WHO DESERVES THE REWARD? SHAPLEY CREDIT OPTIMIZATION FOR MULTI-AGENT LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Integrating Large Language Models (LLMs) with external tools via multi-agent systems offers a promising paradigm for decomposing and solving complex problems. However, training multi-agent LLM systems with tool use remains challenging due to the credit assignment problem, as it is often unclear which agent interactions contribute to the success or failure of decision trajectories. Existing LLM agent training methods typically rely on globally broadcast rewards or augment them with heuristic or LLM-judge-based local feedback, failing to capture marginal contributions from individual agents and leading to inefficient reinforcement learning. To address these limitations, we introduce the Shapley-based Hierarchical Attribution for Reinforcement Policy (SHARP), a credit-aware optimization framework tailored to multi-agent LLM reinforcement learning with tool use. SHARP stabilizes training by normalizing agent-specific advantages across trajectory groups and decomposes the reward into three components: a global broadcast accuracy reward for task alignment, a Monte Carlo Shapley based marginal credit reward for agent-level attribution, and a tool format reward for valid tool invocation. Extensive experiments across real-world LLM agent benchmarks demonstrate that SHARP significantly outperforms recent baselines, achieving average match improvements of 35.01% and 24.97% over single-agent and multi-agent LLM agent approaches, respectively. The implementation is available at `https://anonymous.4open.science/r/SHARP-DDB4/`.

1 INTRODUCTION

Large Language Models (LLMs) have attracted attention in complex real-world scenarios in recent years, promoting a fundamental shift from static knowledge retrieval to dynamic, tool-augmented interactions Lewis et al. (2020); Wang et al. (2024). A popular paradigm for solving complex problems requiring reasoning, planning, and execution is to integrate LLMs with tools via Multi-Agent Systems (MAS) Zhang et al. (2025b); Ferrag et al. (2025); Shen et al. (2023). Unlike single-agent architectures that often struggle with context overflow and noisy tool feedback, LLM-based MAS enables multiple role-conditioned agents to collaborate through decomposition, reasoning, tool use, and response aggregation Li et al. (2025b); Wu et al.; Hong et al.; Qian et al. (2024).

Existing LLM-agent reinforcement learning methods commonly address credit assignment in two ways. The first relies primarily on globally broadcast rewards, where the same trajectory-level signal is assigned to all participating agents Zhang et al. (2025b); Wu et al. (2026); Sun et al. (2026). This strategy encourages global task alignment, but treats the multi-agent trajectory as a monolithic unit and cannot distinguish the marginal contribution of each agent. The second augments the global reward with local feedback for specific agents or intermediate steps. Such feedback is typically obtained from LLM-judge-based signals Hong et al. (2025); Li et al. (2025a); Xue et al. (2025); Lu et al. (2025); Mo et al. (2025) or manually designed task- and role-specific rewards Xu et al. (2025); Park et al. (2025); Wang et al. (2025). Although these local signals provide denser supervision while preserving outcome-level alignment, they are often tied to specific tasks, tools, prompts, or judge designs. Moreover, evaluator-based rewards may introduce additional inference cost and bias, while heuristic local rewards often evaluate agents or steps in isolation without explicitly modeling coalition-dependent marginal contributions. These limitations motivate a reward assignment mechanism that

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.

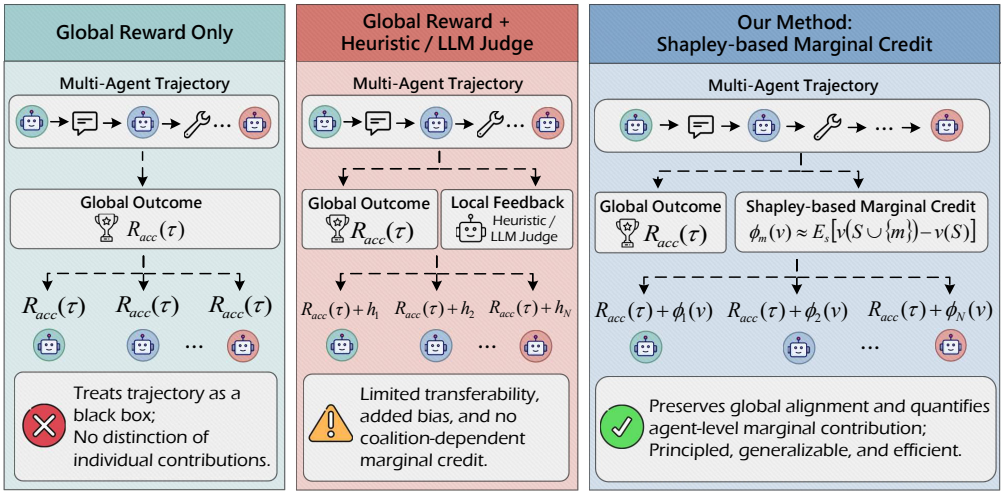

Figure 1: Comparison of reward assignment strategies in tool-using multi-agent LLM reinforcement learning. Global reward methods assign the same trajectory-level outcome to all agents, while heuristic or LLM-judge-based methods add task-specific local feedback. In contrast, SHARP preserves global task alignment and explicitly estimates agent-level marginal credit through Shapley-based attribution.

jointly preserves global task alignment and explicitly models agent-level marginal credit without relying on workflow-specific reward engineering or black-box LLM judges. Figure 1 illustrates the difference between these reward assignment strategies.

To address this problem, we propose Shapley-based Hierarchical Attribution for Reinforcement Policy (SHARP), a credit-aware optimization framework tailored to tool-using multi-agent LLM reinforcement learning. SHARP jointly preserves global task alignment and provides marginal credit assignment for agent-level feedback. The global broadcast accuracy reward anchors optimization to the task outcome, while the Monte Carlo Shapley-based marginal credit reward adapts the Shapley marginal contribution principle to long-horizon LLM-agent trajectories involving reasoning traces, tool calls, and intermediate observations. Concretely, each agent's credit is estimated by averaging counterfactual value differences over sampled agent coalitions, allowing SHARP to account for collaboration-dependent effects among agents rather than evaluating each agent in isolation. Together with a tool-format reward for tool invocation and group-relative advantage normalization, SHARP provides stable and fine-grained optimization signals for tool-using multi-agent LLM systems.

We evaluate SHARP on several LLM-agent benchmarks, including *MuSiQue*, *GAIA-text*, *WebWalkerQA*, and *FRAMES*. Our method outperforms existing baselines significantly, achieving average match improvements of 35.01% and 24.97% over single-agent and multi-agent baselines, respectively. Furthermore, SHARP demonstrates strong cross-task generalization on the *DocMath-Eval* dataset and exhibits promising scaling behavior, achieving 18.23 points of improvement over single-agent models on an 8B backbone. Coordination analysis further shows that SHARP improves hierarchical LLM-agent coordination by reducing harmful subagent interactions from 5.48% to 3.65%.

Our main contributions can be summarized as follows:

- We propose SHARP, a credit-aware reinforcement learning framework tailored to tool-using multi-agent LLM systems. SHARP leverages Monte Carlo Shapley-based marginal credit estimation to provide fine-grained agent-level attribution and stabilizes optimization through group-normalized agent-specific advantages.
- We introduce a tripartite reward decomposition mechanism for LLM-agent training, comprising global task alignment, Shapley-based marginal credit, and tool-call format validity signals. This design targets long-horizon tool-integrated trajectories where final task success, agent-level contribution, and tool-call executability must be jointly considered.
- Extensive experiments across diverse LLM-agent benchmarks show that our proposed method with a Qwen3-8B backbone significantly outperforms existing single-agent and multi-agent baselines, yielding average match gains of 35.01% and 24.97%, respectively.

2 Related Works

LLM-based Multi-Agent Reinforcement Learning. LLM-based multi-agent reinforcement learning extends methods such as GRPO Shao et al. (2024), but faces a credit assignment challenge: existing rewards rarely model collaboration-aware marginal contributions among agents. Methods such as G-Designer Zhang et al. (2025b), CARD Wu et al. (2026), and AgentPO Sun et al. (2026) mainly rely on task-level supervision to optimize multi-agent coordination. Other works introduce denser feedback for intermediate behaviors. M-GRPO Hong et al. (2025), CoMAS Xue et al. (2025), MATPO Mo et al. (2025), and COA Li et al. (2025a) use LLM-judge-based signals, while WebGen-Agent Lu et al. (2025) uses VLM and GUI-agent feedback. Another line of work designs task- or role-specific rewards: AceSearcher Xu et al. (2025) rewards decomposition and search, MAPoRL Park et al. (2025) uses a verifier score for answer correctness and corrective discussion, and CURE Wang et al. (2025) rewards programmer and unit-test agents. Although these methods provide supervision, their signals are global, evaluator-based, or workflow-specific, and do not estimate how each agent's contribution changes under different collaborations. To address this gap, SHARP estimates collaboration-aware agent-level marginal contributions across diverse tool-using multi-agent LLM tasks, without workflow-specific reward engineering or LLM-judge supervision.

Explicit Marginal Credit Assignment. Explicit marginal credit assignment attributes each agent's contribution through a defined rule rather than an implicit evaluator. In classical MARL, COMA Foerster et al. (2018) uses counterfactual baselines, while VDN Sunehag et al. (2017) and QMIX Rashid et al. (2020) decompose global values into agent-level utilities. These methods provide explicit feedback but often depend on structured state-action spaces, synchronized decisions, or learned critics, limiting their transfer to long text-based LLM-agent trajectories. Shapley values offer another explicit attribution principle by assigning credit according to expected marginal contributions over coalitions. Prior MARL studies, such as SQDDPG Wang et al. (2020) and SCC Li et al. (2021), have applied Shapley-based credit assignment with approximation or sampling strategies. However, these methods are primarily designed for classical MARL settings and do not directly address long-horizon LLM-agent trajectories involving reasoning traces, tool calls, and intermediate observations. SHARP adapts this explicit attribution perspective to tool-using multi-agent LLM systems, estimating agent-level credit from sampled coalition-level counterfactual value differences without relying on black-box LLM judges or workflow-specific heuristic rewards.

3 Preliminaries

3.1 Tool-Integrated Interaction Trajectory

We model tool-integrated reasoning as a sequential process where an LLM interacts with external tools over multiple turns Motwani et al. (2024); Dong et al. (2025); Qian et al. (2025). Given a query q, a trajectory is defined as

$$\tau \triangleq \{a_1, s_1, a_2, s_2, \ldots, a_T, s_T\}, \tag{1}$$

where a_t denotes a model action, and s_t denotes the corresponding tool observation. At step t, the action is sampled from $a_t \sim \pi_\theta(\cdot \mid p_{\text{sys}}, q, a_{<t}, s_{<t})$ and the tool returns $s_t \sim P_{\text{Tool}}(\cdot \mid a_t)$, where p_{sys} specifies system prompt and tool schemas. The trajectory probability factorizes as

$$P_\theta(\tau) = \prod_{t=1}^{T} \pi_\theta(a_t \mid p_{\text{sys}}, q, a_{<t}, s_{<t})\, P_{\text{Tool}}(s_t \mid a_t). \tag{2}$$

Since tool responses are independent of θ, optimization acts only on generated actions.

3.2 Shapley Value for Credit Assignment

For a set of agents $\mathcal{N}$ and a coalition value function $v : 2^{\mathcal{N}} \to \mathbb{R}$, the marginal contribution of agent m to a coalition $S \subseteq \mathcal{N} \setminus \{m\}$ is

$$\Delta_m(S) = v(S \cup \{m\}) - v(S). \tag{3}$$

The Shapley value assigns credit by averaging this contribution across all collaboration contexts:

$$\phi_m(v) = \sum_{S \subseteq \mathcal{N} \setminus \{m\}} \frac{|S|!(|\mathcal{N}| - |S| - 1)!}{|\mathcal{N}|!} \Delta_m(S). \tag{4}$$

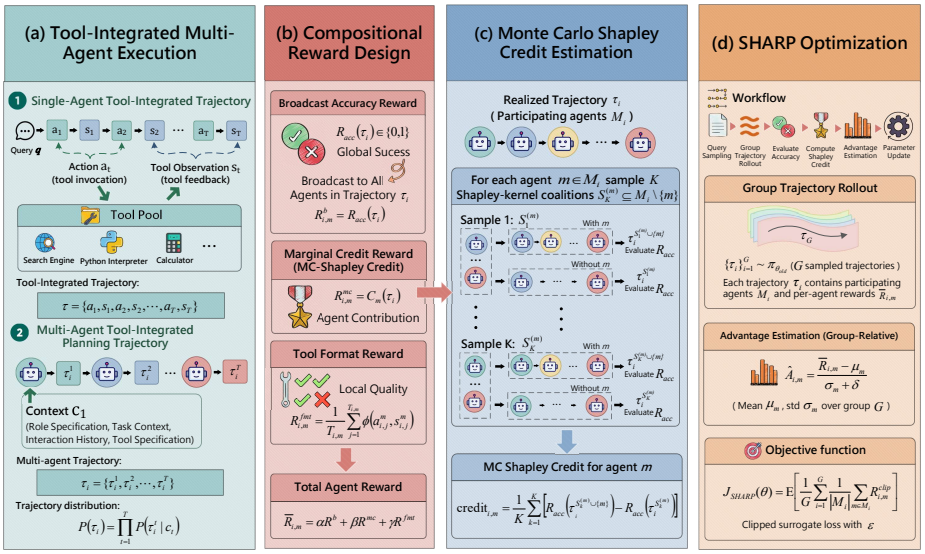

Figure 2: Overview of SHARP framework. The pipeline involves (a) tool-integrated multi-agent execution; (b) tripartite reward system integrating global accuracy, explicit marginal credit, and tool format rewards; (c) marginal credit mechanism isolating agents' contribution via Shapley values; (d) SHARP workflow using group-relative policy for stable alignment.

This formulation captures how an agent's contribution changes with different collaborators, making it suitable for marginal credit assignment. Since exact enumeration is expensive, we use a Monte Carlo approximation for tool-integrated multi-agent trajectories in Section 4.3.

4 METHODOLOGY

In this work, we study Tool-Integrated Planning (TIP), where multiple role-conditioned agents collaboratively solve a query through reasoning, tool use, and response generation. Each agent is conditioned on its role, context, and interaction history, producing reasoning steps, tool invocations, or responses. The workflow is summarized in Algorithm A and illustrated in Figure 2.

4.1 TOOL-INTEGRATED MULTI-AGENT EXECUTION

Given a user query q, the system produces a multi-agent trajectory

$$\tau = (\tau^1, \tau^2, \ldots, \tau^T), \tag{5}$$

where τ^t is the segment generated by the agent invoked at step t, containing reasoning tokens, tool calls, observations, or intermediate responses. The invoked agent receives a context c_t containing its role specification, task state, interaction history, and tool schemas, and generates τ^t conditioned on c_t. The trajectory distribution is factorized as $P(\tau) = \prod_{t=1}^{T} P(\tau^t \mid c_t)$. This provides a unified trajectory-level view for multi-agent reasoning, tool use, and reward assignment.

4.2 COMPOSITIONAL REWARD DESIGN

Reward design for collaborative MAS must balance global task success, agent-level attribution, and tool-call executability. Broadcast rewards cannot disentangle individual contributions, while terminal signals alone cannot capture malformed tool calls in tool-integrated settings. We therefore organize the reward design around three practical principles.

Multi-Agent Reward Design Principles

In multi-agent reinforcement learning, the surrogate reward assigned to each participant $m \in \mathcal{M}$ should satisfy the following three principles:
(i) Global Alignment (Effectiveness). The signal should be grounded in the final task outcome to align local decisions with overall trajectory success.
(ii) Marginal Attribution (Credit Faithfulness). The signal should identify each agent's specific contribution, enabling precise credit assignment.
(iii) Format Compliance (Execution Validity). The signal should encourage tool calls to conform to valid and executable formats.

Following these principles, we decompose the reinforcement signal into three components: *broadcast accuracy reward*, *marginal credit reward*, and *tool format reward*, which capture terminal success, agent-level contribution, and tool-call validity, respectively.

Globally Broadcast Accuracy Reward. For a complete trajectory τ, define a binary terminal signal

$$R_{\mathrm{acc}}(\tau) \triangleq r_{\mathrm{acc}}(\tau) \in \{0, 1\}, \tag{6}$$

which reflects whether the final response aligns with the ground truth Rafailov et al. (2023). Under GRPO, G trajectories $\{\tau_i\}_{i=1}^{G}$ are sampled per query, and the terminal reward is broadcast to all participating agents:

$$R_{i,m}^{\mathrm{b}} \triangleq R_{\mathrm{acc}}(\tau_i), \quad m \in \mathcal{M}_i, \tag{7}$$

where $\mathcal{M}_i$ denotes the agents involved in τ_i.

Marginal Credit Reward. Broadcast rewards alone cannot disentangle different agents' contributions. We assign agent-specific feedback by

$$R_{i,m}^{\mathrm{mc}} \triangleq \mathcal{C}_m(\tau_i;\, R_{\mathrm{acc}})\,, \quad m \in \mathcal{M}_i, \tag{8}$$

where $\mathcal{C}_m(\cdot)$ maps a trajectory to per-agent credit conditioned on task success Hong et al. (2024). We instantiate $\mathcal{C}_m$ via the counterfactual marginal contribution formulation in Eq. equation 14, estimating how the outcome changes under different agent collaborations.

Tool Format Reward. Malformed tool calls may prevent execution even when reasoning is correct. For trajectory τ_i, let $T_{i,m}$ be the number of tool calls made by agent m. The format reward is

$$R_{i,m}^{\mathrm{fmt}} \triangleq \begin{cases} \dfrac{1}{T_{i,m}} \displaystyle\sum_{j=1}^{T_{i,m}} \psi\big(a_{i,j}^{m}\big)\,, & T_{i,m} > 0, \\ 0, & T_{i,m} = 0, \end{cases} \tag{9}$$

where $\psi(a_{i,j}^{m}) = 1$ if the j-th tool call is syntactically valid and schema-compliant, and 0 otherwise.

Aggregate Rewards. The total surrogate reward for agent m on trajectory τ_i is

$$\bar{R}_{i,m} \triangleq \alpha\, R_{i,m}^{\mathrm{b}} + \beta\, R_{i,m}^{\mathrm{mc}} + \gamma\, R_{i,m}^{\mathrm{fmt}}, \quad m \in \mathcal{M}_i, \tag{10}$$

where α, β, and γ balance terminal effectiveness, credit faithfulness, and tool-call validity.

4.3 Shapley-based Marginal Credit Assignment

In this work, the credit functional is instantiated using an approximation of the Shapley value from cooperative game theory Chen et al. (2023); Napolitano & Cagliero (2025). Given a set of agents $\mathcal{N}$ and a value function $v(\cdot)$, the Shapley value of agent m is defined as

$$\phi_m(v) = \sum_{S \subseteq \mathcal{N} \setminus \{m\}} \omega(S)\,\big(v(S \cup \{m\}) - v(S)\big)\,, \text{ where } \omega(S) = \frac{|S|!(|\mathcal{N}| - |S| - 1)!}{|\mathcal{N}|!}. \tag{11}$$

Notably, $\phi_m(v)$ quantifies the expected marginal contribution of agent $m \in \mathcal{N}$ across all possible coalitions. Given the inherent complexity of multi-agent TIP, exact Shapley-based credit assignment is computationally intractable. Specifically, computing the Shapley value for each agent requires

evaluating $2^{|\mathcal{N}|-1}$ possible coalitions, leading to exponential complexity in the number of agents. This brings computational burden in large-scale MAS settings and motivates the following acceleration strategy with sampling-based approximation.

To address the exponential cost of exact Shapley enumeration, we estimate the marginal contribution using Monte Carlo sampling. For each agent m, we sample K coalitions $\{S_k^{(m)}\}_{k=1}^K$ according to the Shapley kernel. Specifically, we first sample the coalition size uniformly from $\{0, \ldots, |\mathcal{N}|-1\}$ and then uniformly sample a subset of $\mathcal{N} \setminus \{m\}$ with that size. This induces

$$\Pr(S_k^{(m)} = S) = \frac{1}{|\mathcal{N}|} \frac{1}{\binom{|\mathcal{N}|-1}{|S|}} = \frac{|S|!(|\mathcal{N}| - |S| - 1)!}{|\mathcal{N}|!} = \omega(S), \tag{12}$$

for any $S \subseteq \mathcal{N} \setminus \{m\}$. Hence, the resulting estimator

$$\widehat{\phi}_m(v) = \frac{1}{K} \sum_{k=1}^{K} \left[v(S_k^{(m)} \cup \{m\}) - v(S_k^{(m)}) \right] \tag{13}$$

is an unbiased Monte Carlo estimator of the Shapley value. This reduces the cost from exhaustive enumeration to $O(K)$ marginal evaluations per agent.We provide a detailed analysis of the sampling budget K and its cost–performance trade-off in Appendix E.

For trajectory-level attribution, we instantiate the agent set as the participating agents $\mathcal{M}_i$ in trajectory τ_i and define the value function as $v(\tau_i^S) \triangleq R_{\text{acc}}(\tau_i^S)$, where τ_i^S is the counterfactual trajectory induced by coalition S. We construct τ_i^S via an agent-exclusion rerollout procedure that preserves the original temporal order and dependency structure; details are provided in Appendix C.

Given a realized trajectory τ_i and a participating agent $m \in \mathcal{M}_i$, we quantify its credit by Monte Carlo Shapley estimation:

$$\text{credit}_{i,m} \triangleq \widehat{\phi}_{i,m} = \frac{1}{K} \sum_{k=1}^{K} \left[R_{\text{acc}} \left(\tau_i^{S_k^{(m)} \cup \{m\}} \right) - R_{\text{acc}} \left(\tau_i^{S_k^{(m)}} \right) \right], \tag{14}$$

where $S_k^{(m)} \subseteq \mathcal{M}_i \setminus \{m\}$ denotes the k-th sampled agent coalition excluding agent m, and τ_i^S denotes the counterfactual trajectory induced by coalition S, where agents outside S are replaced by their corresponding default counterfactual actions.

This construction follows the standard counterfactual credit assignment view, where an agent's contribution is measured by the value difference induced by its presence or replacement. Consequently, the marginal credit reward for each participating agent is assigned as

$$R_{i,m}^{\text{mc}} \triangleq \text{credit}_{i,m}, \quad m \in \mathcal{M}_i. \tag{15}$$

4.4 SHARP Optimization via GRPO

To achieve efficient, stable optimization in the presence of sparse rewards and multi-agent coordination, we use the improved group relative policy gradient Nauman & Cygan (2023); Shao et al. (2024) to normalize agentic rewards.

Agentic Group Relative Advantage. For G sampled trajectories $\{\tau_i\}_{i=1}^G$, we collect the corresponding rewards $\{\bar{R}_{i,m}\}_{i=1}^G$, and compute the mean μ_m and standard deviation σ_m for each individual agent m. The group relative advantage $\hat{A}_{i,m}$ of agent m on trajectory τ_i is defined as

$$\hat{A}_{i,m} \triangleq \frac{\bar{R}_{i,m} - \mu_m}{\sigma_m + \delta}, \tag{16}$$

where δ is a constant controlling the stability.

SHARP Clipped Surrogate Objective. Let $\{a_{i,j}^m\}_{j=1}^{T_{i,m}}$ be the sequence of actions generated by agent m on trajectory τ_i, and let $\text{ctx}_{i,j}^m$ be the corresponding context. We define the ratio between the current and old policies as

$$R_{i,m} \triangleq \exp\left(\sum_{j=1}^{T_{i,m}} \log \frac{\pi_\theta\left(a_{i,j}^m \mid \text{ctx}_{i,j}^m\right)}{\pi_{\theta_{\text{old}}}\left(a_{i,j}^m \mid \text{ctx}_{i,j}^m\right)} \right). \tag{17}$$

Table 1: Comparison across benchmarks with respect to multi-agent support (MAS), training-based optimization (TRN), broadcast outcome reward (BOR), and marginal credit modeling (MCR). ✓ and ✗ indicate support and non-support. †, ‡, and no superscript denote methods using LLaMA-3.1-8B, Qwen2.5-7B, and Qwen3-8B, respectively. All benchmark results are reported as the mean and standard deviation over five independent runs.

Method	MAS	TRN	BOR	MCR	MuSiQue	GAIA-text	WebWalkerQA	FRAMES	AVG
LLaMA-3.1-8B RAG	✗	✗	✗	✗	$7.36_{\pm 0.42}$	$8.64_{\pm 0.51}$	$0.81_{\pm 0.07}$	$5.96_{\pm 0.38}$	5.69
Qwen3-8B RAG	✗	✗	✗	✗	$8.43_{\pm 0.56}$	$15.72_{\pm 0.83}$	$1.18_{\pm 0.09}$	$6.91_{\pm 0.44}$	8.06
Plan-Search †	✗	✗	✗	✗	$27.11_{\pm 1.24}$	$9.86_{\pm 0.68}$	$3.41_{\pm 0.21}$	$10.53_{\pm 0.73}$	12.73
Plan-Search	✗	✗	✗	✗	$35.91_{\pm 1.53}$	$27.86_{\pm 1.16}$	$6.61_{\pm 0.38}$	$28.91_{\pm 1.22}$	24.82
Search-R1 ‡	✗	✓	✗	✗	$18.44_{\pm 0.94}$	$13.87_{\pm 0.72}$	$2.39_{\pm 0.15}$	$11.08_{\pm 0.61}$	11.44
Single-agent GRPO	✗	✓	✗	✗	$45.32_{\pm 1.87}$	$28.41_{\pm 1.28}$	$7.33_{\pm 0.42}$	$30.74_{\pm 1.35}$	27.95
Hierarchical MAS †	✓	✗	✗	✗	$35.68_{\pm 1.42}$	$13.15_{\pm 0.76}$	$5.69_{\pm 0.31}$	$20.87_{\pm 0.95}$	18.84
Hierarchical MAS	✓	✗	✗	✗	$37.81_{\pm 1.36}$	$27.91_{\pm 1.19}$	$7.56_{\pm 0.34}$	$31.74_{\pm 1.28}$	26.26
G-Designer	✓	✓	✓	✗	$38.94_{\pm 1.47}$	$27.72_{\pm 1.13}$	$4.82_{\pm 0.29}$	$28.61_{\pm 1.05}$	25.02
CARD	✓	✓	✓	✗	$44.51_{\pm 1.66}$	$33.27_{\pm 1.21}$	$7.51_{\pm 0.36}$	$26.94_{\pm 1.08}$	28.06
COA	✓	✓	✓	✗	$44.72_{\pm 1.58}$	$31.61_{\pm 1.18}$	$7.09_{\pm 0.33}$	$32.52_{\pm 1.14}$	28.98
AceSearcher†	✓	✓	✓	✗	$36.04_{\pm 1.39}$	$20.36_{\pm 0.84}$	$6.91_{\pm 0.35}$	$25.86_{\pm 1.06}$	22.29
MATPO	✓	✓	✓	✗	$47.43_{\pm 1.72}$	$31.24_{\pm 1.17}$	$7.61_{\pm 0.37}$	$36.68_{\pm 1.31}$	30.74
SHARP †	✓	✓	✓	✓	$46.31_{\pm 1.12}$	$23.09_{\pm 0.84}$	$7.67_{\pm 0.26}$	$27.76_{\pm 0.98}$	26.21
SHARP	✓	✓	✓	✓	$\mathbf{54.58}_{\pm 1.05}$	$\mathbf{37.63}_{\pm 0.88}$	$\mathbf{12.85}_{\pm 0.24}$	$\mathbf{37.48}_{\pm 1.02}$	**35.63**

The clipped surrogate objective for agent m on trajectory τ_i is formulated by

$$R_{i,m}^{\text{clip}} \triangleq \min\Big(R_{i,m}\,\hat{A}_{i,m}, \text{clip}(R_{i,m},\,1-\epsilon,\,1+\epsilon)\,\hat{A}_{i,m}\Big)\,. \tag{18}$$

Overall Optimization Objective. The SHARP objective $J_{\text{SHARP}}(\theta)$ aggregates clipped advantages across trajectories and participating agents, where

$$J_{\text{SHARP}}(\theta) \triangleq \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{M}_i|}\sum_{m\in\mathcal{M}_i} R_{i,m}^{\text{clip}}\right], \tag{19}$$

where policy parameters θ are updated by maximizing $J_{\text{SHARP}}(\theta)$, enabling coordinated refinement of role-conditioned agent behaviors. Corresponding optimization steps are present in Algorithm 1.

5 Experimental Evaluation

We evaluate SHARP's effectiveness, stability, scalability, and interpretability through three research questions: **RQ1:** How does SHARP perform compared to existing single-agent and multi-agent baselines across diverse benchmarks? **RQ2:** How stable and scalable is SHARP across heterogeneous tasks, model sizes, and training steps? **RQ3:** How does SHARP characterize and reshape hierarchical multi-agent coordination? Extended analyses and cases are left in Appendices D–G.

Benchmarks. Benchmarks. We utilize five distinct benchmarks, including MuSiQue Trivedi et al. (2022), GAIA-text Mialon et al. (2023), WebWalkerQA Wu et al. (2025), FRAMES Krishna et al. (2025) and DocMath-Eval Zhao et al. (2024). Please see Appendix B for more details.

Baselines and Setup. We compare our method against four categories of baselines. (i) **Vanilla LLMs**, including Llama-3.1-it 8B Grattafiori et al. (2024) and Qwen-3 8B Yang et al. (2025). (ii) **Prompt based planning and search methods**, such as Plan-Search Yu et al. (2023). (iii) **Single agent reinforcement learning methods**, including Search R1 Jin et al. (2025) and Single agent GRPO Shao et al. (2024). (iv) **Multi-agent systems without marginal credit modeling**, including untrained hierarchical MAS, global-reward MAS methods such as G-Designer Zhang et al. (2025b) and CARD Wu et al. (2026), and RL methods using local rewards such as AceSearcher Xu et al. (2025), COA Li et al. (2025a), and MATPO Mo et al. (2025). Unless otherwise specified, SHARP uses $K = 3$; see Appendices C and E for implementation details and cost–performance analysis.

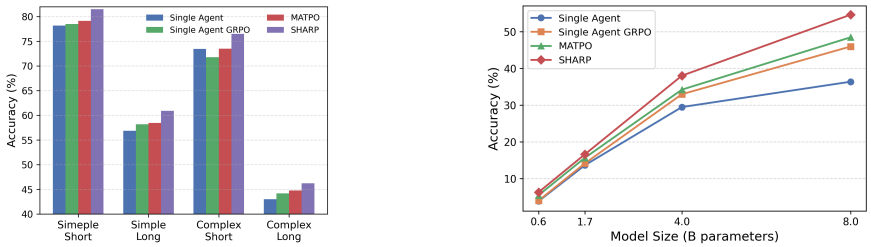

Figure 3: Generalization and scalability analysis. Left: performance on DocMath-Eval. Right: parameter scalability on MuSiQue from 0.6B to 8B. SHARP shows consistent improvement as the model size increases and achieves a larger advantage over the baselines at larger scales.

5.1 RQ1: Overall Performance

Motivation: Multi-agent LLM systems vary across intertwined factors such as architecture, training-based optimization, and reward design, making the source of performance gains unclear. A controlled comparison is therefore required to assess the contribution of marginal credit modeling.

Observation 1: Marginal credit modeling consistently delivers the best overall performance. As shown in Table 1, SHARP (Qwen) achieves the highest average performance of 35.63, surpassing the strongest non-marginal multi-agent baseline MATPO by 4.89 points. Overall, SHARP yields average match gains of 35.01% and 24.97% over single-agent and multi-agent baselines, respectively. Appendix C details the computation and the results demonstrate consistent improvements across diverse benchmarks. Taken together, these results indicate that modeling marginal credit is a key factor behind SHARP's consistent gains, beyond architectural design or optimization strategy alone.

Observation 2: Optimization strategy outperforms structural complexity alone. The results in Table 1 reveal a progressive performance trend: zero-shot RAG baselines score 5.69–8.06, while untrained Plan-Search improves scores to 12.73–24.77. Single-agent GRPO further raises the average score to 27.95. Untrained Hierarchical MAS yields limited gains of 18.84–26.26, while MAS RL baselines without marginal credit modeling reach 28.06–30.74. By incorporating marginal credit into MAS optimization, SHARP achieves the best average score of **35.63**. This progression indicates that both MAS architectures and reinforcement learning are crucial for enhancing planning and execution capabilities in realistic scenarios.

> **Insight 1:** Marginal credit modeling provides a stronger performance boost than architectural structure or optimization strategy alone, consistently yielding superior overall accuracy across benchmarks in heterogeneous multi-agent LLM systems.

5.2 RQ2: Stability and Scalability Analysis

Motivation: Practical MAS require robustness to task heterogeneity, model scaling, and long-horizon training, motivating a systematic analysis of stability and scalability.

Observation 1: Generalization across diverse reasoning tasks. We evaluate four DocMath-Eval scenarios: simple-short, simple-long, complex-short, and complex-long Zheng et al. (2025); Tang et al. (2025). Figure 3(left) shows that SHARP performs best across all scenarios, exceeding MATPO by 1.46–2.98 points. SHARP therefore generalizes across document reasoning tasks with varied context lengths and complexities.

Observation 2: Scaling with model parameters. We examine performance across Qwen3 backbones from 0.6B to 8B on MuSiQue. Single-agent accuracy rises from 3.85 to 36.35, while Single-agent GRPO and MATPO improve from 4.01 and 5.47 at 0.6B to 45.93 and 47.43 at 8B, respectively. Figure 3(right) shows that SHARP scales more effectively, rising from 6.29 at 0.6B to 54.58 at 8B. The difference between SHARP and single-agent models widens with model size, reaching 18.23 points at 8B. This trend verifies that SHARP becomes more effective on stronger backbones.

Observation 3: Training stability and sampling-budget scaling. Figure 4(left) analyzes GAIA-text training over 180 optimization steps. Single-agent GRPO and MATPO fluctuate noticeably, whereas both SHARP variants achieve stronger final performance. Increasing the sampling budget from

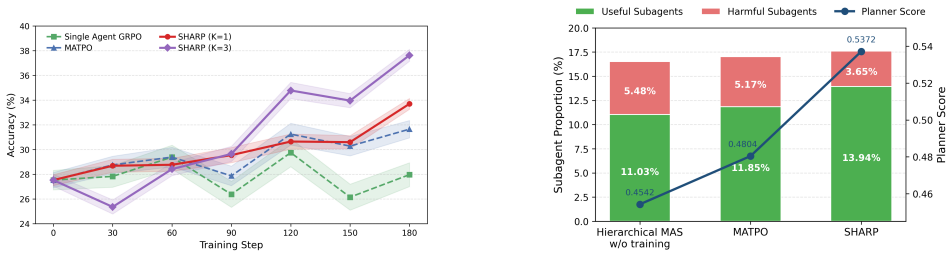

Figure 4: Left: GAIA-text training-step scalability across Monte Carlo Shapley budgets. SHARP with $K = 1$ and $K = 3$ is compared with Single-agent GRPO and MATPO; shading denotes 95% confidence intervals. Right: coordination metrics based on marginal Shapley credit. Subagents with positive marginal credit are categorized as useful, while negative ones are categorized as harmful.

$K = 1$ to $K = 3$ raises final accuracy from 33.70 to 37.63, showing that more accurate marginal credit estimates provide stronger optimization signals. Explicit marginal credit therefore improves training effectiveness, with $K = 3$ achieving the strongest final performance here.

> **Insight 2:** Explicit marginal credit modeling yields robust generalization across task distributions, scales effectively with model size, and stabilizes long-horizon training, indicating that SHARP benefits more from stronger backbones and extended optimization.

5.3 RQ3: Coordination Analysis

Motivation: Training strategies reshape multi-agent coordination, but their impact on agent interactions remains unclear. We analyze Shapley-based credit signals in a hierarchical planner-worker setting on MuSiQue, using the planner's average marginal credit as the planning coordination score and labeling subagents with positive or negative marginal credit as useful or harmful, respectively.

Observation 1: Improved planning and subagent selection. Figure 4(right) shows planner scores and subagent utility across training schemes. SHARP increases the planner score to 0.5372, outperforming the vanilla baseline at 0.4542 and MATPO at 0.4804. Simultaneously, useful subagents increase from 11.03% to 13.94%. These results indicate that SHARP guides the planner to generate more effective subtasks and select workers more strategically.

Observation 2: Filtering harmful interactions. Marginal credit assignment identifies and penalizes counterproductive actions. SHARP reduces harmful subagent calls Chan et al. (2025) from 5.48% to 3.65%. By assigning lower credit to workers who degrade final outcomes, SHARP cleans the coordination trace and encourages more reliable agent interactions. Appendix G presents qualitative cases of useful and harmful subagent behaviors.

Observation 3: Common limits in systemic efficiency. Despite these improvements, *useful* subagents remain a minority of total calls in the tested systems. This confirms that much of current multi-agent tool usage could be redundant or neutral. The observation reveals a fundamental bottleneck in multi-agent design and motivates more aggressive pruning of low-utility execution paths.

> **Insight 3:** Shapley-based marginal credit modeling reshapes planner–worker coordination by improving subtask selection, filtering harmful interactions, and exposing inefficiencies in multi-agent coordination.

6 Conclusions

We present SHARP, a principled framework for credit assignment in tool-using multi-agent LLM systems. By combining globally aligned rewards with Shapley-based marginal attribution and lightweight format constraints, SHARP provides stable learning signals for collaborative policy optimization. Empirically, SHARP consistently improves both task performance and inference efficiency across benchmarks. Further analysis reveals a clear trade-off between the fidelity of credit estimation and computational cost, where moderate approximation achieves the best balance. Overall, SHARP offers a scalable and interpretable approach to credit-aware optimization, providing a foundation for efficient coordination in complex multi-agent systems.

AI USE STATEMENT

Generative AI tools were used to improve grammar, spelling, and word choice during manuscript editing. The authors reviewed all AI-assisted edits and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

Section 4.3 specifies the SHARP objective and Monte Carlo Shapley credit estimator, while Appendix A provides pseudocode. Appendix B details the benchmarks, data splits, and evaluation protocols. Appendix C reports model backbones, optimization settings, reward weights, counterfactual trajectory construction, and tool-execution setup. Appendix H provides the system prompts and tool schemas. An anonymous implementation link is provided in the abstract.

REFERENCES

Alan Chan, Kevin Wei, Sihao Huang, Nitarshan Rajkumar, Elija Perrier, Seth Lazar, Gillian K Hadfield, and Markus Anderljung. Infrastructure for ai agents. *arXiv preprint arXiv:2501.10114*, 2025.

Hugh Chen, Ian C Covert, Scott M Lundberg, and Su-In Lee. Algorithms to estimate shapley value feature attributions. *Nature Machine Intelligence*, 5(6):590–601, 2023.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025.

Mohamed Amine Ferrag, Norbert Tihanyi, and Merouane Debbah. From llm reasoning to autonomous ai agents: A comprehensive review. *arXiv preprint arXiv:2504.19678*, 2025.

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Haoyang Hong, Jiajun Yin, Yuan Wang, Jingnan Liu, Zhe Chen, Ailing Yu, Ji Li, Zhiling Ye, Hansong Xiao, Yefei Chen, et al. Multi-agent deep research: Training multi-agent systems with m-grpo. *arXiv preprint arXiv:2511.13288*, 2025.

Jiwoo Hong, Noah Lee, and James Thorne. Orpo: Monolithic preference optimization without reference model. *arXiv preprint arXiv:2403.07691*, 2024.

Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiawu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. Metagpt: Meta programming for a multi-agent collaborative framework. In *The twelfth international conference on learning representations*.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4745–4759, 2025.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Jiahui Li, Kun Kuang, Baoxiang Wang, Furui Liu, Long Chen, Fei Wu, and Jun Xiao. Shapley counterfactual credits for multi-agent reinforcement learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 934–942, 2021.

Weizhen Li, Jianbo Lin, Zhuosong Jiang, Jingyi Cao, Xinpeng Liu, Jiayu Zhang, Zhenqiang Huang, Qianben Chen, Weichen Sun, Qiexiang Wang, et al. Chain-of-agents: End-to-end agent foundation models via multi-agent distillation and agentic rl. *arXiv preprint arXiv:2508.13167*, 2025a.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025b.

Zimu Lu, Houxing Ren, Yunqiao Yang, Ke Wang, Zhuofan Zong, Junting Pan, Mingjie Zhan, and Hongsheng Li. Webgen-agent: Enhancing interactive website generation with multi-level feedback and step-level reinforcement learning. *arXiv preprint arXiv:2509.22644*, 2025.

Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. Gaia: a benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhanfeng Mo, Xingxuan Li, Yuntao Chen, and Lidong Bing. Multi-agent tool-integrated policy optimization. *arXiv preprint arXiv:2510.04678*, 2025.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip HS Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. Malt: Improving reasoning with multi-agent llm training. *arXiv preprint arXiv:2412.01928*, 2024.

Davide Napolitano and Luca Cagliero. Lightningshap: A cost-effective approach to local shapley values estimation. *IEEE Transactions on Artificial Intelligence*, 2025.

Michal Nauman and Marek Cygan. On many-actions policy gradient. In *International Conference on Machine Learning*, pp. 25769–25789. PMLR, 2023.

Chanwoo Park, Seungju Han, Xingzhi Guo, Asuman E Ozdaglar, Kaiqing Zhang, and Joo-Kyung Kim. Maporl: Multi-agent post-co-training for collaborative large language models with reinforcement learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30215–30248, 2025.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. Chatdev: Communicative agents for software development. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 15174–15186, 2024.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder De Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

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

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. *Advances in Neural Information Processing Systems*, 36:38154–38180, 2023.

Lin Sun, Chuang Liu, Can Zhang, Yubin Wu, Weijia Lu, and Ning Wu. Agentpo: Enhancing multi-agent collaboration via reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *arXiv preprint arXiv:1706.05296*, 2017.

Zichen Tang, Jiacheng Liu, Zhongjun Yang, Rongjin Li, Zihua Rong, Haoyang He, Zhuodi Hao, Xinyang Hu, Kun Ji, Ziyan Ma, et al. Finmmr: make financial numerical reasoning more multimodal, comprehensive, and challenging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3245–3257, 2025.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. Musique: Multihop questions via single-hop question composition. *Transactions of the Association for Computational Linguistics*, 10:539–554, 2022.

Hongru Wang, Yujia Qin, Yankai Lin, Jeff Z Pan, and Kam-Fai Wong. Empowering large language models: Tool learning for real-world interaction. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2983–2986, 2024.

Jianhong Wang, Yuan Zhang, Tae-Kyun Kim, and Yunjie Gu. Shapley q-value: A local reward approach to solve global reward games. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7285–7292, 2020.

Yinjie Wang, Ling Yang, Ye Tian, Ke Shen, and Mengdi Wang. Co-evolving llm coder and unit tester via reinforcement learning. *arXiv preprint arXiv:2506.03136*, 2025.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. Autogen: Enabling next-gen llm applications via multi-agent conversations. In *First conference on language modeling*.

Tongtong Wu, Yanming Li, Ziye Tang, Chen Jiang, Linhao Luo, Guilin Qi, Shirui Pan, and Gholamreza Haffari. Card: Towards conditional design of multi-agent topological structures. In *The Fourteenth International Conference on Learning Representations*, 2026.

Ran Xu, Yuchen Zhuang, Zihan Dong, Jonathan Wang, Yue Yu, Joyce C Ho, Linjun Zhang, Haoyu Wang, Wenqi Shi, and Carl Yang. Acesearcher: Bootstrapping reasoning and search for llms via reinforced self-play. *arXiv preprint arXiv:2509.24193*, 2025.

Xiangyuan Xue, Yifan Zhou, Guibin Zhang, Zaibin Zhang, Yijiang Li, Chen Zhang, Zhenfei Yin, Philip Torr, Wanli Ouyang, and Lei Bai. Comas: Co-evolving multi-agent systems via interaction rewards. *arXiv preprint arXiv:2510.08529*, 2025.

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.

Xiao Yu, Maximillian Chen, and Zhou Yu. Prompt-based monte-carlo tree search for goal-oriented dialogue policy planning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7101–7125, 2023.

Boyang Zhang, Yicong Tan, Yun Shen, Ahmed Salem, Michael Backes, Savvas Zannettou, and Yang Zhang. Breaking agents: Compromising autonomous llm agents through malfunction amplification. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34952–34964, 2025a.

Guibin Zhang, Yanwei Yue, Xiangguo Sun, Guancheng Wan, Miao Yu, Junfeng Fang, Kun Wang, Tianlong Chen, and Dawei Cheng. G-designer: Architecting multi-agent communication topologies via graph neural networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 76678–76692. PMLR, 2025b.

Yilun Zhao, Yitao Long, Hongjun Liu, Ryo Kamoi, Linyong Nan, Lyuhao Chen, Yixin Liu, Xiangru Tang, Rui Zhang, and Arman Cohan. Docmath-eval: Evaluating math reasoning capabilities of llms in understanding long and specialized documents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pp. 16103–16120, 2024.

Da Zheng, Lun Du, Junwei Su, Yuchen Tian, Yuqi Zhu, Jintian Zhang, Lanning Wei, Ningyu Zhang, and Huajun Chen. Knowledge augmented complex problem solving with large language models: A survey. *arXiv preprint arXiv:2505.03418*, 2025.

Appendix

A **Algorithm Workflow** 15

B **Information of Training and Test Dataset** 16

C **Implementation Details** 16

D **Reward Weighting and Component Contribution** 17

E **Effect of Shapley Sampling Budget on Cost and Performance** 18

F **Catastrophic Forgetting Analysis** 19

G **Case Study** 20

H **System Prompts** 34

I **Impact Statement** 38

J **Limitations and Discussions** 39

A ALGORITHM WORKFLOW

Algorithm 1 Workflow of SHARP for Tool-Integrated Planning

Input: Query q; current policy π_θ; old policy $\pi_{\theta_{\text{old}}}$; group size G; Monte Carlo sample size K; clip ϵ; constant δ; reward weights α, β, γ; tool format scorer $\psi(\cdot)$.
Output: Updated parameters θ.

for $i \leftarrow 1$ **to** G **do**
  // Collect TIP trajectory
  Initialize trajectory context $c_{i,1}$ with query q Initialize trajectory $\tau_i \leftarrow \emptyset$ and participating agent set $\mathcal{M}_i \leftarrow \emptyset$
  for $t \leftarrow 1$ **to** T_i **do**
    Invoke an agent $m_{i,t}$ conditioned on context $c_{i,t}$ Sample local trajectory segment $\tau_i^t \sim \pi_{\theta_{\text{old}}}(\cdot \mid c_{i,t})$ Append τ_i^t to τ_i and update $\mathcal{M}_i \leftarrow \mathcal{M}_i \cup \{m_{i,t}\}$ Update context $c_{i,t+1}$ with the generated segment and tool observations
  // Compute compositional rewards
  Compute terminal accuracy reward $R_{\text{acc}}(\tau_i) \in \{0, 1\}$
  foreach $m \in \mathcal{M}_i$ **do**
    $R_{i,m}^{\text{b}} \leftarrow R_{\text{acc}}(\tau_i)$
    if $T_{i,m} > 0$ **then**
      $R_{i,m}^{\text{fmt}} \leftarrow \frac{1}{T_{i,m}} \sum_{j=1}^{T_{i,m}} \psi(a_{i,j}^m)$
    else
      $R_{i,m}^{\text{fmt}} \leftarrow 0$
  // Monte Carlo Shapley marginal credit
  foreach $m \in \mathcal{M}_i$ **do**
    $\text{credit}_{i,m} \leftarrow 0$
    for $k \leftarrow 1$ **to** K **do**
      Sample Shapley-kernel coalition $S_k^{(m)}$;
      Construct counterfactual trajectory $\tau_i^{S_k^{(m)}}$ induced by coalition $S_k^{(m)}$ Construct counterfactual trajectory $\tau_i^{S_k^{(m)} \cup \{m\}}$ induced by coalition $S_k^{(m)} \cup \{m\}$
      $\text{credit}_{i,m} \leftarrow \text{credit}_{i,m} + R_{\text{acc}}\left(\tau_i^{S_k^{(m)} \cup \{m\}}\right) - R_{\text{acc}}\left(\tau_i^{S_k^{(m)}}\right)$
    $\text{credit}_{i,m} \leftarrow \frac{1}{K}\text{credit}_{i,m}$ $R_{i,m}^{\text{mc}} \leftarrow \text{credit}_{i,m}$
  foreach $m \in \mathcal{M}_i$ **do**
    $\bar{R}_{i,m} \leftarrow \alpha R_{i,m}^{\text{b}} + \beta R_{i,m}^{\text{mc}} + \gamma R_{i,m}^{\text{fmt}}$
  // Group-relative advantages
foreach *agent identity* m *appearing in the sampled group* **do**
  $\mu_m \leftarrow \text{mean}(\{\bar{R}_{i,m} : m \in \mathcal{M}_i\}_{i=1}^G)$ $\sigma_m \leftarrow \text{std}(\{\bar{R}_{i,m} : m \in \mathcal{M}_i\}_{i=1}^G)$
// SHARP policy update
for $i \leftarrow 1$ **to** G **do**
  foreach $m \in \mathcal{M}_i$ **do**
    $\hat{A}_{i,m} \leftarrow \frac{\bar{R}_{i,m} - \mu_m}{\sigma_m + \delta}$
    $R_{i,m} \leftarrow \exp\left(\sum_{j=1}^{T_{i,m}} \log \frac{\pi_\theta\left(a_{i,j}^m \mid \text{ctx}_{i,j}^m\right)}{\pi_{\theta_{\text{old}}}\left(a_{i,j}^m \mid \text{ctx}_{i,j}^m\right)}\right)$
    $R_{i,m}^{\text{clip}} \leftarrow \min\left(R_{i,m}\hat{A}_{i,m},\ \text{clip}(R_{i,m}, 1-\epsilon, 1+\epsilon)\hat{A}_{i,m}\right)$
$J_{\text{SHARP}}(\theta) \leftarrow \frac{1}{G}\sum_{i=1}^G \frac{1}{|\mathcal{M}_i|}\sum_{m \in \mathcal{M}_i} R_{i,m}^{\text{clip}}$
Update $\theta \leftarrow \theta + \eta \nabla_\theta J_{\text{SHARP}}(\theta)$

B Information of Training and Test Dataset

Benchmarks. We consider five benchmarks covering multi-hop reasoning, tool-assisted interaction, and structured decision making:

- **MuSiQue** Trivedi et al. (2022): a multi-hop question answering benchmark constructed by composing connected single-hop questions, designed to require genuine multi-step reasoning.
- **GAIA-text** Mialon et al. (2023): evaluates general AI assistants on real-world questions requiring reasoning, web browsing, and tool-use proficiency.
- **WebWalkerQA** Wu et al. (2025): benchmarks multi-step web traversal and evidence aggregation through structured navigation.
- **FRAMES** Krishna et al. (2025): a unified benchmark for evaluating factuality, retrieval, and reasoning in end-to-end retrieval-augmented generation.
- **DocMath-Eval** Zhao et al. (2024): evaluates document-grounded numerical reasoning over long-context documents with text and tables.

Training and Test Splits. We use a subset of these benchmarks for training and evaluation. For MuSiQue, we train our models on 5,975 instances and evaluate on a held-out test set of 200 instances. For GAIA-text, WebWalkerQA, FRAMES, and DocMath-Eval, we do not use any training data and evaluate our models directly on their released test sets following the official evaluation protocols.

Data Release. To ensure reproducibility and facilitate future research, we will release all training and test splits used in our experiments, together with preprocessing scripts and evaluation protocols, upon publication.

C Implementation Details

As stated in the main manuscript, SHARP outperforms both single-agent and multi-agent baselines, achieving empirical gains of 35.01% and 24.97%, as confirmed by comparisons with baselines built on the Qwen3-8B backbones for fairness. The involved baselines include Plan-Search, Single-agent GRPO, Hierarchical MAS, CARD, COA and MATPO. Those models designed with LLaMA-3.1-8B or Qwen2.5-7B, or purely typology strategy (G-Designer) are not involved for fairness.

Training Setup. Our optimization framework leverages pre-trained weights from the *Qwen-3-8B* and *LLaMA-3.1-8B* instruction-tuned models. We employ the AdamW optimizer with a fixed learning rate of 10^{-5} and a global batch size of 256. To enhance trajectory diversity during the learning phase, we generate 8 rollouts per input instance. The training process is executed over 180 gradient steps across a cluster of 64×A100 GPUs, using data parallelism to maintain throughput. Hyperparameters remain consistent across all benchmark evaluations unless otherwise noted, with reward coefficients specifically tuned to $\alpha = 0.9$, $\beta = 0.9$, and $\gamma = 0.1$.

Counterfactual Trajectory Construction. To evaluate the value of a sampled coalition S, we construct a coalition-conditioned counterfactual trajectory τ_i^S by applying an agent-exclusion intervention to the realized trajectory τ_i. Specifically, only the trajectory segments generated by agents in S are retained, while segments generated by agents outside S are removed from the interaction history and replaced with a null segment $\varnothing$, which produces no reasoning content, tool invocation, or tool observation. For each removed segment, we identify downstream agents whose contexts depend on the removed content and reroll out only those downstream agents that belong to S under the modified context. The resulting trajectory is evaluated using the same final-answer aggregation and accuracy evaluation procedure as the original trajectory, yielding $R_{\text{acc}}(\tau_i^S)$. This procedure defines the coalition value relative to the specified null-segment absence baseline.

SHARP Optimization. To facilitate fine-grained attribution, we utilize the SHARP optimization protocol with Monte Carlo Shapley-based marginal credit estimation. For each generated trajectory

τ_i and each participating agent $m \in \mathcal{M}_i$, we sample K Shapley-kernel coalitions coalitions $S_k^{(m)} \subseteq \mathcal{M}_i \setminus \{m\}$ and estimate the agent's marginal credit as

$$\text{credit}_{i,m} = \frac{1}{K} \sum_{k=1}^{K} \left[R_{\text{acc}}\left(\tau_i^{S_k^{(m)} \cup \{m\}}\right) - R_{\text{acc}}\left(\tau_i^{S_k^{(m)}}\right) \right],$$

where τ_i^S denotes the counterfactual trajectory induced by coalition S. This replaces single-agent leave-one-out masking with a coalition-based approximation of Shapley marginal contribution.

We estimate group-wise normalization statistics (μ_m, σ_m) across each mini-batch, maintaining stability with parameters $\epsilon = 0.2$ and $\delta = 10^{-6}$ throughout all experimental runs.

Tool Execution. Environment interactions are mediated through a role-scoped service interface built on FastMCP. This architecture enforces a strict separation of concerns, where the primary agent manages delegation and offloads execution to specialized auxiliary agents with non-overlapping toolsets. For the computational sandbox, A dedicated Python agent connects to a remote environment via the python-mcp-server. It exposes an *execute_python_code* tool that allows the safe execution of dynamic code with configurable timeouts. Moreover, for the object of web exploration, the Browsing agent interfaces with a *duck-mcp-server*, providing a sequential pipeline of *duckduckgo_search* and scrape for real-time information retrieval.

D Reward Weighting and Component Contribution

SHARP combines three reward components: the globally broadcast accuracy reward, the marginal credit reward, and the tool format reward. In all main experiments, we set the reward weights to $\alpha = 0.9$, $\beta = 0.9$, and $\gamma = 0.1$ for the three components, respectively. This configuration reflects their different roles in optimization: the global reward and marginal credit reward directly optimize task-level success and agent-level attribution, while the tool format reward provides auxiliary supervision for valid tool invocation.

The smaller weight for the tool format reward is motivated by its relatively narrow supervision target. Specifically, the tool format reward only evaluates whether generated tool calls are syntactically valid and conform to the required schema. This capability is typically learned quickly during training and tends to saturate earlier than task-level rewards. Therefore, assigning a smaller weight to this component is sufficient to encourage executable tool calls while avoiding excessive interference with outcome-driven optimization.

To quantify the contribution of each reward component, we conduct an ablation study on GAIA-text. Table 2 reports the performance under different reward configurations.

Table 2: Ablation study of reward components on GAIA-text. The full SHARP setting uses $\alpha = 0.9$, $\beta = 0.9$, and $\gamma = 0.1$.

Setting	Performance
w/o global accuracy reward	30.80
w/o marginal credit reward	31.65
w/o tool format reward	35.57
Full SHARP, equal weights	36.05
Full SHARP, $\alpha = 0.9, \beta = 0.9, \gamma = 0.1$	**37.63**

The results show that all reward components contribute to the final performance. Removing the global accuracy reward causes the largest degradation, indicating that terminal task success remains the primary optimization signal. Removing the marginal credit reward also leads to a clear performance drop, confirming the importance of agent-level credit attribution. The tool format reward provides a smaller but consistent gain, suggesting that format-level supervision improves tool-use reliability without dominating the learning signal. Compared with equal weighting, the adopted configuration further improves performance, supporting the choice of assigning stronger weights to task-level and credit-level rewards while using a lighter weight for tool-call format validity.

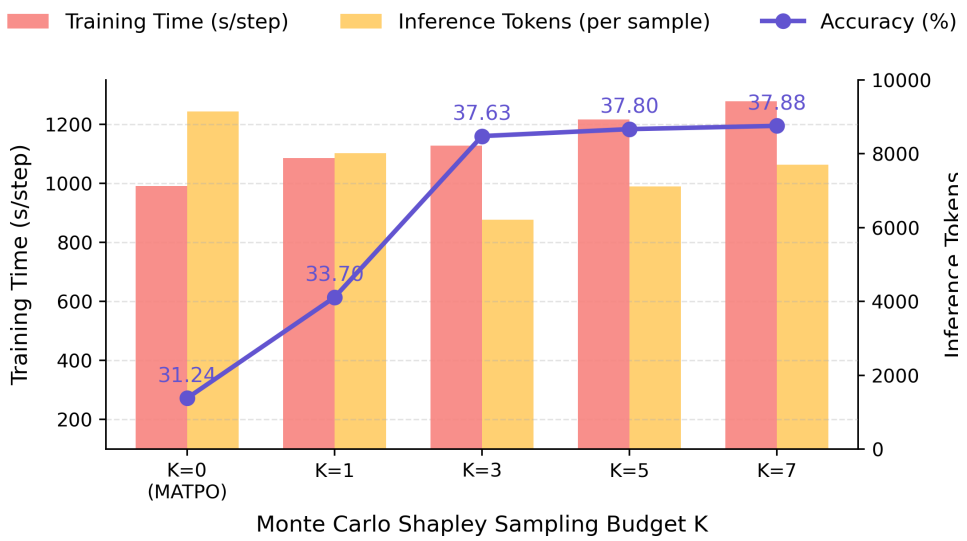

Figure 5: **Cost–performance trade-off under different Shapley sampling budgets.** We vary the Monte Carlo Shapley sampling budget K, where $K = 0$ denotes the non-marginal MATPO baseline. Red bars show training latency, yellow bars show inference tokens, and the purple line shows normalized accuracy with original values annotated. $K = 3$ reaches a near-saturated accuracy level while achieving the lowest inference token consumption; larger K yields marginal accuracy gains at higher training cost.

E Effect of Shapley Sampling Budget on Cost and Performance

Setup. To evaluate the cost and efficiency of Monte Carlo Shapley-based credit estimation, we conduct this analysis on GAIA-text and vary the sampling budget K used for marginal credit computation. Here, K denotes the number of sampled counterfactual agent collaborations used to estimate each agent's marginal contribution. When $K = 0$, no Shapley-based marginal credit is computed, corresponding to the MATPO-style non-marginal credit baseline. When $K > 0$, SHARP estimates marginal credit by averaging counterfactual value differences over K sampled agent collaborations. Thus, a larger K gives a more fine-grained estimate of agent-level marginal credit, but also requires more counterfactual evaluations during training. We report training latency per step, inference token usage per sample, and final GAIA-text accuracy to analyze how credit-estimation fidelity affects training cost, deployment efficiency, and task performance.

Effect of the Sampling Budget K. As shown in Figure 5, increasing K initially improves both accuracy and inference efficiency. Compared with the $K = 0$ baseline, using $K = 1$ increases accuracy from 31.24 to 33.70, suggesting that even a small number of counterfactual samples provides useful marginal credit signals. Increasing the budget to $K = 3$ further improves accuracy to 37.63 and reduces inference token consumption from 9140 to 6213 tokens. This indicates that moderate sampling improves the reliability of credit estimation, allowing the policy to better identify useful agent interactions and suppress redundant reasoning or tool calls. In this regime, additional credit computation during training translates into more concise and effective inference behavior. A qualitative case study in Appendices G.3 and G.4 provides further evidence for this behavior.

Diminishing Returns for Large K. The benefit of increasing K becomes limited after $K = 3$. When K increases from 3 to 5 and 7, training time continues to rise from 1127s/step to 1216s/step and 1278s/step, respectively. Accuracy increases only marginally from 37.63 to 37.80 and 37.88, while inference token usage increases from 6213 to 7115 and 7701 tokens. Larger budgets more closely approximate the Shapley value through additional counterfactual evaluations, yet provide limited gains in final accuracy. Once sampled collaborations sufficiently identify dominant agent contributions, extra samples predominantly increase credit-estimation cost with limited new optimization signal. Therefore, K controls a trade-off between credit-estimation fidelity and optimization efficiency, and $K = 3$ provides the most favorable empirical balance in this study.

Table 3: **Assessment of Knowledge Retention via MMLU.** We report MMLU accuracies to monitor potential catastrophic forgetting following domain-specific training. Both MATPO and SHARP demonstrate that specialized optimization does not degrade general-domain knowledge. Instead, the trained variants exhibit marginal performance gains over the inference-only baseline, with SHARP achieving the highest overall accuracy.

Method	MMLU (Accuracy ↑)
Hierarchical MAS	93.04
MATPO	94.00
sharp	94.65

Mechanistic Insight. The improvement at moderate K stems from more informative agent-level attribution. Compared with global reward assignment, Shapley-based marginal credit provides a clearer signal about which agents contribute positively to the final outcome. This helps reduce unnecessary agent invocations, redundant intermediate reasoning, and inefficient tool use during inference, which explains the reduction in token consumption at $K = 3$. However, once the marginal credit signal is sufficiently informative, further increasing K may add redundant counterfactual evaluations rather than useful new information. This explains why $K = 5$ and $K = 7$ incur higher training cost while providing only marginal accuracy gains.

Computational Tractability. Although Monte Carlo Shapley estimation introduces additional training-time computation, the overhead remains practical. Counterfactual evaluations for different sampled collaborations can be parallelized, and the practical trade-off setting $K = 3$ increases training time only moderately from 990s/step to 1127s/step. In return, it improves accuracy from 31.24 to 37.63 and reduces inference token usage substantially from 9140 to 6213. This indicates a favorable cost–benefit trade-off: a modest increase in offline training cost leads to both higher task performance and much lower deployment-time inference cost. We further observe that the reduction in inference tokens comes from more concise and efficient collaboration patterns, where agents avoid redundant reasoning and unnecessary tool use. A qualitative case study in Appendices G.3 and G.4 provides further evidence for this behavior.

Takeaway. The sampling budget K determines how much counterfactual evidence is used to estimate marginal credit. A small K provides limited but useful credit signals, and a moderate K improves attribution quality and coordination efficiency. Larger K values better approximate the Shapley value, but their accuracy gains are limited relative to the added training overhead. Among the evaluated settings, $K = 3$ offers the most favorable cost–performance trade-off, demonstrating that SHARP does not require exhaustive Shapley estimation to obtain effective marginal credit for tool-using multi-agent LLM training.

F Catastrophic Forgetting Analysis

Setup. A common concern when training on a specialized task distribution (e.g., search/tool-heavy trajectories) is *catastrophic forgetting* of general domain knowledge. To assess the impact of search-domain optimization on the model's underlying knowledge base, we evaluate the Qwen3 backbone on the MMLU benchmark across three distinct configurations: (i) **Hierarchical MAS (Baseline)**, an inference-only hierarchical setup without further training; (ii) **MATPO**, representing a standard policy optimization approach trained on the search dataset; and (iii) **SHARP**, our proposed framework utilizing Shapley-based marginal credit on the same training distribution.

Results and discussion. As detailed in Table 3, specialized training on search-centric trajectories does not lead to a regression in general-domain performance. On the contrary, both trained models surpass the inference-only Hierarchical MAS baseline (93.04%). MATPO shows a modest improvement (94.00%), while SHARP achieves the most significant gain (94.65%). These results provide empirical evidence that our training protocol preserves, rather than erodes, the pre-trained knowledge base. The observed performance uplift suggests a degree of *positive transfer*, where the refined reasoning and coordination capabilities acquired during search-domain training generalize to improve zero-shot knowledge retrieval.

Why forgetting is mitigated? We hypothesize that the alleviation of catastrophic forgetting is rooted in the nature of the SHARP optimization objective. Unlike standard fine-tuning that might necessitate substantial representation shifts to accommodate new factual content, SHARP primarily refines *agent coordination and credit assignment* logic. In broadcast-based reinforcement learning, the global reward signal is often confounded, potentially leading to noisy and indiscriminate parameter updates across the model. In contrast, SHARP utilizes Shapley-based marginal credit to provide a high-fidelity, targeted signal that localizes the causal contribution of specific subagent actions. By improving the signal-to-noise ratio of the updates, SHARP minimizes unnecessary perturbations to the model's latent representations, thereby safeguarding its general-purpose knowledge. Collectively, these findings underscore the robustness of our framework in specialized environments without sacrificing the foundational capabilities of the backbone LLM.

G Case Study

G.1 Case Study 1: Useful Sub-agent

Problem Statement

By 1931, what percentage of the world's Jews were the group of Jews that represented the bulk of modern Jewry?

Full Response Trajectory

<think> Okay, the user is asking about the percentage of the world's Jews in 1931 that belonged to the group representing the bulk of modern Jewry. First, I need to understand what group that is. Modern Jewry is mostly made up of Ashkenazi Jews, right? They are of Central and Eastern European origin. But I'm not 100% sure, so I should verify that.
Wait, the question is about the percentage in 1931. So I need to find out the distribution of Jewish populations by region in that year. The main regions were Europe, the US, and others. Ashkenazi Jews were predominant in Europe, but by 1931, there might have been significant migration, especially due to the rise of antisemitism and the start of the Holocaust. Maybe the percentage had shifted?
I should check historical data on Jewish populations in 1931. The tool provided is search_and_browse from the browsing_agent server. I need to use that to find the specific percentage. The subtask should be to find the percentage of world Jews in 1931 that were part of the dominant group (Ashkenazi).
But wait, maybe the user is referring to the group that became the majority in modern times, which might not be the same as the 1931 distribution. Let me make sure. Modern Jewry is largely Ashkenazi, but the question is about the percentage in 1931. So the subtask is to find that percentage.
I should use the search_and_browse tool with the subtask: "What percentage of the world's Jews in 1931 were Ashkenazi Jews, representing the bulk of modern Jewry?" Then, the tool will return the answer. Once I get that, I can present it as the final answer. </think>

```
<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{
"subtask": "What percentage of the world's Jews in 1931 were
Ashkenazi Jews, representing the bulk of modern Jewry?"
}
</arguments>
</use_mcp_tool>
```

Summarized Sub-agent Trajectory

Final Response: Percentage of World Jews Identified as Ashkenazi in 1931

Conclusion:
By 1931, **92%** of the world's Jews were Ashkenazi Jews, representing the bulk of modern Jewry. This percentage is widely cited in historical records and aligns with the peak demographic dominance of Ashkenazi Jews prior to the Holocaust.

Supporting Information:

- **New World Encyclopedia**: Explicitly states, "By 1931, Ashkenazi Jews accounted for nearly 92 percent of world Jewry."
- **LinkedIn Post**: References "92% of the world's Jews" as the figure for Ashkenazi Jews in 1931, contextualized within historical demographic trends.
- **Historical Forum Discussion**: Mentions that "at their peak in 1931, Ashkenazi Jews accounted for 92 percent of the world's Jews," emphasizing this as a significant historical threshold.

Observations:

- The figure of 92% is consistently repeated across multiple independent sources, suggesting a high degree of consensus in historical scholarship.
- The percentage reflects the pre-Holocaust demographic distribution, which is critical for understanding the demographic shifts that followed.
- Minor variations in statistical interpretation exist, but the 92% figure is the most frequently referenced and reliable estimate.

Contribution to Main Task:
This subtask directly answers the main task by identifying Ashkenazi Jews as the group constituting **92%** of the world's Jews in 1931. This information provides the foundational demographic data required to address the broader question. The main agent may now proceed to analyze post-1931 demographic changes or explore the historical significance of this percentage in shaping modern Jewry.

Answer:

$\boxed{92\%}$

Counterfactual Response Trajectory

<think> Okay, let me try to figure this out. The question is asking about the percentage of the world's Jews in 1931 that were Ashkenazi Jews, who represent the bulk of modern Jewry. First, I know that Ashkenazi Jews are a major group within the Jewish population, traditionally from Central and Eastern Europe. Before the Holocaust, they made up a significant portion of the global Jewish population. However, I need to find the exact percentage for the year 1931. I remember that historical data shows that Ashkenazi Jews were the largest group, often around 80–90% of the total Jewish population in the early twentieth century. But I should verify this. Since the user mentioned using the tool, but the tool's response was masked, I have to rely on known historical facts.
In 1931, before the Holocaust, the majority of Jews were indeed Ashkenazi. The percentage is typically cited as around 80% to 90%, but I need to be precise. Some sources state that by the early 1900s, Ashkenazi Jews constituted about 85% of the world's Jewish population. However, exact figures can vary slightly depending on the source.

Given that the user wants the answer in a box, I should provide the most commonly accepted figure. Based on historical records, the percentage is approximately 85%. Therefore, the answer is 85%. </think>

The group of Jews that represented the bulk of modern Jewry by 1931 was the Ashkenazi Jews. Historical data indicates that Ashkenazi Jews accounted for approximately **85%** of the world's Jewish population during this period.
Answer:

$$\boxed{85}$$

Example This case illustrates the practical usefulness of sub-agents in our multi-agent framework. When sub-agents are enabled, the system produces the correct answer: Ashkenazi Jews constituted approximately 92% of the world's Jewish population by 1931. The sub-agent actively retrieves historical evidence and verifies the numerical figure using multiple sources, leading to a precise and well-grounded final response.

In contrast, the counterfactual setting without sub-agents fails to reach the correct conclusion. Although the main agent correctly recognizes Ashkenazi Jews as the dominant group, it relies on coarse prior knowledge and outputs an approximate but incorrect estimate (85%). This reflects insufficient evidence grounding and limited numerical precision.

The comparison clearly shows that sub-agents contribute useful and non-redundant information. By performing targeted retrieval and cross-checking, the sub-agent directly resolves the main agent's uncertainty and changes the outcome from incorrect to correct. This case demonstrates that sub-agents improve reasoning accuracy rather than merely increasing computational cost.

G.2 Case Study 2: Harmful Sub-agent

Problem Statement

Which party held the 1781 governorship of Virginia?

Full Response Trajectory

<think>
The user is asking which party held the governorship of Virginia in 1781. To approach this question, it is first necessary to identify who served as the governor of Virginia during that year.
Because this specific historical information may not be immediately available from memory, the appropriate strategy is to consult an external information retrieval tool. The available tool for this purpose is search_and_browse provided by the browsing_agent server.
The immediate subtask is therefore to identify the governor of Virginia in 1781. The tool requires a clearly specified subtask in structured form. Accordingly, the query should be framed to retrieve not only the governor's identity but also their political affiliation.
However, the user explicitly asks about the *political party*, not merely the individual office-holder. This introduces an additional complication, as formal political parties may not yet have existed during the period in question. Nevertheless, the subtask can be formulated as: *"Identify the political party of the governor of Virginia in 1781."*
If the tool returns only the governor's name, a secondary reasoning step can be used to infer political alignment from historical context. The process therefore proceeds by issuing this subtask and extracting the relevant information from the result.
</think>

<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>

```
{
"subtask": "Identify the political party of the governor of
Virginia in 1781."
}
</arguments>
</use_mcp_tool>
```


Summarized Sub-agent Trajectory

Final Response: Political Party of the 1781 Virginia Governor

Conclusion. The governor of Virginia in 1781 was **Thomas Jefferson**. He was associated with Anti-Federalist and early Republican political principles. Importantly, no formal political party structure existed in Virginia or the United States at that time.

Supporting Information.

- **Thomas Jefferson's role:** Served as governor of Virginia from 1779 to 1781 during the American Revolutionary War.
- **Political context:** Formal political parties such as the Democratic–Republican Party and the Federalist Party emerged later, primarily in the 1790s.
- **Factional alignment:** Jefferson opposed a strong centralized federal government, a position later associated with the Democratic–Republican tradition.

Observations.

- The question applies a modern concept of political parties to a period that predates formal party systems.
- Jefferson's later party affiliations should not be retroactively applied to his 1781 governorship.
- Historical records from this period typically describe political alignments in terms of factions rather than parties.

Contribution to Main Task. This analysis establishes that the governorship of Virginia in 1781 was not held by any formal political party. Instead, it was occupied by an individual aligned with Anti-Federalist and early Republican ideological currents. This clarification allows the main task to correctly conclude that the notion of a ruling party in 1781 is historically anachronistic.

No formal party; associated with Anti-Federalist/Republican factions

Counterfactual Response Trajectory

<think>
The question to be addressed is: *"Which party held the 1781 governorship of Virginia?"*
To answer this, it is necessary to consider the political landscape of Virginia in the late eighteenth century. During this period, the United States was still in its formative years, and political parties were not yet formalized in the modern sense. Political alignment was primarily ideological rather than institutional.
The two dominant ideological groupings that later evolved into formal parties were the Federalists and the Democratic–Republicans. However, these factions were loosely organized and did not operate as structured parties with formal membership or official control of offices.
In 1781, the governor of Virginia was Thomas Jefferson. Jefferson was a prominent leader of the American Revolutionary period and a strong advocate of states' rights, limited central

authority, and individual liberties. These views later became closely associated with the Democratic–Republican tradition.
Although Jefferson is historically linked to the Democratic–Republican Party, it is important to emphasize that this party did not yet exist as a formal institution in 1781. Political identity at the time was fluid, and officeholders were not elected under formal party labels.
Nevertheless, if the question is interpreted through a retrospective party-alignment lens, Jefferson's ideological position aligns most closely with the Democratic–Republican faction. On this basis, one may attribute the 1781 governorship to that ideological grouping, while recognizing the historical limitations of such a classification.
</think>

The governor of Virginia in 1781 was Thomas Jefferson. His political views and later affiliations place him within the Democratic–Republican ideological tradition. Accordingly, when interpreted through a retrospective party framework, the governorship may be associated with the Democratic–Republican faction.

Democratic–Republican

Example This case highlights a failure mode where a sub-agent becomes *harmful* and pushes the system toward an incorrect final answer Zhang et al. (2025a); Chan et al. (2025). The question asks: *"Which party held the 1781 governorship of Virginia?"* In the **counterfactual** trajectory *without* sub-agents, the main agent answers **Democratic–Republican**, which matches the expected label in the benchmark.

However, in the **full** trajectory *with* a sub-agent, the final answer is diverted to **"No formal party; associated with Anti-Federalist/Republican factions"**, which is marked incorrect by the task. Although the sub-agent's reasoning is historically nuanced (formal parties were not fully institutionalized in 1781), it introduces an anachronism caveat and shifts the response away from the dataset's labeling scheme. As a result, the sub-agent injects *unhelpful* complexity and overrides the simpler, benchmark-aligned answer, turning a correct outcome into an incorrect one.

This contrast shows that sub-agents are not always beneficial: when they over-emphasize side conditions or redefine the question, they can *harm* performance. It motivates the need for credit assignment (and training signals) that penalize sub-agents whose contributions reduce task success, rather than rewarding all additional computation equally.

G.3 Model Responses with Shapley-based Training

Problem Statement

Who is the mother of Tsarevich Ivan Ivanovich of the country that the spouse of Kseniya Boguslavskaya is from?

Full Response Trajectory

<think> Okay, let's try to figure out this question. The user is asking for the mother of Tsarevich Ivan Ivanovich from the country where the spouse of Kseniya Boguslavskaya is from. Hmm, that's a bit complex. Let me break it down step by step.
First, I need to find out who Tsarevich Ivan Ivanovich is. I remember that Tsarevich refers to a title for a male heir to the throne, so this might be from Russian history. Ivan Ivanovich could be a historical figure, maybe from the 16th or 17th century. But I'm not entirely sure. Let me check that.
Next, the question mentions Kseniya Boguslavskaya. I need to find out who her spouse is and which country that spouse is from. Once I have the country, I can then find the mother of Tsarevich Ivan Ivanovich from that country.

Wait, but how do I connect these two? Maybe Kseniya's spouse is a historical figure from a specific country, which would be the same country as Tsarevich Ivan Ivanovich's origin. So, first, I need to determine the spouse of Kseniya Boguslavskaya. Let me try to find that information.
I think Kseniya Boguslavskaya was a noblewoman from the Russian Empire. Her husband might have been a notable figure. Let me use the search_and_browse tool to look up her spouse.
Using the tool to search for Kseniya Boguslavskaya's spouse. The subtask would be to find the spouse of Kseniya Boguslavskaya. Once I get that information, I can determine the country. Then, I need to find Tsarevich Ivan Ivanovich from that country and identify his mother.
Wait, but if the country is Russia, then Tsarevich Ivan Ivanovich might be a historical figure from there. But I'm not certain. Let me check the first part: spouse of Kseniya Boguslavskaya. Let's say the spouse is from a certain country, then proceed accordingly.
Once I have the country, I can then search for Tsarevich Ivan Ivanovich from that country and find his mother. Let me start with the first step: finding Kseniya's spouse. </think>

```
<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{
"subtask": "Identify the spouse of Kseniya Boguslavskaya and
determine their country of origin."
}
</arguments>
</use_mcp_tool>
```

Summarized Sub-agent Trajectory

Final Response: Identifying Kseniya Boguslavskaya's Spouse and Country of Origin

Conclusion:
The spouse of Kseniya Boguslavskaya could not be identified through available resources. No credible information was found regarding her marital status or the country of origin of her spouse. This subtask remains unresolved.

Supporting Information:

- **Search Attempts**: Multiple queries (e.g., "Kseniya Boguslavskaya spouse", "Kseniya Boguslavskaya spouse Russia") yielded results unrelated to the subject, including corporate recruitment pages and translation service links.
- **Irrelevant Sources**: All retrieved content pertained to modern entities (e.g., Kaiseng Haofeng Agricultural Group, Youdao translation tools) and lacked historical or personal data about Kseniya Boguslavskaya.

Observations:

- The absence of historical records or biographical data suggests either a misidentification of the name, a lack of public information, or restricted access to relevant sources.
- The name "Kseniya Boguslavskaya" does not appear in major historical databases or public records, indicating potential inaccuracies in the query.

Contribution to Main Task:
This subtask is critical to resolving the main task, as identifying the spouse's country of origin is necessary to determine the mother of Tsarevich Ivan Ivanovich. Without this information, the main task cannot proceed. The main agent should consider verifying the accuracy of the name or exploring alternative sources for historical data.

<think>
Okay, let's see. The user asked about the mother of Tsarevich Ivan Ivanovich from the country where the spouse of Kseniya Boguslavskaya is from. The first step was to find Kseniya's spouse and their country, but that didn't work. So now I need to figure out what to do next. Since the first part didn't give any results, maybe I should check if there's another way to connect these two parts. Maybe the name Kseniya Boguslavskaya is misspelled or not a real historical figure. Alternatively, perhaps Tsarevich Ivan Ivanovich is from a specific country where the spouse's country can be inferred through some other means. But without knowing the spouse's country, how can I find the mother?
Wait, maybe I should try a different search. Let me think: if the spouse of Kseniya Boguslavskaya is from a certain country, and that country is where Tsarevich Ivan Ivanovich is from, maybe I can find Tsarevich Ivan Ivanovich's mother directly? But that seems circular. Alternatively, maybe there's a common country or historical context that links them.
ßAlternatively, maybe the user made a mistake in the names. For example, Kseniya Boguslavskaya might be a fictional character or a name that's not well-documented. In that case, perhaps I should look for historical figures with similar names. Or maybe Kseniya Boguslavskaya is related to a different country. Let me try searching for "Tsarevich Ivan Ivanovich" to see which country he's associated with. If I can find that, maybe the spouse's country is the same. Let me try that next.
<think>

<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{

"subtask": "Identify Tsarevich Ivan Ivanovich's country of origin and determine his mother."
}
</arguments>
</use_mcp_tool>

Summarized Sub-agent Trajectory

Final Response: Identifying Tsarevich Ivan Ivanovich's Country of Origin and Mother

Conclusion:
Tsarevich Ivan Ivanovich was of **Russian** origin, and his mother was **Anastasia Romanovna**. This information directly addresses the subtask and provides critical data for the main task.

Supporting Information:

- **Anastasia Romanovna**: Confirmed as the mother of Tsarevich Ivan Ivanovich through multiple historical sources, including Wikipedia and educational platforms like Kids' Facts.
- **Russia**: Identified as his country of origin, consistent across all verified historical records.
- **Tsarevich Ivan Ivanovich**: Second son of Ivan the Terrible (Ivan IV Vasilyevich), as stated in https://en.wikipedia.org/wiki/Tsarevich_Ivan_Ivanovich_of_Russia and https://kids.kiddle.co/Tsarevich_Ivan_Ivanovich_of_Russia.

Observations:

- All sources independently corroborate the identity of Ivan Ivanovich's mother and his Russian heritage, indicating high reliability.
- No conflicting information was encountered, and the data aligns with established historical accounts.

Contribution to Main Task:
This subtask resolves the **country of origin** for Tsarevich Ivan Ivanovich, which is a critical component of the main task. The next step for the main agent is to determine the country of origin of **Kseniya Boguslavskaya's spouse** and verify if it relates to the already identified Russian connection. This will enable the final linkage between the spouse's country and Tsarevich Ivan Ivanovich's mother.

<think>
Okay, let's see. The user's original question was about finding the mother of Tsarevich Ivan Ivanovich from the country where the spouse of Kseniya Boguslavskaya is from.
First, I tried to find Kseniya Boguslavskaya's spouse and their country, but couldn't find any credible information. Then, I shifted focus to Tsarevich Ivan Ivanovich. I found out he was Russian and his mother was Anastasia Romanovna.
Now, the next step is to check if Kseniya Boguslavskaya's spouse is from Russia. If they are, then the answer would be Anastasia Romanovna. But since the first search didn't find the spouse's info, maybe there's a connection I'm missing. Perhaps Kseniya Boguslavskaya is linked to Russian nobility, making her spouse also from Russia. Alternatively, maybe the name Kseniya Boguslavskaya is a red herring, and the answer is still based on the Russian origin.
Since the user's question requires the country of the spouse to determine the mother, but we only have the mother from Russia, I need to confirm if the spouse's country is indeed Russia. If I can't find that, maybe the answer defaults to the information we have. But the user might expect a direct answer based on the available data. Let me check if there's any other way to

link them. Maybe Kseniya Boguslavskaya is a historical figure, and her spouse is known. But earlier searches didn't find that.
Since the previous attempts didn't resolve the spouse's country, but the mother is known as Anastasia Romanovna from Russia, perhaps the answer is Anastasia Romanovna, assuming the spouse's country is Russia. But I should make sure that the spouse's country is indeed Russia. Alternatively, there might be a missing link here.
Given that the user's question requires the country of the spouse, but we can't find that, maybe the answer can't be fully determined. However, since the mother is known from Russia, and the spouse's country isn't found, perhaps the answer is still Anastasia Romanovna, assuming the country is Russia. I'll proceed with that.
<think>
Answer:

$$\boxed{AnastasiaRomanovna}$$

G.4 Model Responses without Shapley-based Training

Problem Statement

Who is the mother of Tsarevich Ivan Ivanovich of the country that the spouse of Kseniya Boguslavskaya is from?

Full Response Trajectory

<think> Okay, let's try to figure out this question step by step. The user is asking about the mother of Tsarevich Ivan Ivanovich, but they mention the spouse of Kseniya Boguslavskaya's country. Hmm, first, I need to break down the question.
So, the main question is: Who is the mother of Tsarevich Ivan Ivanovich of the country that the spouse of Kseniya Boguslavskaya is from?
Wait, that's a bit complex. Let me parse it again. The country in question is the one where the spouse of Kseniya Boguslavskaya is from. Once I find that country, then I need to find Tsarevich Ivan Ivanovich's mother from that country.
First, I need to find who Kseniya Boguslavskaya's spouse is. Once I have that, I can determine the country of origin. Then, I need to find Tsarevich Ivan Ivanovich from that country and his mother.
But wait, I'm not sure who Kseniya Boguslavskaya is. Maybe she's a historical figure or a fictional character? Let me start by searching for Kseniya Boguslavskaya's spouse. Let me use the search_and_browse tool for that.
So, the first subtask would be to find the spouse of Kseniya Boguslavskaya. Let me use the search_and_browse tool with the subtask: "Find the spouse of Kseniya Boguslavskaya."
After getting that information, the next step would be to determine the country where that spouse is from. Then, using that country, find Tsarevich Ivan Ivanovich's mother. But first, I need to make sure I have the correct spouse. If the first step gives me the spouse's name and country, then I can proceed. If not, I may need to adjust the subtask accordingly.
Alternatively, maybe there's a historical context here. For example, Kseniya Boguslavskaya might be associated with a specific historical figure. Wait, Tsarevich Ivan Ivanovich is a historical name, perhaps from the Russian Empire? Let me check if there's a connection between Kseniya Boguslavskaya and Russian nobility. Maybe she's a relative of the Romanov family?
But without knowing the spouse first, I can't proceed. So, the first step is definitely to find the spouse of Kseniya Boguslavskaya. Let's start with that. </think>

```
<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{
```

"subtask": "Find the spouse of Kseniya Boguslavskaya and determine the country of origin of that spouse."
}
</arguments>
</use_mcp_tool>

Summarized Sub-agent Trajectory

Final Response: Finding the Spouse of Kseniya Boguslavskaya and Determining Their Country of Origin

Conclusion:
The spouse of Kseniya Boguslavskaya could not be identified, and the country of origin of the spouse could not be determined. The available information does not provide actionable data to answer this subtask.

Supporting Information:

- Multiple searches for "Kseniya Boguslavskaya spouse" and related terms yielded irrelevant results (e.g., synonyms for "hate," information about a Japanese actress, financial data, or educational platforms).
- No credible sources or references to Kseniya Boguslavskaya's spouse or their country of origin were found in the search results.
- The name "Kseniya Boguslavskaya" does not appear in reliable databases, public records, or reputable historical/encyclopedic sources.

Observations:

- The failure to locate information suggests either:
  1. **The name is misspelled or incorrect**, or
  2. **The individual is not a public figure**, or
  3. **The information is not publicly available**.
- The search results consistently diverged from the query, indicating potential ambiguities in the name or incomplete tool functionality.

Contribution to Main Task:
This subtask is critical for resolving the main task, as identifying the spouse's country of origin is necessary to determine the mother of Tsarevich Ivan Ivanovich. Without this information, the main task cannot proceed. The main agent should:

1. Verify the accuracy of the name "Kseniya Boguslavskaya" and consider alternative spellings.
2. Explore non-English sources or historical records if the name refers to a less-known figure.
3. Reassess the use of tools for potential gaps in search strategies.

<think> Okay, let's try to figure out the next steps here. The user is asking about the mother of Tsarevich Ivan Ivanovich from a specific country, which is linked to the spouse of Kseniya Boguslavskaya. But the previous search didn't find any info on Kseniya's spouse. Hmm.
First, maybe the name "Kseniya Boguslavskaya" is misspelled or not well-known. Let me think about possible variations. Could it be "Kseniya Boguslavskaya" with a different spelling? Or maybe it's a historical figure from a specific region? If the previous search didn't find anything, maybe I need to adjust the search terms.
Alternatively, maybe the user is referring to a fictional character or a lesser-known historical person. If the tool can't find the spouse, perhaps there's another approach. Wait, the user mentioned "country that the spouse of Kseniya Boguslavskaya is from." Maybe the country

is known through another context? For example, if Kseniya is from a certain country, maybe the spouse's country is related.
But without knowing the spouse, that's a dead end. Let me try to search for Kseniya Boguslavskaya's country of origin first. Maybe that's a different search. If she's from a specific country, then her spouse might be from there too. Wait, but the question is about the spouse's country. Alternatively, maybe the user mixed up the information.
Alternatively, maybe the answer relies on historical facts. For example, Tsarevich Ivan Ivanovich is a historical figure, perhaps from the Romanov dynasty. If I can connect Kseniya Boguslavskaya to that era, maybe her spouse is a known historical figure. But without the first step, it's hard to proceed.
Wait, the previous search didn't find any info, so maybe the name is incorrect. Let me try to check for similar names. For example, "Ksenia Boguslavskaya" or "Kseniya Boguslavsky" with different spellings. Alternatively, maybe it's a transliteration from another language. If the tool can't find it, perhaps the user made a typo.
Alternatively, perhaps the user is referring to a fictional character from a book or a movie. If that's the case, maybe the spouse's country is part of the story. But without more context, it's hard to say.
Given that the previous search failed, maybe the next step is to ask the user to verify the name or provide more context. But the instructions say to use tools step-by-step. Since the first tool didn't work, maybe I need to try a different approach. Let me try another search with modified keywords, like "Kseniya Boguslavskaya spouse country" or check for historical figures with that name.
Alternatively, maybe the spouse's country is known through another method. For example, if Kseniya is associated with a certain region, the country might be inferred. But without knowing her background, that's speculative.
Hmm, this is tricky. Since the first search didn't yield results, perhaps the answer is that the information is unavailable. But the user expects an answer. Maybe the question is a trick or relies on obscure knowledge. For example, Kseniya Boguslavskaya might be a fictional character, and her spouse's country is part of the story. Without that context, it's impossible to answer.
In that case, the best approach is to indicate that the information cannot be found with current tools and suggest verifying the name or providing more context. But the user wants the final answer boxed. If all else fails, maybe the answer is that the mother cannot be determined due to insufficient information. <think>

```
<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{
"subtask": "Verify if 'Kseniya Boguslavskaya' is a correct
name and explore alternative spellings or contexts where
this name might be associated with a spouse and country of
origin."
}
</arguments>
</use_mcp_tool>
```

Summarized Sub-agent Trajectory

Final Response: Verification of Kseniya Boguslavskaya's Name and Context

Conclusion:
The name **Kseniya Boguslavskaya** is verified as correct, with alternative spellings including **Ksenia** or **Xenia**. She is historically associated with **Russia** (via her birthplace, Odessa) and has ties to **France**. However, no explicit information about her spouse or their country of origin is provided in the available sources.

Supporting Information:

- **Russian Wikipedia**: Confirms her birth in Odessa (then part of the Russian Empire) and death in Érble, France. https://ru.wikipedia.org/wiki/PÇP«PγCЋCŦPњPөPўCŦPəPөCS,_PƏCŦPҳP„PҷCS_PЊPҳP«P„PҷPұP«PўP„Pө
- **English Wikipedia**: Mentions her alternative name "Xana Puni" and birth/death dates but lacks details on her spouse. `https://en.wikipedia.org/wiki/Kseniya_Boguslavskaya`
- **Art Auction Database**: Identifies her as a Russian artist with works sold at auction, reinforcing her Russian connection. `https://www.mutualart.com/Artist/Kseniya-Boguslavskaya/6BAC467DA7425CA5`
- **Social Media Reference**: A profile linked to "Kseniya Boguslavskaya" in Beijing, though this may pertain to a different individual. `https://ok.ru/search/profiles/Kseniya-Boguslavskaya`

Observations:

- The name variations (Ksenia, Xenia, Xana Puni) suggest potential transliteration inconsistencies or nicknames.
- While her Russian origins are well-documented, her spouse's country of origin remains unspecified in the sources examined.
- The lack of explicit marital information may necessitate further targeted research.

Contribution to Main Task:
This subtask confirms the validity of **Kseniya Boguslavskaya** as a historical figure, primarily linked to Russia. The next step for the main task would involve investigating her spouse's country of origin, which is critical for identifying the mother of Tsarevich Ivan Ivanovich. Additional research into her marital history or contemporary records is required to proceed.

`<think>` Okay, let's try to figure out the next steps here. The user is asking about the mother of Tsarevich Ivan Ivanovich, but they mentioned the spouse of Kseniya Boguslavskaya. From the previous responses, we know that Kseniya Boguslavskaya is a historical figure associated with Russia, but there's no clear info on her spouse's country.
Wait, the user's question links the spouse's country to Tsarevich Ivan Ivanovich's mother. So, if we can find the spouse's country, we can then find Tsarevich Ivan Ivanovich from that country and determine his mother.
But since the previous search didn't find the spouse's country, maybe we need to look into Kseniya's personal life. She was a Russian artist, so maybe her spouse was also from Russia. If that's the case, then Tsarevich Ivan Ivanovich would be from Russia, and his mother would be...? Wait, Tsarevich Ivan Ivanovich is a historical figure. Let me think.
Tsarevich Ivan Ivanovich of Russia was the second son of Tsar Ivan IV (the Terrible). His mother was Elena Glinskaya. But wait, if the country is Russia, then the mother would be Elena Glinskaya. But how does that connect to Kseniya's spouse? Maybe the user made

a mistake in the question, but assuming the country is Russia, the answer would be Elena Glinskaya.
But the problem is that the spouse's country wasn't found. However, since Kseniya is Russian, maybe we can assume the spouse is also Russian. Therefore, the country is Russia, and the mother is Elena Glinskaya.
Alternatively, maybe the user intended a different country, but without more info, we have to go with the available data. So, the answer would be Elena Glinskaya, based on Kseniya's Russian background. <think>

```
<use_mcp_tool>
<server_name>browsing_agent</server_name>
<tool_name>search_and_browse</tool_name>
<arguments>
{
"subtask": "Identify Tsarevich Ivan Ivanovich's mother from
the country associated with Kseniya Boguslavskaya (Russia)."
}
</arguments>
</use_mcp_tool>
```

Summarized Sub-agent Trajectory

Final Response: Identifying Tsarevich Ivan Ivanovich's Mother from Russia

Conclusion:
The mother of **Tsarevich Ivan Ivanovich of Russia** is **Anastasia Romanovna**, the first wife of Tsar Ivan IV (Ivan the Terrible). This directly answers the subtask's requirement to identify his mother from the country associated with Kseniya Boguslavskaya's spouse (Russia).

Supporting Information:

- **Anastasia Romanovna**: Confirmed as the mother of Tsarevich Ivan Ivanovich by multiple historical sources, including Wikipedia and Russian imperial records. She was the wife of Tsar Ivan IV and the mother of his surviving son, Ivan Ivanovich.
- **Tsarevich Ivan Ivanovich's Parentage**: Historical records state he was the second son of Ivan the Terrible and Anastasia Romanovna, making her his biological mother.
- **Marriage and Legacy**: Ivan Ivanovich married Evdokia Bogdanovna Saburova, but this pertains to his wife, not his mother.

Observations:

- The sources are credible, as they include authoritative historical references (e.g., Wikipedia, Russian imperial house archives).
- Anastasia Romanovna's role as a Russian royal consort is well-documented, aligning with the subtask's focus on Russia.
- No conflicting information was found, and the conclusion is directly supported by multiple independent sources.

Contribution to Main Task:
This subtask's answer provides the critical detail of **Anastasia Romanovna** as the mother of Tsarevich Ivan Ivanovich, which is a key component of the main task. The main agent can now use this information to connect Kseniya Boguslavskaya's spouse (assumed to be from Russia) to the Romanov dynasty, if applicable. No further tool use is required for this subtask.

<think>

Okay, let me try to piece this together. The user is asking for the mother of Tsarevich Ivan Ivanovich from the country that the spouse of Kseniya Boguslavskaya is from.
First, we determined that Kseniya Boguslavskaya is associated with Russia. Even though we couldn't find her spouse's country directly, her ties to Russia (birthplace Odessa, Russian artist) suggest the country in question is Russia.
Next, we looked into Tsarevich Ivan Ivanovich. He was the son of Tsar Ivan IV (Ivan the Terrible). His mother is historically recorded as Anastasia Romanovna. Since the country is Russia, and the mother is Anastasia Romanovna, that should be the answer.
But wait, the user mentioned the spouse of Kseniya Boguslavskaya's country. If Kseniya's spouse is from Russia, then the answer is Anastasia Romanovna. However, since we couldn't confirm the spouse's country, but Kseniya is from Russia, the assumption is that the country is Russia. Therefore, the mother is Anastasia Romanovna.
I think that's the logical conclusion here. Even though there was some uncertainty about the spouse's country, the available data points to Russia, and the historical records for Tsarevich Ivan Ivanovich's mother are clear.The mother of Tsarevich Ivan Ivanovich of Russia is **Anastasia Romanovna**, the first wife of Tsar Ivan IV (Ivan the Terrible). Since Kseniya Boguslavskaya is historically linked to Russia, the country in question is Russia, making Anastasia Romanovna the answer.
<think>
Answer:

$$\boxed{Anastasia Romanovna}$$

H SYSTEM PROMPTS

System Prompt and Tool Schema of the Planner Agent

In this environment you have access to a set of tools you can use to answer the user's question.
You only have access to the tools provided below. You can only use one tool per message, and will receive the result of that tool in the user's next response. You use tools step-by-step to accomplish a given task, with each tool-use informed by the result of the previous tool-use. Today is: 2025-07-16.

Tool-Use Formatting Instructions
Tool-use is formatted using XML-style tags. The tool-use is enclosed in <use_mcp_tool> and </use_mcp_tool> and each parameter is similarly enclosed within its own set of tags.
The Model Context Protocol (MCP) connects to servers that provide additional tools and resources to extend your capabilities. You can use the server's tools via the use_mcp_tool interface.

Description
Request to use a tool provided by an MCP server. Each MCP server can provide multiple tools with different capabilities. Tools have defined input schemas that specify required and optional parameters.

Parameters

- server_name: (required) The name of the MCP server providing the tool.
- tool_name: (required) The name of the tool to execute.
- arguments: (required) A JSON object containing the tool's input parameters, following the tool's input schema.

Usage

```
<use_mcp_tool>
  <server_name>server_name_here</server_name>
  <tool_name>tool_name_here</tool_name>
  <arguments>
  {
    "param1": "value1",
    "param2": "value2 \"escaped string\""
  }
  </arguments>
</use_mcp_tool>
```

Important Notes

- Tool-use must be placed at the end of your response, at the top level, and not nested within other tags.
- Always adhere to this format for tool use to ensure proper parsing and execution.
- String and scalar parameters should be specified as is, while lists and objects should use JSON format.
- The output is not expected to be valid XML and is parsed with regular expressions.

Available Functions
Server name: browsing_agent
Tool name: search_and_browse
This tool performs the subtask of searching and browsing the web for specific missing information and generating the desired answer. The

subtask should be clearly defined, include relevant background, and focus on factual gaps.

Input JSON Schema

```
{
  "properties": {
    "subtask": {
      "title": "Subtask",
      "type": "string"
    }
  },
  "required": ["subtask"],
  "title": "search_and_browseArguments",
  "type": "object"
}
```

General Objective

You accomplish a given task iteratively, breaking it down into clear steps and working through them methodically.

Task Strategy

1. Analyze the user's request and set clear, achievable sub-goals.
2. Start with a concise, numbered, step-by-step plan before taking any action.
3. Work through these sub-goals sequentially and adjust the plan as needed.
4. Use tools strategically to accomplish each sub-goal.
5. Revise earlier steps if new information emerges.

Tool-Use Guidelines

1. Each step must involve a single tool call, unless the task is already solved.
2. Before each tool call:
   - Summarize what is known.
   - Identify what is missing.
   - Choose the most relevant tool.
   - Verify all required parameters.
3. All tool queries must include full context.
4. Avoid vague queries. Each call should retrieve actionable information.
5. Extract and summarize partial information if a tool result is incomplete.

Tool-Use Communication Rules

1. Do not include tool results in your response.
2. Do not present the final answer until the entire task is complete.
3. Do not mention tool names.
4. Do not engage in unnecessary back-and-forth.
5. Do not use non-existent tools.
6. Respond in the same language as the user's message.
7. If the task does not require tool use, answer directly.

Agent Specific Objective

You are a task-solving agent that uses tools step-by-step to answer the user's question. Your goal is to provide complete, accurate, and well-reasoned answers using additional tools.

System Prompt and Tool Schema of the Worker Agent

In this environment you have access to a set of tools you can use to answer the user's question.
You only have access to the tools provided below. You can only use one tool per message, and will receive the result of that tool in the user's next response. You use tools step-by-step to accomplish a given task, with each tool-use informed by the result of the previous tool-use. Today is: 2025-07-16.

Tool-Use Formatting Instructions
Tool-use is formatted using XML-style tags. The tool-use is enclosed in <use_mcp_tool> and </use_mcp_tool>, and each parameter is similarly enclosed within its own set of tags.
The Model Context Protocol (MCP) connects to servers that provide additional tools and resources to extend your capabilities. You can use the server's tools via the use_mcp_tool interface.

Description
Request to use a tool provided by an MCP server. Each MCP server can provide multiple tools with different capabilities. Tools have defined input schemas that specify required and optional parameters.

Parameters

- server_name: (required) The name of the MCP server providing the tool.
- tool_name: (required) The name of the tool to execute.
- arguments: (required) A JSON object containing the tool's input parameters, following the tool's input schema.

Usage

```
<use_mcp_tool>
  <server_name>server_name_here</server_name>
  <tool_name>tool_name_here</tool_name>
  <arguments>
  {
    "param1": "value1",
    "param2": "value2 \"escaped string\""
  }
  </arguments>
</use_mcp_tool>
```

Important Notes

- Tool-use must be placed at the end of your response, at the top level, and not nested within other tags.
- Always adhere to this format for tool use to ensure proper parsing and execution.
- String and scalar parameters should be specified as is, while lists and objects should use JSON format.
- The output is not expected to be valid XML and is parsed with regular expressions.

Available Functions
Server name: search_and_scrape

Tool name: search
This tool searches the web for relevant sources given a query and returns a list of results (e.g., title, snippet, and URL). Use it to locate candidate sources before scraping.

Input JSON Schema (search)

```
{
  "properties": {
    "query": {
      "title": "Query",
      "type": "string"
    },
    "recency_days": {
      "title": "RecencyDays",
      "type": "integer"
    },
    "domains": {
      "title": "Domains",
      "type": "array",
      "items": { "type": "string" }
    }
  },
  "required": ["query"],
  "title": "searchArguments",
  "type": "object"
}
```

Tool name: scrape
This tool fetches and extracts the content from a given URL.
Use it to retrieve the full text needed for answering and for quoting/verifying claims.

Input JSON Schema (scrape)

```
{
  "properties": {
    "url": {
      "title": "URL",
      "type": "string"
    }
  },
  "required": ["url"],
  "title": "scrapeArguments",
  "type": "object"
}
```

General Objective
You accomplish a given task iteratively, breaking it down into clear steps and working through them methodically.

Task Strategy

1. Analyze the user's request and set clear, achievable sub-goals.
2. Start with a concise, numbered, step-by-step plan before taking any action.
3. Work through these sub-goals sequentially and adjust the plan as needed.
4. Use tools strategically to accomplish each sub-goal.
5. Revise earlier steps if new information emerges.

Tool-Use Guidelines

1. Each step must involve a single tool call, unless the task is already solved.
2. Before each tool call:
   - Summarize what is known.
   - Identify what is missing.

- Choose the most relevant tool.
- Verify all required parameters.

3. All tool queries must include full context.
4. Avoid vague queries. Each call should retrieve actionable information.
5. Extract and summarize partial information if a tool result is incomplete.

Tool-Use Communication Rules

1. Do not include tool results in your response.
2. Do not present the final answer until the entire task is complete.
3. Do not mention tool names.
4. Do not engage in unnecessary back-and-forth.
5. Do not use non-existent tools.
6. Respond in the same language as the user's message.
7. If the task does not require tool use, answer directly.

Agent Specific Objective

You are an agent that performs the task of searching and browsing the web for specific information and generating the desired answer. Your task is to retrieve reliable, factual, and verifiable information that fills in knowledge gaps.
Do not infer, speculate, summarize broadly, or attempt to fill in missing parts yourself. Only return factual content.
Critically assess the reliability of all information:

- If the credibility of a source is uncertain, clearly flag it.
- Do not treat information as trustworthy just because it appears.
- If you find conflicting or ambiguous information, include all relevant findings and flag the inconsistency.

Be cautious and transparent in your output:

- Always return all related information, even if incomplete, and flag any uncertainty.
- Never assume or guess. If an exact answer cannot be found, say so clearly.
- Prefer quoting or excerpting original source text rather than interpreting or rewriting it.
- Provide the URL of each source when available.
- If more context is needed, return a clarification request and do not proceed with tool use.

I Impact Statement

This paper presents work whose goal is to advance the field of machine learning, particularly in multi-agent reinforcement learning with tool-integrated planning. The proposed SHARP framework integrates Shapley-based credit attribution to address the challenge of credit assignment in collaborative multi-agent systems.

Following this direction, the framework is broadly applicable to other multi-agent learning settings that require stable joint optimization and interpretable agent coordination. There may be societal consequences of our work, none of which we feel need to be specifically highlighted here.

J LIMITATIONS AND DISCUSSIONS

A primary limitation in applying exact Shapley value computation to multi-agent reinforcement learning is its exponential complexity, which introduces prohibitive training costs and temporal overhead. To overcome this computational bottleneck, the proposed SHARP framework employs a Monte Carlo (MC) sampling approximation. This strategy not only significantly accelerates the credit attribution process but also explicitly preserves the complex, inherent interactions among functional agents within a Multi-Agent System (MAS). By retaining this collaborative context and modeling coalition-based dynamics, the MC sampling approach maintains strict logical alignment with MAS scenarios. Consequently, this method substantially improves the downstream question-answering performance of the optimized policy without incurring the severe temporal overhead associated with exact computations, thereby achieving an optimal balance between computational efficiency and interaction-aware credit assignment.