8%), exceeding offline by 4.1%. Recall@75 increased from 0.634 to 0.708 (+11.7%), surpassing offline by 3.8%, highlighting the benefits of AITL system.

**Generation Quality** Applying ORPO (Hong et al., 2024) with AITL fine-tuning further improved generation quality (Table 4): Helpfulness rose from 0.658 to 0.713 (+8.4%), exceeding offline fine-tuning (0.691); Citation accuracy improved significantly from 0.097 to 0.134 (+38.1%), surpassing offline (0.112); Response correctness increased from 0.851 to 0.882 (+3.6%), higher than offline results (0.868). These results highlight the clear advantages of the AITL system over offline annotation pipelines (Appendix E).

**Human Preferences and Adoption.** Pairwise human evaluations (Fig. 7) showed that 60.12% of the fine-tuned model responses were preferred over baseline (33.32%), and 6.57% did not express preference. This improvement also increased the overall adoption rate by 4.5% compared to the baseline. These findings confirm that integrating AITL annotations with ORPO not only improves objective metrics (e.g., Precision@8 and citation correctness), but also aligns with human judgments.

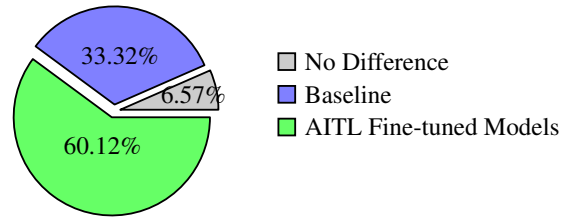


Figure 7: Human preference for end-to-end performance: baseline vs. AITL fine-tuned models.

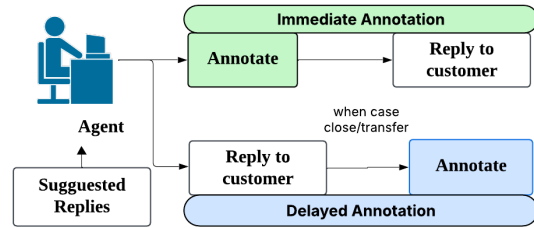


Figure 8: Immediate vs. Delayed Annotation Workflow

## 5 Learnings

### 5.1 Annotation Timing Ablation Study

A key concern is that our system design may not scale effectively to channels with stricter SLA requirements, such as live-chat. To enable annotation with higher SLA channel, we conducted a controlled experiment using the AITL tool, comparing annotations performed immediately during customer interactions (*Immediate Annotation*) with those completed after interactions ended (*Delayed Annotation*). This experiment involved approximately 2,000 cases under identical tooling conditions (Figure 8).

Results (Figure 9) indicate that immediate annotation significantly improves annotator agreement only for the *Missing-Knowledge* step (Step 4), increasing from 63.9% to 76.5% (+12 pp,  $p < 0.05$ ). Conversely, differences for Preference, Adoption, and Knowledge Feedback steps (Steps 1–3) are negligible (complete scores in Appendix G). We therefore recommend adopting a hybrid workflow: perform immediate annotation for Missing-Knowledge when SLA allows brief delays, while delaying the remaining annotation steps after replying to customers to reliably meet stringent SLA requirements.

Retrieval Model	Ranking Model	Recall@75	Precision@8
Baseline	Baseline	0.634	0.357
Offline Data Fine-tuned	Offline Data Fine-tuned	0.670	0.394
AITL Fine-tuned	AITL Fine-tuned	<b>0.708</b>	<b>0.410</b>

Table 3: Performance comparisons on AITL test set. The bolded entries indicate the highest scores.

Generation Model	Retrieval Models	Helpfulness	Citation	Response Correctness
Baseline	Baseline	0.658	0.097	0.851
Offline Fine-tuned (ORPO)	Offline Fine-tuned	0.691	0.112	0.868
AITL Fine-tuned (ORPO)	AITL Fine-tuned	<b>0.713</b>	<b>0.134</b>	<b>0.882</b>

Table 4: Performance of RAG models on the AITL test set. Bold entries indicate the highest scores.

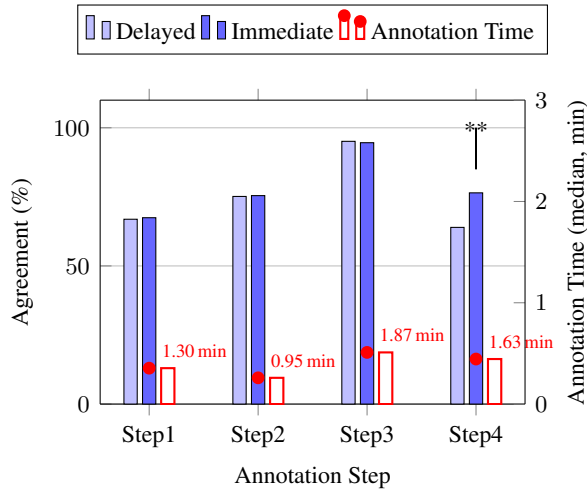


Figure 9: Hybrid agreement (LLM + human) for immediate vs. delayed annotation. Red dots show median labeling time (min). \*\* denotes a significant difference ( $p < 0.05$ ).

## 5.2 Annotation Quality Ablation Study with LLM-based filter

This subsection quantifies the contribution of LLM-based filter used in our data aggregation and filtering stage (Sect. 3.4). The functions as a quality gate that minimizes the lack of prompt adherence and flags potential inconsistencies between the annotation of human agents and their responses to customers (Sect. 3.3). We do not use LLM to originate missing-knowledge labels, which require operational nuance.

**Setup.** We evaluated two otherwise identical pipelines on the same AITL annotation batches: (i) LLM-based prompt-adherence filtering and consistency checks during data aggregation and (ii) this gate disabled. Under LLM-based filtering, 14.3% of examples are removed prior to retriever/ranker training and 34.5% prior to generator training.

Model	Metric	w/ Filtering	w/o Filtering
Retrieval	Recall@75	<b>0.708</b>	0.670
	Precision@8	0.402	0.394
Generation	Helpfulness	0.703	0.696
	Citation	<b>0.131</b>	0.112
	Response Correctness	0.880	0.880

Table 5: VJ filter ablation on identical AITL batches. The only difference is the presence/absence of VJ gating during data filtering (Sect. 3.4). Gains concentrate in retrieval recall and citation accuracy.

**Findings.** Table 5 shows that LLM-based filtering yields consistent gains in Recall@75 (+3.8 pp vs. no-filtering on the same batch) and Citation (+1.9 pp absolute), while Precision@8, Helpfulness, and Response Correctness remain statistically unchanged. This pattern aligns with the LLM’s role as a noise gate: by down-weighting low-prompt-adherence or hallucination-prone supervision during data filtering, it primarily improves retrieval and citation grounding without perturbing other signals.

**Interpretation.** As detailed in Appendix G, LLM and human evaluations are strongly correlated ( $r > 0.90$ ), enabling a cost-effective hybrid reliability score that combines both sources. A notable exception is Step 4 (*Missing-Knowledge*), which benefits uniquely from immediate human annotation (+12 pp agreement; Fig. 9). Accordingly, we position the LLM-based filtering as a validator rather than a generator for this channel, yielding a scalable quality filter while preserving human oversight where domain nuance is essential.

## 5.3 Pairwise Preference Data Effectiveness.

Fine-tuning with plus-level preference data (better or significantly better) boosts helpfulness, but low-

ers the correctness of the citation. Adding an agent adoption as a data filter restores the correctness of the citation to 11.4% and retains a gain of 3.5% in helpfulness, achieving a better balance.

Fine-tune Strategy	Helpfulness	Citation
Baseline	0.694	0.123
Plus Pref	<b>0.766</b>	0.109
Plus Pref + Adopted	0.718	<b>0.137</b>

Table 6: Performance comparison of generation models by different preference dataset

#### 5.4 Continuous Training Strategies.

Retraining the models using a mix of historical and new annotations on the previous checkpoint improves adaptability and robustness, increasing precision @ 8 by 8% in historical data and 4% in recent data compared to training only on new data. (Appendix E) Periodically integrating fresh feedback with diverse historical datasets mitigates overfitting and improves retrieval, ranking, and generation performance. Similar benefits are observed with offline annotation approaches.

#### 5.5 Cross-Model Generalization Study

We replicate AITL on Qwen2.5-32B and the Llama-3 family (3B, 8B, 70B) using the same data split and evaluator as the main study. As shown in Table 7, AITL delivers consistent gains on smaller and medium models. At 70B, where SFT baselines are already strong, AITL shows mixed helpfulness effects but improved/stable citation across variants, suggesting interaction with prior objectives rather than a lack of transfer; overall, these results indicate the AITL data flywheel generalizes across architectures and scales.

### 6 Conclusion and Future Work

We introduced *Agent-in-the-Loop* (AITL), an real-time(near real-time) data flywheel that turns routine customer support operations into continuously improving supervision for retrieval, ranking, and generation. By capturing four signal types including pairwise response preferences, agent adoption and rationales, knowledge relevance, and missing knowledge, AITL closes the gap between evaluation and production reality. Our pilot with U.S.-based agents shows consistent gains in retrieval (Recall@75, Precision@8), generation helpfulness, citation correctness, and agent adoption, while shrinking update cadence from months to weeks.

Model	AITL	Helpfulness	Citation
Qwen2.5-32B-Instruct	No	0.672	0.104
Qwen2.5-32B-Instruct	Yes	<b>0.683</b>	0.104
Llama-3.2-3B-Instruct	No	0.373	0.057
Llama-3.2-3B-Instruct	Yes	<b>0.636</b>	<b>0.061</b>
Llama-3.1-8B-Instruct	No	0.379	0.097
Llama-3.1-8B-Instruct	Yes	<b>0.606</b>	<b>0.114</b>
Llama-3.3-70B-Instruct	No	0.632	0.105
Llama-3.3-70B-Instruct	Yes	<b>0.644</b>	<b>0.122</b>

Table 7: Cross-model results with the same AITL split and evaluator.

Building on these results, we outline three directions for future work:

- **Scaling optional agent feedback.** Replace heavy labels with lightweight micro-annotations (default “skip”), use active sampling for high-uncertainty or disagreement cases, and correct selection bias via inverse-propensity weighting and post-stratification.
- **Product-embedded AITL for efficiency.** Integrate AITL into agent-facing tools; evaluate with a productivity bundle (e.g., CSAT, time-to-resolution, adoption rate, human-edit distance); and study cognitive load, trust calibration, and skill formation across novice and expert agents.
- **Toward fuller automation.** Leverage simulation and judge-based validation to automate dataset curation and preference labeling where appropriate, while preserving human oversight for safety, policy adherence, and domain nuance.

#### Limitation

Although AITL offers clear advantages, there are three key limitations.

First, prolonged use of real-time annotations could lead to increased agent workload and potential annotation fatigue. To mitigate this risk, future implementations could consider strategies such as rotating annotation responsibilities among agents, adaptive workload management, and periodic breaks from annotation duties. In addition, targeted training and incentive programs can further support annotation quality over time.

Second, our study exclusively focused on English-language customer support. This leaves open questions regarding the effectiveness and applicability of AITL in multilingual or culturally

diverse support contexts, which should be investigated in future research.

Finally, the relatively short duration of this study constrains our understanding of how annotation practices might evolve over extended periods and, crucially, how effectively they scale when applied to larger groups of agents.

## Acknowledgments

We are deeply grateful to the many colleagues who contributed to the AITL project. This work was made possible by the GLOW platform, which enabled reproducible workflows, streamlined model development, and rapid end-to-end iterations.

We extend our sincere thanks to the leadership of Airbnb Customer Support for their sponsorship and guidance. We especially thank Hossein Shams, Chloe Zhao, Gen Wang, Chandraprakash Loonker, Mengchen Liang, Jeremy Wang, Jingwen Qiang and Ying Tan for their engineering contributions.

We thank Tony Donisch and Chris Robinson for project ideation and support. We are also grateful to our colleagues in Design including Eric Fensterheim, Stacey Kennelly Nester, whose creativity and insights helped shape the project. In Platform Management, we thank Omar Siddiqui, Eliav Kahan, and Jorge Poblacion for their direction and support. Most importantly, this project could not have been completed without the partnership of our agent support collaborators; we acknowledge Lindsey Oben, David Amador, Isabel Arboleda, Chris Enzaldo, and the CS Labs team for their indispensable support. Finally, we thank the anonymous reviewers for their constructive feedback, which helped us substantially improve this paper.

## References

- Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. 2022a. [Training a helpful and harmless assistant with reinforcement learning from human feedback](#). *Preprint*, arXiv:2204.05862.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. 2022b. [Constitutional ai: Harmlessness from ai feedback](#). *Preprint*, arXiv:2212.08073.
- Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. [Scaling instruction-finetuned language models](#). *Preprint*, arXiv:2210.11416.
- Arthur Câmara, Dinos Papakostas, Mathias Parisot, Fernando Rejon Barrera, and Jakub Zavrel. 2024. [Zeta-alpha-e5-mistral](https://huggingface.co/zeta-alpha-ai/Zeta-Alpha-E5-Mistral). <https://huggingface.co/zeta-alpha-ai/Zeta-Alpha-E5-Mistral>.
- Hui Dai, Ryan Teehan, and Mengye Ren. 2025. [Are LLMs prescient? a continuous evaluation using daily news as the oracle](#). In *Forty-second International Conference on Machine Learning*.
- Jiwoo Hong, Noah Lee, and James Thorne. 2024. [Orpo: Monolithic preference optimization without reference model](#). *Preprint*, arXiv:2403.07691.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Th ophile Gervet, Thibaut Lavril, Thomas Wang, Timoth e Lacroix, and William El Sayed. 2024. [Mistral of experts](#). *Preprint*, arXiv:2401.04088.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K ttler, Mike Lewis, Wen-tau Yih, Tim Rock-t schel, Sebastian Riedel, and Douwe Kiela. 2020. [Retrieval-augmented generation for knowledge-intensive NLP tasks](#). *CoRR*, abs/2005.11401.
- Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Qingwei Lin, Jianguang Lou, Shifeng Chen, Yansong Tang, and Weizhu Chen. 2024. [Arena learning: Build data](#)



flywheel for llms post-training via simulated chatbot arena. *Preprint*, arXiv:2407.10627.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022b. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.

Shariq Shah, Hossein Ghomeshi, Edlira Vakaj, Emmett Cooper, and Shereen Fouad. 2023. [A review of natural language processing in contact centre automation](#). *Pattern Analysis and Applications*, 26(3):823–846.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. [Learning to summarize from human feedback](#). *CoRR*, abs/2009.01325.

Hanchen Su, Wei Luo, Yashar Mehdad, Wei Han, Elaine Liu, Wayne Zhang, Mia Zhao, and Joy Zhang. 2025. [LLM-friendly knowledge representation for customer support](#). In *Proceedings of the 31st International Conference on Computational Linguistics: Industry Track*, pages 496–504, Abu Dhabi, UAE. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *arXiv preprint arXiv:2306.05685*.

## A Annotation Error Types and LLM Verifier Prompt

We identify common annotation error types, each illustrated with concise examples:

- **Preference Mismatch:** Occurs when an agent explicitly prefers response A but ultimately selects response B.
- **Adoption Mismatch:** Occurs when an agent rejects a response due to specific formatting issues but subsequently uses the same formatting in their final reply.

- **Incorrect Knowledge Annotation:** Occurs when the agent provides guidance based on certain policies (e.g., advising customer cancellation) but inaccurately labels relevant policy documents as irrelevant.
- **Omitted Missing Knowledge:** Occurs when the agent mentions essential details (e.g., refund amount) to the customer but fails to annotate this critical contextual information.

Below is a refined example prompt template utilized by the LLM-based verifier for missing knowledge object verification:

**Task:** Evaluate whether the provided knowledge objects annotated by agents sufficiently address the customer’s issue described in the conversation.

Follow these instructions step by step:

- Read and understand the customer’s issue from the provided conversation and contextual information.
- Carefully review the agent’s annotated knowledge objects and their response to the customer.
- Determine if the annotated knowledge items adequately and directly resolve or address the customer’s issue.

Provide your evaluation strictly in the following JSON format:

```
{ "missing_agreement_reason": "Provide a concise yet specific reason why the knowledge objects are sufficient or insufficient.", "missing_agreement": 1 or 0 // 1 if knowledge provided is insufficient, 0 if sufficient }
```

**Conversation:**

[Customer]: what customer said

[Agent]: what agent said

[Customer]: what customer said

**Agent’s Response to Customer:**

agent’s final response provided here

**Provided Knowledge Objects Annotated by Agents:**

1. Knowledge item 1 here

2. Knowledge item 2 here

## B Helpfulness Metric Definition

This section will describe the training process for helpfulness models.

We define *point-wise helpfulness score* to measure the likelihood that internal domain experts (e.g. customer support agents) would prefer a particular response. This score is generated through an ensemble method that combines two distinct approaches:

1. **Reference Evaluator Model:** We employ an internally developed evaluation model based on Mistral-7B, fine-tuned using human-annotated preference data via a pairwise loss objective. This model serves as our reference evaluator, achieving an approximate agreement rate of 80% compared to the ground-truth annotations provided by the expert groups.

2. **GPT-4 Prompt-based Evaluations:** We perform multiple GPT-4 evaluations using carefully designed prompts that embody internal helpfulness criteria. Each evaluation produces a binary indicator (helpful or not), and we sum these indicators across seven distinct prompts to yield an integer score ranging from 0 to 7. By averaging these indicators, we obtain a final continuous helpfulness score. This prompt-based mechanism eliminates the need for ground-truth labels during inference. Evaluation against expert-labeled ground truth confirms an agreement rate of approximately 80%, validating the reliability of this approach.

## C Retrieval and Ranking Metrics

We empirically selected Recall@75 and Precision@8 as our primary retrieval and ranking metrics, guided by operational constraints and experimental insights. Recall@75 evaluates the effectiveness of our retrieval system in identifying relevant articles within the top 75 retrieved candidates. Empirical analysis showed that the recall increased consistently with more retrieved candidates, reaching approximately 70%-80% at topN=75. Beyond 75 candidates, recall improvements plateaued, indicating limited benefits from retrieving additional documents.

Precision@8 was chosen based on practical limitations, as production environments restrict input context lengths for downstream generation models. Our experiments indicated that reranker performance notably surpassed retrieval-only methods from topN=5 onward, with topN=8 providing the optimal trade-off between performance enhance-

ment and operational feasibility. Increasing the number of retrieved snippets beyond 8 led to reduced helpfulness scores due to excessive information dilution. Thus, Precision@8 effectively balances input quality for response generation with practical system constraints.

## D Prior AITL Baseline Details

This appendix details the baseline Retrieval-Augmented Generation (RAG) system, outlining the three core models: generation, retrieval, and ranking.

**Generation Model** Our generation component utilizes an 8×7B Mistral Mixture-of-Experts (MoE) model (Jiang et al., 2024). This model is initially fine-tuned via supervised fine-tuning (SFT) (Ouyang et al., 2022b) on offline human-agent annotations. Subsequently, we apply Odds Ratio Preference Optimization (ORPO) (Hong et al., 2024) to align the model’s generation with agent-approved preferences.

**Retrieval Model** Retrieval is performed by fine-tuning a pre-trained Zeta-Alpha-E5-Mistral 7b embedding model (Cámara et al., 2024). This fine-tuned model transforms article chunks into 1024-dimensional vector embeddings. When a user query is received, it is encoded, and the top- $N$  relevant article chunks are retrieved based on cosine similarity.

**Ranking Model** After retrieval, a ranking model refines the top- $N$  retrieved article chunks to identify the most relevant and helpful content. This ranking is accomplished using an in-house fine-tuned FLAN-T5 model (Chung et al., 2022), adapted for a ranking task. The model takes the user query and each retrieved article chunk’s details as input, and outputs a relevance score, which is then used to rank the chunks.

## E AITL Training Details

This appendix details the post-training processes and parameters for each component of our Retrieval-Augmented Generation (RAG) baseline: the Generation Model, Retrieval Model, and Ranking Model.

### E.1 Generation Model

Fine-tuning commenced from the baseline checkpoint of the 8×7B Mistral Mixture-of-Experts (MoE) model. The model was optimized via Odds Ratio Preference Optimization (ORPO), using a

balanced dataset comprising equal parts newly annotated examples and previously collected training data. Training was conducted for a total of 3 epochs with a batch size of 64, employing an initial learning rate of  $2 \times 10^{-5}$ , which linearly decayed after a warm-up period covering 5% of total training steps. The fine-tuning process utilized 4 NVIDIA A100 GPUs, each equipped with 80GB of memory. After training, the model underwent 4-bit Quantized Low-Rank Adaptation (QLoRA) to substantially decrease computational overhead and memory consumption during inference. Subsequently, the LoRA weights were merged back into the original model weights, ensuring efficient deployment. Parameter selection and alignment loss criteria were informed by comparisons made during previous model iterations.

- **Training:**

- *ORPO Phase:* Incorporates preference signals through pairwise feedback. The model learns to prioritize more 'helpful' or relevant outputs based on labeled data.

- **Observations:**

- ORPO fine-tuning often yields higher coherence and factual correctness than SFT alone.
- We observe slight divergences between purely offline vs. online-labeled preference data through AITL, motivating continuous updates.

## E.2 Retrieval Model

Following the generation model's optimization, we proceed to detail the training process for the retrieval model of our RAG system.

- **Training:**

- *Fine-tuning Methodology:* The Zeta-Alpha-E5-Mistral 7b model underwent fine-tuning employing MultipleNegativesRankingLoss. Positive training samples were comprised of article segments marked as "RELEVANT" or "HELPFUL," supplemented by agent-generated content. Negative samples consisted of both "NOT RELEVANT" or "NOT HELPFUL" labeled chunks (hard negatives) and randomly sampled chunks from the same batch (easy negatives).

- *Training Setup:* Training parameters included a learning rate of  $2 \times 10^{-5}$  and a weight decay of  $2 \times 10^{-6}$ . The training was conducted for 1 epoch, with a training batch size of 1. The model was executed on an A100 GPU cluster consisting of 4 GPUs.

- **Observations:**

- The retrieval model's accuracy benefited from increased training dataset size and demonstrated resilience to data noise. Notably, optimal performance was observed when the model was trained on a combination of high-confidence (annotator and reviewer alignment) and low-confidence (annotator and reviewer disagreement) datasets.
- For RAG in live support conversations, the retrieval models fine-tuned using AITL data exhibited substantially superior performance compared to the baseline, and those fine-tuned using offline datasets.

## E.3 Ranking Model

With the retrieval model established, we now turn our attention to the ranking model, which refines the retrieved results.

- **Training:**

- *SFT:* Training data consisted of positive and negative examples based on agent annotations. Positive examples were constructed from article chunks labeled "RELEVANT" or "HELPFUL," along with agent-generated content. Negative examples comprised chunks labeled "NOT RELEVANT" or "NOT HELPFUL." Each training instance was formatted as **<prompt + user query + article chunk details>** as input, with **<relevant>** as the target for positive examples and **<not\_relevant>** as the target for negative examples. During inference, the relevance score is determined by the probability of the output being "relevant".
- *Training Setup:* The model was trained for 3 epochs on a single A10 GPU, using a learning rate of  $1e - 5$ , a batch size of 16, and gradient accumulation steps of 64.

- **Observations:**

- The ranking model’s performance was optimized when trained on high-confidence annotated datasets, with a balanced 1:2 ratio of historical and newly generated AITL data.
- Utilizing AITL data for fine-tuning ranking models yielded significantly better performance in live support RAG scenarios compared to offline datasets, and also surpassed the performance of models trained only on historical data.

## F Training Experiment Efficiency

To improve the efficiency of offline experiments, we introduced a framework for the generalized offline LLM workflow (GLOW) based on reusable and parameterized fine-tuning, batch scoring and evaluation components. The two main areas of focus for GLOW that benefit this research study include infra-aware LLM developments and integration with template end-to-end workflow.

**Infra-aware LLM Developments** LLM computations are very sensitive to the underlying infrastructure, and there are multiple hyperparameters that can drastically affect the computation capacity:

- Compute optimization: whether to apply lower precision training, Parameter-Efficient Fine Tuning (PEFT) adaptors and model partitioning strategies like DeepSpeed or FSDP
- Input dataset size, batch size and context length
- Model capacity and architecture

A typical LLM offline task that has a specific set of hyperparams must be deployed to a matching infrastructure to avoid failed tasks due to insufficient GPU VRAM, or overprovisioning that led to low GPU utilization rate. To solve this problem, GLOW in its configuration is integrated with the Ray Cluster spec section that combines the LLM task with ephemeral compute cluster provisioned on-demand. This setup not only improves the offline experiment task stability, but also guarantees reproducible results across multiple experiment runs.

**Templated end-to-end Workflow** GLOW offers reusable offline workflow building components for end users to customize their developments, and this

Annotation Step	Immediate (%)		Delayed (%)	
	LLM	Human	LLM	Human
Step 1	62.6	72.3	61.0	72.8
Step 2	76.7	74.2	75.1	75.2
Step 3	91.7	97.5	91.7	98.5
Step 4	76.1	76.8	57.6**	70.3**

Table 8: Annotation accuracy for LLM and human raters. Significance marker \*\*: difference between immediate and delayed is significant ( $p < 0.05$  for humans;  $\Delta > 5$  pp for LLM).

unified API improves production readiness while largely reducing prototype development cycles to production.

## G Immediate and Delayed Comparison

In practice we average the human and LLM scores to form a hybrid metric. Because the two sources correlate strongly ( $r > 0.90$ ; Table 8), this composite score inherits human-level reliability while remaining inexpensive to scale. All subsequent analyses, including the immediate vs. deferred comparison in Figure 9, are therefore reported on this hybrid metric.

Step	Q1	Median	Q3	Mean	Trim. Mean
Step 1	0.583	<b>1.30</b>	2.967	4.017	2.174
Step 2	0.483	<b>0.95</b>	2.733	3.711	2.065
Step 3	1.317	<b>1.87</b>	3.867	4.647	3.332
Step 4	0.721	<b>1.63</b>	3.317	3.409	2.430

Table 9: Annotation-time statistics per step (minutes).

Table 9 shows a pronounced right skew: Means and even 10 % trimmed means sit well above the medians of 1–2 min. Hence, we quote the median as the best indicator of agent annotation time effort, with the trimmed mean offered as a reference.