SCORABLE NEGOTIATION SELF-PLAY FOR NATURAL-LANGUAGE REASONING AND UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Can learning to negotiate improve language models beyond negotiation itself? We answer this question affirmatively with *Language Model Self-Play via Scorable Negotiation Game* (LSSG), a post-training framework that transfers skills from multi-turn bargaining to natural-language reasoning and understanding without any downstream reasoning labels. The key idea is to pair open-ended language interaction with continuous outcome rewards computed directly from transaction prices, with no LLM judge. LSSG warm-starts a single buyer–seller model by generalization-aware behavioral cloning on real dialogues, then refines it by stability-aware self-play, which combines payoff-driven updates with supervised retention and semantic diversity and emotional stability regularization from frozen pretrained scorers. Across seven benchmarks spanning commonsense reasoning, textual inference, logical reasoning, sentiment classification, and knowledge-intensive understanding in English and Chinese, LSSG achieves the highest accuracy among all compared methods in every one of the 14 model–benchmark settings on Qwen3-8B and Qwen3-14B, with macro-average gains of 4.61 and 3.14 percentage points over the vanilla models and up to 8.61 points on a single benchmark. Accuracy rises with every self-play iteration, and LSSG further raises win rate and payoff in two interactive negotiation games, also under adversarial social personas, establishing scorable negotiation as a judge-free training environment whose benefits extend well beyond bargaining.

1 INTRODUCTION

A negotiation can unfold through countless different exchanges, yet its outcome is scored exactly: whether an agreement is reached, and at what price (Liu et al., 2026b). This combination of open-ended language interaction and explicit outcome scoring makes negotiation an unusually well-suited setting for learning from experience (Hu et al., 2025; Oh et al., 2026). Instead of prescribing the correct reply at every turn, a training procedure can let the negotiated outcome assign feedback to entire generated interactions (Liu et al., 2026b). In this paper, we ask and answer a concrete question: *can post-training on automatically scored negotiation experience improve language-model performance on natural-language reasoning and understanding tasks beyond negotiation itself?*

This question sits at the center of a broader challenge: obtaining scalable supervision for post-training. Existing approaches rely on expert-written solutions (Cobbe et al., 2021), preference annotations (Touvron et al., 2023; Zhang et al., 2024), synthetic reasoning corpora (Morishita et al., 2024; Peng et al., 2025), or task-specific verifiers (Shao et al., 2024; Guo et al., 2025), all costly to construct and difficult to scale across domains (Villalobos et al., 2024; Silver & Sutton, 2025). Label-free alternatives exploit a model's own outputs (Kuba et al., 2025): Chain-of-Thought prompting elicits intermediate reasoning at inference time (Wei et al., 2022; Kojima et al., 2022), while self-evaluation and multi-agent debate check or compare candidate answers (Miao et al., 2024; Khan et al., 2024; Estornell & Liu, 2024), presupposing a single verifiable answer. Language games offer a complementary route in which the reward comes from the game itself (Yao et al., 2021; Cheng et al., 2024; Liu et al., 2026a). We advance this direction with price negotiation as a source of continuous, automatically computed outcome feedback, and show that its benefits carry over to tasks far removed from the training environment.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

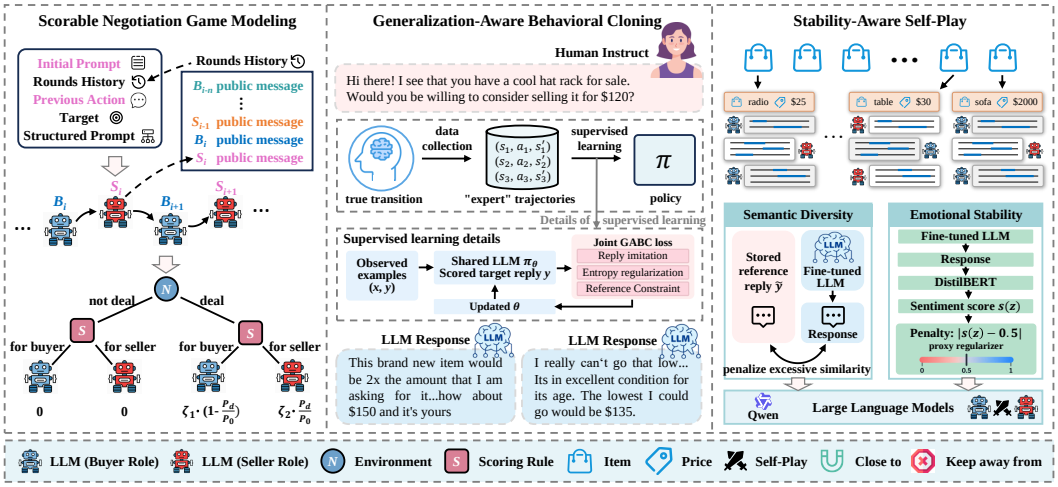


Figure 1: Overview of LSSG: scorable negotiation game modeling (§2) turns bargaining into automatically scored interaction; generalization-aware behavioral cloning (§3.1) warm-starts a shared buyer–seller policy; stability-aware self-play (§3.2) refines it from its own price-based outcomes.

Negotiation is a deliberate choice, motivated by how its training signal is constructed (Oh et al., 2026; Bianchi et al., 2024). Buyer and seller agents exchange natural-language messages and offers until they agree or the interaction ends, and the final transaction price yields a graded, role-specific score: a larger discount raises the buyer's reward, whereas a higher retained price raises the seller's reward (Liu et al., 2026b). This feedback requires no preferred reply, no annotated reasoning trace, and no LLM judgment of dialogue quality, yet the interaction itself demands tracking offers, respecting role-specific constraints, and planning toward agreement over multiple turns (Hong et al., 2026).

We introduce *Language Model Self-Play via Scorable Negotiation Game* (LSSG), a post-training framework that learns from automatically scored negotiation experience without downstream reasoning labels (Figure 1). LSSG combines *generalization-aware behavioral cloning* from real dialogues with *stability-aware self-play*, enabling a single shared buyer–seller model to improve from its own negotiation outcomes. Its central design couples payoff-driven learning with supervised retention and regularization for semantic diversity and emotional stability, while computing every reward directly from transaction prices. On Qwen3-8B and Qwen3-14B, LSSG achieves the highest accuracy among all compared methods in all 14 model–benchmark settings across seven natural-language reasoning and understanding benchmarks, improving macro-average accuracy by 4.61 and 3.14 percentage points over the vanilla models. The same models also win more often and earn higher payoffs in two interactive negotiation games, an advantage that persists under adversarial social personas. Our contributions are as follows:

- We identify *Scorable Negotiation* as a source of post-training supervision that couples open-ended language interaction with explicit, graded, judge-free feedback: transaction-based scores turn self-generated bargaining experience into continuous training feedback with no reasoning labels or learned reward models.
- We propose *Language Model Self-Play via Scorable Negotiation Game* (LSSG), which converts negotiation experience into learning gains well beyond a demonstration warm start. Its stability-aware objective couples role-specific payoff optimization with reference-policy and behavioral regularization, enabling stable, outcome-driven refinement of a shared buyer–seller policy across successive self-play iterations.
- We demonstrate that LSSG delivers the highest accuracy among all compared methods on every one of seven natural-language reasoning and understanding benchmarks at two model scales, with gains that grow with each self-play iteration, while also improving interactive negotiation performance and preserving that advantage under adversarial personas.

2 SCORABLE NEGOTIATION GAME MODELING

We formulate the *Scorable Negotiation Game* (SG) to turn open-ended bargaining into automatically scorable interaction with a continuous, role-specific reward and no external judge. A buyer B and a seller S negotiate over an item with an initial listing price P_0: the buyer seeks a discount, whereas the seller seeks to retain a higher price. The agents communicate through freely generated replies, while the agreement status and final transaction price determine their role-specific rewards.

Interaction rules. In SG, the buyer and seller take turns making offers and exchanging free-form messages. At step t, the active role $i_t \in \{B, S\}$ observes the dialogue history τ_t, giving the interaction state $s_t = (t, i_t, \tau_t)$, where $\tau_t = \{(i_k, a_k^{i_k})\}_{k=1}^{t-1}$ and the initial history is empty. The item description and initial listing price form a fixed context for each negotiation; all states, policies, and rewards are implicitly conditioned on this context. The language model receives a prompt x containing its role, the item information, and the history, and generates a reply token sequence y. The environment interprets the decoded reply as a free-form message together with a negotiation action:

$$\mathcal{A}^i = \{\texttt{offer}(P), \texttt{accept}, \texttt{quit}, \texttt{continue}\} \times \Sigma, \qquad i \in \{B, S\},$$

where $P > 0$ is a proposed price and Σ is the space of messages. Here, `continue` covers replies without a parsed offer, acceptance, or quitting action, including greetings and rejections without a new offer; it is an environment abstraction, not a required output token. A parsed `accept` or `quit` ends the interaction. Otherwise, the transition function $\mathcal{T}$ appends the reply to the history and switches the active role, unless the step limit has been reached. The seller acts first, and each game lasts at most 15 rounds of one seller reply and one buyer reply, or $T = 30$ steps.

Outcome-based scoring. A valid agreement is an accepted transaction with valid positive initial and final prices, $P_0, P_d > 0$. For a completed trajectory τ with a valid agreement, the buyer and seller reward functions assign

$$R_B(\tau) = \zeta_1 \left(1 - \frac{P_d}{P_0}\right), \qquad R_S(\tau) = \zeta_2 \frac{P_d}{P_0},$$

where ζ_1 and ζ_2 are reward-scaling coefficients. The buyer's score measures the relative discount, while the seller's score measures the fraction of the listing price retained. If the interaction ends without a valid agreement, both rewards are zero. These price-based scores measure the buyer's relative discount and the seller's retained price; the buyer's reward is zero at $P_d = P_0$ and negative at $P_d > P_0$. This construction provides continuous-valued, interpretable outcome feedback without an LLM judge, a learned reward model, or a prescribed correct reply at any turn.

Terminal rewards and the game objective. The environment delivers the outcome reward only at the terminal step T_τ; all intermediate rewards are zero. With no temporal discounting ($\gamma = 1$), each role aims to maximize its expected terminal payoff:

$$J_i(\pi^i; \pi^{-i}) = \mathbb{E}_{\tau \sim \Pi}\left[\sum_{t=1}^{T_\tau} r_t^i\right] = \mathbb{E}_{\tau \sim \Pi}[R_i(\tau)], \qquad i \in \{B, S\},$$

where $\Pi = \pi^B \times \pi^S$ is the joint role-conditioned policy and π^{-i} denotes the other role's policy. Appendix H describes how the terminal outcome supplies reply-level return targets for self-play training.

3 LANGUAGE MODEL SELF-PLAY VIA SCORABLE NEGOTIATION GAME

SG provides automatically scored outcomes; LSSG turns these outcomes into an effective post-training signal for the language model. The framework first initializes negotiation behavior through *generalization-aware behavioral cloning* (GABC), then learns from self-generated bargaining experience through *stability-aware self-play* (SASP). Both stages use a single parameter vector θ, with buyer and seller roles specified by their prompts. Thus, π^B and π^S denote two role-conditioned uses of the same model, and their losses contribute to a shared parameter update.

3.1 GENERALIZATION-AWARE BEHAVIORAL CLONING

Human dialogues provide a natural starting point for negotiation, while subsequent self-play requires the policy to generate interactions beyond the demonstrations. GABC therefore combines reply imitation with entropy maximization and a frozen-reference constraint. For a demonstration dialogue, let $\mathcal{T}_i$ denote the decision steps belonging to role $i \in \{B, S\}$, and let π_0^i denote the corresponding role-conditioned pretrained reference policy. The policy-level objective has the same form for both roles:

$$\mathcal{L}^i_{\mathrm{GABC}} = \frac{1}{|\mathcal{T}_i|} \sum_{t \in \mathcal{T}_i} \Big[-\log \pi^i(a^i_t \mid s_t) - \beta^i_H H\big(\pi^i(\cdot \mid s_t)\big) + \beta^i_{\mathrm{KL}} D_{\mathrm{KL}}\big(\pi^i(\cdot \mid s_t) \,\|\, \pi^i_0(\cdot \mid s_t)\big) \Big], \quad (1)$$

where $(\beta^B_H, \beta^B_{\mathrm{KL}}) = (\beta_1, \beta_2)$ and $(\beta^S_H, \beta^S_{\mathrm{KL}}) = (\beta_3, \beta_4)$. The joint policy-level objective averages the two role losses:

$$\mathcal{L}_{\mathrm{GABC}} = \tfrac{1}{2}\mathcal{L}^B_{\mathrm{GABC}} + \tfrac{1}{2}\mathcal{L}^S_{\mathrm{GABC}}.$$

Imitation fits demonstrated replies, entropy regularization discourages an overly concentrated policy, and the reference constraint penalizes departure from the pretrained policy. Together, these terms balance adaptation to bargaining with the flexibility that self-play requires.

Equation 1 specifies the policy-level components. In implementation, imitation uses the mean log-probability of valid reply tokens, entropy is computed over the vocabulary at prediction positions, and the reference constraint uses current and frozen model scores on the same prompt–reply pair. Buyer and seller samples are pooled for training; shared-parameter batching is described in Appendix K. The reference π_0 remains fixed throughout this warm start.

3.2 STABILITY-AWARE SELF-PLAY

SASP shifts the main learning signal from matching human replies to the payoffs of the model's own negotiations; this shift is what carries performance beyond imitation (Sections 4.1 and 4.4). It alternates trajectory collection with offline policy updates, combining role-specific outcome feedback with reference-policy and behavioral regularization.

Collecting negotiation experience. Following Cheng et al. (2024), each iteration freezes a snapshot π_{old} of the current shared model and uses it to play both buyer and seller. We retain valid agreements and split their trajectories into role-specific state–reply datasets $\mathcal{D}^B$ and $\mathcal{D}^S$; filtering criteria are given in Appendix H. The shared policy π_θ is updated offline on these fixed datasets with π_{old} frozen, and the snapshot is refreshed after each training phase.

Assigning role-specific returns. Each retained trajectory supplies a terminal payoff $R_i(\tau)$ for each role, as defined in Section 2. Using undiscounted returns ($\gamma = 1$) and a zero baseline, every reply of role i receives that role's terminal return $\hat{A}^i_t = R_i(\tau)$. Appendix H provides the return estimation details.

Reference-anchored policy updates. For a retained state–reply pair (s, a), let (x, y) be its role-conditioned prompt and reply tokens. We score the reply using its mean token log-probability $\ell_\theta(x, y)$, defined in Equation 8. The corresponding length-normalized reply score is $\tilde{\pi}^i_\theta(a \mid s) = \exp(\ell_\theta(x, y))$, with the reference score $\tilde{\pi}^i_{\mathrm{old}}(a \mid s)$ defined analogously. The difference from the frozen reference, $\Delta\ell_\theta = \ell_\theta(x, y) - \ell_{\mathrm{old}}(x, y)$, defines both the reply-score ratio and the adjusted return weight:

$$\rho^i_\theta(s, a) = \frac{\tilde{\pi}^i_\theta(a \mid s)}{\tilde{\pi}^i_{\mathrm{old}}(a \mid s)} = \exp(\Delta\ell_\theta), \qquad w^i_\theta(s, a) = \hat{A}^i_t - \beta_{\mathrm{sp}}\Delta\ell_\theta,$$

where ρ^i_θ compares length-normalized reply scores and w^i_θ incorporates a KL-inspired reference penalty. We define the base SASP objective for each role as

$$\mathcal{L}^{i,\mathrm{base}}_{\mathrm{SASP}}(\theta) = -\mathbb{E}_{(s,a)\sim\mathcal{D}^i}\left[\rho^i_\theta(s, a) w^i_\theta(s, a)\right], \qquad i \in \{B, S\}, \quad (2)$$

where $\mathcal{D}^i$ denotes the empirical distribution of retained replies for role i. To optimize this objective, we apply clipping to the ratio-weighted update (Schulman et al., 2017), yielding the per-reply loss:

$$\mathcal{L}^{i,\mathrm{reply}}_{\mathrm{SASP}}(\theta; s, a) = -\min\Big\{ \rho^i_\theta(s, a) w^i_\theta(s, a), \mathrm{clip}\big(\rho^i_\theta(s, a), 1 - \epsilon, 1 + \epsilon\big) w^i_\theta(s, a) \Big\}. \quad (3)$$

The coefficients ϵ and β_{sp} control ratio clipping and reference adjustment, respectively. We obtain the role-specific losses $\mathcal{L}^{B}_{\text{SASP}}$ and $\mathcal{L}^{S}_{\text{SASP}}$ by aggregating the clipped reply losses with role-balancing weights, as specified in Appendix I. These losses enter the full training objective in Equation 5.

Supervised retention. To help retain general instruction-following behavior during payoff-driven training, we include an auxiliary instruction corpus $\mathcal{D}_{\text{SFT}}$. For instruction–response pairs (u, v), the retention objective is $\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(u,v)\sim\mathcal{D}_{\text{SFT}}}[\ell_\theta(u, v)]$, using the same mean reply-token log-probability.

Semantic diversity and emotional stability. We supplement price feedback with two text-based regularizers. Let c be an auxiliary generation context, z a generated continuation, and $\tilde{y}$ the reference reply stored in the corresponding training sample. A frozen sentence-BERT encoder δ and a frozen DistilBERT sentiment classifier define

$$C_{\text{SemDi}}(z, \tilde{y}) = \max\big(0, \cos\big(\delta(z), \delta(\tilde{y})\big) - m\big), \qquad C_{\text{EmoSt}}(z) = |s(z) - \bar{s}|,$$

where $m = 0.8$, $s(z)$ is the probability of positive sentiment, and $\bar{s} = 0.5$. The semantic cost penalizes similarity to the stored reply above the margin, while the emotional cost encourages positive-sentiment probabilities toward the midpoint target.

We optimize the auxiliary objectives using score-function surrogates computed from sampled continuations and frozen scorer outputs. For $k \in \{\text{SemDi}, \text{EmoSt}\}$,

$$\widehat{\mathcal{L}}^{\text{PG}}_{k}(\theta; \mathcal{B}) = \frac{1}{Z_{\mathcal{B}}} \sum_{j\in\mathcal{B}_{\text{sp}}} \omega_j \ \text{sg}(C_{k,j})\, \ell_\theta(c_j, z_j), \tag{4}$$

where $C_{k,j}$ is the corresponding cost of continuation z_j, sg stops gradients, $\mathcal{B}_{\text{sp}}$ contains the self-play samples in the batch, and ω_j is the role-balancing weight. $Z_{\mathcal{B}}$ is the batch weight normalizer; its calculation and the role weights are specified in Appendix I. Sampling and gradient computation are detailed in Appendix J.

Joint optimization. The full objective combines the clipped payoff contributions, supervised retention, and the two auxiliary surrogates:

$$\mathcal{L}_{\text{LSSG}}(\theta; \mathcal{B}) = \alpha_1\mathcal{L}^{B}_{\text{SASP}} + \alpha_2\mathcal{L}^{S}_{\text{SASP}} + \alpha_3\mathcal{L}_{\text{SFT}} + \alpha_4\widehat{\mathcal{L}}^{\text{PG}}_{\text{SemDi}} + \alpha_5\widehat{\mathcal{L}}^{\text{PG}}_{\text{EmoSt}}. \tag{5}$$

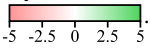
The coefficients $\alpha_1, \ldots, \alpha_5$ control the five contributions to each shared-parameter update. Algorithm 1 summarizes the warm start and the complete self-play training loop.

4 Experiments

We evaluate transfer on seven natural-language reasoning and understanding benchmarks, WinoGrande (Sakaguchi et al., 2021) and CommonsenseQA (CSQA) (Talmor et al., 2019) for commonsense reasoning, CommitmentBank (CB) (De Marneffe et al., 2019) and LogiQA2 (Liu et al., 2023) for textual and logical inference, SST2 (Socher et al., 2013) for sentiment classification, and MedMCQA (Pal et al., 2022) and CMMLU (Li et al., 2024) for knowledge-intensive understanding, together with interactive negotiation in two NegotiationArena games, on Qwen3-8B and Qwen3-14B under the setup in Appendix C.

4.1 Transfer to Language Reasoning and Understanding

Table 1 shows that the benefits of LSSG extend well beyond negotiation. On both Qwen3-8B and Qwen3-14B, LSSG_3 achieves the highest accuracy among all compared methods in every one of the 14 model–benchmark settings, with macro-average gains of 4.61 and 3.14 percentage points over the vanilla models. The margin over the alternatives is decisive: neither CoT prompting nor SFT improves on the vanilla model by more than one percentage point on any benchmark, and CoT lowers WinoGrande accuracy at both scales, whereas LSSG_3 improves every benchmark at both scales.

Table 1: Accuracy (%) on seven language reasoning and understanding benchmarks for Qwen3-8B and Qwen3-14B (±: uncertainty reported by lm-evaluation-harness on its evaluation split). Bold: best per row, LSSG_3 in all 14 settings; accuracy rises from G-SG to LSSG_3 in every row. **V**: Vanilla Qwen3 Model, **CoT**: Chain-of-Thought, **SFT**: SFT directly on V, **G-AG**: Generalization-Aware Behavioral Cloning-Adversarial Taboo Game, **G-NG**: Generalization-Aware Behavioral Cloning-Negotiation Game, **G-SG**: Generalization-Aware Behavioral Cloning-Scorable Negotiation Game, $\textbf{LSSG}_i$: Language Model Self-play via Scorable Negotiation Game trained with i iterations of stability-aware self-play. Table cell colors are assigned relative to the V baseline, in accordance with the colorbar 

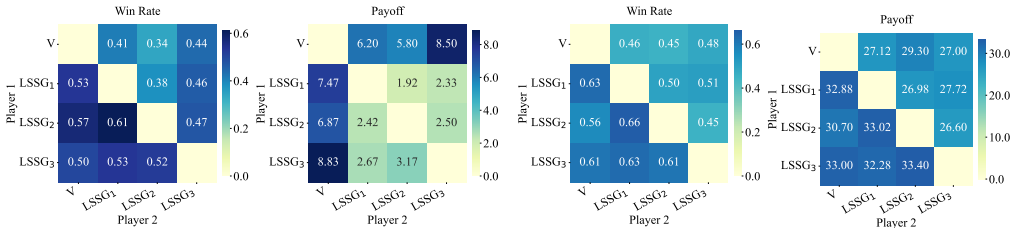
.

Domain	Model	V	CoT	SFT	G-AG	G-NG	G-SG	LSSG_1	LSSG_2	LSSG_3
WinoGrande	Qwen3-8B	67.72 ± 1.31	64.81 ± 1.30	67.83 ± 1.28	70.46 ± 1.31	71.56 ± 1.27	71.98 ± 1.26	72.45 ± 1.25	72.67 ± 1.26	**72.92** ± 1.27
	Qwen3-14B	72.72 ± 1.33	70.18 ± 1.31	72.91 ± 1.30	71.42 ± 1.33	71.88 ± 1.35	72.18 ± 1.39	72.59 ± 1.38	72.91 ± 1.39	**73.24** ± 1.36
CSQA	Qwen3-8B	78.30 ± 1.18	78.55 ± 1.15	78.81 ± 1.13	79.69 ± 1.18	80.42 ± 1.10	81.00 ± 1.12	83.05 ± 1.07	83.41 ± 1.08	**84.73** ± 1.10
	Qwen3-14B	80.26 ± 1.44	80.82 ± 1.41	80.74 ± 1.42	77.55 ± 1.43	83.60 ± 1.44	84.35 ± 1.43	86.04 ± 1.40	86.61 ± 1.44	**86.99** ± 1.43
CB	Qwen3-8B	71.43 ± 6.09	71.26 ± 6.01	71.86 ± 6.05	72.61 ± 5.57	75.10 ± 6.01	76.07 ± 5.46	78.57 ± 5.53	79.79 ± 6.03	**80.04** ± 5.58
	Qwen3-14B	78.57 ± 5.53	77.47 ± 5.51	78.90 ± 5.55	78.11 ± 5.60	79.04 ± 5.54	79.74 ± 5.56	81.00 ± 5.64	82.19 ± 5.61	**82.62** ± 5.59
SST2	Qwen3-8B	90.04 ± 0.93	90.31 ± 0.91	90.75 ± 0.90	90.50 ± 0.94	90.75 ± 0.90	90.83 ± 0.98	91.28 ± 0.96	91.36 ± 0.98	**91.54** ± 0.99
	Qwen3-14B	91.42 ± 0.90	91.72 ± 0.90	91.94 ± 0.91	91.81 ± 0.87	92.18 ± 0.94	92.20 ± 0.96	92.67 ± 0.92	92.74 ± 0.95	**92.90** ± 0.92
LogiQA2	Qwen3-8B	36.72 ± 1.22	35.75 ± 1.27	37.16 ± 1.23	36.15 ± 1.24	37.54 ± 1.22	38.68 ± 1.23	39.05 ± 1.25	40.42 ± 1.22	**41.91** ± 1.20
	Qwen3-14B	37.85 ± 1.22	38.78 ± 1.17	38.30 ± 1.18	39.20 ± 1.20	39.68 ± 1.13	39.85 ± 1.12	40.22 ± 1.16	41.42 ± 1.11	**42.60** ± 1.14
MedMCQA	Qwen3-8B	59.07 ± 0.76	59.36 ± 0.72	59.64 ± 0.71	57.76 ± 0.74	58.53 ± 0.78	58.69 ± 0.76	59.41 ± 0.75	60.54 ± 0.78	**61.69** ± 0.74
	Qwen3-14B	62.04 ± 0.74	62.33 ± 0.78	62.93 ± 0.79	61.74 ± 0.71	61.19 ± 0.77	62.34 ± 0.72	63.19 ± 0.77	63.43 ± 0.74	**63.86** ± 0.70
CMMLU	Qwen3-8B	76.03 ± 0.38	76.58 ± 0.40	76.71 ± 0.42	76.61 ± 0.35	77.04 ± 0.39	77.31 ± 0.38	77.70 ± 0.39	78.12 ± 0.37	**78.76** ± 0.38
	Qwen3-14B	80.08 ± 0.36	80.31 ± 0.39	80.66 ± 0.33	80.83 ± 0.33	80.96 ± 0.34	81.02 ± 0.37	81.65 ± 0.35	82.17 ± 0.32	**82.74** ± 0.34

Commonsense and textual reasoning. The largest improvements appear on the tasks that most directly demand reasoning over context. Over the vanilla 8B and 14B models, CSQA gains 6.43 and 6.73 percentage points, CB gains 8.61 and 4.05 points, and LogiQA2 gains 5.19 and 4.75 points; WinoGrande improves by 5.20 points at 8B and by a smaller 0.52 points at 14B. Transfer is thus consistent across all four reasoning benchmarks at both scales, with the size of the gain depending on the task and backbone.

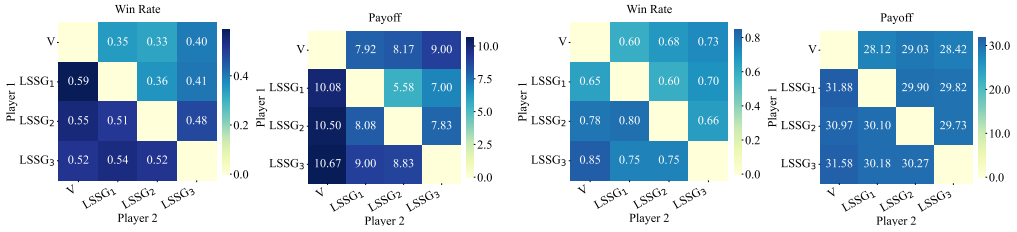
Gains beyond behavioral cloning. The stage-wise comparisons show that improvement continues well beyond the demonstration warm start. G-SG already outperforms the other game-based cloning baselines, G-AG and G-NG, in every setting. More importantly, accuracy increases from G-SG through LSSG_1, LSSG_2, and LSSG_3 on every benchmark at both scales: in macro-average terms, self-play adds 2.43 points at 8B and 1.90 points at 14B on top of G-SG, exceeding the 2.18 and 1.25 points that behavioral cloning itself adds over the vanilla model. This progression also covers the settings where G-SG falls below the vanilla model, WinoGrande at 14B and MedMCQA at 8B; in both, self-play recovers the loss and carries LSSG_3 past the vanilla baseline. The additional gains therefore come from outcome-driven self-play rather than imitation alone, as the ablation in Section 4.4 confirms.

Broader language understanding. The benefits extend to knowledge-intensive and cross-lingual understanding: LSSG_3 improves MedMCQA by 2.62 and 1.82 points and CMMLU by 2.73 and 2.66 points at 8B and 14B, so self-play on English bargaining dialogues transfers to medical multiple-choice questions and to Chinese multitask understanding. These gains are smaller than those on CSQA, CB, and LogiQA2, yet positive in every setting; SST2 also improves at both scales, noting that the auxiliary sentiment scorer was fine-tuned on SST-2 (Appendix E).



(a) Qwen3-8B. Seller & Buyer Game. Win Rate and Payoff.

(b) Qwen3-8B. Resource Exchange Game. Win Rate and Payoff.



(c) Qwen3-14B. Seller & Buyer Game. Win Rate and Payoff.

(d) Qwen3-14B. Resource Exchange Game. Win Rate and Payoff.

Figure 2: Interactive negotiation on NegotiationArena (rows: Player 1 model; columns: Player 2 opponent; cells: Player 1's win rate or average payoff over 60 runs). LSSG_3 obtains a higher win rate and payoff against vanilla than vanilla obtains in the reverse pairing, in both games at both scales.

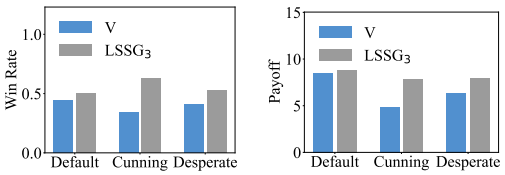
4.2 Negotiation Performance

We next evaluate LSSG through direct interaction with other models, complementing the downstream benchmarks in Section 4.1. Following Bianchi et al. (2024), we use two NegotiationArena scenarios, the Seller & Buyer Game and the Resource Exchange Game, with 60 runs per setting on Qwen3-8B and Qwen3-14B. Both games differ from the training environment in rules and payoff structure, so this evaluation tests generalization to new negotiation formats rather than performance on the training game. In Figure 2, rows identify the model used by Player 1 and columns its Player 2 opponent; each visible cell reports Player 1's win rate or average payoff for that ordered pairing.

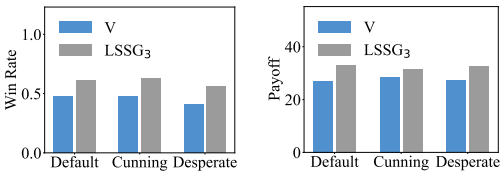
Seller & Buyer. The Seller & Buyer Game involves asymmetric private information: the seller observes production cost, while the buyer observes willingness to pay. In Figure 2(a,c), LSSG_3 as Player 1 obtains a higher win rate and payoff against the vanilla model than the vanilla model obtains in the reverse pairing, at both scales and with a larger gap at 14B. The checkpoint comparisons also show why both metrics matter: against the vanilla opponent at 14B, payoff increases from LSSG_1 through LSSG_3 whereas win rate decreases, so later checkpoints negotiate more profitable agreements on average even when they do not win more often.

Resource Exchange. The Resource Exchange Game requires negotiating trades across multiple resources under different initial endowments, a format that departs entirely from price bargaining, yet LSSG_3 improves both metrics here as well (Figure 2(b,d)). For Qwen3-8B, LSSG_3 as Player 1 achieves a win rate of 0.61 and a payoff of 33.00 against vanilla, compared with 0.48 and 27.00 for vanilla as Player 1 in the reverse pairing, a 13-point win-rate margin and a 22% higher payoff; the 14B entries favor LSSG_3 on both metrics as well.

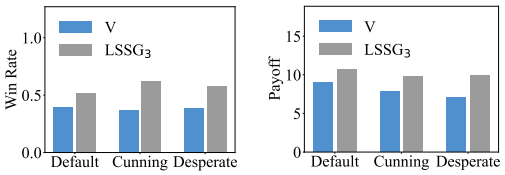
Overall, LSSG_3 outperforms the vanilla model on both metrics in both scenarios at both scales.



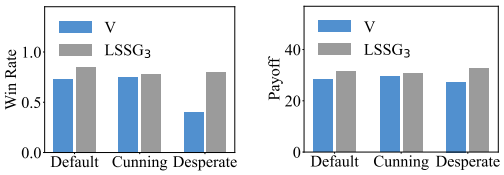
(a) Qwen3-8B. Seller & Buyer Game. Win Rate and Payoff.



(b) Qwen3-8B. Resource Exchange Game. Win Rate and Payoff.



(c) Qwen3-14B. Seller & Buyer Game. Win Rate and Payoff.



(d) Qwen3-14B. Resource Exchange Game. Win Rate and Payoff.

Figure 3: Negotiation stability of LSSG_3 versus the vanilla model (V) under default, cunning, and desperate personas. LSSG_3 retains higher win rate and payoff than V in every condition, game, and scale.

4.3 Negotiation Stability

We examine whether the negotiation advantage observed in Section 4.2 persists when the social style of the interaction changes. Following Bianchi et al. (2024), we evaluate two adversarial personas in addition to the default setting: *cunning*, which instructs agents to include an insult while making a strategic offer or response, and *desperate*, which asks them to express desperation in pursuit of a better deal. Both agents receive the same persona prompt, so the comparison isolates robustness to the social perturbation; we compare LSSG_3 with the vanilla model on both backbones and both games using the same metrics as before.

Figure 3 shows that LSSG_3 retains higher win rates and average payoffs than the vanilla model in every persona condition, in both games, and at both scales. The two models respond to the perturbations in opposite directions: in the Seller & Buyer Game, the vanilla model's win rate drops under the cunning persona relative to the default setting at both scales, whereas LSSG_3's win rate rises. The sharpest contrast appears under the desperate persona in the 14B Resource Exchange Game, where the vanilla model's win rate falls substantially while LSSG_3 stays close to its default level. LSSG is therefore not only stronger under neutral conditions but also markedly more robust to adversarial social behavior.

4.4 Ablation Study

We examine how the SASP objective and the two auxiliary regularizers contribute to downstream performance. **LSSG_3** uses the complete training objective with three self-play iterations. **w/o** $\mathcal{L}_{\text{SASP}}$ removes the buyer and seller SASP loss terms by setting $\alpha_1 = \alpha_2 = 0$, while retaining supervised retention and both auxiliary regularizers during subsequent training. **w/o** $\mathcal{L}_{\text{SemDi}}$ sets $\alpha_4 = 0$ and retains SASP, supervised retention, and emotional stability regularization. **w/o** $\mathcal{L}_{\text{EmoSt}}$ sets $\alpha_5 = 0$ and retains SASP, supervised retention, and semantic diversity regularization. **w/o** $\mathcal{L}_{\text{SemDi}} + \mathcal{L}_{\text{EmoSt}}$ sets $\alpha_4 = \alpha_5 = 0$, retaining SASP and supervised retention. Figure 4 compares the full three-iteration LSSG_3 model with its ablated variants on Qwen3-8B across all seven benchmarks. Supervised retention is kept in every variant; Table 1 separately reports the SFT-only baseline.

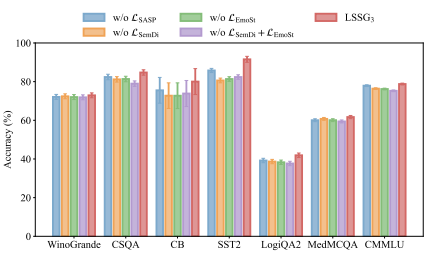


Figure 4: Ablation on Qwen3-8B: removing SASP, either regularizer, or both lowers accuracy on every benchmark.

SASP is essential beyond auxiliary training. Removing the buyer and seller SASP loss terms ($\alpha_1 = \alpha_2 = 0$) lowers accuracy on every one of the seven benchmarks. This variant starts from the

same G-SG initialization and follows the same training schedule while retaining supervised retention and both regularizers, so the payoff-driven objective is the only difference: with everything else held fixed, outcome-based self-play raises accuracy on all seven benchmarks.

Both regularizers add further gains. Removing semantic diversity or emotional stability regularization also reduces accuracy across all seven benchmarks even with SASP retained, and the full model outperforms the variant with both regularizers removed; each term thus contributes on top of the other, with task-dependent magnitude, so every component of the LSSG objective is needed for its full effect.

5 Related Work

Reasoning through feedback and post-training. Chain-of-thought prompting and self-consistency improve inference through intermediate reasoning and answer aggregation (Wei et al., 2022; Wang et al., 2023), while Self-Refine and Reflexion use model-generated feedback to revise outputs or guide later attempts (Madaan et al., 2023; Shinn et al., 2023). Recent work also incorporates feedback into parameter updates: ReGenesis synthesizes reasoning paths for post-training (Peng et al., 2025), and DeepSeek-R1 develops reasoning through reinforcement learning, including rule-based rewards for verifiable tasks (Guo et al., 2025). However, these approaches obtain feedback from candidate solutions, generated critiques, or task-specific verification; they do not develop negotiated economic outcomes as a source of post-training supervision.

Self-play and language-game training. SPIN contrasts demonstrations with model-generated responses (Chen et al., 2024); SPAG trains through self-play in Adversarial Taboo (Cheng et al., 2024). Absolute Zero couples task proposal and solving through code-execution rewards (Zhao et al., 2026), and Language Self-Play develops competitive training without additional data (Kuba et al., 2025). SPIRAL demonstrates reasoning transfer from multi-turn zero-sum games, including Simple Negotiation (Liu et al., 2026a). STRATAGEM explicitly promotes transfer through trajectory weighting and rewards for reasoning evolution (Feng et al., 2026). Playpen studies learning from dialogue-game feedback (Horst et al., 2025), while GameTalk adapts fine-tuning methods to rewards defined over entire conversations (Conchello Vendrell et al., 2026). However, these studies leave the joint role of response-diversity and sentiment regularization in price-based self-play insufficiently explored, particularly in relation to transfer beyond the training games.

Negotiation-based learning and evaluation. CraigslistBargain supports learning from human negotiation dialogues (He et al., 2018), and NegotiationArena evaluates bargaining behavior and social influences (Bianchi et al., 2024). ASTRO optimizes strategy planners through self-play for non-cooperative dialogues (Hu et al., 2025). MERIT combines economically grounded utility feedback with prompting and fine-tuning to improve bargaining (Oh et al., 2026). Liu et al. (2026b) train sellers in multi-buyer markets using verifiable rewards tied to transaction surplus, demonstrating the value of economic outcomes as training signals. However, their evaluations center on negotiation performance and strategy, leaving the transfer of bargaining-trained policies to broader natural-language reasoning and understanding tasks untested. Appendix A provides the full discussion of related work.

6 Conclusion

We introduced LSSG, a demonstration-warm-started self-play framework that turns automatically scorable negotiation outcomes into an effective post-training signal. Multi-turn bargaining provides continuous outcome-based supervision, with every reward computed directly from transaction prices without an LLM judge or downstream reasoning labels, and stability-aware self-play combines these rewards with supervised retention and auxiliary regularization from frozen pretrained scorers. On Qwen3-8B and Qwen3-14B, LSSG achieves the highest accuracy among all compared methods in all 14 model–benchmark settings across seven natural-language reasoning and understanding benchmarks, with macro-average gains of 4.61 and 3.14 percentage points and improvements that grow with each self-play iteration, and it wins more often and earns higher payoffs in two interactive negotiation games, also under adversarial personas. These findings establish scorable negotiation self-play as a label-free, judge-free route to improving language models well beyond the setting in which they are trained.

AI USE STATEMENT

We used generative AI tools for language polishing and code assistance. The authors take full responsibility for the final content of this work.

REFERENCES

Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. How well can llms negotiate? NegotiationArena platform and analysis. In *Forty-first International Conference on Machine Learning*, pp. 3935–3951, 2024.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*, Proceedings of Machine Learning Research, pp. 6621–6642, 2024.

Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. Self-playing adversarial language game enhances llm reasoning. *Advances in Neural Information Processing Systems*, 37:126515–126543, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Victor Conchello Vendrell, Max Ruiz Luyten, and Mihaela van der Schaar. GameTalk: Training LLMs for strategic conversation. *arXiv preprint arXiv:2601.16276*, 2026.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, pp. 107–124, 2019.

Andrew Estornell and Yang Liu. Multi-llm debate: Framework, principals, and interventions. *Advances in Neural Information Processing Systems*, 37:28938–28964, 2024.

Xiachong Feng, Deyi Yin, Xiaocheng Feng, Yi Jiang, Libo Qin, Yangfan Ye, Lei Huang, Weitao Ma, Qiming Li, Yuxuan Gu, Bing Qin, and Lingpeng Kong. STRATAGEM: Learning transferable reasoning via trajectory-modulated game self-play. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 19595–19614, 2026.

Yao Fu, Hao Peng, Tushar Khot, and Mirella Lapata. Improving language model negotiation with self-play and in-context learning from AI feedback. *arXiv preprint arXiv:2305.10142*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 07 2024. URL `https://zenodo.org/records/12608602`.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. Decoupling strategy and generation in negotiation dialogues. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2333–2343, 2018.

Joey Hong, Anca Dragan, and Sergey Levine. Planning without search: Refining frontier LLMs with offline goal-conditioned RL. In *Advances in Neural Information Processing Systems*, pp. 10032–10056, 2026.

Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. Playpen: An environment for exploring learning from dialogue game feedback. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29854–29891, 2025.

Yikuan Hu, Chen Huang, and Wenqiang Lei. ASTRO: Automatic strategy optimization for non-cooperative dialogues. In *Findings of the Association for Computational Linguistics: ACL*, pp. 388–408, 2025.

Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 23662–23733, 2024.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35: 22199–22213, 2022.

Jakub Grudzien Kuba, Mengting Gu, Qi Ma, Yuandong Tian, Vijai Mohan, and Jason Chen. Language self-play for data-free training. *arXiv preprint arXiv:2509.07414*, 2025.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in chinese. In *Findings of the Association for Computational Linguistics ACL 2024*, pp. 11260–11285, 2024.

Austen Liao, Nicholas Tomlin, and Dan Klein. Efficacy of language model self-play in non-zero-sum games. *arXiv preprint arXiv:2406.18872*, 2024.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, et al. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, pp. 35407–35434, 2026a.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962, 2023.

Shuze Daniel Liu, Claire Chen, Jiabao Sean Xiao, Xin Chen, and David Simchi-Levi. Strategic bargaining in multi-buyer markets: Reinforcement learning from verifiable rewards for LLM negotiations. *arXiv preprint arXiv:2607.05863*, 2026b.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. *arXiv preprint arXiv:2303.17651*, 2023.

Ning Miao, Yee Whye Teh, and Tom Rainforth. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. In *Proceedings of the Twelfth International Conference on Learning Representations (ICLR 2024)*. International Conference on Learning Representations, ICLR, 2024.

Terufumi Morishita, Gaku Morio, Atsuki Yamaguchi, and Yasuhiro Sogawa. Enhancing reasoning capabilities of llms via principled synthetic logic corpus. *Advances in Neural Information Processing Systems*, 37:73572–73604, 2024.

Jihwan Oh, Murad Aghazada, Yooju Shin, Se-Young Yun, and Taehyeon Kim. MERIT feedback elicits better bargaining in LLM negotiators. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 25202–25223, 2026.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on health, inference, and learning*, pp. 248–260. PMLR, 2022.

Xianyu Peng, Congying Xia, Xinyi Yang, Caiming Xiong, Chien-Sheng Wu, and Chen Xing. Regenesis: Llms can grow into reasoning generalists via self-improvement. In *The Thirteenth International Conference on Learning Representations*, 2025.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *arXiv preprint arXiv:2303.11366*, 2023.

David Silver and Richard S Sutton. Welcome to the era of experience. *Google AI*, 1, 2025.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, 2019.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Pablo Villalobos, Anson Ho, Jaime Sevilla, Tamay Besiroglu, Lennart Heim, and Marius Hobbhahn. Position: Will we run out of data? limits of llm scaling based on human-generated data. In *Forty-first International Conference on Machine Learning*, 2024.

Ru Wang, Wei Huang, Qi Cao, Yusuke Iwasawa, Yutaka Matsuo, and Jiaxian Guo. Self-harmony: Learning to harmonize self-supervision and self-play in test-time reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Yuxiang Wei, Zhiqing Sun, Emily McMilin, Jonas Gehring, David Zhang, Gabriel Synnaeve, Daniel Fried, Lingming Zhang, and Sida Wang. Toward training superintelligent software agents through self-play swe-rl. In *The Forty-Third International Conference on Machine Learning*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yuan Yao, Haoxi Zhong, Zhengyan Zhang, Xu Han, Xiaozhi Wang, Kai Zhang, Chaojun Xiao, Guoyang Zeng, Zhiyuan Liu, and Maosong Sun. Adversarial language games for advanced natural language intelligence. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 14248–14256, 2021.

Xuan Zhang, Chao Du, Tianyu Pang, Qian Liu, Wei Gao, and Min Lin. Chain of preference optimization: Improving chain-of-thought reasoning in llms. *Advances in Neural Information Processing Systems*, 37:333–356, 2024.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In *Advances in Neural Information Processing Systems*, pp. 105816–105879, 2026.

Appendix

Table of Contents

A Related Work 15

B Scope of Natural-Language Reasoning and Understanding 16

C Experimental Setup 16

D Limitations 17

E Training and evaluation sources used in LSSG 18

F Prompt Details 18

G Self-Play Examples 19

H Role-Specific Return and Baseline Estimation 20

I Clipped Offline Self-Play Objective 21

J Optimization of Auxiliary Losses 22

K Shared Parameters and GABC Batch Aggregation 23

A Related Work

Inference-time reasoning and feedback. Chain-of-thought prompting elicits intermediate reasoning steps, including through zero-shot instructions (Wei et al., 2022; Kojima et al., 2022). Self-consistency samples multiple reasoning paths and aggregates their answers (Wang et al., 2023). Self-Refine generates feedback and revises outputs iteratively (Madaan et al., 2023), while Reflexion stores textual reflections in episodic memory to guide subsequent attempts (Shinn et al., 2023). Self-Check evaluates intermediate reasoning steps (Miao et al., 2024), and debate introduces competing arguments for an evaluator to assess (Khan et al., 2024). These approaches differ in how they elicit and assess reasoning, but their feedback is used primarily to select or revise responses at inference time. They do not turn the economic outcomes of repeated bargaining into parameter updates that can be evaluated on separate reasoning tasks.

Post-training with synthetic and verifiable feedback. Synthetic-data methods construct explicit reasoning supervision: Additional Logic Training uses program-generated logical deductions (Morishita et al., 2024), while ReGenesis synthesizes reasoning paths through self-improvement (Peng et al., 2025). Chain of Preference Optimization derives step-level preferences from tree-search trajectories (Zhang et al., 2024). DeepSeekMath and DeepSeek-R1 employ reinforcement learning to improve reasoning, with DeepSeek-R1 using rule-based correctness and format rewards for its initial reasoning training (Shao et al., 2024; Guo et al., 2025). Self-Harmony extends adaptation to test time, deriving pseudo-labels from answer agreement across original and reframed questions (Wang et al., 2026). These methods reduce different forms of annotation dependence, but their supervision remains tied to reasoning examples, solution comparisons, or answer verification and consistency. They leave economic outcomes from open-ended bargaining unexplored as a source of reasoning post-training signals.

Self-generated curricula and self-play. SPIN iteratively contrasts human demonstrations with responses from earlier model iterations, retaining the demonstration distribution as its training target (Chen et al., 2024). Absolute Zero jointly trains a task proposer and solver, using code execution to validate generated tasks and answers (Zhao et al., 2026). Language Self-Play formulates model improvement as competitive self-play without additional training data (Kuba et al., 2025). Self-play SWE-RL extends this direction to software engineering: an agent learns to inject and repair bugs in sandboxed repositories, with bugs specified through test patches (Wei et al., 2026). Together, these studies show how models can generate new learning experience, but they do not investigate the combination of price-based bargaining payoffs, response-diversity regularization, and sentiment regularization for transfer to natural-language reasoning tasks.

Language games and reasoning transfer. SPAG trains language models through self-play in Adversarial Taboo and evaluates downstream reasoning (Cheng et al., 2024). SPIRAL develops online, multi-agent, multi-turn reinforcement learning with role-conditioned advantage estimation; its games include TicTacToe, Kuhn Poker, and Simple Negotiation, with transfer evaluated on reasoning benchmarks (Liu et al., 2026a). STRATAGEM further targets transfer by weighting trajectories according to reasoning transferability and rewarding reasoning evolution (Feng et al., 2026). Playpen compares supervised fine-tuning, DPO, and GRPO using dialogue-game feedback and examines generalization (Horst et al., 2025). GameTalk adapts GRPO, DPO, and STaR to rewards defined over complete strategic conversations (Conchello Vendrell et al., 2026). These results establish game-based learning as a route to broader capabilities. However, the joint contribution of response-diversity and sentiment regularization during repeated price-bargaining updates remains insufficiently explored, including its effect on downstream reasoning and understanding.

Learning negotiation strategies. CraigslistBargain provides human buyer–seller dialogues and separates strategic decisions from language generation (He et al., 2018). Fu et al. (2023) improve negotiated prices through self-play and in-context learning from an LLM critic's textual feedback. Liao et al. (2024) study repeated filtered behavioral cloning in Deal or No Deal under cooperative, competitive, and mixed objectives. ASTRO generates strategy sets and optimizes a planner through self-play reinforcement learning for non-cooperative dialogues (Hu et al., 2025). More recently, MERIT combines economically grounded utility feedback with prompting and fine-tuning to improve bargaining performance (Oh et al., 2026). Liu et al. (2026b) train sellers in multi-buyer markets with

verifiable terminal rewards based on transaction surplus, learning to balance market exploration and deal closure. These approaches establish several ways to learn from negotiation, but their evaluations focus on bargaining outcomes and strategy rather than testing transfer to broader natural-language reasoning and understanding benchmarks.

Negotiation evaluation and multi-turn planning. NegotiationArena evaluates language models through interactive negotiation games and studies how social behaviors affect outcomes (Bianchi et al., 2024). MERIT's AgoraBench expands evaluation to challenging bargaining settings and economically grounded measures of agent utility (Oh et al., 2026). Hong et al. (2026) use goal-conditioned value functions to guide reasoning in multi-turn interactive tasks, providing a complementary approach to long-horizon decision-making. These evaluations and planning methods characterize interactive competence, but success within a negotiation environment alone does not establish that training on its outcomes improves performance on independent reasoning tasks.

B Scope of Natural-Language Reasoning and Understanding

Table 2: Evaluation scope. LSSG is evaluated across five task groups covering commonsense reasoning, textual and logical inference, sentiment understanding, knowledge-intensive understanding in medical and Chinese-language domains, and interactive negotiation; LSSG_3 obtains the best accuracy on every benchmark and outperforms the vanilla model in every interaction setting.

Evaluation focus	Benchmarks	Task demands	Interpretation of results
Commonsense reasoning	WinoGrande, CSQA	Resolve coreference or answer questions using context and commonsense knowledge.	Accuracy on the evaluated commonsense tasks.
Textual and logical reasoning	CB, LogiQA2	Infer a proposition's status or select an answer supported by natural-language premises.	Accuracy on the evaluated textual inference and logical reasoning tasks.
Sentiment understanding	SST2	Classify the sentiment expressed in a text.	Sentiment-classification accuracy.
Knowledge-intensive understanding	MedMCQA, CMMLU	Answer medical questions or subject-specific questions in Chinese.	Broader transfer involving knowledge use and language understanding.
Interactive negotiation	Seller & Buyer, Resource Exchange	Bargain over prices or resource exchanges under role-specific objectives.	Win rate and payoff in the evaluated interaction settings.

C Experimental Setup

Backbones. We evaluate LSSG on two recent open-weight backbones at different scales, Qwen3-8B and Qwen3-14B (Yang et al., 2025), and observe consistent gains at both. For all LSSG variants, buyer and seller are implemented as a single shared-parameter language model conditioned on role prompts, as described in Section 3. For downstream benchmark evaluation, a single shared model produces predictions under a fixed inference prompt.

Training data. For negotiation training, we build on the CraigslistBargain dataset (He et al., 2018), which contains real buyer–seller bargaining dialogues. To increase the diversity of items used during self-play, we further crawl 105,267 product records from `sfbay.craigslist.org`, each containing an item name and an initial listing price. The real dialogues are used to construct state–role–action triples for generalization-aware behavioral cloning, while the expanded product pool is used to instantiate new Scorable Negotiation Game episodes during self-play. We additionally use Alpaca (Taori et al., 2023) as a small auxiliary instruction corpus for supervised retention. This corpus is used only to preserve general instruction-following behavior and does not contain labels from any downstream evaluation benchmark. The label status for each data source is shown in Appendix E.

Downstream language reasoning and understanding tasks. We evaluate transfer on seven natural-language benchmarks chosen to span distinct task demands, languages, and domains. WinoGrande

(Sakaguchi et al., 2021) and CSQA (Talmor et al., 2019) assess commonsense coreference and commonsense question answering; CB (De Marneffe et al., 2019) and LogiQA2 (Liu et al., 2023) assess textual inference and logical reading comprehension. We further evaluate broader transfer using SST2 (Socher et al., 2013) for sentiment classification, MedMCQA (Pal et al., 2022) for medical multiple-choice question answering, and CMMLU (Li et al., 2024) for Chinese multitask language understanding. Together these groups cover reasoning, language understanding, and knowledge-intensive task demands in both English and Chinese, none of which is targeted by the negotiation training data.

Baselines. We compare LSSG with three groups of baselines. First, we include general model baselines: **Vanilla Model (V)**, the unmodified Qwen3 (Yang et al., 2025) backbone; **SFT**, which directly fine-tunes the vanilla model on the auxiliary instruction corpus; and **Chain-of-Thought (CoT)**, a Chain-of-Thought prompting baseline (Wei et al., 2022). Second, we include game-based behavioral-cloning baselines that isolate the effect of the interaction environment: **G-AG**, which applies the same generalization-aware behavioral cloning objective to the Adversarial Taboo Game (Cheng et al., 2024); **G-NG**, which applies it to the classic Negotiation Game (He et al., 2018); and **G-SG**, which applies it to our Scorable Negotiation Game. Third, we evaluate **LSSG**$_i$ for $i \in \{1, 2, 3\}$, where i denotes the number of stability-aware self-play iterations after G-SG initialization. This comparison cleanly separates the contributions of supervised retention, behavioral cloning, the scorable game formulation, and iterative self-play refinement, so that each gain reported in Table 1 can be attributed to a specific stage of LSSG.

Evaluation settings and metrics. For downstream benchmark evaluation, we report accuracy. Following standard practice, we use a 3-shot setting for CB, a 5-shot setting for CMMLU, and zero-shot evaluation for all remaining tasks. For interactive negotiation evaluation, we use the Seller & Buyer Game and the Resource Exchange Game from NegotiationArena (Bianchi et al., 2024). We report win rate and average payoff over 60 runs for each setting. Win rate measures the fraction of games in which Player 1 obtains a strictly higher utility than Player 2, with ties excluded; average payoff measures the mean utility obtained per game.

Implementation details. Unless otherwise specified, our training setup follows prior self-play language-game training practice (Cheng et al., 2024). In the scorable negotiation modeling phase, $\zeta_1 = \zeta_2 = 1$. In the generalization-aware behavioral cloning phase, we set $\beta_1 = \beta_3 = 0.01$, $\beta_2 = \beta_4 = 0.1$ and the learning rate to 5e-6. In the stability-aware self-play phase, we set $\beta_{\text{sp}} = 0.2$, $\epsilon = 0.2$, $\alpha_1 = \alpha_2 = 1$, $\alpha_3 = 0.5$, $\alpha_4 = \alpha_5 = 0.2$ and the learning rate to 2e-6. The clipping threshold ϵ and reference-policy deviation coefficient β_{sp} have distinct roles; their definitions and the role-weighted update are given in Appendix I. For semantic diversity, we set the margin to $m = 0.8$; for emotional stability, we use the sentiment-probability midpoint target of 0.5. All experiments are conducted on a workstation with 8 NVIDIA GeForce RTX 3090 GPUs, each with 24GB memory. We evaluate downstream benchmark performance using lm-evaluation-harness (Gao et al., 2024) and interactive negotiation performance using NegotiationArena (Bianchi et al., 2024).

Negotiation trajectory sampling. For negotiation trajectory generation, we use stochastic decoding with temperature 1.2 and a maximum of 256 newly generated tokens per reply. The collection script selects $1{,}000$ item records per pass and generates one negotiation trajectory per record through alternating buyer–seller turns.

D LIMITATIONS

Despite the strong and consistent results, our work has limitations that point to natural extensions. First, the supervision signal is instantiated in a specific scorable bargaining setup, where rewards are defined through price concessions and deal outcomes. This design provides continuous, interpretable, and judge-free feedback, and our results already show that it transfers to two negotiation formats not used in training and to seven benchmarks across five task groups; extending the same scoring principle to other scorable language games and domains is a promising direction that we have not yet validated. Second, LSSG adds training complexity through behavioral cloning, self-play, and auxiliary regularization. This increases computational cost relative to a single supervised pass and requires hyperparameter tuning, although a single configuration sufficed for all benchmarks and both backbones in our experiments. We leave a more thorough analysis of training efficiency, robustness across additional model families, and broader transfer settings to future work.

Evaluation scope and future analysis. Our experiments cover two scales of the Qwen3 family, seven natural-language reasoning and understanding benchmarks spanning five task groups in two languages, two interactive negotiation games, and three social persona conditions. Future work will extend evaluation to additional model families and to mathematical and symbolic reasoning, and will use process-level analyses to examine how state tracking, constraint satisfaction, planning, and knowledge use develop during transfer.

E TRAINING AND EVALUATION SOURCES USED IN LSSG

Table 3: Training and evaluation sources used in LSSG, including supervised warm-start and retention data, judge-free negotiation outcome rewards, and frozen pretrained scorers for auxiliary regularization.

Source	Used for	Supervision type	Feedback source
CraigslistBargain dialogues	Behavioral-cloning warm-start	Human dialogue demonstrations / action supervision	Human bargaining dialogues
Craigslist product records	Self-play game instantiation	Item metadata, no reasoning labels	Product listings
Self-play negotiation outcomes	Policy improvement	Automatic price-based outcome reward	Transaction prices and game rules
Alpaca	Supervised retention	Instruction–response supervision	Alpaca instruction pairs
Frozen sentence encoder	Semantic similarity scoring	Pretrained sentence-pair representation learning	Pretrained sentence encoder
Frozen sentiment classifier	Emotional stability scoring	SST-2 supervised fine-tuning	SST-2 training annotations
Downstream benchmark examples	Direct use in evaluation	Benchmark labels used for evaluation	Benchmark annotations

LSSG uses judge-free negotiation outcome rewards computed from transaction prices. Auxiliary regularization uses the frozen sentence encoder `all-MiniLM-L6-v2` and the frozen sentiment classifier `distilbert-base-uncased-finetuned-sst-2-english`. The sentiment classifier was fine-tuned on SST-2 and provides an indirect source of task supervision through its scores on generated continuations.

F PROMPT DETAILS

We provide the seller and buyer prompt templates used for Scorable Negotiation Game interactions.

The Prompt of Seller

Welcome to the Scorable Negotiation Game! This game involves two players: a buyer and a seller. At the start, the seller provides item details, including its name and starting price, which the buyer reviews.

The buyer's aim is to secure the item for less than the starting price or negotiate additional benefits. Reaching an agreement is crucial. If no valid agreement is reached, both players receive zero reward. The seller, on the other hand, tries to sell the item close to its original price, possibly by offering minor incentives. Both players aim to reach an agreement. If no valid agreement is reached, both players receive zero reward.

When one side says `{"action": "accept"}`, the game concludes and both players receive rewards determined by the final agreed price. The game can continue for up to {max_turns} turns, but if no valid agreement is reached, both players receive zero reward.

Game History: {history}.

You are the seller. The name of the item is {item}. The price of the item is {price}. Provide your response for the next turn.

The Prompt of Buyer

Welcome to the Scorable Negotiation Game! This game involves two players: a buyer and a seller. At the start, the seller provides item details, including its name and starting price, which the buyer reviews.

The buyer's aim is to secure the item for less than the starting price or negotiate additional benefits. Reaching an agreement is crucial. If no valid agreement is reached, both players receive zero reward. The seller, on the other hand, tries to sell the item close to its original price, possibly by offering minor incentives. Both players aim to reach an agreement. If no valid agreement is reached, both players receive zero reward.

When one side says `{"action": "accept"}`, the game concludes and both players receive rewards determined by the final agreed price. The game can continue for up to {max_turns} turns, but if no valid agreement is reached, both players receive zero reward.

Game History: {history}.

You are the buyer. The name of the item is {item}. The price of the item is {price}. Provide your response for the next turn.

G SELF-PLAY EXAMPLES

Table 4: A self-play example between a buyer and a seller generated by the G-SG model based on Qwen3-8B.

Item	3ft tall modern table floor lamp open iron cage with shade
Listed Price	$48.0
Seller	Hi how are you.
Buyer	I am good. How can I help you?
Seller	I have a 3 foot table lamp for sale, interested?
Buyer	I am very interested. Is there any way you could come down to 42.50?
Seller	Yes I could come down to 45
Buyer	Ok, it is a deal {'action': 'offer', 'price': 45.0}
Seller	{'action': 'accept'}

Table 5: A self-play example between a buyer and a seller generated by the LSSG_1 model based on Qwen3-8B.

Item	Monkey Sleepying Bag
Listed Price	$25.0
Seller	hi
Buyer	hi..
Seller	how are you? any questions about the sleeping bag?
Buyer	what is the lowest you can go..
Seller	I was thinking $22 was lowest I could go, what do you think?
Buyer	deal 22$...! I accept {'action': 'accept'}

Table 6: A self-play example between a buyer and a seller generated by the LSSG_2 model based on Qwen3-8B.

Item	Air Fryer Toaster Oven Combo - Fabuletta 10-in-1 Countertop Convection
Listed Price	$98.0
Seller	Hello.
Buyer	Hey you still selling the air feyer?
Seller	Yes, I am!
Buyer	Would you be willing to accept $92 cash?
Seller	Deal!
Buyer	{'action': 'accept'}

Table 7: A self-play example between a buyer and a seller generated by the LSSG_3 model based on Qwen3-8B.

Item	Vintage Multi-Stone Chunky Beaded Necklace – Earthy Gemstone Statement
Listed Price	$25.0
Seller	Hey there
Buyer	Hi, that is a beautiful chain! Would you be willing to come down in price to $15?
Seller	I was thinking around $23 but will go a little lower
Buyer	{'action': 'offer', 'price': 20.0}
Seller	{'action': 'accept'}

H Role-Specific Return and Baseline Estimation

General definition and implementation. Let Π_{old} denote the joint role-conditioned policy used to collect trajectories, and let G_t^i be role i's return from state s_t onward. The action-value function and baseline-adjusted weight are

$$\begin{aligned} Q_{\text{old}}^i(s_t, a_t^i) &= \mathbb{E}_{\tau\sim\Pi_{\text{old}}}\left[G_t^i \mid s_t, a_t^i\right], \\ A_b^i(s_t, a_t^i) &= Q_{\text{old}}^i(s_t, a_t^i) - b^i(s_t), \end{aligned} \tag{6}$$

where $b^i(s_t)$ is an action-independent baseline; A_b^i is the usual advantage function when b^i equals the state-value function. These quantities define the general learning signal; training uses sampled terminal returns with a fixed zero baseline. Cheng et al. (2024) motivate an immediate-reward advantage approximation through a temporal-difference residual and an approximate zero-value assumption. For our terminal-reward negotiation setting, we use a sampled return with the baseline explicitly set to zero.

Terminal-return assignment. Let T_τ be the terminal step of a negotiation trajectory. Intermediate environment rewards are zero, and the terminal reward is the role-specific outcome $R_i(\tau)$ defined in Section 2. The implemented assignment is undiscounted ($\gamma = 1$), so a completed rollout gives

$$\hat{G}_t^i = \sum_{k=t}^{T_\tau} r_k^i = R_i(\tau), \qquad \hat{A}_t^i = \hat{G}_t^i - b^i(s_t) = R_i(\tau), \tag{7}$$

with $b^i(s_t) \equiv 0$. For a valid agreement at price P_d, every buyer reply receives $\zeta_1(1 - P_d/P_0)$ and every seller reply receives $\zeta_2 P_d/P_0$. Within each retained trajectory, every reply from the same role receives that role's full, undiscounted terminal return. Environment rewards are delivered at termination, and the return target is assigned to each role-specific reply for training. Thus, $\hat{A}_t^i$ denotes the sampled return under the chosen zero baseline.

Trajectory filtering. Only successful agreements with positive, valid initial and transaction prices are retained for the offline update. Failed negotiations receive zero outcome reward in the environment and are excluded from the training dataset. Retained samples use their raw role-specific terminal returns. The offline update optimizes a surrogate objective on the retained successful trajectories.

Reference-policy adjustment. Let x_t^i denote the role-conditioned prompt and dialogue history, and let y_t^i be the corresponding tokenized reply, including its end-of-sequence token. The implementation computes the mean log-probability over valid reply tokens,

$$\ell_\theta(x, y) = \frac{1}{|y|} \sum_{k=1}^{|y|} \log p_\theta(y_k \mid x, y_{<k}), \tag{8}$$

with prompt and padding tokens excluded from the likelihood average. Before the clipped offline policy update, the return surrogate is adjusted to

$$w_{\theta,t}^i = \hat{A}_t^i - \beta_{\text{sp}} \big[\ell_\theta(x_t^i, y_t^i) - \ell_{\text{old}}(x_t^i, y_t^i)\big], \tag{9}$$

where ℓ_{old} is computed with the frozen reference policy and β_{sp} controls the reference-policy deviation penalty. The reference penalty is computed from the difference between current and reference mean reply-token log-probabilities. The reference-model scores are detached, whereas the current-policy component of $w_{\theta,t}^i$ remains in the computation graph in the implementation. This adjustment supplies the KL-inspired reference-policy regularization within the SASP return weight.

I Clipped Offline Self-Play Objective

Ratio-weighted update. For a role-conditioned state s and reply a, let (x, y) be the corresponding prompt and reply token sequence. The length-normalized reply score is

$$\tilde{\pi}_\theta^i(a \mid s) = \exp\big(\ell_\theta(x, y)\big), \qquad i \in \{B, S\}, \tag{10}$$

where ℓ_θ is the mean reply-token log-probability in Equation 8; $\tilde{\pi}$ is the geometric mean of the reply-token probabilities. The unclipped base objective is

$$\mathcal{L}_{\text{SASP}}^{i,\text{base}}(\theta) = -\mathbb{E}_{(s,a)\sim\mathcal{D}^i} \left[\frac{\tilde{\pi}_\theta^i(a \mid s)}{\tilde{\pi}_{\text{old}}^i(a \mid s)} \, w_\theta^i(s, a) \right], \tag{11}$$

with the reference-adjusted return weight defined in Equation 9. For each retained reply, define

$$\rho_\theta^i(s, a) = \frac{\tilde{\pi}_\theta^i(a \mid s)}{\tilde{\pi}_{\text{old}}^i(a \mid s)} = \exp\big(\ell_\theta(x, y) - \ell_{\text{old}}(x, y)\big). \tag{12}$$

This ratio compares current and reference length-normalized scores for each complete reply. Using the regularized return weight w_θ^i from Equation 9, the per-reply loss is

$$\begin{aligned} \mathcal{L}_{\text{SASP}}^{i,\text{reply}}(\theta; s, a) = \; -\min\Big\{ & \rho_\theta^i(s, a) w_\theta^i(s, a), \\ & \text{clip}\big(\rho_\theta^i(s, a), 1 - \epsilon, 1 + \epsilon\big) w_\theta^i(s, a) \Big\}. \end{aligned} \tag{13}$$

Clipping is applied to the ratio-weighted term as in PPO-style optimization (Schulman et al., 2017), with reference-policy regularization incorporated into w_θ^i. The threshold ϵ controls clipping, while β_{sp} controls the reference-policy deviation inside w_θ^i. Reference scores are detached, while current-policy scores remain differentiable in both ρ_θ^i and w_θ^i.

Shared-model batching and role weights. Buyer and seller samples train the same parameter vector θ using their corresponding role prompts. Let N_B and N_S be the numbers of retained buyer and seller replies, and let $N = N_B + N_S$. Each sample j of role i_j receives a fixed dataset-level weight $\omega_j = N/(2N_{i_j})$ for a nonempty role. The terminal return $R_i(\tau)$ is shared by replies of role i within a trajectory. The adjusted weight $w_{\theta,t}^i$ also depends on the current and reference scores of each reply, while ω_j is fixed by the dataset-level role counts. For the self-play subset $\mathcal{B}_{\text{sp}}$ of a training batch, define $Z_\mathcal{B} = \max(1, \sum_{j\in\mathcal{B}_{\text{sp}}} \omega_j)$. The two contributions used in Equation 5 are

$$\mathcal{L}_{\text{SASP}}^i(\theta; \mathcal{B}) = \frac{1}{Z_\mathcal{B}} \sum_{\substack{j\in\mathcal{B}_{\text{sp}} \\ i_j = i}} \omega_j \, \mathcal{L}_{\text{SASP}}^{i,\text{reply}}(\theta; s_j, a_j), \qquad i \in \{B, S\}. \tag{14}$$

When the two role coefficients are equal to one, their sum is the normalized weighted mean over self-play replies in the batch. The fixed weights balance role counts across the dataset; individual batches can contain different role proportions and yield different role contributions. Supervised-retention samples form a separate masked subset, over which their loss is normalized. An empty subset contributes zero.

Algorithm 1 Training Procedure of LSSG

Input: Initial shared policy π_θ; real-dialogue data $\mathcal{D}_{\mathrm{BC}}$; item pool $\mathcal{I}$; instruction-retention data $\mathcal{D}_{\mathrm{inst}}$; number of self-play iterations K.
Output: Trained shared policy π_θ.

Freeze a copy $\pi_0 \leftarrow \pi_\theta$ as the GABC reference policy.
for each GABC training minibatch from $\mathcal{D}_{\mathrm{BC}}$ **do**
  Compute reply imitation, token entropy, and frozen-reference contributions; aggregate the GABC loss as in Appendix K.
  Update the shared parameters θ using the GABC loss.
end for
for $k = 1, \ldots, K$ **do**
  Copy and freeze $\pi_{\mathrm{old}} \leftarrow \pi_\theta$.
  Generate negotiation trajectories with π_{old}, using buyer/seller role prompts and items from $\mathcal{I}$.
  Retain valid agreements with $P_0, P_d > 0$; discard other trajectories (Appendix H).
  Construct role-conditioned prompt–reply samples (x_j, y_j, i_j) from retained trajectories.
  Assign each reply its role-specific terminal return by Equation 7, and its fixed role weight ω_j as defined above.
  for each training minibatch $\mathcal{B}$ of retained replies and instruction examples from $\mathcal{D}_{\mathrm{inst}}$ **do**
    Identify the self-play subset $\mathcal{B}_{\mathrm{sp}}$ and retention subset; compute $Z_{\mathcal{B}}$.
    Evaluate current reply scores $\ell_\theta(x_j, y_j)$ and detached reference scores $\ell_{\mathrm{old}}(x_j, y_j)$ for $j \in \mathcal{B}_{\mathrm{sp}}$ (Equation 8).
    Compute $w_\theta^{i_j}$ and $\rho_\theta^{i_j}$ by Equations 9 and 12, retaining gradients through the current-policy scores.
    Form the clipped, reference-regularized role losses $\mathcal{L}^B_{\mathrm{SASP}}$ and $\mathcal{L}^S_{\mathrm{SASP}}$ by Equation 14.
    For each $j \in \mathcal{B}_{\mathrm{sp}}$, sample one auxiliary continuation z_j from the current policy given c_j, with gradients disabled (Appendix J).
    Compute detached costs $C_{\mathrm{SemDi},j}$ and $C_{\mathrm{EmoSt},j}$ with the frozen scorers, using the stored reply $\tilde{y}_j$ as the semantic reference.
    Re-evaluate the same sampled tokens with gradients enabled to obtain $\ell_\theta(c_j, z_j)$ (Equation 16).
    Aggregate the two auxiliary gradient surrogates using Equation 4.
    Compute $\mathcal{L}_{\mathrm{SFT}}$ on the instruction-retention subset.
    Combine the five terms by Equation 5 and update the shared parameters θ.
  end for
end for

Training procedure. Algorithm 1 combines the GABC warm start and the offline self-play iterations. The retained trajectories and reference snapshot are fixed within each outer iteration; the snapshot is refreshed only between iterations. Buyer and seller prompts always condition the same trainable parameters θ. Batch updates below include any configured gradient accumulation before an optimizer step.

J Optimization of Auxiliary Losses

Interpretation of the auxiliary costs. The semantic cost penalizes similarity to the stored reference reply above the specified margin. The emotional cost measures absolute deviation of the binary classifier's positive-sentiment probability from the midpoint target of 0.5.

Expected costs and score-function gradients. For a fixed auxiliary context c and reference reply, write $C_k(z)$ for either cost defined in the main text, suppressing the reference argument for semantic diversity. The intended regularization objective is $\mathcal{L}_k(\theta) = \mathbb{E}_{z\sim\pi_\theta(\cdot|c)}[C_k(z)]$, for $k \in \{\mathrm{SemDi}, \mathrm{EmoSt}\}$. The frozen scorers have no trainable dependence on θ for a fixed generated text. If sampling and likelihood evaluation use the same autoregressive distribution π_θ, the expected-cost objective satisfies

$$\nabla_\theta \mathcal{L}_k(\theta) = \mathbb{E}_{z\sim\pi_\theta(\cdot|c)}\left[C_k(z)\,\nabla_\theta \log \pi_\theta(z \mid c)\right]. \tag{15}$$

Here, $\log \pi_\theta(z \mid c)$ is the sum of token log-probabilities for the generated sequence, including its termination token when present. The identity explains how a discrete text score can provide a learning signal without differentiating through token sampling, decoded text, or either scorer.

Sampled continuations and token likelihoods. The implementation generates one auxiliary continuation per training sample with gradients disabled, then performs a separate forward pass on exactly those sampled tokens with gradients enabled. The context c_j is the fixed token prefix supplied to auxiliary generation, and z_j denotes only the newly generated continuation. For semantic scoring, $\tilde{y}_j$ is the stored reference reply decoded from the training labels after excluding ignored positions; it is distinct from z_j. Let M_{jt} indicate valid continuation tokens and $n_j = \max(1, \sum_t M_{jt})$. The differentiable score is

$$\ell_\theta(c_j, z_j) = \frac{1}{n_j} \sum_t M_{jt} \log p_\theta(z_{jt} \mid c_j, z_{j,<t}). \tag{16}$$

The mask excludes context and padding tokens, retains the first generated end-of-sequence token when present, and excludes all subsequent positions. If generation reaches its length limit without that token, all valid generated tokens contribute. Both auxiliary costs are evaluated on the same decoded continuation, with special tokens omitted from the scorer input.

Detached costs and the update sign. Set $C_{\mathrm{SemDi},j} = C_{\mathrm{SemDi}}(z_j, \tilde{y}_j)$ and $C_{\mathrm{EmoSt},j} = C_{\mathrm{EmoSt}}(z_j)$. These costs, the sampled tokens, and the dataset weights are held fixed during backpropagation. Here, sg stops gradients through the costs, and $\mathcal{B}_{\mathrm{sp}}$, ω_j, and $Z_{\mathcal{B}}$ are the self-play subset, role-balancing weights, and normalizer defined in Appendix I. Gradients reach the model through the re-evaluated token scores. Supervised-retention samples are excluded from these auxiliary terms. The gradient of the implemented surrogate in Equation 4 is therefore

$$\nabla_\theta \widehat{\mathcal{L}}_k^{\mathrm{PG}} = \frac{1}{Z_{\mathcal{B}}} \sum_{j \in \mathcal{B}_{\mathrm{sp}}} \omega_j C_{k,j} \nabla_\theta \ell_\theta(c_j, z_j). \tag{17}$$

Equivalently, defining $r_{k,j} = -C_{k,j}$ gives the per-sample loss $-\mathrm{sg}(r_{k,j})\ell_\theta(c_j, z_j)$. Thus the minimized cost-based surrogate has a positive sign in front of $\mathrm{sg}(C_{k,j})\ell_\theta(c_j, z_j)$: gradient descent penalizes the likelihood of sampled high-cost continuations. The surrogate takes nonpositive scalar values because it multiplies nonnegative costs by mean token log-probabilities. The self-play mask and role weights are shared with the SASP batch aggregation; an empty self-play subset contributes zero to both auxiliary terms.

Relation to the expected-cost objectives. The auxiliary decoder uses temperature 0.95, top-k sampling with $k = 50$, a repetition penalty of 1.2, and a maximum of 64 new tokens. Denote the resulting sampling distribution by q_θ. The re-evaluated log-probabilities in Equation 16 use the model's unmodified logits. Continuations are sampled from q_θ, and the update uses their costs directly without importance correction. Moreover, dividing by n_j changes the weighting relative to the sequence-level score-function identity in Equation 15. Equation 4 therefore defines the practical update through transformed sampling and length-normalized token scores. The full training objective uses these surrogates explicitly, while the expected costs retain their roles as definitions of the intended regularization criteria.

K Shared Parameters and GABC Batch Aggregation

Shared parameters and role conditioning. Each training example contains a role-conditioned prompt and its target reply. Buyer and seller examples are loaded into one training dataset and passed through the same trainable language model; the reference scores are computed with the frozen reference model on the same inputs. The buyer and seller superscripts identify the role associated with an example, while both roles contribute gradients to the shared parameters θ. Minibatches can contain different numbers of buyer and seller examples.

Implemented GABC loss. For a demonstration prompt–reply pair (x_j, y_j), let $\ell_{\theta,j} = \ell_\theta(x_j, y_j)$ and $\ell_{0,j} = \ell_0(x_j, y_j)$ be the mean valid reply-token log-probabilities defined in Equation 8. The reference score is computed with the frozen pretrained model. The sample-level reference contribution and weighted imitation component are

$$\begin{aligned} K_j(\theta) &= \exp(\ell_{\theta,j})(\ell_{\theta,j} - \ell_{0,j}), \\ \mathcal{L}_{\mathrm{GABC}}^{\mathrm{reply}}(\theta; x_j, y_j) &= -\ell_{\theta,j} + \beta_{\mathrm{KL}} K_j(\theta). \end{aligned} \tag{18}$$

The reference contribution is a reply-score proxy for the policy-level reference constraint. For a mixed-role batch $\mathcal{B}$ of N replies and padded input length T, the implemented batch loss is

$$\widehat{\mathcal{L}}_{\mathrm{GABC}}(\theta;\mathcal{B}) = \frac{1}{N}\sum_{j=1}^{N} u_j\, \mathcal{L}^{\mathrm{reply}}_{\mathrm{GABC}}(\theta; x_j, y_j) - \frac{\beta_H}{NT}\sum_{j=1}^{N}\sum_{t=1}^{T} H_{j,t}(\theta). \tag{19}$$

Here, u_j is the stored sample weight, defaulting to 1, and $H_{j,t} = -\sum_{v\in\mathcal{V}} p_{\theta,j,t}(v)\log(p_{\theta,j,t}(v) + 10^{-8})$, where $p_{\theta,j,t}$ is the vocabulary softmax at position t of the model output. The entropy average includes all output positions, including prompt and padding positions, and is added separately from the weighted reply losses. The training script uses $\beta_{\mathrm{KL}} = 0.1$ and $\beta_H = 0.01$ for both roles. The batch mean is taken over pooled replies, so each reply contributes according to its stored weight and dialogues contribute through their retained reply counts.

Return granularity in SASP. As detailed in Appendix H, a retained negotiation assigns its terminal payoff to every reply of the corresponding role. This yields one return weight per reply, whose likelihood score aggregates token log-probabilities through Equation 8. The role-conditioned replies then contribute to the common self-play batch through Equation 14.