# RL-ADA: Co-Evolutionary Adversarial Training for Self-Improving Customer Support Agents Without Human Feedback

Ram Narayanan
Centific
ram.sampath@centific.com

Harshit Rajgarhia
Centific
harshit.rajgarhia@centific.com

Abhishek Mukherji
Centific
abhishek.mukherji@centific.com

## Abstract

Task-oriented dialogue agents require robust tool routing to handle unpredictable user interactions, but training them to withstand adversarial inputs is bottlenecked by the high cost of human preference annotation. Standard self-play methods are poorly suited for the inherently asymmetric nature of customer-agent dialogues. We present RL-ADA, a framework for asymmetric co-evolutionary adversarial training that replaces human labels with world feedback: consequence-based signals derived directly from interaction outcomes. A Customer Support Agent (DA, 3B parameters) and an Adversarial Customer Agent (CA, 7B parameters) train against each other in an adversarial arena guided by a fixed automated judge, with an isolation gym selectively retraining the weaker agent on prior-failure transcripts. The size asymmetry is intentional: the CA faces the harder generative task of producing contextually misleading natural language to elicit misroutes from a smaller, task-focused DA. Training converges autonomously under a win-rate stopping criterion, with the framework requiring no human annotation at any stage. We observe a non-monotonic win-rate trajectory consistent with genuine co-evolutionary pressure, and the emergence of adversarial **Contextual Camouflage**: the CA learns to embed intent within dense, realistic customer detail rather than expressing it directly, a strategy that arises purely from reward pressure without explicit specification.

## Keywords

reinforcement learning, adversarial training, co-evolutionary learning, task-oriented dialogue, world feedback, tool routing, asymmetric self-play, automated evaluation

## 1 Introduction

Task-oriented dialogue agents are increasingly deployed in enterprise settings where they must handle unpredictable, adversarial, and continuously evolving user inputs. Training such agents to be robust requires large volumes of labelled interaction data, yet enterprise conversational logs are privacy-sensitive, domain-specific, and expensive to annotate. The result is a fundamental bottleneck: agents that perform well on evaluation benchmarks but degrade under the adaptive pressure of real users.

Existing approaches only partially address this bottleneck. RLHF [12] reduces agent training to a human-annotation problem that is costly and difficult to scale as user behaviour evolves. Self-play methods [4, 15] avoid annotation by learning from interaction outcomes, but assume symmetric agents with identical action spaces and train both simultaneously, inducing non-stationarity and risking catastrophic forgetting. Neither approach directly addresses the *asymmetric, annotation-free* case native to customer support: a support agent that must correctly route tool calls versus an adversarial customer that elicits misroutes through natural language, each requiring different model capacities, action spaces, and reward structures.

We present **RL-ADA** (Reinforcement Learning with Adversarial Dialogue Agents), a framework that replaces human labels with *world feedback*: consequence-based reward signals derived directly from measurable interaction outcomes. A 3B **Customer Support Agent** (DA) and a 7B **Adversarial Customer Agent** (CA), both implemented as language model policies, train against each other without any human annotation at any stage.

*Contributions.*

- A **co-evolutionary training loop** in which a 3B DA language model and a 7B CA language model train against each other using *world feedback* only: rule-based outcome signals combined with a fixed automated judge whose reliability we evaluate against GPT-4o-mini, with no human annotation at any stage (§3, §3.1).
- An **isolation gym** that retrains the weaker agent on a 70:30 failure/success transcript mix, producing non-monotonic co-evolutionary dynamics (§3.5, §5.1).
- A **macro-level stopping criterion** for asymmetric adversarial training based on rolling win-rate stability, framed as a practical surrogate for empirical $\varepsilon$-Nash convergence (§3.6).

*Instantiation.* We instantiate RL-ADA on a banking customer support setting where the DA must route free-text customer utterances to one of six API tools across 78 distinct customer intents (Appendix A). Customers never name their intent directly: *"I see something odd on my statement"* could require `dispute_charge` or `get_transactions`, and an adversarial customer can phrase requests to trigger the wrong tool. The DA is not an intent classifier and is trained only from call outcomes with no human labels.

## 2 Related Work

*RLHF and AI feedback.* Ouyang et al. [12] establish human preference feedback as the dominant alignment paradigm; Bai et al. [1] partially displace it via RLAIF. RL-ADA replaces both with structured episode scores from interaction outcomes, eliminating annotation dependency at the cost of judge reliability uncertainty (§3.1).

*Self-play and game-theoretic RL.* AlphaGo [15], OpenAI Five [4], and AlphaStar [16] demonstrate symmetric self-play at scale; PSRO [8, 9] formalises iterative best-response. Customer-agent dialogue is

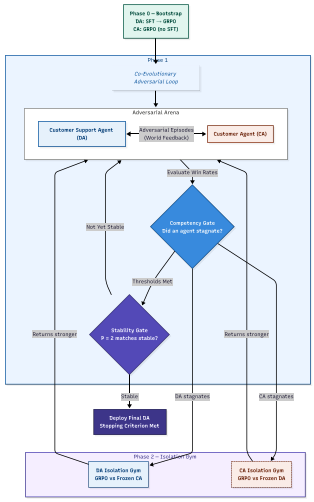


**Figure 1: The RL-ADA three-phase loop. Bootstrap initialises both agents; the Arena measures relative win rates; the Isolation Gym retrains the weaker agent on failure transcripts.**

asymmetric by construction; simultaneous co-training under asymmetry induces cycling dynamics [2, 10] that RL-ADA's freeze-one-train-one structure avoids.

*LLM red-teaming and concurrent work.* Wallace et al. [17] and Perez et al. [13] establish automated adversarial test generation against fixed targets; ARLAS [18] and SPAG [7] apply adversarial self-play to LLM safety and reasoning. RL-ADA differs in that the Adversarial CA co-evolves with the DA rather than attacking a fixed model, targets tool-routing correctness in asymmetric dialogue, and uses asymmetric model sizes (3B vs. 7B) under an explicit convergence criterion.

*Task-oriented dialogue evaluation.* Yao et al. [19] and Barres et al. [3] establish tool-agent-user evaluation benchmarks with simulated cooperative users. RL-ADA differs in that the CA is adversarially trained to cause failures rather than simulate realistic task completion.

## 3 System Architecture

RL-ADA uses three model roles: a **Customer Support Agent** (referred to as DA throughout), an **Adversarial Customer Agent** (CA), and a **Judge**. The DA (Qwen2.5 3B) manages multi-turn clarification, routes tool calls, and ends calls; the CA (Qwen2.5 7B) is an RL-trained language model policy that generates customer-style utterances designed to elicit misroutes; the Judge (Qwen2.5 7B) provides terminal episode quality scores. Training proceeds across three phases (Figure 1).

### 3.1 Automated Judge

Replacing human annotation requires a reliable automated signal for episode quality. Rather than using a proprietary model such as GPT-4o-mini as the judge, which would introduce an external dependency into the training loop, we use a Qwen2.5-7B model (NeutralJudge) as a fixed episode scorer. The key question is whether a local 7B model is sufficiently reliable for this task: reliable enough in resolution detection, quality ranking, and hallucination detection to serve as the sole training signal without systematic bias that would corrupt the reward. We evaluate NeutralJudge against GPT-4o-mini as a reference point on $n$=38 conversations (8 synthetic with tier labels + 30 ABCD [6]; tier Spearman on the 8 synthetic only). NeutralJudge matches GPT-4o-mini in resolution detection (F1 0.807 vs. 0.821), ranks quality tiers in the same order ($\rho$=0.833), and is well-calibrated for relative episode ranking. Hallucination detection is particularly critical: a judge that misses fabricated facts would corrupt the reward signal, making F1 of 1.000 a hard prerequisite for safe RL training (Table 1).

**Table 1: NeutralJudge (7B) vs. GPT-4o-mini ($n$=38).**

| Metric | Value |
|---|---|
| Resolution F1 (NeutralJudge / GPT-4o-mini) | 0.807 / 0.821 |
| Quality-tier rank Spearman ($n$=8 synth.) | 0.833 |
| Cross-judge overall Pearson | 0.779 |
| Score calibration bias | +0.50 |
| Hallucination detection F1 | 1.000 |

### 3.2 Phase 0 – Bootstrap

The DA is first trained via supervised fine-tuning on tool-routing demonstrations derived from Banking77 [5], then refined with GRPO [14] using automated judge reward. The CA receives no SFT initialisation; it learns to produce misleading customer utterances from reward pressure against a fixed DA checkpoint.

The DA operates in a *partially observable environment* implemented as an OpenEnv [11] CustomerEnvironment: at each turn it receives only the conversation history and must decide whether to call a tool, speak, or end the call, without ever observing the CA's hidden intent. The DA reasons explicitly before each action via a structured `thinking` field in its JSON output, performing implicit intent inference as part of its chain-of-thought before committing to a tool call. The 78 customer *intent* strings collapse to 6 tools via a many-to-one map (Appendix A): for example, `unrecognized_charge` and `refund_not_showing_up` both route to `dispute_charge`. The reward is on the *outcome*, not on any intermediate intent label, so the system requires no labelled annotations at training time.

### 3.3 Reward Design

The total DA reward per GRPO completion is $r_{\text{DA}} = r_{\text{format}} + r_{\text{tool}} + r_{\text{env}} + r_{\text{end}}$, summarised in Table 2. $r_{\text{env}}$ is the return from a full CustomerEnvironment forward rollout combining turn-level rule-based signals with a terminal judge score clipped to $[-2, +2]$. Procedural failures (missed identity verification or missing domain tool) trigger mandatory negative deductions ($-1.2$ and $-1.8$ respectively), ensuring the signal is negative whenever the DA fails either required step regardless of other scores.

The CA reward is $r_{\text{CA}} = r_{\text{inv-DA}} + r_{\text{conceal}} + r_{\text{format}}$, where $r_{\text{inv-DA}}$ rewards misroutes and penalises correct DA routing, and $r_{\text{conceal}}$ penalises utterances that directly name the intent. The dense turn-level shaping terms bootstrap the action space; the terminal judge $J$ is the world-feedback signal that drives policy improvement once early-training mode collapse is prevented.[1]

[1]Without shaping, the policy collapses to trivially safe outputs (e.g. always `transfer_to_human`) before $J$ can provide meaningful gradient.

**Table 2: DA reward components.**

| Term | Condition | Value |
|---|---|---|
| $r_{\text{format}}$ | Valid JSON + thinking field | +0.5 |
| | Unparseable JSON | −3.0 |
| $r_{\text{tool}}$ | Correct tool for intent | +1.0 |
| | Wrong / hallucinated tool | −1.0 / −1.5 |
| $r_{\text{env}}$ | `lookup_account` first | +0.5 |
| | Correct domain tool (verified) | +2.0 |
| | speak per turn | −0.05 |
| | Milestone bonus | $\leq$ +1.05 |
| | Late-episode penalty ($s > 7$) | $-0.12(s-7)^{1.4}$ |
| | Terminal judge $J$ | $\in [-2, +2]$ |
| $r_{\text{end}}$ | End-call after full sequence | +1.0 |

## 3.4 Adversarial Arena

Each arena match pits the current DA checkpoint against the current CA checkpoint, the most recently trained version of each agent. Each match runs $N$=18 episodes per scenario (10 scenarios, 180 episodes per match). $W_{\text{DA}}$ is computed over CA-winnable scenarios; safety-critical fraud-escalation scenarios are tracked separately as a reference check. At the end of each match the stopping criterion (§3.6) is evaluated; if training continues, the match outcome determines which agent enters the Isolation Gym (§3.5), where it retrains against a frozen opponent before the next match begins.

## 3.5 Isolation Gym

If the stopping criterion has not fired, the weaker agent enters the Isolation Gym while its opponent's weights remain frozen. The gym constructs a training set from recent arena transcripts using a 70:30 failure/success mix, targeting failure modes from the most recent match while retaining enough successes to prevent catastrophic forgetting. A sliding-window slicer walks each transcript turn by turn: at every agent decision point, all prior dialogue accumulates as a prompt, yielding one GRPO training sample per turn depth.

The DA and CA gyms share this structure but differ in three ways reflecting their asymmetric action spaces: (i) $K$=16 generations per step for the DA versus $K$=8 for the CA, since structured JSON output requires greater rollout diversity; (ii) context and completion lengths are halved for the CA, whose utterances are 1–2 natural-language sentences rather than structured JSON; and (iii) reward functions differ entirely, with the CA gym optimising $r_{\text{realism}} + r_{\text{conceal}} + r_{\text{inv-DA}}$ rather than the DA's tool-routing rewards. Full hyperparameters are in Appendix B.

## 3.6 Stopping Criterion

The stopping criterion is evaluated at the end of each arena match, using the same frozen opponent checkpoints that the match was played against. Training terminates when both a competency gate ($W_{\text{DA}}^{(t)} \geq \tau_{\text{DA}}$=0.60) and a stability gate ($|W_{\text{DA}}^{(t)} - W_{\text{DA}}^{(t-1)}| \leq \varepsilon$=0.05) pass for $P$=2 consecutive matches after $n_{\min}$=3, or when $n_{\max}$=8 matches are exhausted. The win-rate convergence criterion is an empirical surrogate for exploitability (NashConv) [8]: $W_{\text{DA}}$ against a frozen opponent approximates a one-shot best-response value, and we use its inter-match stability as a practical proxy where exact best-response computation is intractable.

**Table 3: Arena match history. $W_{\text{DA}}$ is the mean over 2 independent runs; differences $<$ ∼0.10 are directional only.**

| M | DA | CA | $W_{\text{DA}}$ | $\sigma$ | $\delta$ | Gym |
|---|---|---|---|---|---|---|
| 1 | $\text{DA}_0$ | $\text{CA}_0$ | 0.59 | 0.08 | — | — |
| 2 | $\text{DA}_1$ | $\text{CA}_0$ | 0.68 | 0.08 | +0.09 | DA |
| 3 | $\text{DA}_1$ | $\text{CA}_1$ | 0.56 | 0.08 | −0.12 | CA |
| 4 | $\text{DA}_2$ | $\text{CA}_1$ | 0.64 | 0.08 | +0.08 | DA |
| 5 | $\text{DA}_2$ | $\text{CA}_2$ | 0.62 | 0.08 | −0.02 | CA |

# 4 Experiments

## 4.1 Setup

All training uses 4-bit LoRA on Qwen2.5-3B (DA) and Qwen2.5-7B (CA and Judge); full hyperparameters and training rationale are in Appendix B. Arena matches used 18 episodes per scenario across 10 scenarios, averaged over 2 runs; held-out evaluation uses 12 fixed scenarios.

We use the following checkpoint notation throughout. $\text{DA}_0$ is the baseline DA produced by SFT on Banking77 demonstrations followed by GRPO warm-up with automated judge reward; it enters the first arena match without any isolation gym training. $\text{CA}_0$ is the baseline CA trained from reward pressure against the fixed $\text{DA}_0$ checkpoint, with no SFT initialisation. Subsequent checkpoints $\text{DA}_1$, $\text{DA}_2$ and $\text{CA}_1$, $\text{CA}_2$ are produced by successive isolation gym cycles: each subscript increment represents one gym cycle completed by that agent.

# 5 Results

## 5.1 Arena Progression

Table 3 and Figure 2 show $W_{\text{DA}}$ across five matches. The trajectory is non-monotonic: $W_{\text{DA}}$ drops at matches 3 and 5, each time after the CA has completed a gym cycle, then recovers when the DA retrains. This pattern, visible in Figure 2, is consistent with genuine co-evolutionary pressure rather than independent improvement by each agent, though the small number of matches and high per-match variance (±8pp) limit how strongly this can be interpreted. The stopping criterion fires at match 5 ($|\delta|$=0.02 $\leq \varepsilon$, $P$=2), with three of the eight maximum cycles unused. Per-scenario breakdown is in Appendix E.

## 5.2 DA Improvement on Held-out Evaluation

Table 4 compares $\text{DA}_0$ and $\text{DA}_2$ on the fixed held-out set ($n$=12; treat as indicative). PASS requires correct tool routing, `lookup_account` called first, $r_{\text{DA}} \geq 2.0$, and a clean ending.

After five co-evolutionary cycles, all routing errors are eliminated: $\text{DA}_2$ selects the correct tool on every held-out scenario, up from 75% in $\text{DA}_0$, and the strict PASS rate rises from 25% to 50%. These gains occur without labelled data; the sole training signal is automated arena reward. Although the aggregate FAIL rate is unchanged at 33%, the failure mode shifts entirely: $\text{DA}_0$ fails through routing errors (calling `transfer_to_human` on dispute and transfer scenarios), while $\text{DA}_2$ routes correctly on all scenarios but fails on procedural and conversation-quality criteria. Routing failures are resolved; remaining procedural failures are addressable with additional gym cycles targeting conversation-quality criteria.

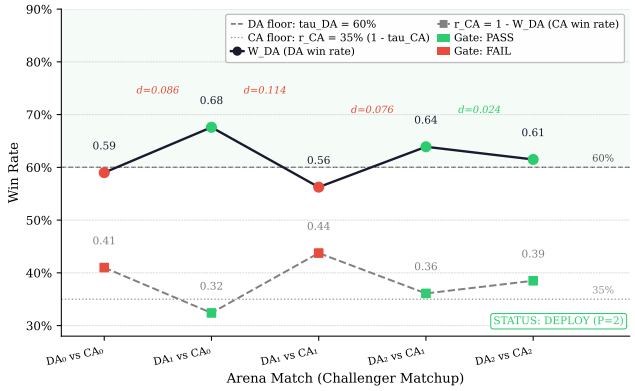


**Figure 2: DA and CA win rates across five co-evolutionary arena matches. $W_{DA}$ drops at matches 3 and 5 each time the CA completes a gym cycle, consistent with adversarial co-evolutionary pressure. Dashed reference lines show competency floors ($\tau_{DA}$=0.60, $r_{CA}$=0.35).**

**Table 4: Held-out evaluation: $DA_0$ (baseline) vs. $DA_2$ (final, after 5 co-evolutionary cycles). 12 fixed scenarios, averaged over 2 runs.**

| Metric | $DA_0$ | $DA_2$ | Δ |
|---|---|---|---|
| Tool-routing accuracy | 75% | 100% | **+25pp** |
| PASS rate (strict) | 25% | 50% | **+25pp** |
| FAIL rate | 33% | 33% | 0 |
| Avg episode reward | +1.58 | +2.16 | **+0.58** |
| Lookup-first rate | 58% | 83% | **+25pp** |

**Table 5: CA opening-utterance tactic distribution, manual classification, $n$≈50 per match. M4 omitted (same CA checkpoint as M3).**

| Tactic | M1 | M2 | M3 | M5 |
|---|---|---|---|---|
| Direct intent | 25 | 34 | 37 | 34 |
| Vague/indirect | 25 | 16 | 13 | 16 |
| Emotional | 2 | 4 | 2 | 2 |
| Role reversal | 0 | 0 | 0 | 0 |

### 5.3 Emergent Behaviour: Contextual Camouflage

We classify the CA's opening utterance via manual inspection of sampled transcripts across $n$≈50 episodes per match (Table 5). The primary emergent behaviour we observe is what we term **Contextual Camouflage**: trained CA models learn to embed the true intent within dense, realistic customer detail, citing specific merchant names, transaction amounts, and incident context.

This is *distinct from vagueness*: Direct Intent increases (25→34→37) while Vague/Indirect decreases (25→16→13). The CA is not becoming more vague; it is becoming more *specifically misleading*, naming the correct domain but layering it in contextual noise that forces the DA to arbitrate between competing signals simultaneously. This behaviour emerges purely from the reward signal of maximising DA misroutes, without any explicit specification. We treat this as a qualitative observation; future work will measure utterance specificity directly via named-entity density or opening-turn token length.

## 6 Limitations

- **Single-chain reward.** The arena scores whether the first domain tool call is correct; real customer calls often require ordered tool chains where all steps must succeed. Extending the reward to full ordered sequences is left to future work.
- **Evaluation scope.** All experiments are in a single banking domain with a small number of matches and held-out scenarios; cross-domain generalization and finer-grained statistical separation are not established. Expanding evaluation scope is ongoing.
- **Ablations and sensitivity.** Relative contributions of reward components are unquantified; stopping thresholds $\tau_{DA}$, $\varepsilon$, $P$ and gym hyperparameters were chosen pragmatically; systematic sweeps and reward ablations are left to future work.
- **Role-reversal not observed.** A stronger CA might learn to impersonate a bank agent; this did not emerge, likely requiring longer gym cycles than the 80 GRPO steps used here.

## 7 Conclusion

RL-ADA trains a 3B Customer Support Agent against a 7B Adversarial Customer Agent using only world feedback (measurable conversation outcomes) with no human annotation. Over five co-evolutionary cycles, all routing errors are eliminated on a 12-scenario held-out benchmark, the strict PASS rate doubles from 25% to 50%, and average episode reward rises from +1.58 to +2.16, driven solely by automated arena reward with no labelled data. A macro-level stopping criterion based on win-rate stability identifies convergence at cycle 5, with three of eight maximum cycles unused. Two findings stand out beyond the headline metrics. First, the arena win-rate trajectory is non-monotonic: DA performance drops whenever the CA completes an isolation gym cycle, consistent with genuine adversarial co-evolutionary pressure. Second, the trained CA develops *Contextual Camouflage*, embedding intent within dense, specific customer detail rather than naming it directly; this adversarial behaviour emerged from reward pressure alone, without explicit specification.

These are preliminary results on a single banking domain. RL-ADA offers a concrete instantiation of the world-feedback paradigm for task-oriented dialogue where annotation is costly and adversarial robustness is required; the framework's components are domain-agnostic and applicable to any setting where interaction outcomes are measurable.

## References

[1] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislaus Fort, Deep Ganguli, Tom Henighan, et al. 2022. Constitutional AI: Harmlessness from AI Feedback. *arXiv preprint arXiv:2212.08073* (2022). Introduces RLAIF: AI-generated preference labels from a principle set, scaling reward signal without human annotation..

[2] David Balduzzi, Sébastien Racanière, James Martens, Jakob Foerster, Karl Tuyls, and Thore Graepel. 2018. The Mechanics of n-Player Differentiable Games. In

*Proceedings of the 35th International Conference on Machine Learning*. 354–363. Also appeared as: arXiv:1802.05642.

[3] Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. 2025. $\tau^2$-Bench: Evaluating Conversational Agents in a Dual-Control Environment. arXiv:2506.07982

[4] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer, Shariq Hashme, Chris Hesse, et al. 2019. Dota 2 with Large Scale Deep Reinforcement Learning. *arXiv preprint arXiv:1912.06680* (2019).

[5] Iñigo Casanueva, Tadas Temčinas, Daniela Gerz, Matthew Henderson, and Ivan Vulić. 2020. Efficient Intent Detection with Dual Sentence Encoders. In *Proceedings of the 2nd Workshop on Natural Language Processing for Conversational AI*. 38–45.

[6] Derek Chen, Howard Chen, Yi Yang, Alexander Lin, and Zhou Yu. 2021. ABCD: A Goal-Oriented Compendium of Behaviours for Dialogue. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 52–64. arXiv:2104.00783. Customer-service dialogue dataset with action-labels and resolution ground truth..

[7] Pengyu Chen, Shibo Luo, Bowen Zhang, Zhidi Liu, Jingcheng Liu, Xipeng Chen, and Wei Li. 2024. Self-Playing Adversarial Language Game Enhances LLM Reasoning. In *Advances in Neural Information Processing Systems*, Vol. 37. arXiv:2404.10642. Adversarial two-agent word game; capability gains transfer to general reasoning benchmarks without human labels..

[8] Marc Lanctot, Vinicius Zambaldi, Audrunas Gruslys, Angeliki Lazaridou, Karl Tuyls, Julien Pérolat, David Silver, and Thore Graepel. 2017. A Unified Game-Theoretic Approach to Multiagent Reinforcement Learning. In *Advances in Neural Information Processing Systems*, Vol. 30. arXiv:1711.00832.

[9] H. Brendan McMahan, Geoffrey J. Gordon, and Avrim Blum. 2003. Planning in the Presence of Cost Functions Controlled by an Adversary. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*. 536–543.

[10] Panayotis Mertikopoulos, Christos Papadimitriou, and Georgios Piliouras. 2018. Cycles in Adversarial Regularized Learning. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 2703–2717. arXiv:1709.02738.

[11] meta-pytorch. 2026. *OpenEnv: An Interface Library for RL Post-Training with Environments*. https://github.com/meta-pytorch/OpenEnv BSD-3-Clause License.

[12] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training Language Models to Follow Instructions with Human Feedback. *Advances in Neural Information Processing Systems* 35 (2022), 27730–27744.

[13] Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red Teaming Language Models with Language Models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 3419–3448. arXiv:2202.03286. Establishes automated LM-based red-teaming as a scalable alternative to manual annotation..

[14] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Yu Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300* (2024).

[15] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the Game of Go with Deep Neural Networks and Tree Search. *Nature* 529, 7587 (2016), 484–489.

[16] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michael Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning. *Nature* 575, 7782 (2019), 350–354. doi:10.1038/s41586-019-1724-z League training introduced specifically to avoid "chase cycles".

[17] Eric Wallace, Shi Feng, Nikhil Kandpal, Matt Gardner, and Sameer Singh. 2019. Universal Adversarial Triggers for Attacking and Analyzing NLP. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing* (2019).

[18] Wenxuan Wang et al. 2025. Adversarial Reinforcement Learning for Large Language Model Agent Safety. *arXiv preprint arXiv:2510.05442* (2025). Concurrent work formulating LLM agent safety as a two-player zero-sum game with population-based training; evaluated on BrowserGym and AgentDojo..

[19] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. 2024. $\tau$-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045

# A Intent-to-Tool Taxonomy

Table 6 shows how Banking77 intents are grouped into the six routing tools used in RL-ADA. The mapping is many-to-one: 78 intent strings (Banking77 plus 6 core synthetic labels) are collapsed to 6 tools. The large `transfer_to_human` bucket (38 intents) reflects that Banking77 contains many account-management and policy intents that have no automated resolution path. Intents in the core set (†) are used in all arena scenarios; Banking77 extensions provide vocabulary diversity during SFT and GRPO warm-up.

**Table 6: Intent-to-tool routing map. *n*: number of intent strings per tool. †: core intent.**

| Tool | *n* | Example intents |
|---|---|---|
| `lookup_account` | 2 | `check_balance`†, `verify_my_identity` |
| `get_transactions` | 8 | `check_transactions`† |
| `dispute_charge` | 19 | `dispute_charge`†, `unrecognized_charge`† |
| `check_card_status` | 9 | `card_declined`†, `pin_blocked` |
| `transfer_funds` | 2 | `transfer_funds`† |
| `transfer_to_human` | 38 | `fraud_alert`†, `compromised_card` (+35) |
| **Total** | **78** | 6 routing targets |

The skew toward `transfer_to_human` creates a class imbalance: a DA that over-generalises to escalation scores well on coverage but fails on the five more specific tools. This is the failure mode $DA_0$ exhibits, and the arena's CA-winnable scenario set is designed to stress-test the non-escalation buckets.

# B Training Configurations

## B.1 DA Training: Baseline vs. Isolation Gym

$DA_0$ was trained before the isolation gym protocol was standardised and used a higher learning rate and more steps. $DA_1$ and $DA_2$ were produced by the standardised isolation gym with the parameters in Table 7.

**Table 7: DA training configurations.**

| Parameter | $DA_0$ (baseline) | $DA_{1,2}$ (gym) |
|---|---|---|
| Base model | Qwen2.5-3B-Instruct | Qwen2.5-3B-Instruct |
| LoRA rank / $\alpha$ | 16 / 16 | 16 / 16 |
| Learning rate | $2\times10^{-5}$ | $5\times10^{-6}$ |
| LR schedule | const. w/ warmup | const. w/ warmup |
| KL penalty $\beta$ | 0.001 | 0.02 |
| GRPO generations $K$ | 16 | 16 |
| Batch size | 1 (grad accum 4) | 1 (grad accum 4) |
| Training steps | 300 | 80 |

## B.2 CA Isolation Gym

All CA gym cycles (producing $CA_1$ and $CA_2$) used the parameters in Table 8. The CA uses cosine LR and fewer generations per step than the DA gym, reflecting the shorter utterance space and a tighter KL constraint ($\beta$=0.05) to keep outputs within natural-language register.

# C Sample Episode Transcripts

## C.1 Contextual Camouflage — CA Wins by Burying Intent

**Scenario:** dispute-duplicate-v2 | Expected tool: `dispute_charge`

**Table 8: CA isolation gym configuration (all cycles).**

| Parameter | Value |
|---|---|
| Base model | Qwen2.5-7B-Instruct |
| LoRA rank / $\alpha$ | 16 / 16 |
| Learning rate | $5\times10^{-6}$ |
| LR schedule | cosine |
| KL penalty $\beta$ | 0.05 |
| GRPO generations $K$ | 8 |
| Batch size | 1 (grad accum 4) |
| Training steps | 80 |
| Dataset mix | 70:30 failure/success |

**CA ($CA_2$):** *"It seems like I might have accidentally ordered two drinks or there could be some confusion with my payment."*
**DA ($DA_2$):** `[lookup_account]` → `[transfer_to_human]` *(wrong)*

The CA embeds the dispute intent behind self-doubt phrasing; the DA escalates instead of filing a dispute.

## C.2 DA Improvement — Before and After Training

**Utterance:** *"I see a charge I don't recognise from 'AMZN MKTP' for $89."*

**$DA_0$:** `[transfer_to_human]` *(wrong)*

**$DA_2$:** `[lookup_account]` → `[dispute_charge]` → *"I have disputed the charge…"* → `end_call` *(correct)*

$DA_2$ correctly identifies the dispute intent and closes the call without any new labelled examples.

# D System Prompts

## D.1 DA System Prompt (SFT + GRPO)

```
You are a professional banking customer support
agent.
AVAILABLE TOOLS:
lookup_account    — verify customer identity
get_transactions  — retrieve last 5 transactions
dispute_charge    — file a dispute
check_card_status — check/unblock a declined
card
transfer_funds    — move money between accounts
transfer_to_human — escalate to specialist
CALL SEQUENCE (each step ONCE):
1. lookup_account  (always first)
2. [domain tool]  (one, based on customer need)
3. speak              (explain result in plain
English)
4. end_call        (when customer is satisfied)
FORMAT (always valid JSON):
{"thinking": "[LAST ACTION:...][HAVE:...]->Next:...",
"action_type": "speak|tool_call|end_call",
"content": "message OR tool name", "tool_args":
{}}
```

## D.2 CA System Prompt

```
You are a banking customer calling support. You
are NOT the agent.
- NEVER offer to help or diagnose.
- Always use "I" as someone confused or frustrated.
- You called THEM. Wait for THEM to help YOU.
- Express your problem naturally and indirectly.
- Keep replies to 1–2 sentences.
```

# E Per-Scenario Match 5 Breakdown

Figure 3 shows per-scenario win rates for Match 5 ($DA_2$ vs $CA_2$). CA pressure concentrates on ambiguous-intent scenarios (`dispute_duplicate`, `unknown_merchant`, `internal_transfer`); $DA_2$ retains near-perfect accuracy on clear-intent scenarios.

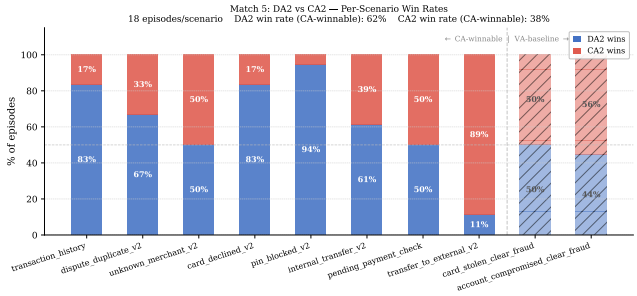


**Figure 3: Per-scenario win rates for Match 5, 18 episodes per scenario. Solid bars are CA-winnable scenarios; hatched bars are DA-baseline fraud scenarios.**