ASU - AI for Business: Creating Smart Business Negotiations Bots

Dhruv Bansal (dbansa11@asu.edu) Bradley Breisch (bbreisc1@asu.edu)
Luan Nguyen (ltnguy58@asu.edu)

Fall 2025 - Spring 2026

Abstract

Automated negotiation with large language models (LLMs) is commercially attractive but technically demanding: a capable agent must track private utilities, adapt strategy across many turns, produce numerically valid proposals, and know when to concede or walk away. We consider two regimes in which LLMs participate in negotiation. In the mixed setting, an LLM faces a human or unstructured counterpart in an assistive or advisory role. In the symmetric multi-agent setting, two LLM agents bargain under a shared formal protocol with private priorities until a deal, impasse, or turn limit is reached. This paper presents a complete pipeline for the symmetric regime: a structured Thought / Talk / Action output format that decouples strategic intent from language generation, a three-stage training procedure combining supervised fine-tuning with reinforcement learning, and a composite reward scheme targeting deal-level outcomes. We train and evaluate on the CaSiNo multi-issue bargaining corpus, pitting the trained agent in head-to-head play against opponents conditioned on diverse personas. Our results show that the trained agent modestly improves aggregate deal rate relative to the untuned base model, while also highlighting that agreement rate and utility are distinct objectives that can move independently. We discuss the implications of these findings for the design and evaluation of negotiation agents in business settings.

1 Introduction

Negotiation is a routine but high-stakes function across procurement, commercial contracting, business development, and resource allocation. Automating it with AI is a compelling opportunity, yet the task resists the kind of single-turn, single-prompt reasoning at which modern LLMs excel. A capable negotiating agent must maintain private utility estimates across many turns, adapt its strategy as the counterpart reveals preferences through offers and rejections, and ultimately produce numerically valid agreements that reflect both tactical strength and situational judgment.

We distinguish two common regimes in which LLMs participate in negotiation. In the **mixed** setting, an LLM occupies one side of the table while the counterpart is a human or a separately-prompted model without a shared protocol. In the **symmetric multi-agent** setting, two LLM agents negotiate under a shared formal protocol, each observing only the public dialogue and their own private priorities, until the episode closes with a deal, a walk-away, or a turn limit.

The central challenge motivating this work is that linguistic fluency and negotiation competence are not the same thing. A systematic evaluation spanning 35 tasks across four negotiation corpora shows that even frontier models produce contextually coherent language while making irrational offers, conceding utility too readily, and failing to model their counterpart's priorities through the

conversation [1]. Models can score well on comprehension and annotation tasks while remaining poor real-time negotiators.

This paper presents our approach to building a locally trainable, open-weight negotiation agent for the symmetric multi-agent setting.

2 Problem Statement

The target task is multi-issue bilateral negotiation under partial information. Two parties divide a fixed pool of resources across several categories. Each holds private utility values that determine how much each allocation is worth to them. Neither party can observe the other's priorities directly. The agent must persist state across multiple turns, generate numerically valid proposals, infer the counterpart's priorities from their observable behavior, and choose when to concede, anchor, or accept. This combination of requirements is what makes the problem hard and what makes naive prompting of an LLM an insufficient solution.

2.1 The Strategy-Language Tradeoff

The first large-scale neural negotiation benchmark [2] highlighted a key tradeoff: RL-trained agents outperform supervised baselines on utility, but optimizing word sequences directly leads to degenerate language, producing repetitive, abrupt, or unnatural interactions that exploit training artifacts rather than genuinely negotiate. The solution is to separate strategy from surface language by expressing each move as a high-level dialogue act (propose, counter, agree) and using RL to optimize these decisions, while leaving language generation to a separate module [3]. This act-level approach stabilizes learning and produces agents that are both more effective and more natural in conversation.

This principle holds at the scale of contemporary LLMs. In buyer-seller bargaining, an Offer Generator that controls the numeric bid, paired with an LLM responsible only for language generation and structured via a Thought / Talk / Action prompt format, substantially raised buyer surplus and eliminated a class of first-bid anchoring failures that end-to-end models could not escape [4]. This design underpins our structured output format.

2.2 Behavioral Control

What an agent does across turns matters as much as how it phrases each response. Work on agent personality in negotiation [5] demonstrates that moderately selfish agents outperform both overly cooperative and overly aggressive ones. Excessive selfishness drives the counterpart to walk away, while excessive cooperation leaves utility on the table. The counterpart adapts to observed agent behavior rather than to linguistic surface cues alone, meaning that strategic behavioral control belongs at the policy level, not just in prompt text.

The EvoEmo framework [6] takes this further, evolving behavioral policies governing traits such as confidence, patience, and aggressiveness using population-based genetic optimization over full multi-turn rollouts, rather than fine-tuning model weights directly. Evolved adaptive policies outperformed both vanilla and fixed-emotion baselines on success rate, negotiation efficiency, and buyer savings, even against stronger opponents. These findings support treating behavioral shaping as an important design lever, which motivates our use of persona-conditioned opponents during training and evaluation.

2.3 Reward Design in Multi-Turn Episodes

In a negotiation that spans many turns, the true outcome is only knowable at the end of the episode. Assigning a single terminal reward uniformly to every preceding move conflates good and bad individual decisions. Recent advances in turn-level reward design address this by combining an episode-level outcome signal with per-turn verifiable checks and LLM-as-judge evaluations, enabling faster learning and more stable optimization.

The REPO framework [7] tackled the complementary problem of reward gaming in multi-turn dialogue: rather than simply summing a preference-trained reward model, an LLM judge, and programmatic checks, REPO uses the secondary signals to modulate the primary reward, preventing the policy from exploiting any single channel. This design outperformed PPO, DPO, and GRPO baselines on dialogue quality and behavioral consistency in a production negotiation setting. Our reward scheme follows the same logic: a points-based utility signal provides the ground-truth target, while per-turn format, length, judge, and strategic-talk rewards supply dense intermediate feedback that guides learning throughout the episode.

2.4 Evaluation Methodology

Existing benchmarks expose important capability gaps but leave live bargaining performance underspecified. The most thorough evaluation framework for LLM negotiators [1] spans 35 tasks covering the start, middle, and end stages of a negotiation, yet most tasks test comprehension and annotation rather than live bargaining. A model can improve steadily on these tasks while remaining a poor real-time negotiator.

NegotiationArena [8] confirms this in live play. Initial offer anchoring and injected behavioral instructions (e.g., "act desperate," "act cunning") substantially shift outcomes between models, but without per-turn metrics there is no mechanism to attribute which moves drove the result. Our evaluation focuses on head-to-head play, tracking deal rate, points accrued, and turns-to-deal against diverse opponent personas as the primary signal, with comprehension metrics retained where analytically useful.

3 Our Approach

3.1 Domain and Data

We train and evaluate on CaSiNo [9], a corpus of 1,030 human-human negotiation dialogues in which two campsite neighbors divide three packages each of food, water, and firewood. Each participant is assigned a private priority ordering over the three item types, which maps to point values: high-priority items are worth 5 points, medium 4, and low 3, for a maximum of 15 points per agent. This setup is a clean instance of the symmetric multi-agent problem: two parties, a shared item pool, private utilities, and a well-defined terminal outcome. We train on the training split and report results on held-out test scenarios to check generalization.

3.2 Structured Output Format

Each model turn is structured into three XML-style sections:

```
<thought>Private reasoning: utilities, beliefs about the partner, tactical plan.</thought>
<talk>Natural language directed at the counterpart.</talk>
```

`<action>[ACTION]</action>`

The `<thought>` block is never transmitted to the opponent and is stripped from the message before it is placed in the counterpart's context. The `<talk>` block is the only natural language the counterpart receives. The `<action>` block carries the structured decision and must contain exactly one of five valid actions:

- **`[TALK]`**: A conversational turn with no associated proposal. The agent's message is delivered via the `<talk>` block; this action signals that the turn contains no offer or terminal decision.
- **`[SUBMIT_DEAL] food:F water:W firewood:FW`**: Propose a division of the item pool. The values F, W, and FW are integers in [0, 3] representing the agent's own allocation. Before the message is placed in the opponent's context, every deal value is flipped to 3 minus the original, so the opponent sees the quantities they would receive under the proposal. Allocations therefore always sum to 3 per item across both parties.
- **`[ACCEPT_DEAL]`**: Accept the counterpart's most recently submitted proposal. The accepted deal is used to compute final point totals.
- **`[REJECT_DEAL]`**: Explicitly decline the counterpart's most recent proposal without walking away; the negotiation continues.
- **`[WALK_AWAY]`**: Terminate the negotiation with no agreement. Both parties receive zero points.

This format makes reward computation tractable: format correctness, deal arithmetic, and action appropriateness can all be checked programmatically against the structured fields, while the `<thought>` block is available for judge-based evaluation of reasoning quality.

3.3 Training Pipeline

Training proceeds in three stages. All stages use Qwen2.5-7B-Instruct [10] as the base model with LoRA adapters [11] (rank 16, alpha 32, all-linear targets) rather than full fine-tuning. Training runs on an HPC cluster (ASU Sol) with A100 GPUs, using the TRL library [12] for all RL stages.

3.3.1 Stage 1: Supervised Fine-Tuning

The CaSiNo corpus contains raw human-human dialogues without internal reasoning traces. Before any RL, we annotate each assistant turn with a synthetic `<thought>` block generated by GPT-5-mini via API, producing complete `Thought / Talk / Action` transcripts for every training dialogue. The base model is then fine-tuned on these annotated conversations using standard SFT with assistant-only loss, so gradients flow only through the model's own turns. This stage teaches the model to reliably follow the output format and to produce grounded language before any reward signal is introduced.

3.3.2 Stage 2: Annotated GRPO

Starting from the SFT checkpoint, we run Group Relative Policy Optimization (GRPO) [13] on the annotated human episodes. For each training prompt, the model generates 8 candidate completions; all reward signals are computed for each, and the group mean serves as the baseline for the policy update. A `prompt_split` parameter controls which assistant turns in each dialogue are held as fixed context and which are treated as optimization targets: a higher split value keeps more of the dialogue as context and produces fewer but higher-quality training prompts per episode, while a lower value increases the number of prompts at the cost of a shorter conditioning prefix. This

stage uses the `dr_grpo` loss variant from TRL, which is less sensitive to outlier reward values than standard GRPO, with asymmetric epsilon clipping (0.2/0.28).

3.3.3 Stage 3: Self-Play GRPO

In the final stage, the learner generates training data by playing negotiation episodes against a frozen opponent checkpoint. The opponent is initialized from the annotated GRPO checkpoint (checkpoint-1200), and its system prompt is conditioned on one of four personas, sampled with equal probability at the start of each episode:

- **Uncompromising:** insists on top-priority items and rarely concedes.
- **Selfish:** anchors on claiming all units of the highest-value item with minimal movement.
- **Anchoring:** opens with an extreme offer and concedes slowly.
- **Cooperative:** prioritizes reaching an agreement and responds reasonably to fair proposals.

Episodes run for up to 18 turns with generation temperature 0.7. Termination occurs on `[ACCEPT_DEAL]`, `[WALK_AWAY]`, or a reject-loop condition (three consecutive identical proposals). After each episode, the learner's turns are sliced into GRPO rows using the same prompt-split mechanism as Stage 2, and a policy update is applied. The self-play dataset is cached to disk for reproducibility and fast re-runs. This stage runs with `beta=0.0` (no explicit KL penalty), relying instead on the DR-GRPO clipping mechanism to constrain policy drift. This self-play loop exposes the learner to a range of opponent behaviors without collapsing to a single counter-strategy, which is the main failure mode of standard self-play.

3.4 Reward Design

Five reward signals are computed per turn and combined as a weighted sum. The composite design is motivated by the finding that modulating multiple reward sources stabilizes learning and reduces reward gaming [7]. Reward weights differ between the annotated and self-play stages to reflect their distinct learning objectives.

Signal	Description	Annotated	Self-Play
Format	Binary check for well-formed XML tags (`<thought>`, `<talk>`, `<action>` in order), valid action syntax, and deal arithmetic (allocations in [0, 3]). `[ACCEPT_DEAL]` is only valid when the opponent's last action was `[SUBMIT_DEAL]`.	0.5	0.5

Signal	Description	Annotated	Self-Play
Length	Linear reward for responses between 50 and 750 characters; penalizes very short (-1.0) or excessively long completions with gentle decay.	0.3	0.2
Judge	LLM-as-judge score (0 to 1) evaluating both the strategic coherence of the `<thought>` block and the alignment between the `<talk>` message and the chosen `<action>`. Scored by Claude Haiku 4.5 via API.	1.5	0.5
Points	Per-turn utility signal. For `[SUBMIT_DEAL]` and `[ACCEPT_DEAL]`, computed as `(points / max_points) * 2.0 - 0.5`, rewarding proposals that favor the learner. `[WALK_AWAY]` receives -1.0; other actions receive 0.	1.0	2.5
Strategic talk	Heuristic score (0 to 1) for `[TALK]` and `[REJECT_DEAL]` turns only. Awards credit for mentioning high-value items (anchoring), avoiding priority-leaking language, and referencing low-value items (concession signaling).	–	1.0

Format and length rewards are verifiable and computed deterministically. The judge uses Claude Haiku 4.5 via OpenRouter with retry logic for API errors (3 attempts, exponential backoff). The points reward carries the heaviest weight in self-play (2.5), reflecting the shift from learning format compliance (already achieved in annotated GRPO) to maximizing deal quality. The strategic talk

reward was introduced for the self-play stage to provide a per-turn gradient on conversational moves, which otherwise receive no signal from the points reward.

4 Evaluation

4.1 Comprehension Tasks

We evaluate on the structured task suite from Kwon et al. [1], adapted as a Python evaluation module. The framework spans start, during, and end stages of a negotiation across four datasets (CaSiNo, DealOrNoDeal, JobInterview, and CRA), covering 37 tasks in total. Task types include comprehension (can the model identify item counts, point values, and priorities from a scenario?), annotation (can it classify dialogue acts and negotiation strategies?), partner modeling (can it infer the counterpart's priorities from observed behavior?), and generation (can it produce a contextually appropriate next utterance?). We report on the CaSiNo and DealOrNoDeal subsets, which are the most relevant to our training domain.

The framework supports multiple model backends (OpenAI API, HuggingFace transformers, local checkpoints with LoRA adapter merging, and vLLM-served models), allowing direct comparison between the base model and RL-trained checkpoints on identical prompts. Metrics include accuracy for classification tasks, elementwise accuracy for structured outputs like deal allocations, macro F1 for multi-label annotation, and BLEU/ROUGE for generation.

The framework's key limitation is that all tasks are scored independently: a model is tested on understanding priorities and separately on generating a response, but never on whether it uses that understanding to negotiate better in live play. A model can score well here while being a poor negotiator, which is precisely the gap the head-to-head harness is designed to expose.

4.2 Head-to-Head Negotiation Harness

The comprehension task suite evaluates passive understanding: whether a model can read a negotiation transcript and answer questions about it. This is analogous to testing chess knowledge by asking someone to analyze a recorded game. The critical question is whether the model can actually negotiate: produce well-formatted structured outputs turn after turn, adapt to different opponent styles, close deals, and maximize its own utility in live multi-turn play.

We implement a separate head-to-head evaluation harness (`negotiate.py`) that runs full negotiation episodes between two LLM agents. Each episode loads a CaSiNo scenario, assigns private priorities to both sides, and alternates turns between a learner and an opponent model. The harness enforces the structured output format, strips `<thought>` blocks before forwarding messages to the counterpart, flips deal perspectives so each side sees the allocation from their own point of view, and tracks format compliance per turn. Episodes terminate on `[ACCEPT_DEAL]`, `[WALK_AWAY]`, a reject loop (three consecutive identical `[SUBMIT_DEAL]` proposals), or a configurable turn limit.

This harness is necessary because the comprehension tasks cannot measure the qualities that training is designed to improve. A model that aces priority identification tasks may still fail to leverage those priorities when generating offers. A model that understands deal arithmetic in isolation may produce malformed proposals under the pressure of a multi-turn exchange. The head-to-head harness directly measures deal rate, learner and opponent points, score ratio (learner points divided by the sum of both players' points, where 0.5 indicates parity), turns to deal, format compliance,

and malformed deal rate, all broken down by opponent persona. These are the metrics that reveal whether RL training produced a better negotiator, not just a better test-taker.

We evaluate four models (the GRPO Self-Play checkpoint, the annotated GRPO checkpoint at step 1,200, the SFT checkpoint, and the untuned Qwen base model) in a round-robin format. Each model plays 200 episodes as the learner (first mover) against the SFT checkpoint as the standard opponent, with opponent persona sampled uniformly across the four types. We also run cross-matches where the SFT checkpoint faces the two GRPO checkpoints as opponents, yielding six matchups total.

5 Results

5.1 Head-to-Head Performance

Table 1 summarizes the six head-to-head matchups across 200 episodes each.

	GRPO-SP	GRPO-Ann	SFT	Base
Deal (%)	**93.0**	91.5	90.5	91.5
Pts	**17.90**	17.19	17.23	17.59
Opp	**18.44**	18.97	18.85	18.66
Ratio	**.493**	.475	.478	.485
Turns	6.60	**6.11**	6.51	6.61

Table 1: Head-to-head negotiation results (200 episodes per matchup). Pts/Opp are mean learner and opponent points, Ratio is learner share of joint score. Bold indicates best among the four learners.

All models achieve deal rates above 90%, and all score ratios fall between 0.475 and 0.493. The differences are small: GRPO Self-Play's best-in-class ratio of 0.493 is only 0.008 above the untuned base model's 0.485, and the base model matches the annotated GRPO checkpoint on deal rate while achieving a higher ratio (0.485 vs. 0.475). The learner (first mover) consistently scores fewer points than the opponent across all matchups, suggesting a modest second-mover advantage in this protocol. Format compliance exceeds 93% for all models with malformed deal rates below 1%.

The aggregate numbers mask meaningful variation by opponent persona. Tables 2 and 3 break down deal rate and learner points by persona.

Persona	GRPO-SP	GRPO-Ann	SFT	Base
Anchoring	**100.0**	94.6	**100.0**	92.3
Cooperative	**100.0**	98.0	**100.0**	**100.0**
Selfish	95.6	**100.0**	95.7	98.4
Uncompromising	**79.7**	75.9	67.3	76.0

Table 2: Deal rate (%) by opponent persona. All models play as learner against SFT. Bold indicates highest per row.

Persona	GRPO-SP	GRPO-Ann	SFT	Base
Anchoring	**16.89**	16.51	16.54	15.38
Cooperative	**19.36**	18.75	18.15	19.15
Selfish	**18.16**	17.56	17.02	18.11
Uncompromising	17.51	15.88	17.26	**18.13**

Table 3: Learner points by opponent persona. All models play as learner against SFT. Bold indicates highest per row.

GRPO's clearest gains are persona-specific. Against uncompromising opponents, GRPO-SP achieves 79.7% deal rate versus SFT's 67.3% (+12.4pp) while recovering to 17.51 points from GRPO-Ann's 15.88. Against cooperative opponents, GRPO-SP scores 19.36 points (ratio 0.541), the only case where any model consistently outscores its opponent. Against anchoring opponents, all models score below parity (ratios 0.42–0.46), confirming that anchoring is effective regardless of the counterpart.

The annotated GRPO checkpoint reveals a tradeoff: it scores the lowest points against uncompromising opponents (15.88, ratio 0.439), suggesting that training on human dialogues, where participants generally reach agreement, made the model too accommodating against hard bargainers. Self-play partially corrects this by restoring utility while maintaining the highest deal rate in that category.

5.2 Comprehension Task Performance

Table 4 reports CaSiNo comprehension results across all three checkpoints and the base model. Start-stage tasks (scenario comprehension before any dialogue) are at or near ceiling for all models, with one exception: `max_points` is low across the board (17–33%), and GRPO-SP drops to 16.7%. The small sample size (n=6) limits the diagnostic value of start-stage tasks.

Stage	Task	Metric	GRPO-SP	GRPO-Ann	Base
Start	High priority	Acc.	1.000	1.000	1.000
	Low priority	Acc.	.833	.833	**1.000**
	Point values	Elem.	1.000	1.000	1.000
	Max points	Acc.	.167	**.333**	**.333**
	Item count	Acc.	1.000	1.000	1.000
Mid	Own high priority	Acc.	**.744**	.694	.694
	Own low priority	Acc.	**.579**	**.579**	.537
	Partner high prior.	Acc.	.570	.545	**.628**
	Partner low prior.	Acc.	.281	**.306**	.289
	Strategy classif.	F1	**.442**	.437	.394
	Response gen.	B / R	.143 / **.19**	**.143** / .18	.139 / .18
End	Deal specifics	Elem.	**.920**	.871	.843
	Deal total	Acc.	.661	.719	**.736**
	Deal likeness	Acc.	.372	.364	**.388**
	Deal satisfaction	Acc.	**.430**	.397	.397
	Partner deal like.	Acc.	.471	**.504**	.471
	Partner deal sat.	Acc.	**.446**	.397	.413

Table 4: CaSiNo comprehension task results. Elem. = elementwise accuracy; B / R = BLEU / ROUGE. Start-stage n=6, mid/end n=121–200. Bold indicates the highest score per task.

Mid-stage tasks show the most informative variation. Own high-priority identification is unchanged from base to annotated (both 69.4%) then jumps to 74.4% after self-play, a 5-point gain. Strategy classification follows the same pattern (39.4% to 43.7% to 44.2%). However, partner high-priority inference regresses: the base model leads at 62.8%, dropping to 54.5% (annotated) and recovering only to 57.0% (self-play). Both GRPO checkpoints are worse than the base model at reading the opponent's priorities from dialogue context.

End-stage tasks reveal a striking pattern on deal specifics: GRPO-SP achieves 92.0% elementwise accuracy, a 7.7-point improvement over the base model's 84.3%. This is the largest single improvement across all comprehension tasks and suggests that self-play training sharpened the model's ability to identify the specific terms of a deal. However, deal total accuracy moves in the opposite direction (73.6% base to 66.1% self-play), indicating the model is better at parsing individual allocations but worse at computing the aggregate.

Table 5 reports DealOrNoDeal results, which test out-of-distribution transfer since the model was trained only on CaSiNo.

Stage	Task	Metric	GRPO-SP	GRPO-Ann	Base
Start	Point values	Elem.	.567	**.591**	.583
	Max points	Acc.	.669	.630	**.709**
	Item count	Acc.	1.000	1.000	1.000
Mid	Dialogue acts	F1	**.260**	.255	.245
	Response gen.	B / R	.108 / .21	**.112 / .22**	.107 / .21
End	Deal specifics	Elem.	**.550**	.532	.548
	Deal total	Acc.	.565	**.610**	.560

Table 5: DealOrNoDeal comprehension task results. Start-stage n=127, mid/end n=200. The `mid_full_proposal` task is excluded (n=0 for both GRPO checkpoints). Bold indicates the highest score per task.

On DealOrNoDeal, max points accuracy drops from the base model (70.9% to 63.0% annotated) then partially recovers after self-play (66.9%), while dialogue act classification improves monotonically (24.5% to 25.5% to 26.0%). The changes are small and no model dominates.

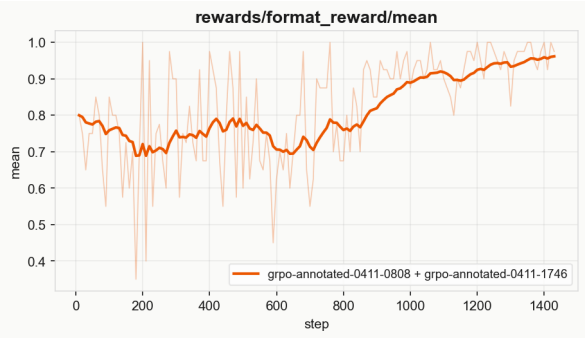
Across all 24 comparable tasks, GRPO-SP leads on 7, GRPO-Ann on 5, the base model on 5, with 7 ties. The overall pattern is that GRPO training does not systematically improve or degrade comprehension. Most changes are under 5 percentage points.

The partner-modeling regression is the most noteworthy finding. Both GRPO checkpoints are worse than the base model at inferring the opponent's high-priority item from dialogue context (-8.3pp for annotated, -5.8pp for self-play vs. base). This aligns with the head-to-head result that annotated GRPO produced a more accommodating model: the training signal rewarded proposals that maximize the learner's own points, which may have de-emphasized opponent modeling. Comprehension scores do not predict head-to-head performance: the base model leads on several comprehension tasks while performing worse in live negotiation, confirming the evaluation gap identified in Section 4.

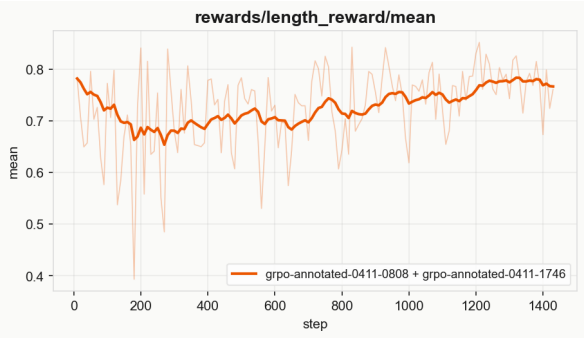
5.3 Training Dynamics

5.3.1 Annotated GRPO

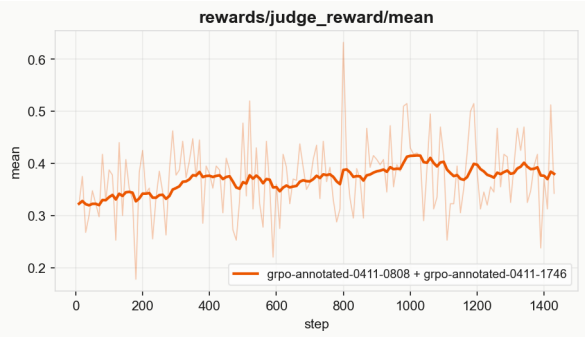
Figure 1 shows the per-component reward means across the full annotated GRPO run (approximately 1,430 steps, merged from two consecutive runs starting from the SFT checkpoint). Each component reveals what the optimizer learned and where it stalled.



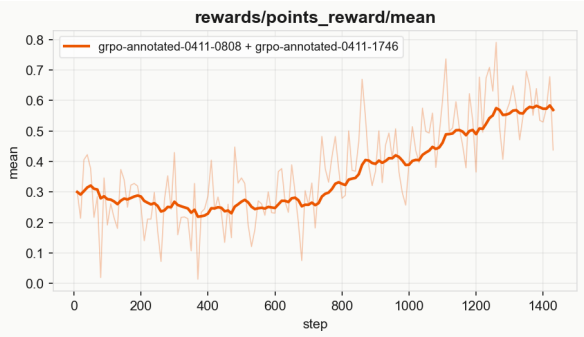
(a) Format reward



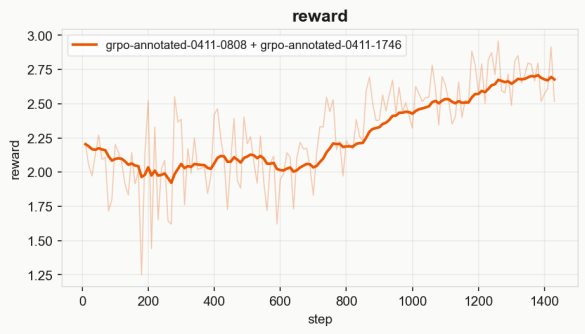
(b) Length reward



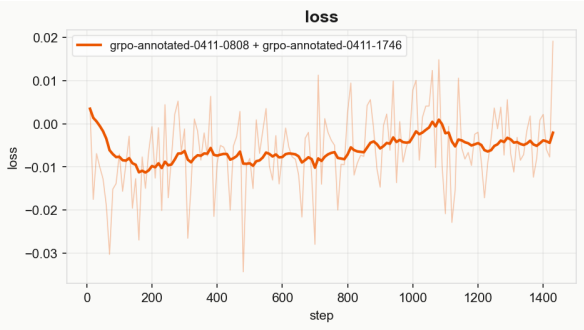
(c) Judge reward



(d) Points reward



(e) Total reward



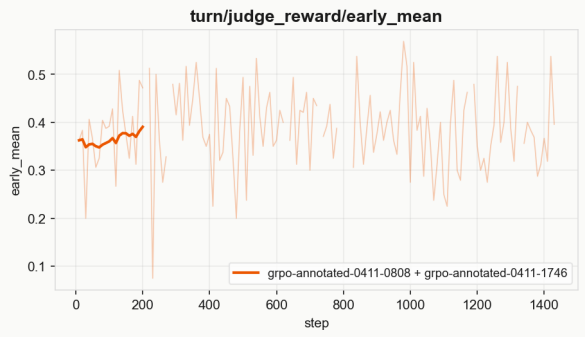
(f) GRPO policy loss

Figure 1: Per-component and total reward means during annotated GRPO training (approximately 1,430 steps). Bold lines show exponential smoothing; faint lines show raw values.

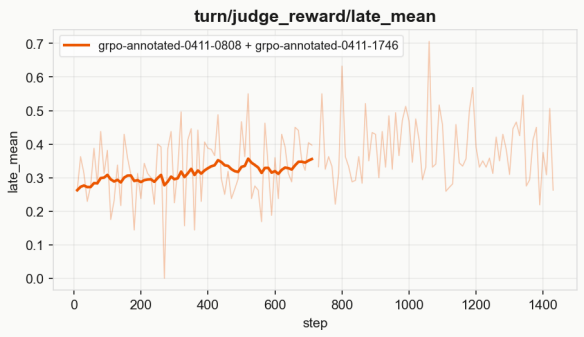
The reward components follow a clear learning sequence. **Format reward** dips from 0.80 to 0.70 in the first 200 steps as the optimizer explores, then climbs steadily to 0.95 by step 1,100. **Length reward** follows a U-shaped path (0.78 to 0.68 to 0.77) with negligible net change. **Judge reward** is the hardest signal to move, gaining only 0.06 over the full run (0.32 to 0.38); the LLM-as-judge signal is too noisy to drive large behavioral changes.

The key structural feature is the **points reward** inflection at step 700: the signal is flat at 0.25–0.30 for the first 700 steps, then climbs steeply to 0.57 by step 1,300. This delay suggests the model needed to stabilize format compliance before the utility signal could take effect. **Total reward** mirrors this trajectory (2.20 to 2.70), dominated by the points component in the second half of training. Policy loss variance increases after step 600, consistent with the model making larger updates as it discovers higher-reward behaviors.

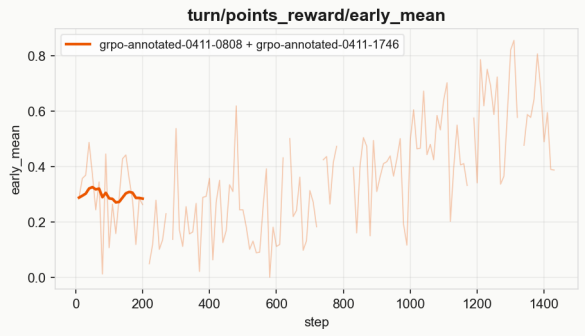
Figure 2 breaks the judge and points rewards down by turn position. Early-turn metrics (turns in the first half of each dialogue) and late-turn metrics (second half) reveal where in the negotiation the model improved most.



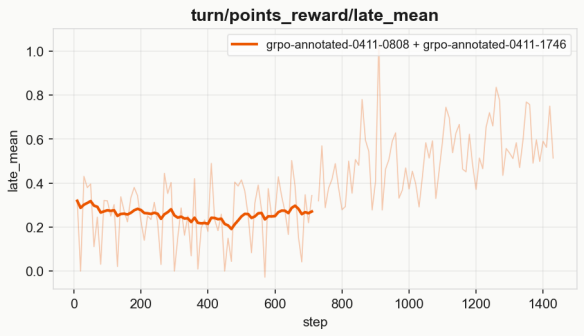
(a) Judge reward, early turns



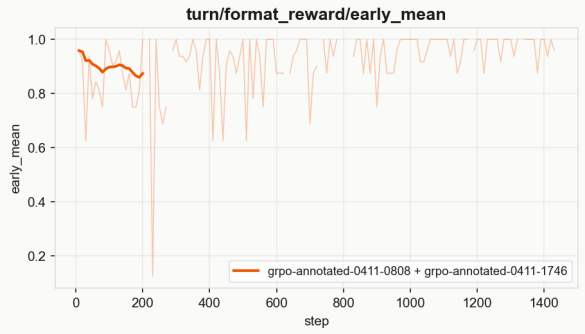
(b) Judge reward, late turns



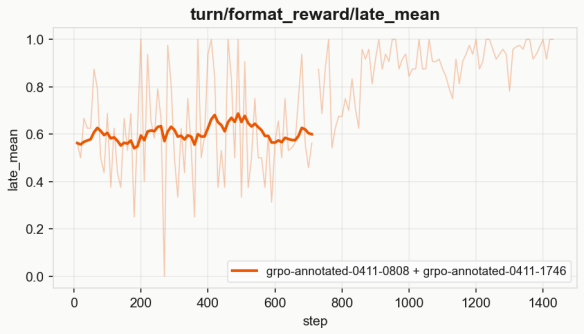
(c) Points reward, early turns



(d) Points reward, late turns



(e) Format reward, early turns



(f) Format reward, late turns

Figure 2: Per-component reward means split by turn position during annotated GRPO training. Early turns are in the first half of each dialogue; late turns are in the second half.

The turn-position breakdown reveals that most learning happens on late turns. **Late-turn format reward** shows the largest single improvement in the run: 0.57 to 0.95 (+0.38), while early-turn

format starts near ceiling (~0.96) and, after a brief early dip, returns there. The model's main challenge was maintaining structural compliance across longer dialogues, not producing it initially. Points reward roughly doubles on early turns (0.28 to 0.60) and more than doubles on late turns (0.30 to 0.75), with the late-turn inflection at step 700 matching the aggregate pattern. Judge reward is flat on early turns (0.36 throughout) and shows only a modest rise on late turns (0.27 to 0.40). Reward standard deviation declines from 0.55 to 0.42 (more consistent outputs), and clip ratio stays below 0.1% after the first 250 steps, confirming conservative policy updates.

5.3.2 Self-Play GRPO

Figure 3 shows the per-component reward means for the self-play GRPO run (grpo-self_play-0413-1759, approximately 870 steps), which starts from the annotated GRPO checkpoint-1200.

The self-play run faces a qualitatively different learning problem: instead of diverse human dialogues, the model generates training data by negotiating against a frozen copy of itself.

Format reward climbs from 0.90 to 0.97, but with standard deviation hitting zero in 37% of steps (all eight GRPO completions receive identical scores), it no longer provides useful gradient. **Length reward** drifts down from 0.53 to 0.48 as the model produces shorter, more formulaic responses. **Judge reward** declines slightly (0.38 to 0.36). **Points reward** stagnates despite carrying the heaviest weight (2.5), fluctuating between 0.29 and 0.46 with no directional trend. The frozen opponent plays identically for each scenario, so once the model finds a reasonable counter-strategy, the reward landscape is flat. **Strategic talk reward** shows the steepest decline (0.20 to 0.11), indicating that GRPO reinforces whatever patterns produce acceptable deals at the cost of conversational diversity. **Total reward** declines from 2.45 to 2.31.

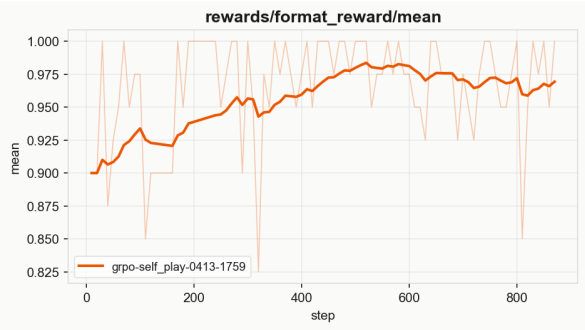
Diagnostic signals confirm the policy is converging rather than collapsing: reward standard deviation declines (0.47 to 0.38) and entropy remains stable (0.50–0.52). The model is not becoming degenerate; it is narrowing to a consistent policy that does not improve on the annotated checkpoint.

The contrast between stages is stark. Annotated GRPO showed a clear points inflection at step 700 and meaningful gains across components. Self-play shows none of these patterns. The difference is not in the reward design but in the data: human dialogues provide diverse strategic contexts that create a rich reward landscape, while a frozen opponent produces repetitive episodes the model quickly learns to handle but cannot learn beyond.

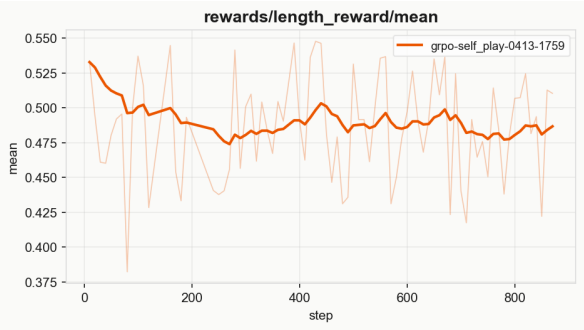
5.4 Discussion

The central question is whether GRPO training produced a meaningfully better negotiator. The aggregate evidence says: marginally. The untuned base model achieves 91.5% deal rate and 0.485 score ratio; GRPO Self-Play reaches 93.0% and 0.493. Two stages of reinforcement learning bought 1.5 percentage points on deal rate and 0.008 on ratio. The base model was already a competent negotiator from pretraining.

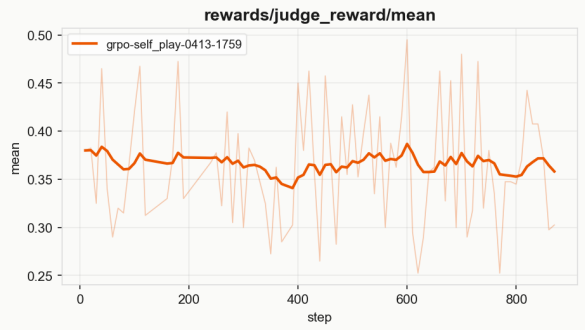
Where GRPO does make a difference is persona-specific robustness. Against uncompromising opponents, GRPO-SP closes 12.4 percentage points more deals than SFT (79.7% vs. 67.3%) while maintaining reasonable utility. Against cooperative opponents, it is the only model that consistently outscores its counterpart (ratio 0.541). The annotated stage drove the largest training-time improvements: a clear points-reward inflection at step 700 and meaningful late-turn format gains. Self-play total reward declined (2.45 to 2.31) as the frozen opponent provided diminishing learning signal, yet the self-play checkpoint outperforms the annotated one on every head-to-head metric



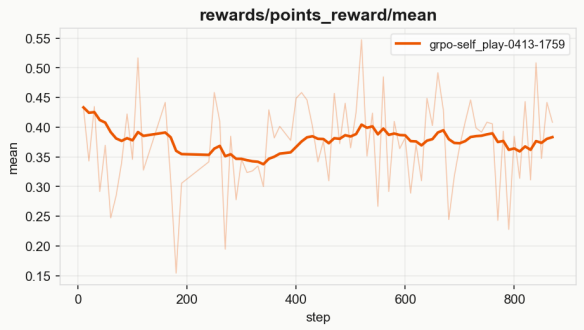
(a) Format reward



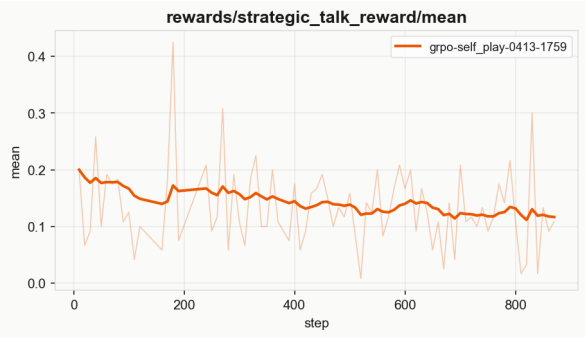
(b) Length reward



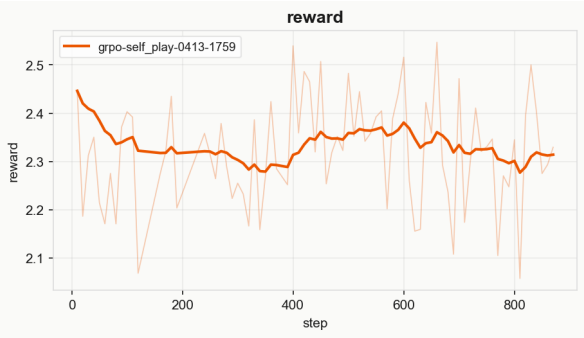
(c) Judge reward



(d) Points reward



(e) Strategic talk reward



(f) Total reward

Figure 3: Per-component and total reward means during self-play GRPO training (approximately 870 steps from checkpoint-1200). Bold lines show exponential smoothing; faint lines show raw values.

(deal rate 93.0% vs. 91.5%, ratio 0.493 vs. 0.475, uncompromising deal rate 79.7% vs. 75.9%). This disconnect between training reward and evaluation performance suggests that self-play exposure to diverse opponent personas improved robustness in ways the reward signal did not capture.

The annotated GRPO checkpoint also reveals a tradeoff between deal rate and utility. It scores the lowest points against uncompromising opponents (15.88, ratio 0.439), suggesting that training on human dialogues, where participants generally reach agreement, biased the model toward accommodation. Self-play partially corrects this, recovering utility while maintaining the highest deal rate in that category.

Credit assignment remains the primary bottleneck. The judge reward, the only signal targeting reasoning quality, improved by 0.06 over 1,430 annotated steps and declined during self-play. The LLM-as-judge gradient is too weak to drive behavioral changes on the timescale of either run. A fine-tuned reward model or a mechanism connecting deal outcomes to specific reasoning steps would be needed for further progress.

Two architectural insights emerge for future work. First, binary verifiable rewards (format, length) saturate quickly and stop contributing gradient; an adaptive weighting scheme that downweights saturated components could improve sample efficiency. Second, self-play against a frozen opponent collapses the strategic diversity that makes annotated GRPO effective. Periodically refreshing the opponent from the learner's own checkpoint would reintroduce pressure, though this risks cyclic dynamics. Adding a KL penalty (beta > 0) during self-play would constrain the policy drift that caused strategic talk and judge rewards to decline.

The comprehension results reinforce a key evaluation finding: passive understanding does not predict live negotiation ability. The base model matches or exceeds the GRPO checkpoint on several comprehension tasks while performing worse in head-to-head play. The partner-modeling regression (up to -8.3pp on CaSiNo partner high-priority inference) suggests that optimizing own-utility rewards may de-emphasize opponent modeling, a tradeoff worth investigating in future work.

References

[1] D. Kwon, E. Weiss, T. Kulshrestha, K. Chawla, G. M. Lucas, and J. Gratch, "Are LLMs effective negotiators? Systematic evaluation of the multifaceted capabilities of LLMs in negotiation dialogues," in *Findings of the association for computational linguistics: EMNLP 2024*, 2024.

[2] M. Lewis, D. Yarats, Y. N. Dauphin, D. Parikh, and D. Batra, "Deal or no deal? End-to-end learning for negotiation dialogues," *arXiv preprint arXiv:1706.05125*, 2017.

[3] H. He, D. Chen, A. Balakrishnan, and P. Liang, "Decoupling strategy and generation in negotiation dialogues," in *Proceedings of the 2018 conference on empirical methods in natural language processing*, Association for Computational Linguistics, 2018, pp. 2333–2343.

[4] T. Xia *et al.*, "Measuring bargaining abilities of LLMs: A benchmark and a buyer-enhancement method," in *Findings of the association for computational linguistics: ACL 2024*, Association for Computational Linguistics, 2024.

[5] K. Chawla, I. Wu, Y. Rong, G. M. Lucas, and J. Gratch, "Be selfish, but wisely: Investigating the impact of agent personality in mixed-motive human-agent interactions," in *Proceedings of the 2023 conference on empirical methods in natural language processing*, 2023.

[6] Y. Long, L. Xu, L. Beckenbauer, Y. Liu, and A. Brintrup, "EvoEmo: Towards evolved emotional policies for adversarial LLM agents in multi-turn price negotiation," in *Proceedings of the 25th international conference on autonomous agents and multiagent systems (AAMAS 2026)*, 2026.

[7] Z. Zhuang, Y. Chen, X. Zeng, C. Luo, L. Liu, and Y. Chen, "Teaching LLM to be persuasive: Reward-enhanced policy optimization for alignment from heterogeneous rewards," *arXiv preprint*, 2025.

[8] F. Bianchi *et al.*, "NegotiationArena: A platform for competitive LLM negotiation," in *Proceedings of the 41st international conference on machine learning*, in ICML'24. 2024.

[9] K. Chawla, J. Ramirez, R. Clever, G. Lucas, J. May, and J. Gratch, "CaSiNo: A corpus of campsite negotiation dialogues for automatic negotiation systems," in *Proceedings of the 2021 conference of the north american chapter of the association for computational linguistics: Human language technologies*, 2021, pp. 3167–3185.

[10] A. Yang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2024.

[11] E. J. Hu, Y. Shen, P. Wallis, *et al.*, "LoRA: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[12] Hugging Face, "TRL: Transformer reinforcement learning." https://github.com/huggingface/trl, 2023.

[13] Z. Shao *et al.*, "DeepSeekMath: Pushing the limits of mathematical reasoning in open language models," *arXiv preprint*, 2024.

Repository. Code and experiment configs: https://github.com/dhruvb26/CSE485-Capstone. Training metrics dashboard: https://huggingface.co/spaces/dhruvb26/negotiation-agent.